ZigBee Cluster Tree Formation For Time-Bounded Data Flows in One Collision Domain

Aasem Ahmad
DCE, FEE, CTU in Prague
Prague, Czech Republic
Email: ahmadaas@fel.cvut.cz

Zdeněk Hanzálek
DCE, FEE, CTU in Prague
Prague, Czech Republic
Email: hanzalek@fel.cvut.cz

***Abstract*—We study one-collision domain ZigBee cluster-tree design problems to satisfy periodic time-bounded data flows. The formation of the cluster-tree topology can be seen as a bounded-degree-and-depth tree which is an NP-complete problem. The objective is to minimize the number of clusters such that all flows can take place and there exists a cluster schedule that meets the deadlines of the flows. For the resulting tree, the cluster schedule is required to be energy efficient, which can be achieved by maximizing the length of the schedule period, and consequently, increasing the lifetime of the network. We present a Cluster-Tree Formation and Energy-Efficient clusters scheduling algorithm, CFEFS, based on the Hungarian algorithm, the Maximum Matching algorithm and the Branch and Bound algorithm to tackle this design problem that integrates the cluster formation and the cluster scheduling in one problem.**

I. Introduction

Wireless sensor network technology has been utilized by industrial monitoring and control systems to improve their functionality and efficiency [1] especially in hard to reach environments [2]. Other requirements, such as timeliness and energy efficiency, are important to be fulfilled.

The IEEE 802.15.4/ZigBee standards [3] are leading technologies for low-cost, low-power and low-rate WSNs. The ZigBee beacon enabled cluster-tree topology (Fig. 1) is suited for the high predictability of performance guarantees [4]. The cluster-tree is formed by the ZigBee coordinator (ZC) which is a router with special functionalities: it is the root of the tree and it initializes the network by setting three important parameters: the maximum number of children of a router (Cm), the maximum number of child routers of a router (Rm), and the depth of the network (Lm). The other devices (i.e. the ZigBee Router (ZR) and the ZigBee End Device (ZED)) join the network by joining with previously joined ZR or with the ZC. Each router forms a cluster and is called the cluster head. Both the ZC and ZR participate in multi-hop routing, which is not the case for the ZED. The joining relation is logically described by the parent-child relation which is performed by a Distributed Address Assignment Mechanism DAAM suggested by ZigBee [3].

In the cluster-tree, the time behavior of each cluster is periodic and the period of each cluster, called the *Beacon Interval* BI, is divided into two portions. The *active* portion, divided into *time slots*, during which the cluster-head enables the transmissions inside its cluster, and the subsequent *inactive* portion to save energy. Hence, each cluster is active only once during the period. The communication between the clusters is given by a set of data flows through the network. Based on the transmission area of the nodes and the topology of the network, collision avoidance is another requirement to be achieved. Thus, the scheduling problem in cluster-tree WSNs is defined by construction of collision free and periodic cluster schedule specifying which cluster will be active at which time and which slot will be assigned to which data flow. When the data flows are time-constrained and the end-to-end delay is longer than the schedule period, the schedule is cyclic [6].

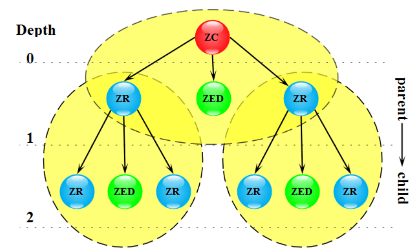

Fig. 1. Cluster-tree topology with (Rm=2, Cm=3, Lm= 2).

A. Problem description

The DAAM joining mechanism is not that efficient [5]. If a node is not able to join the network due to the Cm, Rm, or Lm parameters, then it becomes an orphan node. Although, this mechanism dose not distinguish between candidate parents for the new device that requests joining the network (i.e. simply, the new device joins the parent that have the strongest signal regardless whether the resulting tree meets the various application requirements). So, the particular form of a ZigBee cluster-tree obtained by DAAM may lead to infeasible cluster-schedule when the flows are time-constrained.

In this paper, we omit the DAAM and provide a new joining mechanism driven by the cluster scheduling problem. Thus, we assume a spatially distributed set of ZigBee devices in the sensing area with no parent-child relation between them. We assume As in [5] that the following parameters are given:

- the coordinates of the ZigBee devices and their types.
- the topological parameters (Cm, Rm, Lm).
- the transmission area of all the devices.

We also consider a set of periodic time-constrained flows (each given by parameters such as a source node, sink node and precise end-to-end deadline given in time units). The source and the sink of the flow can be ZC, ZR or ZED. A source node periodically measures a sensed value and reports it to the sink. The req_period parameter defines the maximum value for

978-1-4799-8244-8/15/$31.00 ©2015 IEEE

node measurement period. All the clusters are in one collision domain (i.e. at maximum, one cluster can be active at any given time) and have an equal ***BI***, which is configurable parameter.

The omission of DAAM leads to the formation of the cluster-tree topology for the set of spatially distributed devices based on Rm, Cm, and Lm parameters as a bounded-degree-and-depth tree which is NP-complete problem [5]. The aim behind omitting the DAAM procedure is to (1) overcome the orphan node problem for the nodes that form flow sources or sinks, and consequently, all the flows can be accommodated in the network, and (2) to meet the application requirement through right choosing of parent-child relations (i.e. under the resulting cluster-tree topology, there exists a cluster schedule that meets all the deadlines of the flows). To reduce the data transmission delay and the cost of the network, our first objective is to minimize the number of clusters. This objective is also consistent with flow deadlines satisfaction. The second objective is to maximize the lifetime of the network by maximizing the length of the schedule period (i.e the cluster schedule is an energy efficient) which is equivalent to maximize the time when the nodes are in low power mode. This maximization of the cluster schedule's period is in contradiction with the flow deadlines.

To solve the problem, we present the Cluster-tree Formation and Energy-Efficient Feasible Scheduling algorithm (CFEFS), which is explained in details in Section III.

II. Related works

There are some papers already dealing with the construction of the ZigBee cluster tree. In [5], the authors suggest centralized and distributed algorithms to satisfy the Cm, Rm, and Lm parameters while maximizing the number of devices joining the networks. In [7], the authors propose a ZigBee-compatible adaptive joining mechanism with connection shifting schemes to improve the connectivity of ZigBee networks.

Many papers tackle the scheduling problem in cluster-tree WSNs. Koubaa et al. [8] have proposed an algorithm for collision-free beacon/superframe scheduling in ZigBee cluster-tree networks with no timing requirements for the flows. In [9], the authors introduce a cluster scheduling mechanism for a multi-hop cluster tree WSNs minimizing energy consumption. The scheduling problem is NP-hard while assuming multiple collision domains and precise end-to-end deadlines for data flows. In our previous work [6], the problem became polynomial due to specifying the flow deadlines in terms of the maximum number of crossed periods assuming one-collision domain. These two assumptions enabled the transformation of the cyclic time-constraint scheduling problem to a shortest path problem.

To the best of our knowledge, so far, no previous research has directly integrated the cluster-tree formation and the cluster scheduling problems together to satisfy the flow deadlines.

III. Cluster-tree Formation for feasible and energy efficient cluster schedule

A. Cluster formation and scheduling constraints

For the cluster-tree formation, the desired tree is the one that minimize the number of clusters such that the designing parameters (Cm, Rm, Lm) are satisfied and none of the flow sources and sinks are an orphan node. The cluster scheduling problem is constrained by: BI_{max}: the upper bound of BI, BI_{min} the lower bound of BI, and the flow deadline constraints. As specified by the ZigBee standards [3], the length of BI is defined by the *Beacon Order* (BO) parameter as follows:

$$BI = aBaseSuperframeDuration \cdot 2^{BO} \qquad (1)$$

where $0 \leq BO \leq 14$ and *aBaseSuperframeDuration* = 15.36 ms (assuming the 2.4 GHz frequency band and 250 kbps of bit rate). BI_{max} is determined by the minimum periods of the set of flows. Then BO_{max} is calculated as shown in Eq. 2:

$$\begin{aligned} BI_{max} &= \min_{k}(req\_period_{f_k}) \\ BO_{max} &= \left\lfloor \log_2 \left(\frac{BI_{max}}{aBaseSuperframeDuration} \right) \right\rfloor \end{aligned} \qquad (2)$$

Since we assume a one-collision domain cluster-tree, BI_{min} is given by the sum of the active portions of all clusters which varies according to the cluster-tree topology. So, BO_{min} is defined as a global variable which is set by CFEFS algorithm, for each tree topology. BO_{min} is initialized to 0.

To simplify the problem and in order to be able to apply our scheduling algorithm presented in [6], we transform the flow end-to-end deadlines to the maximum number of crossed periods denoted by h_{f_k} for flow f_k as shown in Eq. 3:

$$h_{f_k} = \begin{cases} \left\lfloor \dfrac{e2e\_deadline_{f_k}}{\lambda} \right\rfloor - 1 & \text{if } e2e\_deadline_{f_k} \geq \lambda, \\ 0 & \text{otherwise.} \end{cases} \qquad (3)$$

where $e2e\_deadline_{f_k}$ is the precise end-to-end deadline for the flow f_k to reach its destination and λ is the length of the schedule period which is equal to BI for some $BO \in [BO_{min}, BO_{max}]$. For energy efficiency, it is required to maximize λ such that the scheduling problem has a feasible solution.

B. CFEFS algorithm

The CFEFS algorithm iterates over the following two parts:

- Formation of a bounded-degree-and-depth ZigBee cluster-tree.
- Feasible and energy-efficient cluster schedule construction for the topology found in the first part.

Let $Mesh(V, E)$ be the graph representation of the connectivity of all devices where $V(Mesh)$ denotes the set of all devices and $E(Mesh)$ is the set of edges. A pair of devices i and j are connected by an edge (i, j) if i and j are in each other's transmission area and i is a router. The pseudo code shown in Alg. 1 constructs the minimum set of nodes denoted by $Active\_nodes$ to reach the T nodes (Alg. 1 line 1) using the Breadth-first search (BFS) algorithm starting from the root node till the nodes at depth Lm. The S_Mesh is a subgraph of the $Mesh$ created by the $CSMesh$ function where $V(S\_Mesh) = Active\_nodes$ while $E(S\_Mesh)$ is constructed in the same manner used in the $Mesh$. Based on the $Active\_nodes$ and the BO, which is initialized to BO_{max}, the CFEFS algorithm presented in Alg. 2 is iteratively called to find the required cluster-tree denoted by the Z_Tree with

Algorithm 1: *Active_nodes* set construction and calling of CFEFS.

```
1  T←Unique set of all sources and sinks ;
2  Active_nodes ←BFS (Mesh,root,T,Lm);
3  if (T ⊆ Active_nodes) then
4      while (1) do
5          BO = BO_max , BO_min = 0;
6          while (BO ≥ BO_min && ∼ feasible) do
7              S_Mesh ←CSMesh(Mesh,Active_nodes);
8              (feasible,Z_Tree,schedule)←
9                  CFEFS(S_Mesh,flows,T,BO,Cm,Rm,Lm);
10             BO = BO − 1;
11         if (feasible) then
12             break;
13         else
14             choose randomly a router R_i such that
               R_i ∉ Active_nodes and R_i has the maximum
               number of edges connected with Active_nodes;
15             if (there exist such R_i) then
16                 add R_i to Active_nodes;
17             else
18                 break;
19 else
20     Lm is not large enough;
```

the corresponding cluster schedule ($Schedule$). The proposed algorithm terminates if the required tree is found, otherwise BO is decreased by 1 as long it does not reach BO_{min}. A new router is added to the set of $Active\_nodes$ (Alg. 1 line 13) in case when no feasible solution is found for the previous set of $Active\_nodes$ and the whole process is repeated again.

In the CFEFS, the tree construction problem is divided into two sub problems. In the first one, Rm and Lm are used to join the routers to the network using the Hungarian algorithm [10] which is applied on the S_Mesh for at most Lm steps. At each step i, the level i denoted by L_i includes the nodes at depth $i-1$ that joined the tree while L_{i+1} includes their neighboring nodes which have not joined the tree yet. Notice that L_1 includes the root. At L_i, each node is duplicated R_m times and the resulting nodes are connected by edges to their neighboring nodes at L_{i+1}. The edges are weighted inversely proportional to the degree of their end nodes at L_{i+1}. When $deg(j) = 1$ and $j \in T$, then j immediately joins its neighboring node (see node 9 in step 2 in Fig. 3a). The Hungarian algorithm is then applied to assign the remaining nodes at L_{i+1} to the nodes at L_i. The S_Mesh is then locally updated by removing (j, k) edges where j and k are joined nodes or $j \in L_i$ or $k \in L_i$. For the second sub problem, every $ZED \in T$ is joined to the network using the Maximum Matching algorithm. The $MaxMatch$ joins this set of ZEDs with the set formed by the ZC and the ZRs that can accept new ZED due to the Cm and the Lm.

For the scheduling part in our proposed algorithm and based on [6], the cluster scheduling problem is infeasible if the inequality graph has negative cycles or the active portions of the clusters do not fit into the schedule period λ. The inequality graph is a graph representation of the cyclic time-constraint scheduling model (an example is shown later in this section and for more details; please refer to [6]). To deal with the infeasible cases caused by the negative cycles in the inequality graph, function $CrSchAlg$ slightly modifies our previous

Algorithm 2: Cluster-tree Formation and Energy-Efficient Feasible Scheduling algorithm (CFEFS).

```
1  R_Tree ←Hungarian (S_Mesh, Rm, Lm);
2  Z_Tree←MaxMatch(S_Mesh,R_Tree, Cm, Lm);
3  if (All flows sources and sinks are in Z_Tree) then
4      (feasible,C_Sch,N_Cycle) ←CrSchAlg(Z_Tree,flows,BO);
5      if (feasible) then
6          return; % stop execution of CFEFS and return
7      else
8          if ∼empty(N_Cycle) then
9              for each topological_edge e_i in N_Cycle do
10                 if (e_i is not the only edge connecting nodes
                   belong to T) then
11                     remove e_i from S_Mesh;
12                     recursive call of the CFEFS;
13         else
14             BO_min = BO;
```

algorithm presented in [6] to return the topological edges (i.e. parent-child edges) which constitute the negative cycles N_Cycle by iteratively calling the Bellman-Ford algorithm. The topological edges determine the potential edges to be removed from the S_Mesh. In this approach, the Branch and Bound algorithm is used to eliminate the negative cycles in the inequality graph (Alg. 2 line 12). When the cluster schedule does not fit into λ (i.e. an infeasible case with no negative cycles), we set BO_{min} to BO.

Example 1: Let us assume the set of flows represented in Fig. 2a where the flow deadlines are given in the maximum number of crossed periods. The $Mesh$ graph is given in Fig. 2b. To simplify the problem, we assume that each node represents a cluster and an edge connects two nodes if the corresponding cluster heads are in each other's transmission area. Let us also assume that $Rm = 2$, $Lm = 4$ and cluster 1 is the root of the required cluster-tree. In this example, we show the infeasibility case caused by the negative cycle in the inequality graph.

Flow_id	Source	Sink	Maximum number of crossed periods
1	1	6	0
2	9	1	2
3	7	8	1

(a)

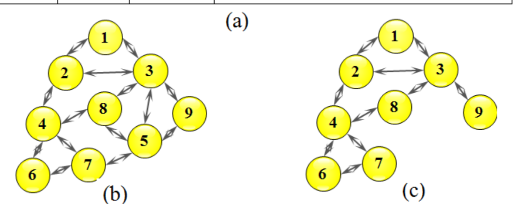

Fig. 2. (a): The flow parameters, (b): Mesh graph, (c): S_Mesh graph.

Based on the set of flows, $T = [1, 6, 9, 7, 8]$ and the S_Mesh is found by the BFS search starting from node 1 as shown in Fig. 2c. The Hungarian algorithm is then applied on the S_Mesh as shown in Fig. 3a. The resulting cluster-tree with flows f_1, f_2, f_3 is shown in Fig. 3b.

Based on [6], the cyclic scheduling constraint model of the cluster-tree presented in Fig. 3b is illustrated in Fig. 4. Each variable N_j represents the number of forward edges from the root of the network represented by node 1 to node j. The forward edge is an edge which goes from the parent node to its child node while the backward edge is an edge that goes

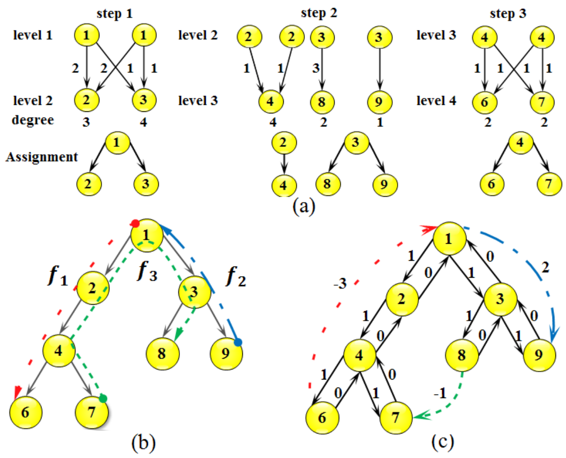

Fig. 3. (a): Hungarian algorithm steps applied on Fig. 2c, (c): Cluster-tree topology found by the Hungarian algorithm, (d): Inequality graph for the constraint model shown in Fig. 4.

from the child node to its parent node. The constraint model is transformed to the inequality graph as shown in Fig. 3c. For each constraint in the form $N_j - N_i \leq const$, we add an edge from node i to node j weighted by $const$.

$0 \leq N_j - N_i \leq 1$	topological constraint
$N_1 - N_6 \leq -3$	for flow f_1
$N_9 - N_1 \leq \ \ 2$	for flow f_2
$N_7 - N_8 \leq -1$	for flow f_3
$N_j \geq 0$	

Fig. 4. The constraint model of the example presented in Fig. 3b.

The N_j value equals the length of the shortest path from node 1 to node j. Based on the Bellman-Ford algorithm, the cycle $(1 - 3 - 8 - 7 - 4 - 6 - 1)$ in Fig. 3c is a negative cycle. Hence, the $N\_Cycle = \{(1,3),(3,8),(4,7),(4,6)\}$. The Branch and Bound algorithm iterates through the edges in the N_Cycle one by one. When the edges $(1,3)$ and $(3,1)$ are removed from S_Mesh in Fig. 2c, the Hungarian algorithm constructs the cluster-tree topology that is shown in Fig. 5a.

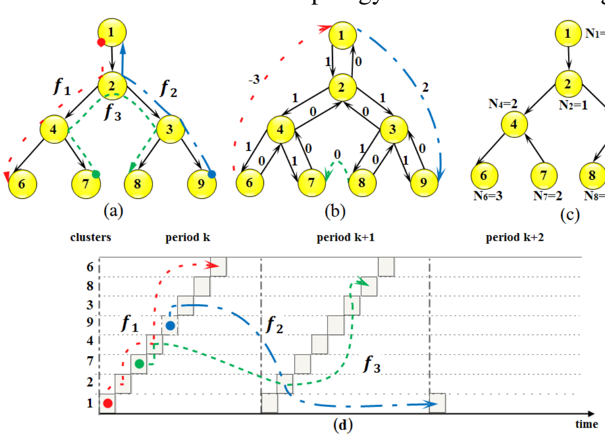

Fig. 5. (a): Cluster-tree topology found by the Hungarian algorithm after removing $(1,3)$ and $(3,1)$ edges from Fig. 2c, (b): Inequality graph for the constraint model shown in Fig. 6, (c): Partial ordering of the clusters activation, (d): Feasible schedule of the clusters.

The constraint model of the resulting tree presented in Fig. 5a is illustrated in Fig. 6. The inequality graph is shown in Fig. 5b. The partial ordering of the cluster activations based on the N_j values is shown in Fig. 5c. For each neighboring nodes pair i and j, if N_i equals N_j, then nodes i and j are connected by a backward edge, otherwise they are connected by forward edge. The topological ordering of the nodes in Fig. 5c gives the cluster schedule as shown in Fig. 5d.

$0 \leq N_j - N_i \leq 1$	topological constraint
$N_1 - N_6 \leq -3$	for flow f_1
$N_9 - N_1 \leq \ \ 2$	for flow f_2
$N_7 - N_8 \leq \ \ 0$	for flow f_3
$N_j \geq 0$	

Fig. 6. The constraint model of the cluster-tree presented in Fig. 5a.

IV. CONCLUSION AND FUTURE WORK

This paper presents a new joining mechanism for ZigBee cluster-tree driven by the cluster scheduling problem. Our proposed CFEFS algorithm seeks to define the parent-child relations between the ZigBee devices to satisfy both the topological constraints defined by the ZigBee standards and the flow deadline constraints.

As future work, the plan is to benchmark our proposed algorithm for some generated instances in addition to design a distributed version of our algorithm.

Acknowledgement

This work was supported by the Grant Agency of the Czech Republic under the Project GACR P103/12/1994.

REFERENCES

[1] M. Chitnis, Y. Liang, J. Y. Zheng, P. Pagano, and G. Lipari, "Wireless Line Sensor Network for Distributed Visual Surveillance," in *Proc. of the 6th ACM International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN)*, Oct. 2009, pp. 71–78.

[2] *IEEE P802.15 Wireless Personal Area Networks: Proposal for Factory Automation*, Working Draft Proposed Standard, Rev. 802.15.4-15/08/0571r0, 2009.

[3] *ZigBee Specification*, ZigBee Standards Org. Std. 053 474r13, 2006.

[4] P. Jurčík, R. Severino, A. Koubâa, M. Alves, and E. Tovar, "Real-Time Communications over Cluster-Tree Sensor Networks with Mobile Sink Behaviour," in *Proc. of the 14th IEEE International Conf. on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, Aug. 2008.

[5] M.-S. Pan, C.-H. Tsai, and Y.-C. Tseng, "The orphan problem in zigbee wireless networks," IEEE Transactions on Mobile Computing, vol. 8, pp. 15731584, 2009

[6] A. Ahmad, Z. Hanzálek, and C. Hanen, "A Polynomial scheduling algorithm for IEEE 802.15.4/ ZigBee cluster tree WSN with one collision domain and period crossing constraint," in *Proc. of 19th International Conf. on Emerging Technologies and Factory Automation (ETFA)*, Sep. 2014.

[7] T.-W. Sung and C.-S. Yang (2010), "An adaptive joining mechanism for improving the connection ratio of ZigBee wireless sensor networks,". Int. J. Commun. Syst., 23: 231251. doi: 10.1002/dac.1067.

[8] A. Koubâa, A. Cunha, M. Alves, and E. Tovar, "TDBS: a time division beacon scheduling mechanism for ZigBee cluster-tree wireless sensor networks," *Real-Time Systems Journal*, vol. 40, no. 3, pp. 321–354, Oct. 2008.

[9] Z. Hanzálek and P. Jurčík, "Energy efficient scheduling for cluster-tree Wireless Sensor Networks with time-bounded data flows: Application to IEEE 802.15.4/ZigBee," *IEEE Transaction on Industrial Informatics*, vol. 6, no. 3, Aug. 2010.

[10] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. Network Flows: Theory, Algorithms, and Applications. Prentice Hall, 1993.