Synthesis of Embedded Networks for Building Automation and Control

Alessandro Pinto*, Massimiliano D'Angelo†, Carlo Fischione*, Eelco Scholte‡, Alberto Sangiovanni-Vincentelli*
* Department of Electrical and Computer Sciences, University of California, Berkeley, CA, USA
{apinto,fischion,alberto}@eecs.berkeley.edu
† University of L'Aquila, L'Aquila, Italy and PARADES GEIE, Rome, Italy
mdangelo@parades.rm.cnr.it
‡United Technologies Research Center, East Hartford, CT 06108, USA
scholte@utrc.utc.com

Abstract— **We present a methodology and a software framework for the automatic design exploration of the communication network among sensors, actuators and controllers in building automation systems. Given 1) a set of end-to-end latency, throughput and packet error rate constraints between nodes, 2) the building geometry, and 3) a library of communication components together with their performance and cost characterization, a synthesis algorithm produces a network implementation that satisfies all end-to-end constraints and that is optimal with respect to installation and maintenance cost. The methodology is applied to the synthesis of wireless networks for an essential step in any control algorithm in a distributed environment: the estimation of control variables such as temperature and air-flow in buildings.**

I. INTRODUCTION

A building automation system is the interconnection of a large number of sensors, actuators and controllers distributed on thousands of square feet. Communication among these components takes place over a network, whose design is subject to several constraints. The control algorithm for applications such as fire detection systems, temperature control, distributed control of air flow in buildings imposes end-to-end communication constraints from the sensors to the controllers, and from these to the actuators. The constraints imposed by the application include latency, bandwidth and packet error rate. The building geometry imposes constraints on the possible location of nodes, on the wires' layout and on the wireless communication between nodes.

The cost of the communication network constitutes a large portion of the overall cost of a building automation system. Therefore, it is desirable to tailor the network architecture to both the control algorithm and the building geometry, avoiding to waste costly communication resources. A variety of network components is available on the market to achieve this goal. In recent years, many wired and wireless protocol standards for building automation have emerged [7], [8], [9], [10], as well as a number of companies providing components that are compliant with these protocol specifications. This is an opportunity for engineers to design networks that are application-specific. *However, the difficulty resides in the ability to match the application and physical constraints to the performance offered by network components, while minimizing the total network cost*. In fact, to avoid long verification cycles, engineers tend to use architectures that have been already tested in previous designs. Moreover, the network is purposely over-designed to make the communication delay negligible compared to the time-scale of the control algorithms. The result is a network that is far from being cost-effective.

Therefore, an automatic synthesis flow that is able to find an optimal network implementation starting from the constraints and the available communication components would provide invaluable help to build cost-effective and correct-by-construction building automation networks. We propose a methodology and a companion software framework that facilitates the design exploration of control networks. The application is captured by a set of point-to-point communication constraints between nodes that have a fixed position in the building. The implementation space (i.e., the set of possible network implementations) is implicitly captured by a library of components characterized by cost and performance models. The building geometry is also taken into account by capturing the position of the walls, restrictions on the positions of nodes, and wiring constraints. This methodology has been used in the context of wired networks for building automation systems; we refer the reader to [6] for the details.

In this paper, we focus on wireless networks. We formulate an optimization problem to find an optimal wireless network implementation that satisfies all the constraints. The implementation is optimal in the sense that it minimizes a cost function that consists of actual dollar cost for components and installation. We derive an Integer Linear Programming (ILP) formulation of the optimization problem and we solve it using CPLEX [12], leaving the development of efficient heuristics to minimize computation time and increase the size of the problems that can be tackled by our approach for future work. We apply the methodology to an essential step in any distributed control algorithm: the distributed estimation of physical control variables such as temperature and air-flow.

II. CASE STUDY: DISTRIBUTED ESTIMATION IN BUILDINGS

We use a distributed estimation example to illustrate our work. Figure 1 shows the input of our design flow, which

978-1-4244-2079-7/08/$25.00 ©2008 AACC.

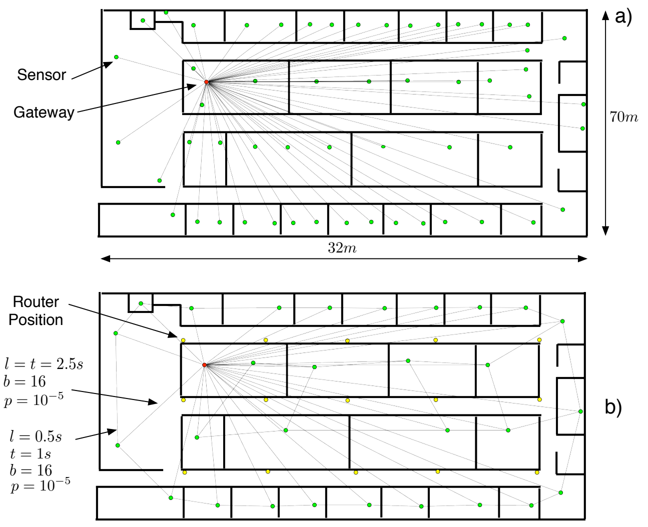

Fig. 1. The two test cases used in our experiments: a) Distributed estimation, b) Centralized estimation.

includes the floor-plan of a building and a set of sensors located in predefined positions.

The centralized estimation dynamics (Figure 1(a)) can be described by the following set of equations:

$$\begin{aligned} x_{k+1} &= f(x_k, u_k, w_k) \\ y_k &= h(x_k, u_k, v_k) \end{aligned}$$

where u_k represents the measure from the sensors, w_k and v_k model noise, and x_k is the state of the system, and k is the time index. The measures u_k are sent to a central gateway that computes the estimate x_{k+1} and, in the case of closed loop control, the output y_k to be sent to actuators. Communications between the sensors and the gateway are represented by connectors annotated with the required quality of service: To guarantee estimation accuracy, the delay and packet error rate of the communication cannot exceed a certain upper bound that depends on the application.

In the distributed estimation case (Figure 1(b)), the dynamics can be described as follows:

$$\begin{aligned} x_{k+1,i} &= f(x_{k,i}, x_{k,\Gamma_k}, u_{k,i}, w_{k,i}) \\ y_{k,i} &= h(x_{k,i}, x_{k,\Gamma_k}, u_{k,i}, v_{k,i}) \end{aligned}$$

where i denotes the index in space (and would be of size 3 for a 3D building). In the previous equations, Γ_k denotes the set of indexes of the states that influence the one in position i (e.g. its nearest neighbors). In the case of a flat hierarchy, the set Γ_k generally contains only its nearest neighbors $\Gamma_k = [i, i+1, i-1]$ (for a single dimension), however, in the case of decentralized estimation [5], possibly augmented with multi-scale consensus [4], the set Γ_k will contain a larger set of states, which are not necessarily close in space, since now nodes might have to communicate at slower scales with locations that are further away. Therefore, a sensor in position i communicates with its neighbors exchanging the value of the state variables.

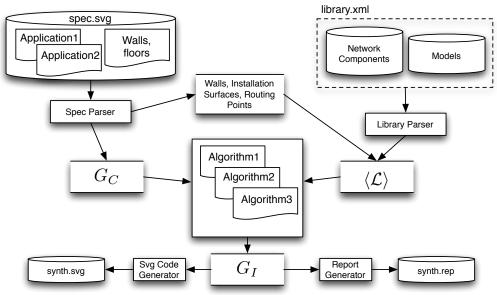

Fig. 2. Design flow for building automation networks implemented in the Communication Synthesis Infrastructure (COSI) [11].

Assuming the estimator and control algorithm can tolerate a maximum delay ΔT_{max}, the requirements for the network can be formulated, for the nominal case, as an end-to-end maximum delay. Additionally, the amount of data, and the required redundancy translate into a minimum bandwidth and maximum packet error rate requirement.

III. Methodology and Synthesis Flow

Figure 2 shows the design flow and software infrastructure that we use to synthesize communication networks. We adopt the platform-based design methodology [3]. This methodology advocates the separation of specification (i.e what the communication network is supposed to do), and implementation *platform* which is the set of realizable networks (i.e., how the specification is implemented). We capture the specification and the platform using a formal model. For the sake of simplicity, in this section we present a simplified view of this model, but a full-fledged model is presented in detail in [2].

In our framework, a *network* is a directed graph $G(V,E)$ together with labeling functions. The vertexes represent network nodes such as sources/destinations, routers, and repeaters, and the edges represent communication links connecting the nodes. Nodes and links together are called *components*. A labeling function is a map $V \cup E \rightarrow D$ where D is the domain of the components' label. A label is a tuple of metrics that characterize a component like latency, packet-error rate and position. For instance, the initial specification of a communication problem is defined by a point-to-point network $G_C(V_C, E_C)$ with associated position and types of the nodes, and bandwidth, latency, message length and packet error rate of the links.

The *network library* $\mathscr{L}$ is a collection of networks. The labeling functions of a library element are used to capture its performance envelope. For instance, a network $G_P(V_P, E_P) \in \mathscr{L}$ can be annotated with the maximum bandwidth (also called capacity) that the links E_P can support. Usually, many labeling functions can characterize the performance of the same library element. For instance, the position of a node can be assigned to many points inside a building.

The elements of $\mathscr{L}$ can be instantiated and composed to form larger networks. The composition of two networks is an important operation in our framework. This operation defines how to obtain the labeling functions of the composite network starting from the labeling functions of the networks being composed. Furthermore, the operator defines when the composition of two networks is valid by imposing constraints that are called *composition rules*. The set of all valid compositions of networks in $\mathscr{L}$ is called *network platform* and an element of the platform is called *network platform instance*.

A synthesis algorithm takes the specification G_C and the network platform and generates a network implementation G_I that minimizes a cost function while satisfying the specification. Different synthesis algorithms can be developed to leverage the particular structure of the communication synthesis problem in a given domain, thus exploring the design space more efficiently.

We take as input a graphical description of the building geometry in SVG format that captures the walls, the positions that are candidate for the installation of routers and a set of cable ladders that are available to layout wires. Communication constraints are also captured in this file. The library and the models are described in a separate XML file. We generate a graphical representation of G_I and a textual report of its cost.

IV. Communication Synthesis

We show how the Communication Synthesis Infrastructure can be used to synthesize a wireless network for building automation systems. We formulate the synthesis problem as an Integer Linear Program. Then, we specialize the synthesis problem to the case of beacon-enabled ZigBee networks.

A. Communication Constraints

We are given a specification $G_C(V_C, E_C)$ where the set of nodes is $V_C = \{v_1, \dots, v_n\}$ and the set of constraints is $E_C = \{e_1, \dots, e_k\}$. We denote by $N = \{1, \dots, n\}$ the index set of the nodes and we simply refer to node i instead of node v_i. Also, we denote by $Q = \{1, \dots, k\}$ the index set of the constraints. For a constraint $e_i \in E_c$, we denote by $s_i, d_i \in V_C$ its source and destination, respectively. The label associated to a constraint e_i is a 4-tuple (t_i, l_i, b_i, p_i) where t_i is the message period, l_i is the maximum latency, b_i is the number of bits per period and p_i is the maximum packet error rate probability.

B. Communication Library and Platform

COSI supports the definition of library components at different abstraction levels and with different granularity. Components are networks whose labeling functions define their performance space. In this article we focus on wireless nodes and wireless links. Performance metrics are captured by performance models that are independent from the components. Therefore, the components described here are representative of a wide class of wireless networks. Sections IV-D and IV-E describe detailed models for Zigbee networks.

Nodes. The library contains two types of nodes: devices and routers. Devices are sensors or actuators and do not have routing capabilities. Routers can receive and forward packets to other nodes according to a routing table. The position of sensors and actuators is given as part of the specification. The position and number of routers to be installed in the final implementation are decision variables. We assume that the set of candidate locations where routers can be installed is restricted to a finite set $\{p_1, \dots, p_m\}$ and we denote by $M = \{n+1, \dots, n+m\}$ its index set. By node $i \in M$ we mean a router in position p_{i-n}.

Links. A link is characterized by its bit-rate (i.e. the maximum number of bits per second that can be sent across the link) and a function that associates to the signal to noise plus interference ratio (*SINR*) a bit error rate. Also, a threshold on the *SINR* is associated to the link, meaning that if the value of the *SINR* at a receiver is below the threshold, a wireless link cannot be instantiated to connect the transmitter to the receiver.

Composition Rules. The set of valid network implementations (i.e. valid compositions of library elements) is restricted by topological constraints. In this paper, the topology of the network is restricted to be a tree. Because the wireless channel is a shared medium, we set an upper bound on the number of routers that can be installed in a certain space. This is particularly important when communications are orthogonal in time (i.e. for TDMA-like systems). The number of incoming connections that a router can serve is also upper bounded. Finally, a wireless link can only connect devices to routers, routers to routers and routers to devices. Section IV-C discusses a linear formulation of all these constraints.

Cost Function. The cost of a wireless network is defined as the sum of the cost of the nodes. Each node has a retail price and an installation cost that may be dependent on its position. Moreover, it has a maintenance cost that is the cost of replacing the batteries over 20 years of operation. If E_{tx} and E_{rx} are the energy to transmit and receive one bit (respectively), and E is the battery capacity (expressed in J), then the total number of batteries to be replaced is $B = (b_{rx}E_{rx} + b_{tx}E_{tx})/E$ where b_{rx} and b_{tx} are the total number of bits received and transmitted over 20 years. The maintenance cost can be calculated as $B \cdot c_B$ where c_B is the sum of the retail price of a battery plus the cost of replacing them.

C. Synthesis Problem

We formulate the synthesis problem for wireless networks as an integer linear program (ILP). We make the following assumptions. Nodes in N are devices, i.e. simple sensors and actuators. The topology of the network is a tree where the leaves are the nodes in N. The network relies on a synchronization service such that each router transmits/receives to/from its children at regular intervals called *beacon intervals*.

Since the number of points where the nodes can be installed is finite (it has size $|N| + |M|$), it is possible to define a node-edge incidence matrix A of the induced graph on those points. An entry of the matrix $A(i,e)$ is equal to $1(-1)$ if a wireless link e can be installed with its source(destination) being i, and 0 otherwise. Let x_i be a binary variable that is equal to 1 if node i is installed and 0 otherwise. Let e_{ij} be a binary variable that is equal to 1 if a wireless link is instantiated between node i and node j, where j is the parent of i. Moreover, let y_{ijq} be a binary variable that is equal to 1 if the path from s_q to d_q uses a wireless link available between node i and node j, and $\mathbf{y}_q$ be the column vector of such variables [1] for all the edges. A path from s_q to d_q is an assignment to the components of $\mathbf{y}_q$ that satisfies $A_q\mathbf{y}_q = \mathbf{b}_q$, where $\mathbf{b}_q$ is a vector such that $\mathbf{b}_q(i) = 1$ for $i = s_q$, $\mathbf{b}_1(i) = -1$ for $i = d_q$ and $\mathbf{b}_1(i) = 0$ otherwise. The components of $\mathbf{y}_q$ that are equal to 1 are the wireless links belonging to the path, thus we simply refer to a solution of $A_q\mathbf{y}_q = \mathbf{b}_q$ as a *path*.

Given a path and given an additive quantity $w(i,j)$ defined on each link, we can compute the end-to-end value of the quantity with a linear expression $w(s_q,d_q) = \sum_{ij} y_{ijq}w(i,j)$. Because we assume synchronization, each wireless link is characterized by an upper bound on the delay that is equal to the beacon interval. Instead of using the packet error rate p as a measure of the quality of a path, we consider the probability of success $p' = 1 - p$, which can be expressed as:

$$p'(s_q,d_q) = \prod_{ij:y_{ijq}=1} p'(i,j) \ ,$$

which is not a linear expression. Therefore we use the logarithm of p' in the formulation of the problem:

$$\log p'(s_q,d_q) = \sum_{ij} y_{ijq} \log p'(i,j) \ .$$

The following ILP models the synthesis problem:

$$\begin{array}{lll}
\mathscr{P}: & \min_{\mathbf{x},\mathbf{y}_q} \ F & \\
 & s.t. & \\
1. & x_i + x_j - 2e_{i,j} \geq 0 & \forall i,j \in N \cup M \\
2. & e_{ij} + e_{ji} \leq 1 & \forall i,j \in N \cup M \\
3. & e_{ij} = 0 & \forall j \in N \\
4. & \sum_{ij} e_{ij} - \sum_i x_i = -1 & \forall i \in N \cup M \\
5. & \sum_i e_{ij} \leq in_{max} & \forall j \in M \\
6. & e_{ij} + e_{ji} - y_{ijq} \geq 0 & \forall i,j \in N \cup M, \forall q \in Q \\
7. & e_{ij} + e_{ji} - y_{jiq} \geq 0 & \forall i,j \in N \cup M, \forall q \in Q \\
8. & A_q\mathbf{y}_q = \mathbf{b}_q & \forall q \in Q \\
9. & \sum_q y_{ijq}(b_q + O) \leq b_{max} & \forall i,j \in N \cup M \\
10. & \sum_{i \in M} x_i \leq n_{max} & \\
11. & \sum_{ij} y_{ijq} l(i,j) \leq l_q & \forall q \in Q \\
12. & \sum_{ij} y_{ijq} \log p'(i,j) \leq \log(1 - p_q) & \forall q \in Q \\
13. & x_i, \ e_{ij}, \ y_{ijq} \in \{0,1\} & \forall i,j \in N \cup M, \forall q \in Q
\end{array}$$

where the cost function can be expressed as follows:

$$F = \sum_i c_i x_i + \sum_{ij} c_{ij} \sum_q y_{ijq} \ ,$$

The cost c_i includes the cost to install a node while c_{ij} contribute to the maintenance cost of nodes i and j (that obviously depends on the number of bits transmitted by i and received by j).

Constraints 1 through 5 define the tree topology of the network. Constraint 1 says that a link can be installed only between two installed nodes. Constraint 2 imposes a unidirectional parent-children relationship while constraint 3 says that sensors cannot be parent nodes. Constraint 4 forces the topology to be a tree by imposing the number of edges to be one less than the number of nodes. Constraint 5 limits the maximum number of children per node. Constraint 6 and 7 say that a path can be routed on a link only if that link is installed. Constraint 8 is the classical balance equation. Constraint 9 is a constraint on the maximum utilization of a link (where O denotes the protocol overhead) and constraint 10 limits the maximum number of routers that can be installed. Constraints 11 and 12 are the end-to-end constraints on the delay and packet error rate. Finally constraint 13 requires all variables to be binary. Notice that this problem is still very general and can be directly written by interpreting the constraints given by the platform in terms of the available components, the building structure and the performance and cost models of each component. In fact, the ILP formulation represents a class of problems. One instance of the ILP corresponds to one particular building and one particular network configuration. The rest of this section describes how to determine the parameters of problem $\mathscr{P}$ for ZigBee [10] networks and for a general channel model.

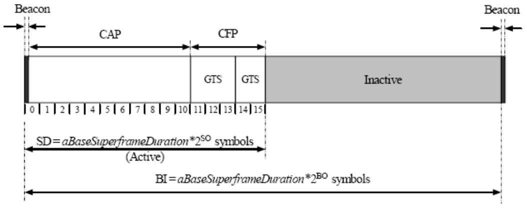
Beacon
Beacon
CAP
CFP
GTS
GTS
Inactive
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
SD = aBaseSuperframeDuration*2^SO symbols
(Active)
BI = aBaseSuperframeDuration*2^BO symbols

Fig. 3. Structure of a superframe as defined by the ZigBee protocol standard.

D. Performance models for ZigBee networks: MAC and Network layers

The protocol stack of a ZigBee node is composed of the physical layer and Medium Access Control (MAC) Layer described in the IEEE802.15.4 standard, and a network layer and an application framework defined by the ZigBee Alliance. At the physical layer, the IEEE802.15.4 standard offers a total of 27 channels, with a peak rate of 250Kbit/s (parameter b_{max} in problem $\mathscr{P}$). At the MAC layer, nodes are grouped in PANs (Personal Area Networks). While simultaneous transmissions in different PANs can not collide, because they take place on different channels, intra-PAN transmissions need to be coordinated. The *superframe structure* (see Fig.3) is a flexible way to manage medium access control inside a PAN. A PAN is started by a node that assumes the role of PAN Coordinator, which establishes the values of a set of configuration parameters of the superframe. These parameters have to be adopted by all the nodes that want to be associated with such a PAN. The coordinator fixes the physical channel, the Beacon Order (BO) and Superframe Order (SO) of the superframe structure.

The PAN coordinator can periodically transmit a beacon frame (beacon-enabled mode). The time interval between

[1] We assume a fixed ordering of the edges.

two consecutive beacons is called Beacon Interval (*BI*) and it is defined as $BI = aBaseSuperframeDuration \times 2^{BO}$ *symbols*. The *aBaseSuperframeDuration* has a constant duration of 960 symbols. The beacon order *BO* can range from 0 to 14 (and $BO = 15$ means that no beacon has to be transmitted, i.e. non beacon-enabled mode). The beacon interval *BI* is composed of an active part and an (optional) inactive part. The duration of the active part is determined by the Superframe Duration (SD), which is defined as $SD = aBaseSuperframeDuration \times 2^{SO}$ *symbols*. The superframe order *SO* can range from 0 to *BO* (no inactive period).

The active period can be further divided in two parts:

- A Contention Access Period, in which transmissions are ruled by a slotted CSMA/CA algorithm and therefore collisions can occur. A minimum length (440 symbols) of the contention access period has to be guaranteed for the transmission of management frames.
- An (optional) Contention Free Period, which is composed of up to 7 Guaranteed Time Slots (GTSs) (parameter in_{max} in problem $\mathscr{P}$). A GTS can be used by a device for transmission (reception) of data to (from) the coordinator. Transmissions in the GTS is uniquely allocated to a device, and concurrent transmissions by other devices in the same PAN are forbidden.

In the inactive part, nodes can put the transceiver in a sleep state and save energy. Optionally, a device can assume the role of coordinator, which has to adopt the same *BO* and *SO* as the PAN coordinator. In a beacon-enabled PAN ($BO \neq 15$), such a device will start transmitting its own beacon. Its active part must not overlap with the active part of other coordinators in the network. Therefore, the maximum number of coordinators in a beacon-enabled PAN, including the PAN coordinator itself, can be no more than BI/SD (parameter n_{max} in problem $\mathscr{P}$).

ZigBee supports three network topologies. In the star topology configuration the Zigbee coordinator controls the entire network and all devices directly communicate with the ZigBee coordinator. In the mesh topology, peer-to-peer communications between devices are allowed. The ZigBee coordinator is still responsible for starting the network that can be extended by adding ZigBee routers. When a mesh topology is used, coordinators shall not send beacons. In the tree topology configuration, ZigBee routers move data and control messages through the network using a hierarchical routing strategy. Beacon-enabled communications within the PAN are allowed which makes possible to synchronize communications and implement a contention-free PAN. This is our preferred configuration since it allows to use duty-cycling to save energy and to precisely characterize the delay between nodes.

E. Performance models for ZigBee networks: the physical layer

Consider a node i of the network transmitting packets with a radio power level P_i toward node j. The distance between node i and j is denoted with $d_{i,j}$. We denote with $\mathrm{PL}(d_{i,j})_{dB}$ the path loss attenuation between the transmitter and the receiver. For example, for the Telos Sky wireless sensors [15], the following generic yet representative model of the path-loss can be used [13]:

$$\mathrm{PL}(d_{i,j})_{dB} = \mathrm{PL}(d_0)_{dB} + 10\beta \log_{10}\left(\frac{d_{i,j}}{d_0}\right) + \Omega_{i,j} + \mathrm{PL}_{mw},$$

where $\mathrm{PL}(d_0)$ is the path loss computed at a reference distance d_0, β is the path loss exponent, and $\Omega_{i,j}$ is the shadowing attenuation, which is modeled as a Gaussian random variable having zero average and variance $\sigma^2_{i,j}$. We adopt a multi-wall model [1] to account for the path loss due to the presence of walls between a transmitter and a receiver. Therefore, $\mathrm{PL}_{mw} = L_C + n_W L_W$ where L_C is a constant, n_w is the number of walls intersected by the line of sight between the transmitter and the receiver, and L_w is 3.4dB or 6.9dB depending on the thickness of the wall.

The Signal to Interference plus Noise Ratio (SINR) in dB can be modeled as follows:

$$10\log_{10}\mathrm{SINR}_{i,j} = 10\log_{10}P_{i,j} - P_{n\,dB}, \qquad (1)$$

where $P_{i,j}(d)$ is the radio power received at the node j from the node i:

$$10\log_{10}P_{i,j} = 10\log_{10}P_i - \mathrm{PL}(d_{i,j})_{dB},$$

In Equation (1), we make the assumption that nodes are not simultaneously transmitting (i.e the network operates in beacon-enabled mode and its topology is a tree), so that the collision probability can be neglected and P_n summarizes the thermal noise and the power of the interference coming from co-channel radio systems (as, e.g., WIFI networks). Hence, we assume that the power of the noise term is a constant term N_0. A typical value for the power of the thermal noise for the Telos Sky receivers is $N_0 = -170$dBm.

The bit error probability of the link from node i to node j can be expressed as

$$p_b(\mathrm{SINR}_{ij}) \triangleq f_1(\mathrm{SINR}_{ij}), \qquad (2)$$

where $f_1(\cdot)$ is a function that accounts for the relation among the modulation format, the statistical distribution of the SINR, and the bit error rate. The bit error probability for O-QPSK modulation (also adopted by the Telos Sky nodes) with coherent demodulation in a slow Rayleigh fading environment (corresponding to slow moving objects), which exhibits non-selective behavior both in frequency and in time, can be expressed by [13]

$$f_1(\mathrm{SINR}_{i,j}) \approx \frac{1}{2}\left(1 - \sqrt{\frac{\mathrm{SINR}_{i,j}}{1+\mathrm{SINR}_{i,j}}}\right),$$

Using (2), it is possible to express the packet loss probability. Assume that a packet at the data-link layer is composed of O bits of protocol overhead and a payload of b_i bits. Under the assumption that the CRC code is always able to detect erroneous packets (see [14] for an experimental support), the packet loss probability, without any retransmission mechanism, can be expressed as

$$p'(i,j) \triangleq f_2(\mathrm{SINR}_{ij}) = 1 - [1 - p_b(\mathrm{SINR}_{ij})]^{O+b_i}. \; (3)$$

Equation (3) can be easily extended to include FEC.

BO	SO	Cost ($ in 20 yr)	T_{cpu} (s)	Gap
7	3	23760	1400	5%
8	3	15150	590	4.9%
8	4	22900	1400	6%

TABLE I

RESULTS FOR THE CENTRALIZED ESTIMATION CASE.

V. A CASE STUDY: DISTRIBUTED ESTIMATION IN BUILDINGS

As a case study, we consider the centralized and distributed estimation of a physical quantity in the building shown in Figure 1. This building is one floor of a business premises with 25 rooms and a total dimension of 32×70 m^2. Green dots represent sensors, the red dot corresponds to a central gateway and yellow dots are the positions where routers can be installed. A connector between two dots corresponds to a communication constraint with associated latency, period, number of bits and packet error rate requirements. The number of sensors in the distributed case is less than the number of sensors in the centralized case but there is mutual interaction between sensors corresponding to neighboring states. The mutual interaction is considered bidirectional and takes place at a higher rate than the communication with the central gateway.

Table I shows the results of the optimization for the centralized case. We report the beacon order BO, the superframe order SO, the total network cost over 20 years, the computation time T_{cpu} in seconds (on an Intel Xeon @ 2.8 GHz, 512 MB of RAM) and the optimality gap. The problem is unfeasible for $SO < 3$ due to the bandwidth requirements and for $BO > 8$ due to the latency requirements. The minimum difference between BO and SO must be 4 because the minimum number of routers needed for this network is greater than 8. This is due not only to the number of sensors, but also to the packet error rate constraint and the building geometry that limits the maximum length of a hop intersecting multiple walls. We also mention that the wired network implementation for this application has a total cost of $18000.

In the distributed estimation case, the number of sensors is reduced by half. However, the stringent latency requirements between neighbor nodes limits the maximum beacon order to 5. The reduced bandwidth requirement at the gateway allows a superframe order of 1 making the duty-cycle equal to 6.25%. This instance of the problem has been solved to the global optimum for a total cost of $23400. We notice that this is the only combination of values for BO and SO that is feasible.

VI. CONCLUSIONS AND FUTURE WORK

We presented a general framework for the design exploration of wireless communication infrastructures for building automation. We captured the communication requirements as end-to-end constraints among sensors, actuators and computational units. The building geometry is also captured to account for the degradation of the communication channel due to radio power attenuation. We formulated an optimization problem whose decision variables are the routers and the wireless links to be installed. The problem was specialized to the case of ZigBee networks for distributed estimation. The methodology is very general; it was also used to synthesize wired networks in buildings [6]. The proposed framework will allow designers to populate the library with network components and to use automatic synthesis algorithms to decide the best network technology for a specific application and a specific building. Being based on a formal model, the design flow generates solutions that are correct-by-construction, relieving designers from the burden of long manual verification cycles. We plan to extend this methodology to the case of non beacon-enable networks taking into account several interactive radio parameters such as power levels, modulation formats and coding.

VII. ACKNOWLEDGMENTS

A. Pinto, C. Fischione and A. Sangiovanni-Vincentelli acknowledge the support of the NSF ITR CHESS, the MICRO program of the State of California, UTRC and the GSRC. M. D'Angelo acknowledges the support of the Fondazione Ferdinando Filauro of the University of L'Aquila, Italy.

REFERENCES

[1] European COST Action 231. "Digital Mobile Radio Towards Future Generation Systems (COST 231 Final Report)". 1999, Available at http://www.lx.it.pt/cost231/,

[2] A. Pinto, L. P. Carloni, and A. L. S. Vincentelli. "A methodology and an open software infrastructure for the constraint-driven synthesis of on-chip communications". Technical Report UCB/EECS-2007-130, University of California, Berkeley, November 2007.

[3] Alberto L. Sangiovanni Vincentelli "Quo Vadis SLD: Reasoning about Trends and Challenges of System-Level Design", Proceedings of the IEEE, pp. 467–506, March 2007

[4] J.-H. Kim, M. West, E. Scholte and S. Narayanan "Multi-scale Consensus for Decentralized Estimation and its Application to Building Systems", American Control Conference, 2008.

[5] R. Olfati-Saber, "Distributed Kalman Filter with Embedded Consensus Filters," in Proc. of 44th IEEE Conference on Decision and Control, and the European Control Conference, pp. 8179-8184, Dec. 2005.

[6] A. Pinto, L. P. Carloni, A. L. Sangiovanni-Vincentelli, "A Communication Synthesis Infrastructure for Heterogeneous Networked Control Systems and Its Application to Building Automation and Control", Proceedings of the Seventh International Conference on Embedded Software (EMSOFT), October, 2007.

[7] W. Kastner and G. Neugschwandtner and S. Soucek and H. Michael Newman, "Communication systems for building automation and control", Proceedings of the IEEE, pp. 1178–1203, June 2005.

[8] LonWorks Core Technology (http://www.echelon.com/developers/lonworks/default.htm).

[9] Steven T. Bushby, "BACnet^{TM} - A standard communication infrastructure for intelligent buildings", Automation in Construction, pp. 529–540, vol.6, 1997.

[10] ZigBee Alliance (http://www.zigbee.org).

[11] COSI (http://embedded.eecs.berkeley.edu/cosi/).

[12] ILOG CPLEX Optimization tool (http://www.ilog.com/products/optimization/archive.cfm).

[13] G. L. Stüber, "Principles of Mobile Communication", Kluwer Academic Publishers, 1996.

[14] J. Jeong, and C. T. Ee, "Forward Error Correction in Sensor Networks", University of California at Berkeley, 2003.

[15] "Tmote Sky Data Sheet", Moteiv, San Francisco, CA, 2006 (http://www.moteiv.com/products/docs/tmote-sky-datasheet.pdf).