Correct-by-Construction and Optimal Network Synthesis for Distributed Control/Embedded Systems

S. Xu and R. Kumar, Dept. of Elec. & Comp. Eng., Iowa State Univ. (ISU), and
A. Pinto, United Technologies Research Center (UTRC)

Abstract— **In this paper we develop a formal approach for the synthesis of a cost-effective and correct-by-construction communication network for distributed control/embedded applications subject to a set of end-to-end communication constraints of latency, bandwidth and error-rate, together with the constraints of the network protocols and the desired geographical placement of the network. We also develop a software platform to implement the proposed approach for network synthesis, and apply it to a practical wireless network synthesis for centralized as well as distributed state estimation in building automation.**
Keywords: network synthesis, ZigBee, Integer Linear Programming, optimization, building automation

Note to Practitioners

Network synthesis begins by specifying a set of point-to-point quality-of-service requirements of latency, throughput and error-rate. For a ZigBee based wireless network (one of the popular networks), we present a mathematical approach to its synthesis for a given set of service specifications along with the topographical information about the location of network placement. The approach formalizes the synthesis problem that can be adopted for other types of networks, and guarantees its correctness as well as optimality. The approach has been illustrated by an application to building automation.

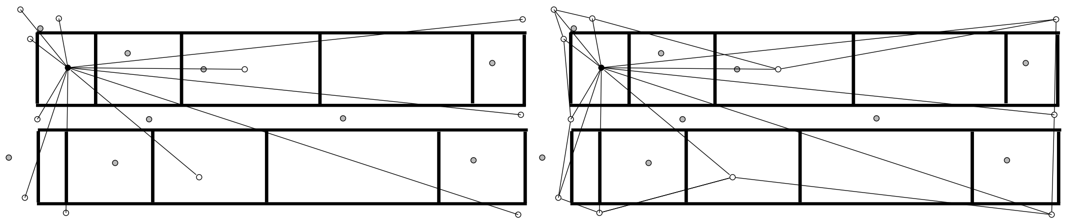
I. Introduction

A distributed cyberphysical system, consisting of a distributed physical system (plant) and a distributed cybersystem (computing system), requires a communication network to share information for the application at hand. Considering the application of a distributed control/embedded system as an example, there exists flow of information among physically distributed sensors, controllers and actuators. Compared to the conventional point-to-point communication based control architecture, the introduction of a communication network in such a control loop reduces the setup and the operating costs, and adds to the fault-tolerance of communication. Owing to these attractive features, network-based distributed control/embedded systems have found a wide spread adoption in industrial, military and commercial applications such as automobiles, aircrafts, chemical plants, power systems, building automation systems, and multi-robotic systems [17].

The communication network, which acts as a backbone for a distributed control/embedded system, can be wireless, wired or a mix of the two (a distinction of the physical layer), and can schedule messages in a time-triggered or an event-triggered fashion (a distinction at the medium access layer) [9], [6]. Certain applications such as HVAC (heating, ventilating and air-conditioning control) for building automation consist mainly of discrete-time control and generate periodic traffic through periodic sampling of the sensors. For such applications, time-triggered medium-access is a natural choice. (Probably the most important reason to achieve synchronization in a wireless network is energy efficiency: If nodes are synchronized, then they know when to sleep/wake up.) In this paper, we study the synthesis of a wireless network with time-triggered medium-access, in particular, the ZigBee network in its beacon-enabled mode.

The control application requirements of sampling rate and measurement accuracy (which depends on quantization-accuracy as well as channel-reliability), impose certain end-to-end constraints on the communication network. For example, [16] shows that a sufficient condition for achieving stabilizability for a linear system, under control over a communication channel, is that the packet-size times the success-rate (which is the capacity of a lossy-channel) exceeds the sum of the logarithms of the magnitudes of the unstable eigenvalues. This puts a lower bound on quantization-accuracy (packet-size) and channel reliability (packet error rate). These end-to-end constraints include maximum latency (which must not exceed the sampling-period), minimum packet-size (to ensure a minimum quantization accuracy), and a maximum error-rate (to ensure a maximum signal-distortion). (Note latency and packet-size

The research was supported in part by the National Science Foundation under the grants NSF-ECS-0601570, NSF-ECCS-0801763, NSFCCF-0811541, and NSF-ECCS-0926029.



(a) Centralized case

(b) Distributed case

Fig. 1. Application of network synthesis.

can equivalently be specified as latency and bandwidth, and we write the two types of specifications interchangeably.) Furthermore, additional constraints arise due to a desired geographical placement of the network (router and link locations), and also other performance constraints may exist (such as maximum utilization). Since the cost of a communication network constitutes a large portion of the overall distributed control/embedded system, care needs be taken in optimizing the network setup and operating cost subject to the aforementioned constraints.

With the complexity of distributed control/embedded systems continuing to evolve, synthesizing an application-specific cost-optimal network subject to certain constraints, based on selected network protocol standards and hardware components that are compliant with these protocols, has become more challenging. Heuristic and experience-based approaches are no longer viable. An automated network synthesis methodology guaranteeing correctness by construction, and the corresponding software platform for the synthesis of a low-cost network to achieve the given quality of service specifications is highly desired.

Motivated by this, in our previous works [13], [14], we have been developing an approach for cost-effective and correct-by-construction communication network synthesis for distributed control/embedded systems to meet the end-to-end specifications of latency, bandwidth, and error-rate. Our approach is based on mathematical optimization: The network synthesis issues of router-placement, connection-routing and router/connection scheduling are formulated as an instance of an Integer Linear Programming (ILP) problem. The scheduling here plays a dual role: It helps to synchronize the nodes—Beacon-synchronization is an active area of research, and it implements the precedence constraints among the nodes introduced by the routing decisions. In [13], the scheduling of routers/connections is assumed given. However since the routes for the desired connections introduce precedence constraints among the routers, the scheduling of routers/connections is very much required to resolve which router will occupy what time-slots, and for which connections. In [14], we extended our work in [13] by including scheduling of connections as part of the network synthesis problem. In this article, we refine the scheduling constraints of [14] so as to accurately capture the restrictions of the wireless network protocol (ZigBee in the present case). Binary decision variables are introduced to resolve which router and which of the connections will occupy what macroslot and which connections will occupy the same minislot. An objective function is formulated to optimize the network installation and operational costs subject to the placement, routing, scheduling, protocol-format and quality-of-service constraints.

In order to illustrate our approach, we apply it to a practical wireless network synthesis problem for centralized as well as distributed state estimation in building automation as shown in Figure 1. For the centralized state estimation, the measurements are sent to a central gateway to compute the estimates of the system state, whereas for the distributed state estimation, the distributed state estimates are computed based on the measurements reported from the neighboring nodes. In the case of centralized state estimation, a wireless network is required to support 11 connections among various sensors and a centralized gateway, whereas 22 such connections are required to be supported in the distributed setting. In each case, there is a total of 9 candidate router locations to choose from, and the beacon-enabled ZigBee protocol in tree-topology is used for networking. The main contribution of our work is we formalize the router-placement, connections-routing, and router/connections-scheduling problem for a beacon-enabled ZigBee network in tree-topology as an *integrated* integer-programming problem so that all protocol/topology constraints as well as all quality-of-service constraints are formally captured. For our application, each connection has a latency requirement of 1.5 sec for the centralized state estimation and 1 sec for the distributed setting, packet-size requirement of 64 bits, and error-rate requirement of 10^{-4}. Such requirements are typical of building automation since there the time scales are slower. Yet the nature of the network synthesis problem is the same (the correctness and optimality requirements are not bounded by the scaling of the time), and the chosen application serves to illustrate our theory to the practical size problems. More details of the application are given in Section IV.

The rest of the paper is organized as follows. Section II overviews the ZigBee protocol. Section III presents the formulation of network synthesis problem. Section IV presents the application of the practical setting of centralized as well as distributed state-estimation in building automation. Section V discusses the software platform to implement the proposed approach for network synthesis as well as the corresponding simulation results. Section VI concludes the work.

II. Overview of ZigBee Network

For our application we use the beacon-enabled ZigBee network as the networking platform. ZigBee provides a low-power, low-cost, wireless network solution, and has gained considerable acceptance within industry. We first overview ZigBee protocol to facilitate the understanding of the formalization of the restrictions of ZigBee protocol on medium-access, routing and scheduling. The readers are referred to [3], [2] for further details about ZigBee/IEEE 802.15.4 specifications and standards.

A ZigBee network consists of three types of devices: (i) ZigBee coordinator, also referred to as PAN (Personal Area Network) Coordinator, (ii) ZigBee router, and (iii) ZigBee end device. ZigBee coordinators and ZigBee routers are full-function devices, allowing the association of other devices. In contrast, a ZigBee end device is a reduced-function device which does not allow such association. In each ZigBee PAN, there is a unique ZigBee coordinator. A ZigBee coordinator initializes network formation and acts as IEEE 802.15.4 coordinator and also as a ZigBee router once the network is formed. A ZigBee router discovers and associates with the ZigBee coordinator or another ZigBee router which has already been associated with the PAN. The former node is called the child node of the latter node, which is called its parent node. A ZigBee router also manages the local address allocation/deallocation besides participating in routing. A ZigBee end device, having no routing functionality, associates with the ZigBee coordinator or a ZigBee router.

The protocol stack of a ZigBee network is composed of the physical and Medium Access Control (MAC) Layers described in the IEEE 802.15.4 standard, and network and application layers defined by the ZigBee Alliance. At the physical layer, the IEEE 802.15.4 standard offers a total of 27 channels, with a peak data-rate of 250Kbits/s. At the MAC layer, nodes are grouped into PANs. A PAN is started by a node, which assumes the role of PAN Coordinator. IEEE 802.15.4 MAC supports two operation modes: (i) beacon-enabled mode, and (ii) non beacon-enabled mode. In a beacon-enabled mode, beacons are periodically broadcasted by routers to their children for synchronization and communication, whereas in a non beacon-enabled mode, non-slotted CSMA/CA is adopted for communication.

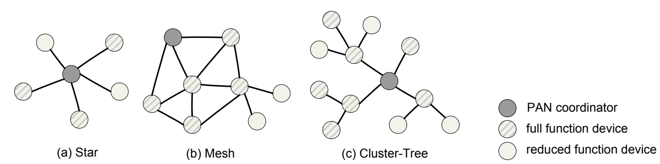


Fig. 2. ZigBee network topology.

Three network topologies are supported by ZigBee: (i) star, (ii) mesh, and (iii) cluster-tree (simplified as tree), as shown in Figure 2. In a star topology, all devices communicate with each other via the ZigBee coordinator, requiring all the nodes to be within the radio range of the ZigBee coordinator. In a mesh topology, peer-to-peer communication among devices is allowed: the nodes within each other's radio range can directly communicate. In a mesh topology, traffic is distributed over a typically large number of sensors and traffic locality is exploited. Moreover, the existence of multiple paths between the same pair of nodes improves reliability through path diversity. In comparison, the central node of a star topology network needs to always relay packets from all other nodes, thereby becoming a single point of failure together with high power consumption and, therefore, a shorter life.

To increase the lifetime of a wireless sensor network, sleeping disciplines are used [18], [8]. During a sleep cycle, a node is (almost completely) powered off. To implement these power saving techniques efficiently, nodes need to be synchronized. It is difficult to achieve synchronization in a mesh topology which is asynchronous in nature (since no beacons are used). Cluster-tree topology is a special mesh topology in which any pair of nodes can only be connected by a single routing path. The allowance of beacon-enabled communications within the tree topology makes possible, synchronization and a contention-free PAN, thereby energy-efficient networking and a precise characterization of delays. We have chosen to focus on the synthesis of a Zigbee

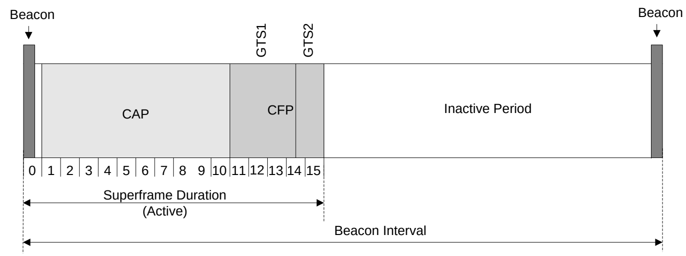


Fig. 3. Superframe structure assignable to a single router.

network in its tree topology.

The simultaneous transmissions in different PANs do not collide since the transmissions take place on different channels. However, intra-PAN transmissions need to be coordinated. To manage medium access, superframe structures consisting of active and inactive periods, as shown in Figure 3, are used. Each router is associated with one or more superframe structures, phased so that the corresponding active periods don't overlap. ZigBee coordinator establishes the values of a set of configuration parameters of the superframe, which have to be adopted by all the nodes associated with the PAN. In a superframe, the time interval between two consecutive beacons is called Beacon Interval (BI) and is defined as $BI = aBaseSuperframeDuration \times 2^{BO}$*symbols*, where BO represents Beacon Order and $aBaseSuperframeDuration$ has a constant duration of 960*symbols* $= 960$*symbols* $\times$ 4*bits/symbol* $\times$ $1/(250K)$*secs/bit* $= 15.36$*msec*, corresponding to the minimum duration of the superframe (case of $BO = 0$). The beacon order BO can range from 0 to 14 ($BO = 15$ means that no beacon will be transmitted, i.e., the non beacon-enabled mode). A superframe structure is composed of an active period, called a macroslot, and an (optional) inactive period. The duration of the active period (macroslot) is determined by the Superframe Duration (SD), defined as $SD = aBaseSuperframeDuration \times 2^{SO}$*symbols*, where the superframe order SO can range from 0 (minimum-sized macroslot) to BO (maximum-sized macroslot and no inactive period). Superframe structures are shifted in phase by multiples of a SD (so their active periods don't overlap) and are individually assignable to the routers. Thus up to $BI/SD = 2^{BO-SO}$ superframe structures with non-overlapping active periods (macroslots) can fit within a BI; the actual number can be smaller which we denote as n_{max} in the paper. Superframe Duration is divided into 16 equal-sized minislots, classified into Contention Access Period (CAP) and Contention Free Period (CFP). During the contention access period, transmissions are governed by a slotted CSMA/CA based medium access, and thus collisions can occur. A minimum length (440 symbols) of the contention access period has to be reserved for the transmission of the management frames. We let the contention free period to span the entire macroslot, and in which case it can consist of up to 16 Guaranteed Time Slots (GTSs). The actual number of GTSs used, denoted m_{max} in the paper, may be smaller than the maximum allowed. Transmissions in the GTSs are uniquely allocated to devices, and concurrent transmissions by the devices in the same PAN are forbidden. During the inactive period, nodes can put their transceivers in the sleep modes to 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. Beacon-scheduling is an active area of research [11], [7], [19], and our scheduling algorithm automatically ensures non-overlapping scheduling, complementing the above works.

III. Network Synthesis with Scheduling

In this section, we formulate the network synthesis problem, specialized to the case of beacon-enabled ZigBee networks, as an instance of an Integer Linear Programming problem. To present the ILP formulation, we need the following notions.

A. Network Synthesis Parameters

The nodes/links and the connections requirements of a communication network can be specified as a directed graph (N, C) consisting of a set of *nodes* N and a set of *connections* $C \subseteq N^2$. N is further partitioned into $E \cup R$, the set of end devices E and candidate router locations R. The set of end devices E consists of the sets of source locations S and destination locations D. Sources generate messages, destinations consume messages, and routers transport messages. The network is synthesized to serve a set of connections, i.e., a set of source-destination pairs $C \subseteq S \times D \subseteq N^2$. Each connection $c \in C$ is labeled with a quality-of-service requirement which is a 3-tuple (l_c, b_c, p_c): l_c is the maximum latency (which must not exceed the sampling-period), b_c is the number of bits per message (to ensure a minimum quantization accuracy), and p_c is the maximum packet error rate probability (to ensure a maximum signal-distortion). Note latency l_c (with the unit of seconds/message) together with b_c (with the unit of number of bits/message) implicitly specifies the minimum required bandwidth b_c/l_c (with the unit of bits/second).

Given a pair of nodes i and j, the packet error rate probability, denoted by $p(i,j)$, can be computed as follows. Suppose the node i transmits packets with a radio power level P_i. Let the distance between node i and j be denoted by $d_{i,j}$. We denote the path loss attenuation between the transmitter and the receiver by $\mathrm{PL}(d_{i,j})_{\,dB}$. For example, for the Telos Sky wireless sensors [4], the following generic yet representative model of the path-loss can be used [15]:

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

where $\mathrm{PL}(d_0)$ denotes the path loss computed at a reference distance d_0, β denotes the path loss exponent, $\Omega_{i,j}$ denotes the shadowing attenuation modeled as a Gaussian random variable having zero average and variance $\sigma_{i,j}^2$, and PL_{mw} is the path-loss due to multiple-walls. We adopt a multi-wall model [5] to account for the path loss due to the presence of walls between a transmitter and a receiver: $\mathrm{PL}_{mw} = L + n_w L_w$, where L 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 a constant depending on the thickness of the wall.

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

$$10 \log_{10} SINR_{i,j} = 10 \log_{10} P_{i,j} - P_{n\ dB},$$

where $P_{i,j}$ is the radio power received at the node j from the node i, and P_n summarizes the thermal noise and the power of the interference coming from co-channel radios. We make the assumption that nodes are not simultaneously transmitting (i.e the network operates in beacon-enabled mode), so the formula for $P_{i,j}$ is given by:

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

whereas P_n is simply a constant N_0 representing the power of thermal noise. A typical value for the power of the thermal noise for the Telos Sky receivers is $N_0 = -170\mathrm{dBm}$.

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

$$p_b(SINR_{ij}) \triangleq f_1(SINR_{ij})$$

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 time, can be expressed by [15]:

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

Assume that a packet at the data-link layer is composed of O bits of protocol overhead and a payload of b_i bits and the CRC code is always able to detect erroneous packets (see [10] for an experimental support). Then the packet error rate probability, without any retransmission mechanism, can be modeled by:

$$p(i,j) \triangleq f_2(SINR_{ij}) = 1 - [1 - p_b(SINR_{ij})]^{O+b_i}\ .$$

The optimization objective for a network is defined as its installation and operation costs: The router installation cost for node i is c_i, whereas the operational cost per connection c, per link it uses, is $w_c = (e_t + e_r)(b_c + O)(T_{life}/BI)(c_B/e_B)$, where $e_t + e_r$ is the energy consumed per bit by transmitter-receiver pair, $b_c + O$ is total number of data and overhead bits for connection c, T_{life}/BI is the total number of rounds of communication in network's life assuming each round fits within one BI, and c_B/e_B is the cost of battery per unit energy stored in the battery. For our application we choose $c_i = \$7002, \forall i$, $e_t = 236nJ$, $e_r = 132nJ$, $e_B = 30KJ$, $c_B = \$40$, $T_{life} = 20\mathit{years}$, and $BI = 15.36\mathit{msec} \times 2^{BO} = 15.36\mathit{msec} \times 32 = 0.49\mathit{sec}$.

We use B_{max} (b_{max}) to denote the bit capacity of a macro- (mini-) slot.

B. Network Synthesis Formulated as ILP

In order to synthesize a cost-optimal network to offer the desired QoS for each connection, we need to make the following three decisions: (i) placement of routers, (ii) routing of connections, and (iii) scheduling of routers/connections. We discuss these decision issues in further detail below.

Router-placement. The locations of sources and destinations are given as part of the connections specification C, whereas the placement of routers to support the connections needs to be determined. For a practical application such as building automation, the number of candidate locations can be taken to be finite: Based on the communication range of routers, the area required to be connected can be divided into a finite number of zones, where each zone can have at most one router. A binary variable x_i is associated with each candidate location, which equals one if and only if a router is installed at location i. The objective is to place as few and cost-effective routers as necessary.

Connection-routing. For each desired connection (source-destination pair), a routing path from source to destination needs to be determined. For this, we need to determine which pairs of nodes will have active links between them. A binary variable ℓ_{ij} is associated with a pair of nodes, which equals one if and only if a link is installed between i and j, and i is j's parent. A node-link incidence matrix I consisting of elements of 1, -1, and 0 is constructed, where an entry $(k,(ij))$ equals 1 (resp., -1) if node k is a source (resp., destination) of link ij, and 0 otherwise. Then a route from source s_c to destination d_c of a connection c can be obtained as a solution of a classical balance equation involving the node-link incidence matrix and binary variables y_{ijc} which equals 1 if and only if link ij is used in the route for connection $c \in C$. Let y_c be a vector of size $|N|^2$ obtained by stacking the entries y_{ijc}, and $\mathbf{b}_c$ be a vector of size $|N|$ such that $\mathbf{b}_c(i) = 1$ for $i = s_c \in S$, $\mathbf{b}_c(i) = -1$ for $i = d_c \in D$ and $b_c(i) = 0$ otherwise. Then a solution of $Iy_c = \mathbf{b}_c$ provides the values of the decision variables y_{ijc} that ensure the existence of a route (a sequence of links) for a connection $c \in C$. In addition, the routes for the connections must be chosen in such a way that the aggregate error-rate across a route is below the required error-rate of a connection using that route.

Router/Connection-scheduling. In a time-triggered setting, the sources (sensors or processors) generate periodic messages, and so the communication from sources to destinations occurs in rounds. Accordingly, the time-line is divided into beacon-intervals, and all communications of a round must fit within a number of beacon-intervals. Each connection is broken down into a sequence of links according to the routing decision as above, and accordingly each link carries traffic for a certain number of connections. Each beacon-interval is further divided into macroslots, and each macroslot is assigned to a router so that a subset of its children can communicate data for a subset of connections routed through the links to those children. A router may be assigned multiple macroslots to allow communication of the entire set of connections across the entire set of its children. The reason that all such data may not be routed within a single macroslot is that each connection-route introduces a certain precedence constraint among the nodes of the route, and all different precedence constraints of all the connections must be respected. Moreover the limitation on the bit capacity of a macroslot as well as of a minislot, and also the limit on the number of minislots must also be respected.

Binary decision variable g_{ijsc} (resp., g'_{ijsc}) is associated with each router i, each node j, and each time-slot s for connection c, which equals 1 if and only if router i sends (resp., receives) data to (resp., from) node j in macroslot s for connection c. These scheduling variables are constrained to capture the precedence constraints of the routes, the capacity and the number constraints of the macro- and mini-slots. Further the placement, routing, and scheduling variables are constrained together to ensure the requirements of latency and data-sizes.

To summarize, the following decision variables and parameters are used in our ILP formulation of the network synthesis problem.

- Binary decision variables:
  - x_i: 1 if and only if a device (end device/router) is installed at location i.
  - ℓ_{ij}: 1 if and only if a link between node i and j is installed, and i is j's parent.
  - y_{ijc}: 1 if and only if the route for connection c uses a link from node i to node j.
  - z_{isc}: 1 if and only if node i is assigned macroslot s for connection c.
  - g_{ijsc}/g'_{ijsc}: 1 if and only if node i sends/receives data to/from j in macroslot s for connection c (a derived variable that depends on z_{isc} and y_{ijc}).
  - g_{ijs}/g'_{ijs}: 1 if and only if node i sends/receives data to/from j in macroslot s (a derived variable obtained as projection of g_{ijsc}/g'_{ijsc}).
- Network parameters:
  - E: the set of end devices
  - R: the set of routers
  - N: the set of nodes, where $N = E \cup R$
  - C: the set of connections
  - BO: beacon order
  - SO: superframe order
  - b_{max}: the bit capacity of a minislot
  - B_{max}: the bit capacity of a macroslot
  - n_{max}: the number of macroslots in one beacon

interval, where $n_{max} \leq 2^{BO-SO}$

  - m_{max}: the number of minislots in a macroslot, where $m_{max} \leq 7$
  - $p(i,j)$: the packet error rate probability of link ij

- Specification parameters:
  - c_i: the location-i router installation cost
  - w_c: the connection-c operation cost per link
  - b_c: the number of bits per message of connection c
  - p_c: the maximum packet error rate probability of connection c
  - l_c: the maximum latency of connection c

The proposed ILP formulation for the network synthesis problem is as follows.

$P:\quad \min\ \sum_{i\in R} c_i x_i + \sum_{i,j\in N}\sum_{c\in C} w_c y_{ijc}$

$s.t.$

1. $x_i = 1,\ \forall i \in E$
2. $\ell_{ij} + \ell_{ji} \leq 1,\ \forall i,j \in N$
3. $\ell_{ij} = 0,\ \forall i \in E, j \in N$
4. $x_i + x_j \geq 2\ell_{ij},\ \forall i,j \in N$
5. $\sum_{i\in N} \ell_{ij} \leq 1,\ \forall j \in N$
6. $\sum_{j} \ell_{ij} \leq m_{max},\ \forall i,j \in N$
7. $\ell_{ij} + \ell_{ji} \geq y_{ijc},\ \forall i,j \in N, c \in C$
8. $\ell_{ij} \leq \sum_{c\in C}(y_{ijc} + y_{jic}),\ \forall i,j \in N, c \in C$
9. $Iy_c = \mathbf{b}_c,\ \forall c \in C$
10. $\sum_{s=1}^{n_{max}} z_{isc} \geq y_{ijc} + \ell_{ij} - 1, \forall i \in R, j \in N, c \in C$
    $\sum_{s=1}^{n_{max}} z_{isc} \geq y_{jic} + \ell_{ij} - 1,$
11. $\sum_{s=1}^{n_{max}} z_{isc} \leq 1, \forall i \in R, c \in C$
12. $z_{isc} + z_{jsc'} \leq 1, \forall i \neq j \in R, s \in [1, n_{max}], c, c' \in C$
13. $\sum_{s=1}^{p} z_{isc} - \sum_{s=1}^{p} z_{jsc} \geq y_{ijc} + \sum_{s=1}^{n_{max}} z_{isc} + l_{ji} - 3,$
    $\forall c \in C, i,j \in R, p \in [1, n_{max}]$
    $\sum_{s=1}^{p} z_{isc} - \sum_{s=1}^{p} z_{jsc} \geq y_{ijc} + \sum_{s=1}^{n_{max}} z_{jsc} + l_{ij} - 3,$
14. $\sum_{i,j\in N} y_{ijc} \log(1 - p(i,j)) \geq \log(1 - p_c), \forall c \in C$
15. $z_{isc} s \leq \min\{n_{max}, \lfloor l_c/SD \rfloor\},$
    $\forall i \in R, s \in [1, n_{max}], c \in C$
16. $g_{ijsc} \geq y_{ijc} + \ell_{ij} + z_{isc} - 2,$
    $g'_{ijsc} \geq y_{jic} + \ell_{ij} + z_{isc} - 2,$
    $\forall i \in R, j \in N, s \in [1, n_{max}], c \in C$
17. $\sum_c g_{ijsc}(b_c + O) \leq b_{max}, \forall i \in R, j \in N, s \in [1, n_{max}]$
    $\sum_c g'_{ijsc}(b_c + O) \leq b_{max},$
18. $g_{ijs} \geq g_{ijsc}, \forall i \in R, j \in N, s \in [1, n_{max}], c \in C$
    $g'_{ijs} \geq g'_{ijsc},$
19. $\sum_j g_{ijs} + \sum_j g'_{ijs} \leq m_{max}, \forall i \in R, j \in N,$
    $s \in [1, n_{max}]$

The meaning of the cost function was presented in the previous subsection, and the meaning of the constraints is as follows: Constraint 1: End device nodes must be sited. Constraint 2: Parent-child relation is asymmetric. Constraint 3: End devices cannot be parents. Constraint 4: If i is the parent of j, then i and j are both sited. Constraint 5: A node has at most one parent. Constraint 6: Each node has at most m_{max} children (limited by the number of guaranteed minislots per macroslot as well as by selecting a ZigBee parameter). Constraint 7: Connection c is routed via node i to j, then either i is the parent or the child of j. Constraint 8: If i is the parent of j, then some connection is routed between i and j. Constraint 9: Each connection is routed, i.e., the balance equations hold. Constraint 10: If j receives/sends data from/to its parent i for connection c, then i gets a macroslot for c. Constraint 11: For each connection, each node is assigned at most one macroslot. Constraint 12: Each macroslot is assigned to at most one node. Constraint 13: If i sends data to its parent/child j for connection c, which it receives/sends from/to its child/parent, then i's macroslot for c precedes j's macroslot for c. Constraint 14: Success rate for a connection exceeds its specification. Constraint 15: Latency requirement is met for each connection. Constraint 16: Defines auxiliary variables g_{ijsc}/g'_{ijsc}—If parent i sends/receives data to/from node j for connection c, and has macroslot s, then $g_{ijsc}/g'_{ijsc} = 1$. Constraint 17: Data sent/received by node i to/from node j in a common macroslot s must fit the bit capacity of the minislot. Constraint 18: Defines auxiliary variables g_{ijs}/g'_{ijs}—If parent i sends/receives data to/from node j, and has macroslot s, then $g_{ijs}/g'_{ijs} = 1$. Constraint 19: Number of communications between parent i and its children in any single macroslot cannot exceed the number of guaranteed minislots m_{max}.

Certain constraints require additional discussion, which we present next. In a ZigBee network of tree-topology, communications in the beacon enabled mode are performed between the parent nodes and their children in the time-slots assigned to the parent nodes. Therefore if a link between node i and j is used by a connection and i is the parent of j, then certain macroslot(s) should be assigned to i so as to allow i to communicate with j. This is captured by Constraint 10. Constraint 13 enforces the precedence among the routers for each connection. Here two cases, as shown in Figure 4, are considered: router i must be scheduled before router j if a communication from i to j is needed by a connection, and either (i) i needs to collect data from a child before forwarding to its parent j (Figure 4 (a)), or (ii) j needs to forward data collected from its parent i to one of its children (Figure 4 (b)).

Remark 1: Although the above formulation considers the case when a round of communication fits a single beacon interval, there is nothing in the formulation that

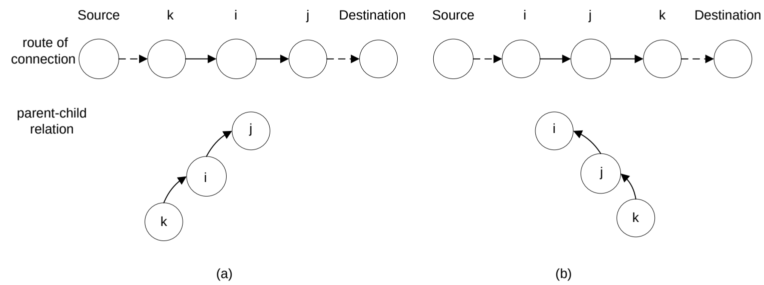


Fig. 4. Two cases considered for modeling precedence constraints.

limits it to a single beacon interval: Simply replace $n_{\max}$ with $n_{max}N$, where N denotes the number of beacon intervals needed to fit one round of communication. The value of N can be found through a geometric search that tries the increasing powers of 2 as the choice for N.

IV. Application to Building Automation

The proposed approach is demonstrated through estimation applications for building automation. Estimation of certain quantities, such as occupancy is very important for an effective HVAC (heating, ventilating, and air-conditioning) control. The state of the system needs to be estimated from sensor measurements so that cooling and heating requirements can be determined by the control algorithm. The performance of the closed loop system depends on the quality of service provided by the network that interconnects sensors, processing boards and actuators.

Given a certain required control performance (such as stability, reaction time), quality of service requirements can be computed in terms of maximum delay, maximum error-rate and bandwidth. Note there is a tradeoff to explore at this level. For example, the time scales of many building automation applications are slow, which implies that the bandwidth and delay requirements are not stringent. However, when data are sent rarely, it is important to deliver the data reliably. On the other hand, one could relax reliability requirements and at the same time increase network bandwidth so as to send packets more frequently. Note this relies on time redundancy. Furthermore, in addition to the performance constraints, a network is also subject to the constraints arising due to a building layout such as the possible locations of nodes (sensors, actuators, and routers).

The centralized state estimation dynamics can be described by the following set of equations:

$$x_{k+1} = f(x_k, u_k, w_k); \quad y_k = h(x_k, u_k, v_k).$$

Here u_k represents measure from the sensors, w_k and v_k denote noise components, x_k is the state of the estimator, 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 outputs y_k are sent to the actuators. Communications between the sensors and the gateway are represented by connections annotated with the desired QoS, that guarantee the required estimation accuracy.

In the distributed estimation case, the estimation dynamics can be described as:

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

Here i denotes the index in space and Γ_i is a set of spatial indices representing the neighborhood set of i. In the case of a geographical one-dimensional neighborhood, the set Γ_i contains only its nearest neighbors $\Gamma_i = [i, i+1, i-1]$, whereas in a general case the set Γ_i can contain the elements which are not necessarily geographically close in space.

Figure 1 illustrates the input of the network synthesis (1(a) and 1(b) for centralized and distributed state estimation respectively), together with the floor-plan of a building and the locations of sensors (there are no actuators because of no control, only estimation). In Figure 1, the white nodes are the sensor locations, the black solid node is the estimation point, and the remaining gray nodes are the locations where routers can be installed. The edges represent the required connections (source/destination pairs). In the centralized case, the number of connections required to be supported is 11, whereas the number is 22 for the decentralized case. In each case, there is a total of 9 router locations to choose from.

Assuming the estimation and control algorithm can tolerate a maximum delay T_{max}, the latency requirements for each connection is formulated as the end-to-end maximum delay. Additionally, the amount of data, the required redundancy, and the maximum allowable signal distortion can be translated into a minimum band-

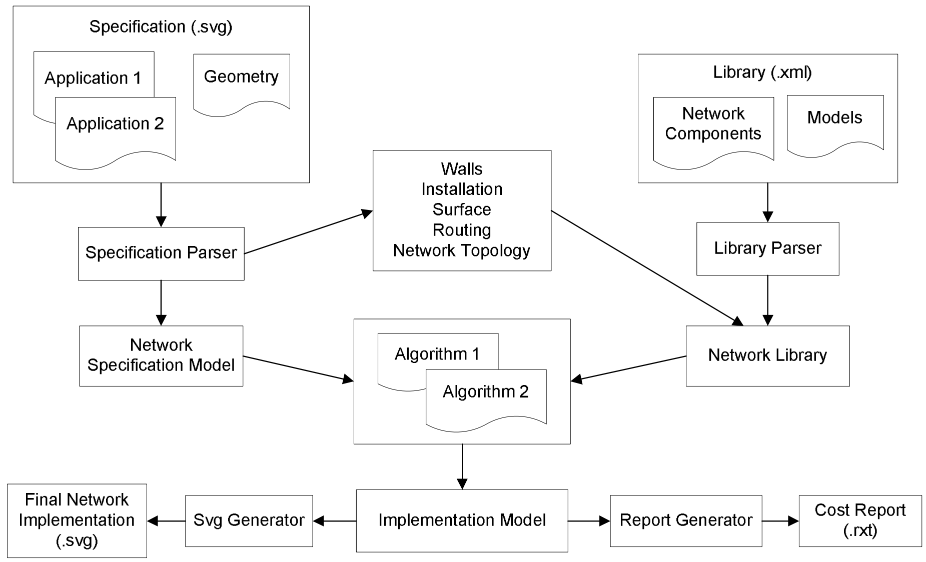


Fig. 5. A software platform for automatic network synthesis flow.

width and maximum packet error rate requirements. For the centralized case $T_{max} = 1.5s$, whereas $T_{max} = 1s$ for the decentralized case. The packet size is 64 bits and the packet error rate probability for each case is 10^{-4}. Then subject to the set of end-to-end latency, bandwidth and error-rate constraints for the desired set of connections, the building geometry, and cost characterization, a network synthesis (router-placement, connection-routing, and router/connection scheduling), satisfying the constraints and optimality with respect to installation and operating cost, can be performed by the proposed automated network synthesis approach.

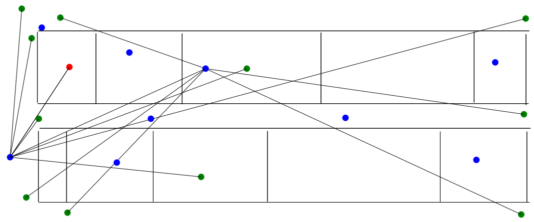
V. Software Platform and Simulation Results

We have developed a software platform to implement the proposed network synthesis approach building on the Communication Synthesis Infrastructure [12] framework. Keeping the building automation application in mind, the software takes a graphical description of building floor-plan and connections layout in SVG format showing the walls, the desired connections, and the candidate router positions. The SVG description is parsed into an internal representation that includes a data structure to represent a network of components and a data structure to represent the building geometry. The building geometry is stored as a set of walls that are represented as surfaces in a three-dimensional space with an associated property of thickness. This representation is used to compute the number of walls that can be traversed by the line of sight between two nodes in the network. This information is then used by the multi-wall model presented in Section III-A. The description of the available nodes to build the network, their properties and associated models is captured in a separate XML file.

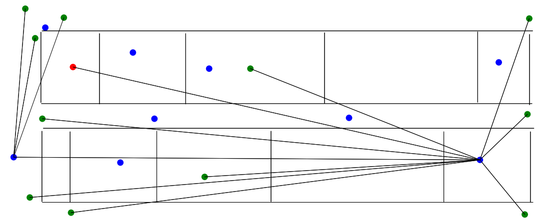
Given the specifications captured in the SVG and the XML files, the algorithm module, written in C++, encodes the network synthesis problem as an instance of an Integer Linear Programming optimization problem. We resolve the proposed ILP optimization formulation by using an open source solver SAT4J [1]. (Note the proposed ILP formulation consists of binary variables and thus can be resolved using a pseudo boolean solver such as PBSolver provided by SAT4J.) Based on the solution of the ILP formulation, a graphical representation of a synthesized network in SVG format and a textual report of its cost is generated. The corresponding software platform is shown in Figure 5.

Remark 2: The developed software platform allows designers, modelers and component developers of distributed embedded systems to collaborate by using the same data model and interfaces. The library of components can be generalized to include wired and wireless network components.

For our application, in the case of centralized estimation, each sensor sends/receives packets to/from the central gateway. We assume a maximum delay for each packet of $1.5sec$, a packet length of 64 bits and an overhead of 176 bits, and a maximum probability of error to be 10^{-4}. Based on the simulation results (see



(a) Centralized



(b) Distributed

Fig. 6. Synthesized networks for centralized and distributed state estimation.

Remark 3 and Table I), we pick $SO = 3$, $BO = 5$, $n_{max} = 2^{BO-SO} = 4$, $m_{max} = 7$, and so $B_{max} = 960$*symbols* $\times$ 4*bits/symbol* $\times$ $2^{SO} = 31720$*bits*, and $b_{max} = B_{max}/16 = 1920$*bits*. With the above selected parameters, all the connections can be correctly routed and scheduled within one beacon interval, and the corresponding synthesized network is shown in Figure 6(a) that installs 2 routers and 13 links. It has the cost of \$15994 (considering $c_i = \$7002, \forall i$, $e_r = 132nJ$, $e_t = 236nJ$, $e_B = 30KJ$, $c_B = \$40$, $T_{life} = 20$*years*, and $BI = 0.49$*sec*).

In the distributed estimation case, the communication among the neighbors happens with a period of $1sec$ which is also their maximum delay. Communication with the central estimation point happens every $2sec$, which is the corresponding maximum delay. We pick $m_{max} = 10$ (the other parameters remain the same as in the centralized case) by taking into account the tradeoff between performance and computation (see Remark 3 and Table I). The synthesized network is shown in Figure 6(b) that installs 2 routers and 13 links, and its cost is \$18396. Note a higher cost compared to the centralized case is expected since a double number of connections need be supported.

Remark 3: Care should be taken while picking the values for n_{max} and m_{max} for a given set of connections specification: A large m_{max} trivializes the problem since small number of routers suffice as each router can have large number of children, while a small m_{max} increases computation burden as combination of multiple routers must be explored which increases complexity for routing and scheduling. Similarly, a large n_{max} makes computation for optimization more involved, while a small n_{max} may cause the problem infeasible. The optimization results for various choices for n_{max} and m_{max} are as shown in Table I, in which "$-$" denotes "no solution obtained" (since SAT4J is unable to decide whether there exists a solution to the given ILP formulation).

VI. Conclusion

Distributed control/embedded systems are gaining a wide spread acceptance in industry owing to the cost and fault-tolerance advantages, replacing the point-to-point communication networks. The complexity of distributed control/embedded systems continues to rise due to a larger number of nodes that must be interconnected. Therefore it is greatly desirable that automated methods be developed for the synthesis of embedded networks that are cost-effective and correct-by-construction. Here cost is that of installation as well as operation, and correctness is to ensure that all connections are routed and receive the desired quality of service for latency, throughput and error rate. The synthesis problem is formalized as an integer linear program to capture these requirements, the constraints of the ZigBee protocol as well as the geographical placement of the network. Decisions are made for router placement, connection routing and router/connection scheduling. We also applied the proposed approach to the synthesis of wireless networks for centralized and distributed state estimation in building automation, and presented a software implementation as well as simulation results for a practical sized building automation network. Future research will consider synthesis for heterogeneous networks as well as strategies for computationally efficient near-optimal solutions.

References

[1] http://www.sat4j.org.

[2] IEEE 802.15.4 standard. http://standards.ieee.org/getieee802/download/802.15.4-2006.pdf.

[3] Zigbee alliance. http://www.zigbee.org.

[4] Tmote Sky Data Sheet. http://www.moteiv.com/products/docs/tmote-sky-datasheet.pdf, 2006.

[5] European COST Action 231. Digital mobile radio towards future generation systems (COST 231 Final Report). In *Available at http://www.lx.it.pt/cost231*, 1999.

[6] B. Brahimi, E. Rondeau, and C. Aubrun. Comparison between networked control system behavior based can and switched ethernet networks. In *2nd NeCST Workshop on Networked Control Systems and Fault Tolerant Control*, Rende, Italy, 2006.

TABLE I

SIMULATION RESULTS FOR VARIOUS n_{max}, m_{max}.

$n_{max} = 8$	Centralized			Decentralized		
	$m_{max} = 15$	$m_{max} = 10$	$m_{max} = 7$	$m_{max} = 15$	$m_{max} = 10$	$m_{max} = 7$
number of variables	40413	40413	40413	80877	80877	80877
number of constraints	98543	98543	98543	221459	221459	221459
cost ($)	4371	8006	-	5271	9199	-
running time (s)	10	13936	3205	35	21305	3583
number of routers installed	1	2	-	1	2	-

$n_{max} = 4$	Centralized			Decentralized		
	$m_{max} = 15$	$m_{max} = 10$	$m_{max} = 7$	$m_{max} = 15$	$m_{max} = 10$	$m_{max} = 7$
number of variables	22737	22737	22737	45489	45489	45489
number of constraints	54299	54299	54299	120263	120263	120263
cost ($)	8623	15772	15994	10422	18396	-
running time (s)	6	12875	14807	21	20396	5347
number of routers installed	1	2	2	1	2	-

[7] J. Cho and S. An. An adaptive beacon scheduling mechanism using power control in cluster-tree wpans. *Wireless Personal Communications: An International Journal*, 50(2):143–160, 2009.

[8] T.van Dam and K. Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. In *SENSYS*, 2003.

[9] K. Etschberger, R. Hofmann, J. Stolerg, C. Schlegel, and S. Weiher. *Controller Area Network: basics, protocols, chips and applications*. Weingarten: IXXAT Press, 2001.

[10] J. Jeong and C.T. Ee. Forward error correction in sensor networks. In *University of Berkeley*, 2003.

[11] A. Kouba, A.Cunha, M. Alves, and E. Tovar. Tdbs: a time division beacon scheduling mechanism for zigbee cluster-tree wireless sensor networks. *Real-Time Systems*, 40(3):321–354, 2007.

[12] A. Pinto, L. Carloni, and A.L. Sangiovanni-Vincentelli. A communication synthesis infrastructure for heterogeneous networked control systems and its application to building automation and control. In *Proceedings of the Seventh International Conference on Embedded Software (EMSOFT'07)*, Salzburg, Austria, 2007.

[13] A. Pinto, M. D'Angelo, C. Fischione, E. Scholte, and Al. Sangiovanni-Vincentelli. Synthesis of embedded networks for building automation and control. In *Proceedings of American Control Conference (ACC08)*, Seattle, Washington, 2008.

[14] A. Pinto, R. Kumar, and S. Xu. Synthesis of wireless time-triggered embedded networks for networked control systems. In *Proceedings of IEEE Conf. on Automation Science and Engineering*, Bangalore, India, 2009.

[15] G. L. Stuber. Principles of mobile communication. Kluwer Academic Publishers, 1996.

[16] S. Tatikonda and S. Mitter. Control over noisy channels. *IEEE Transactions on Automatic Control*, 49(7):1196–1201, 2004.

[17] F. Wang and D. Liu. *Networked Control Systems Theory and Applications*. Springer Verlag, 2008.

[18] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient mac protocol for wireless sensor networks. In *IEEE INFOCOM*, 2002.

[19] L. Yeh, M. Pan, and Y. Tseng. Two-way beacon scheduling in zigbee tree-based wireless sensor networks. In *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC 2008)*, Taiwan, 2008.