New Approach Construction for Wireless ZigBee Sensor Based on Embedding Pancake Graphs

S. Femmam[1, 2], M. F. Zerarka[2] & M. I. Benakila[3]

[1] University of Haute Alsace, UHA 68093, France

[2] Polytechnic Engineering School, EPF 92330, France

[3] University of Versailles UVSQ, Av. des Etats-Unis, France

Correspondence: S. Femmam, University of Haute Alsace UHA 68093, IEEE Senior member, France. E-mail: femmam@ieee.org

Received: September 19, 2011 Accepted: Novermber 29, 2011 Online Published: June 1, 2012

doi:10.5539/nct.v1n1p7 URL: http://dx.doi.org/10.5539/nct.v1n1p7

This research is financed by Polytechnic Engineering School EPF Labs

Abstract

Wireless Sensor Networks (WSN) based on the IEEE 802.15.4 standard are constantly expanding. Applications like production control, building control are more and more based on WSN because of their energy efficiency, self organizing capacity and protocol flexibility. However, the construction of Cluster-Tree networks based on the beacon mode Pancake graphs is still undefined by the IEEE 802.15.4 standard. In order to enable the construction of such topology, i.e., Beacon Cluster-Tree based on Pancake graphs, we present, in this paper, a new topology construction approach.

The Pancake is one of the Cayley graphs that were proposed as alternative to the Hypercube for interconnecting processors in parallel computers. This network offers attractive and desirable properties: Vertex symmetry, small degree and diameter, extensibility, high connectivity, easy routing, regularity of topology, fault-tolerance, and embed ability of other topologies. We present in this work the many-to-one embedding of Multiply-Twisted Hypercube into the Pancake networks with dilation 5 as a new approach for wireless networks. The presented approach is based on the exploitation of RF front-end capabilities in treating multipath signals and, thus, avoiding the introduction of beacon or Super Frames scheduling algorithms. Avoiding the introduction of scheduling algorithms ensures a simple solution that could be easily implemented and executed by ZigBee sensor nodes.

Keywords: beacon, ZigBee Cluster-Tree, wireless sensors, Cayley graph, embedding, Multiply-Twisted Hypercube networks, Pancake networks, dilation

1. Introduction

A sensor network is an infrastructure comprised of sensing, computing, and communication elements that gives an administrator the ability to observe, and react to events and phenomena in a specified environment (Ilyas & Mahboub, 2005).

This kind of network is characterized by low energy consumption (Dondi et al., 2008), low rate, and cost. Applications based on Wireless Sensor Networks (WSN) are growing every day and cover diversified application fields such as: ambient assisted living, building automation, or factory automation.

The beacon mode has been designed to work with a star topology. However, no mechanisms have been defined in the IEEE 802.15.4 standard to enable the beacon mode using a mesh topology or based on Pancake graphs and it is not clearly defined how to establish a beacon Cluster-Tree topology based on Pancake graphs. One of the advantages of WSNs is that they allow the establishment of large networks with a considerable amount of nodes (up to 65000 nodes for ZigBee standard) covering large areas (Gungor & Hancke, 2009; Yoo et al., 2010). In this context, beacon Cluster-Tree networks are the most suitable, such a network grants, due to the GTS mechanism, a certain QoS for time sensitive applications (Chalhoub & Misson, 2010; Junghee, 2009).

IEEE 802.15.4/ZigBee standard defines the Cluster-Tree topology as a special case of a peer-to-peer network (Part 15.4, 2006). But the realization of beacon Cluster-Tree networks based on Pancake graphs is not defined in the standard. Some work has been done in order to model Cluster-Tree topologies (Kouba et al., 2006; Mhatre & Rosenberg, 2004), failure recovery (Gupta & Younes, 2003) and to allow the construction of Beacon Cluster-Tree networks, but never the model based on Pancake graphs was studied in our knowledge. We give an overview of the MAC sub-layer and the physical layer specification of the IEEE 802.15.4. And we present the open source platform Zigbee compliant RF transceiver (CISCO, 2008).

Figure 1. Crossbow's TelosB mote

We conceived a real experimentation using the TelosB motes. Crossbow's TelosB mote (Figure 1) is an open source platform designed to enable cutting-edge experimentation for the research community. This notes offer: An IEEE 802.15.4/ZigBee compliant RF transceiver, a globally compatible ISM band, a 250 kbps data rate, and low power consumption. The signal RF is transmitted based on a model system of OQPSK modulation. This model was achieved with the Matlab software. The spectrum obtained has a main lobe occupying a frequency range of 3MHz (Figure 2). The "holes" in the spectrum are due to the fact that only 16 orthogonal codes of 32 chips are used. In the available bandwidth (between 2.405GHz and 2.48GHz), the ZigBee standard set is 11 channels with a channel spacing of 5MHz the spectrum of the modulation is given in Figure 2.

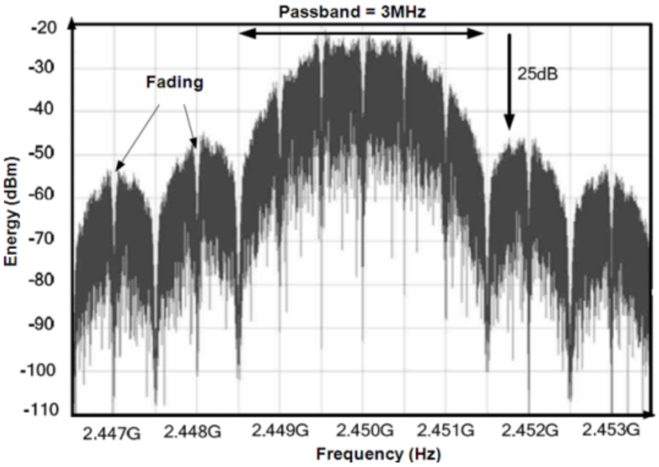


Figure 2. Spectrum modulation IEEE 802.15.4

Finally point out that this standard operates in half duplex for which there can be simultaneously receiving and transmitting data, contrary to what is done in the mode full duplex. The main functions of the physical layer are spreading, dispreading, modulation and demodulation of the signal. The IEEE 802.15.4 defines three frequency bands in which LRWPANs could be deployed: 868 MHz, 915 MHz, and 2.4 GHz. This paper focuses on the 2.4 GHz frequency band because it has been worldwide. The study of graph embedding arises naturally in a number of computational problems: portability of algorithms across various parallel architectures, layout of circuits in VLSI. Akers and Krishnamurthy (Akers & Krishnamurthy, 1989) proposed the Pancake as alternative to the Hypercube and their variations for interconnecting processors in parallel computers. This network has desirable proprieties: Small diameter and fixed degree, (n-1) regular, high connectivity, vertex symmetry, Hamiltonian, fault tolerance, extensibility and embeddability of other topologies (Akers & Krishnamurthy, 1989; Heydari et al., 1997; Hsieh et al., 1998; Kavensky et al., 1995; Hung et al., 2003). The embedding capabilities are important in evaluating an interconnection wireless networks. The many-to-one embedding of the guest graph G into a host graph *H* is a mapping from many vertices of *G* into one vertex of *H* and mapping each edge of *G* into one path of *H*. Graph embedding is useful because an algorithm designed for H can be applied to G directly (Abraham et al., 1991; Cheng-Kuan et al., 2008; Lin et al., 2010; Xu et al., 2010). In comparison with a Multiply-Twisted Hypercube, the Pancake configuration offers good and simple simulations of other interconnection wireless networks (Miller et al., 1994; Senoussi et al., 1997).

This paper is organized as follows: First of all we introduce the definitions which will be used including the definition and properties of Multiply-Twisted Hypercube and also the definition and properties of a Pancake network. In section three we present an algorithm of many-to-one embedding Multiply-Twisted Hypercube into a Pancake. In the section four we show that a dilation of many-to-one embedding of Multiply-Twisted Hypercube into a Pancake is equal to 5. In section five and six we present the most known approaches for resolving this problem and the core of the proposed approach. Simulation results are presented in section seven and eight, and finally, we conclude.

2. Preliminaries

2.1 Definition 1

The *n*-dimensional Hypercube Q_n and the Multiply-Twisted Hypercube $TQ_n = (V,U)$ have a same set of vertices: *V*. We represent the address of each vertex in Q_n or TQ_n as a binary string of length *n*. With this structure, we do not need to distinguish between vertices and their binary address. In Q_n two vertices are adjacent if and only if, their binary labels differ only in a one bit position (Leighton, 1992). For the Multiply-Twisted Hypercube TQ_n, adjacency requirement is little more complex. The embedding Twisted Hypercube TQ_3 is represented in Figure 3.

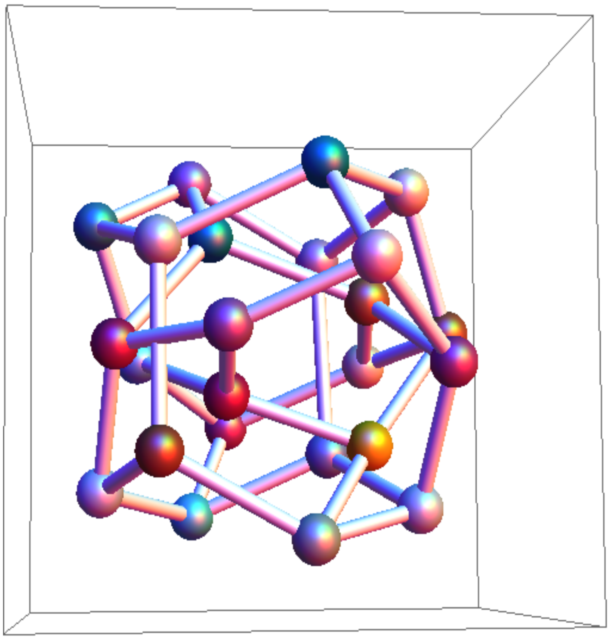


Figure 3. Embedding twisted hypercube TQ3

Definition: Two binary strings $x=x_1x_0$ and $y=y_1y_0$ made up of a length of two are said to be pair-related uniquely if, (x, y) Є {(00, 00), (10, 10), (01, 11), (11, 01)}. The Multiply-Twisted Hypercube TQ_n is recursively defined as follows: TQ_1 is the complete graph based on two vertices labeled 0 and 1. TQ_n consists of two sub-cubes $0TQ_{n-1}$ and $1TQ_{n-1}$ the most significant bit of the labels of the vertices in $0TQ_{n-1}(1TQ_{n-1})$ is 0(1). U is the set of vertices $u=u_{n-1}u_{n-2}...u_1u_0$ Є $0TQ_{n-1}$ with $u_{n-1}=0$ and $v=v_{n-1}v_{n-2}...v_1v_0$ Є $1TQ_{n-1}$ with $v_{n-1}=1$, *u* and *v* are joined by an edge in TQ_n if and only if:

$$u_{n-2} = v_{n-2} \tag{1}$$

if *n* is even and $(u_{2i+1}u_{2i}, v_{2i+1}v_{2i})$ are pair related.

The Multiply-Twisted Hypercube $TQ_n = (V,U)$, with a *V* set of vertices and a *U* set of edges as an alternative to the Hypercube, has the same number of vertices *V* and the same degree as the Hypercube.

The Multiply-Twisted Hypercube is one of the variations of the Hypercube which is derived with some twisted edges. Due to these twisted edges, the diameter of TQ_n is only half of that of the Hypercube. Some interesting studies on properties of TQ_n can be found in (Huan et al., 2002; Xu et al., 2010) include a relatively small degree, embedding capabilities, scalability, robustness and the fault-tolerant hamiltonicity of TQ_n. The Multiply-Twisted Hypercube graph is not a vertex-transitive for $n \geq 5$ (Kulasinghe et al., 1995).

2.2 Definition 2

Cayley graphs were originally proposed as a generic theoretic model for analyzing symmetric interconnection networks. The most notable feature of Cayley graphs is their universality. The Cayley graph represents a class of high performance interconnection networks with a low degree and low diameter, good connectivity and simple routing algorithms. The Pancake is in fact a Cayley graph.

Let $I = (1, 2, 3, \dots, n)$ and $p = (p_1, p_2, \dots p_n)$, $p_i \in I$ and $p_i \neq p_j$ for $i \neq j$, where p is the permutation of I.

A Pancake graph $G_n=(P_n, E_n)$ is defined as follows:

$$P_n = \{(p_1, p_2, \dots, p_n) | \, p_i \in I, \, p_i \neq p_j \text{ for } i \neq j\} \tag{2}$$

and

$$E_n=\{((p_1, p_2, \dots, p_{j-1}, p_j, p_{j+1}, \dots, p_n), (p_j, p_{j-1}, \dots, p_2, p_1, p_{j+1}, \dots, p_n)) | (p_1, p_2, \dots, p_n) \in P_n \quad \text{for } 2 \leq j \leq n \tag{3}$$

In other words, the set of P_n of all permutations constitutes the nodes of the vertices of G_n. Two nodes u and v are joined by an (undirected) edge if and only if, the permutation corresponding to the node v can be obtained from u by flipping the objects in positions 1 through j. Therefore for each permutation we can flip any number of objects between the first position and position j, or $2 \leq j \leq n$, G_n is a $(n-1)$ regular graph, $|P_n| = n!$, $|E_n| = (n-1)n!/2$.

The Pancake graphs proposed by (Akers & Krishnameurthy, 1989) are an important family of interconnecting networks. An example of n-Pancake topology for n=4 is represented in Figure 4. One of the main properties is their symmetry. It is built using Cayley graph groups with simple routing algorithms. The Pancake graph has many other attractive features, they are strongly hierarchical, maximally fault-tolerant, Hamiltonian (Akers & Krishnameurthy, 1989; Kanevsky et al., 1995; Chen-Kuan et al., 2008). And have a small diameter compared to the Hypercube graph (Morales et al., 1996). The G_n is made of n copies of G_{n-1} namely $G_n[n, k]$ for $1 \leq k \leq n$. We consider each $G_n[n, k]$ as a super node. Hence $G_n[n, s]$, $G_n[n, t]$ are connected by a collection of edges of the form $((t, p_2, p_3, \dots, p_{n-1}, s), (s, p_{n-1}, \dots, p_2, t))$ thus, there are $(n-2)!$ edges connecting $G_n[n, s]$ and $G_n[n, t]$ (Chen et al., 2008). G_n is a complete graph on the super nodes connected by the super edges.

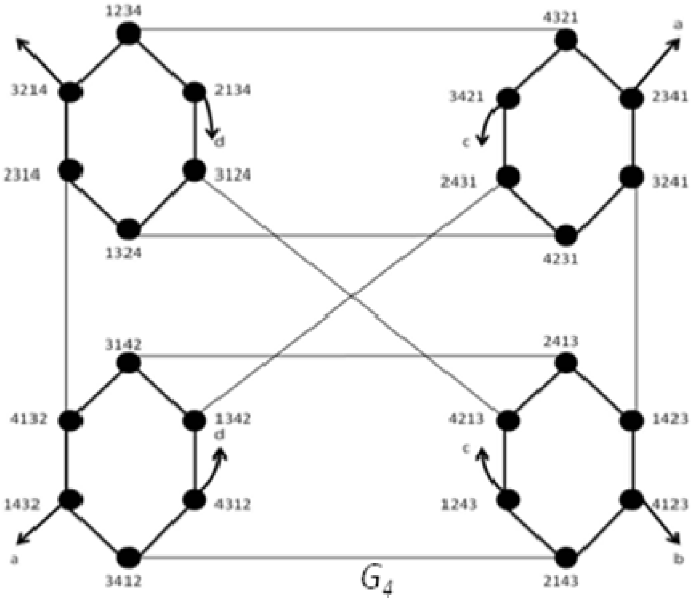


Figure 4. Example of n-pancake graphs n=4

3. Embedding Multiply-Twisted Hepercube Graph into a Pancake Graph

In this section, we present a new function, the many-to-one embedding of a Multiply-Twisted Hypercube graph denoted by TQ_n into a Pancake graph denoted by G_n. Steps for such an embedding function are as follows: *Step* 1: find the first node of TQ_n and the first node of G_n, The Twisted Hypercube and the Pancake graphs for 3 dimensions are represented in Figure 5. For example: 000 of TQ_3 and 123 of G_3. S*tep* 2: Embedding the vertex of TQ_n into G_n using the Embed_node(node) algorithm given in subsection 3.1. *Step* 3: Embedding edges of TQ_n into the path of G_n using the Embed_edge(nodedep, nodearr) algorithm given in subsection 3.2.

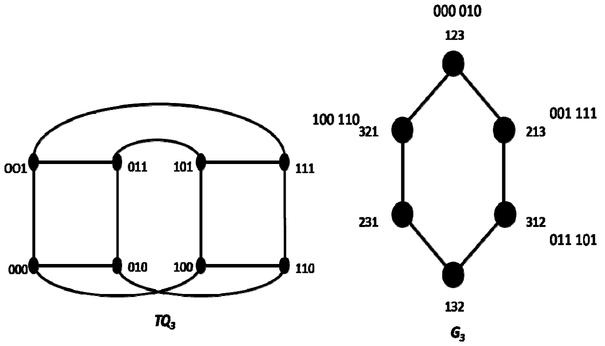


Figure 5. Twisted Hypercube and pancake of 3 dimensions

3.1 Embed_Node(Node) Algorithm

Embed_node(node) algorithm applies all actions specified in Tables 1, 2 and 3.

Note that: *Suffixe*(X) is a function which extracts the l characters from a string X starting from the first character of the least significant weight with $l = n/2$ and *Sufi* = *Suffixe*(X_i). Where X_1 is a node of $G_{n\text{-}1}[n\text{-}1, n\text{-}1]$, X_2 is a node of $G_{n\text{-}1}[n\text{-}1, 1]$, X_3 is a node of $G_{n\text{-}1}[n\text{-}1, 3]$, and X_4 is a node of $G_{n\text{-}1}[n\text{-}1, 2]$.

Pref is a function which extracts the sub-string of (n-8) characters, starting from the third character of the most significant weight. *Suffix*1=*Suffixe*(X) and l=((n/2)–1) when $n > 5$.

The Pancake graph G_n is made by i copies of $G_{n\text{-}1}[n\text{-}1, k]$, for k=1, i. Embed_node(node) algorithm use only i super nodes containing 2^l components G_3 with l=n/2 and i=2 if n is even and i=4 if n is odd. There are two stated situations: the first one is when n is even, we use $G_{n\text{-}1}[n\text{-}1, n\text{-}1]$ for embedding all nodes of $0TQ_{n-1}$, and $G_{n\text{-}1}[n\text{-}1, 1]$ for embedding all nodes of $1TQ_{n-1}$. The second situation is when n is odd or n=2m+1(m Є □), the TQ_n nodes are those of $0TQ_{2m}$ and $1TQ_{2m}$. Let N=2m, the TQ_n nodes are that of $0TQ_N$ and $1TQ_N$, that is to say the nodes of $00TQ_{N-1}$, $01TQ_{N-1}$ and $10TQ_{N-1}$,$11TQ_{N-1}$. We use four super nodes $G_{N\text{-}1}[N\text{-}1, N\text{-}1]$, $G_{N\text{-}1}[N\text{-}1, 1]$, $G_{N\text{-}1}[N\text{-}1, 2]$, $G_{N\text{-}1}[N\text{-}1, 3]$.

The first super node is used for embedding all nodes of $00TQ_{N-1}$, the second one for embedding all nodes of $10TQ_{N-1}$, the third one for embedding all nodes of $11TQ_{N-1}$ and the last one is for embedding all nodes of $01TQ_{N-1}$. An example of embedding graph of TQ_5 is represented in Figure 6.

Table 1. Embedding all nodes of TQ4 into G4

0*TQ*₃	***G*₄[4,4]**	**1*TQ*₃**	***G*₄[4,1]**
0000	$x_1x_2x_3x_4$	1000	$x_4x_3x_2x_1$
0001	$x_2x_1x_3x_4$	1001	$x_3x_4x_2x_1$
0010	$x_1x_2x_3x_4$	1010	$x_4x_3x_2x_1$
0011	$x_3x_1x_2x_4$	1011	$x_2x_4x_3x_1$
0100	$x_3x_2x_1x_4$	1100	$x_2x_3x_4x_1$
0101	$x_3x_1x_2x_4$	1101	$x_2x_4x_3x_1$
0110	$x_3x_2x_1x_4$	1110	$x_2x_3x_4x_1$
0111	$x_2x_1x_3x_4$	1111	$x_3x_4x_2x_1$

Table 2. Embedding all nodes of TQ_5 into G_5

00*TQ*$_3$	**G_4[4,4]**	**10*TQ*$_3$**	**G_4[4,1]**	**01*TQ*$_3$**	**G_4[4,3]**	**11*TQ*$_3$**	**G_4[4,2]**
0	$x_1x_2x_3x_4$	10000	$x_4x_3x_2x_1$	1000	$x_3x_4x_1x_2$	11000	$x_2x_1x_4x_3$
1	$x_2x_1x_3x_4$	10001	$x_3x_4x_2x_1$	1001	$x_4x_3x_1x_2$	11001	$x_1x_2x_4x_3$
10	$x_1x_2x_3x_4$	10010	$x_4x_3x_2x_1$	1010	$x_3x_4x_1x_2$	11010	$x_2x_1x_4x_3$
11	$x_3x_1x_2x_4$	10011	$x_2x_4x_3x_1$	1011	$x_1x_3x_2x_4$	11011	$x_4x_2x_1x_3$
100	$x_3x_2x_1x_4$	10100	$x_2x_3x_4x_1$	1100	$x_1x_4x_3x_2$	11100	$x_4x_1x_2x_3$
101	$x_3x_1x_2x_4$	10101	$x_2x_4x_3x_1$	1101	$x_1x_3x_4x_2$	11101	$x_4x_2x_1x_3$
110	$x_3x_2x_1x_4$	10110	$x_2x_3x_4x_1$	1110	$x_1x_4x_1x_2$	11110	$x_4x_1x_2x_3$
111	$x_2x_1x_3x_4$	10111	$x_3x_4x_2x_1$	1111	$x_4x_3x_1x_2$	11111	$x_1x_2x_4x_3$

Table 3. Embedding all nodes of TQ_n in G_n for n >5 and A=00, 01, 10, 11

***APref*00*TQ*$_3$**	**G_{n-1}[*n*-1,*n*-1]**	***APref*10*TQ*$_3$**	**G_{n-1}[*n*-1,1]**	***APref01TQ3***	**G_{n-1}[*n-1,3*]**	***APref11TQ3***	**G_{n-1}[*n-1,2*]**
*APref*000000	$x_1x_2x_3x_4$*suf1*	*APref*100000	$x_4x_3x_2x_1$*Suf2*	*APref010000*	$x_3x_4x_1x_2$*Suf3*	*APref110000*	$x_2x_1x_4x_3$*Suf4*
*APref*000001	$x_2x_1x_3x_4$*suf1*	*APref*100001	$x_3x_4x_2x_1$*Suf2*	*APref010001*	$x_4x_3x_1x_2$*Suf3*	*APref110001*	$x_1x_2x_4x_3$*Suf4*
*APref*000010	$x_1x_2x_3x_4$*Suf1*	*APref*100010	$x_4x_3x_2x_1$*Suf2*	*APref010010*	$x_3x_4x_1x_2$*Suf3*	*APref110010*	$x_2x_1x_4x_3$*Suf4*
*APref*000011	$x_3x_1x_2x_4$*suf1*	*APref*100011	$x_2x_4x_3x_1$*Suf2*	*APref010011*	$x_1x_3x_2x_4$*Suf3*	*APref110011*	$x_4x_2x_1x_3$*Suf4*
*APref*000100	$x_3x_2x_1x_4$*suf1*	*APref*100100	$x_2x_3x_4x_1$*Suf2*	*APref010100*	$x_1x_4x_3x_2$*Suf3*	*APref110100*	$x_4x_1x_2x_3$*Suf4*
*APref*000101	$x_3x_1x_2x_4$*suf1*	*APref*100101	$x_2x_4x_3x_1$*Suf2*	*APref010101*	$x_1x_3x_4x_2$*Suf3*	*APref110101*	$x_4x_2x_1x_3$*Suf4*
*APref*000110	$x_3x_2x_1x_4$*suf1*	*APref*100110	$x_2x_3x_4x_1$*Suf2*	*APref010110*	$x_1x_4x_1x_2$*Suf3*	*APref110110*	$x_4x_1x_2x_3$*Suf4*
*APref*000111	$x_2x_1x_3x_4$*suf1*	*APref*100111	$x_3x_4x_2x_1$*Suf2*	*APref010111*	$x_4x_3x_1x_2$*Suf3*	*APref110111*	$x_1x_2x_4x_3$*Suf4*

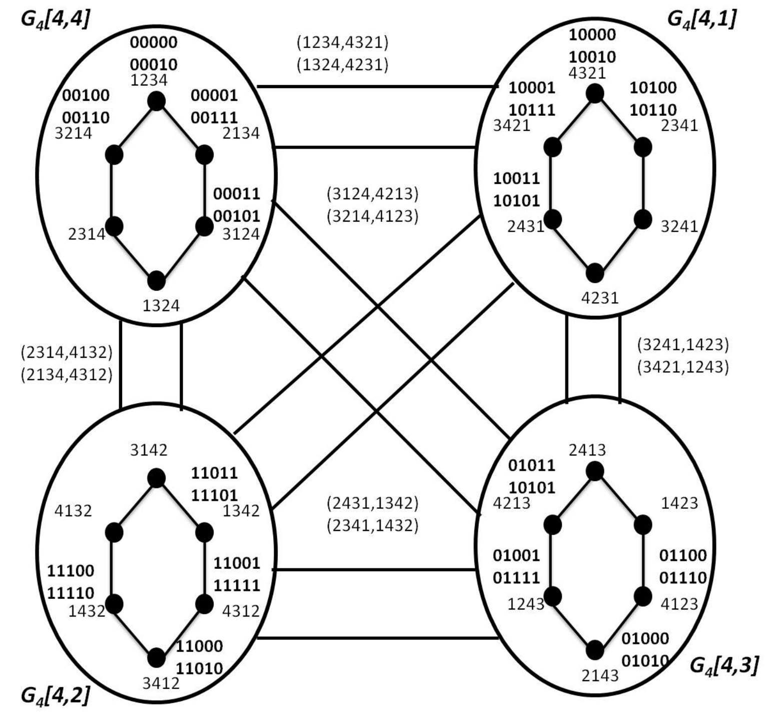


Figure 6. Embedding graph of TQ_5 into G_5

3.2 Embed_Edge(Nodedep, Nodearr) Algorithm

The Embed_edge (nodedep, nodearr) algorithm is given as follows:

```
Begin;
        S1:= Suffixe(nodedep);    S2 := Suffixe(nodearr);
        S3:= Suffix1(nodedep);    S4 := Suffix1(nodearr);
        If S1=S2 then
        Embed1_edge(nodedep, nodearr)
        Else
        If S3=S4 then
        Embed2_edge(nodedep, nodearr)
        Else
        Embed3_edge(nodedep, nodarr)
        Endif;
        Endif;
End;
```

3.2.1 Embed1_Edge(Nodedep, Nodearr) Algorithm

The Embed1_edge(nodedep, nodearr) algorithm is used when the paths are in the same G_3 of a super node. This algorithm applies the following different cases as represented in Table 4.

Table 4. Results of Embed1_edge(nodedep, nodearr) algorithm

$APref_2$000-$APref_2$001 by $x_1x_2x_3x_4Suffix$-$x_2x_1x_3x_4Suffix$.
$APref_2$000-$APref_2$010 by $x_1x_2x_3x_4Suffix$-$x_1x_2x_3x_4Suffix$.
$APref_2$000-$APref_2$100 by $x_1x_2x_3x_4Suffix$-$x_3x_2x_1x_4Suffix$.
$APref_2$*001*-$APref_2$011 by $x_2x_1x_3x_4Suffix$-$x_3x_1x_2x_4Suffix$.
$APref_2$001-$APref_2$111 by $x_2x_1x_3x_4Suffix$-$x_2x_1x_3x_4Suffix$.
$APref_2$010-$APref_2$011 by $x_1x_2x_3x_4Suffix$-$x_2x_1x_3x_4Suffix$-$x_3x_1x_2x_4Suffix$.
$APref_2$010-$APref_2$110 by $x_1x_2x_3x_4Suffix$-$x_3x_2x_1x_4Suffix$.
$APref_2$011-$APref_2$101 by $x_3x_1x_2x_4Suffix$-$x_3x_1x_2x_4Suffix$.
$APref_2$100-$APref_2$101 by $x_3x_2x_1x_4Suffix$-$x_1x_2x_3x_4Suffix$-$x_2x_1x_3x_4Suffix$-$x_3x_1x_2x_4Suffix$.
$APref_2$101-$APref_2$111by $x_3x_1x_2x_4Suffix$-$x_2x_1x_3x_4Suffix$.
$APref_2$*110*-$APref_2$111 by $x_3x_2x_1x_4Suffix$-$x_1x_2x_3x_4Suffix$-$x_2x_1x_3x_4Suffix$.

Note that A=(00, 10) and for this situation *Suffix*=*Suffixe*(X) with X is a node of the concerned super node and $Pref_2$ is a function which extracts the sub-string of (n-5) characters starting from the third character of the highest weight. The symmetric paths are applied when A=(01, 11), for instance $A\,Pref_2$000-$A\,Pref_2$001 by $x_1x_3x_2x_4Suffix$-$x_3x_1x_2x_4Suffix$, $A\,Pref_2$010-$A\,Pref_2$011 by $x_1x_3x_2x_4Suffix$-$x_3x_1x_2x_4Suffix$-$x_2x_1x_3x_4Suffix$ and $A\,Pref_2$100-$A\,Pref_2$101 by $x_2x_3x_1x_4Suffix$-$x_1x_3x_2x_4Suffix$-$x_3x_1x_2x_4Suffix$-$x_2x_1x_3x_4Suffix$.

3.2.2 Embed2_Edge(Nodedep, Nodearr) Algorithm

The Embed2_edge(nodedep, nodearr) algorithm performs the embedding of the edge of Multiply-Twisted Hypercube into Pancake, if suffix1(nodedep) and suffix1(nodearr) differ exactly in the fourth position. Four cases arise in this situation.

In the first case, the edge of the Multiply-Twisted Hypercube is $Pref_1 00a_{n-2}a_{n-1}a_n$- $Pref_1 01a_{n-2}a_{n-1}a_n$, in the second case it is $Pref_1 00a_{n-2}a_{n-1}a_n$- $Pref_1 10a_{n-2}a_{n-1}a_n$, in the third case it is $Pref_1 01a_{n-2}a_{n-1}a_n$- $Pref_1 11a_{n-2}a_{n-1}a_n$, and finally in the last case it is $Pref_1 10a_{n-2}a_{n-1}a_n$- $Pref_1 11a_{n-2}a_{n-1}a_n$. The Embed2_edge(nodedep, nodearr) algorithm applies exactly the following actions in the first case as represented in Table 5. For example: $Pref_1$ is a function which extracts (n-5) characters starting from the first character of the most significant weight.

Table 5. Embed2_edge(nodedep, nodearr) algorithm

$Pref_1$00010-$Pref_1$01010 or $Pref_1$*00000*-$Pref_1$01000 by $x_1x_2x_3x_4Suffix-x_3x_2x_1x_4Suffix-x_2x_1x_4x_3Suffix-x_2x_1x_4x_3Suffix$.
$Pref_1$00001-$Pref_1$01011 or $Pref_1$00101-$Pref_1$01111 by $x_1x_2x_3x_4Suffix-x_2x_1x_3x_4Suffix-x_4x_3x_2x_1Suffix$.
$Pref_1$00100-$Pref_1$01100 or $Pref_1$00110-$Pref_1$01110 by $x_1x_2x_3x_4Suffix-x_4x_3x_2x_1Suffix$.
$Pref_1$00011-$Pref_1$01001or $Pref_1$00111-$Pref_1$01101 by $x_1x_2x_3x_4Suffix-x_4x_3x_2x_1Suffix$- $x_2x_3x_4x_1Suffix$.

3.2.3 Embed3_Edge(Nodedep, Nodarr) Algorithm

This procedure is used for $n>5$ and all paths are between two different G_4 in the different super nodes. Let A=(a_1a_2) and B=(b_1b_2), where (a_1a_2, b_1b_2) = (00, 01), (00, 10), (01, 11), (10, 11). For $n>5$, Embed3_edge(nodedep, nodarr) algorithm performs the different actions specified in the four stated following cases. Except the case when $n=6$, $A\,Pref_1$ is replaced by 0, $B\,Pref_1$ is replaced by 1 and when $n=7$, ($A\,Pref_1$, $B\,Pref_1$) = (00, 01), (00, 10), (01, 11), (10, 11).

TheEmbed3_edge(nodedep, nodearr) algorithm applies exactly the following actions as represented in Table 6, results for the case 1 are shown in Table 6 : Note that for the sake of simplicity the cases 2, 3 and 4 are not given in this paper.

Table 6. Embed3_edge(nodedep, nodarr) algorithm

$APref_1$00000-$BPref_1$00000 or $APref_1$00010-$BPref_1$00010 by $x_1x_2x_3x_4x_5Suffix-x_3x_2x_1x_4x_5Suffix$.
$APref_1$00001-$BPref_1$00011 or $APref_1$00101-$BPref_1$00111 by $x_1x_2x_3x_4x_5Suffix-x_2x_3x_1x_4x_5Suffix-x_5x_4x_1x_3x_2Suffix-x_4x_5x_1x_3x_2Suffix$- $x_1x_5x_4x_3x_2Suffix$.
$APref_1$00100-$BPref_1$00100 or $APref_1$0011*0*-$BPref_1$00110 *by* $x_1x_2x_3x_4x_5Suffix-x_3x_2x_1x_4x_5Suffix-x_5x_4x_1x_2x_3Suffix-x_1x_4x_5x_2x_3Suffix$.
$APref_1$0001*1*-$BPref_1$00001 or $APref_1$00111-$BPref_1$00101 by $x_1x_2x_3x_4x_5Suffix-x_2x_1x_3x_4x_5Suffix-x_5x_4x_3x_1x_2Suffix-x_3x_4x_5x_1x_2Suffix-x_1x_5x_4x_3x_2Suffix$.
$APref_1$01000-$BPref_1$01000 or $APref_1$01010-$BPref_1$01010 *by* $x_1x_2x_3x_4x_5Suffix-x_4x_3x_2x_1x_5Suffix-x_5x_1x_2x_3x_4Suffix$.
$APref_1$01001-$BPref_1$01011 or $APref_1$01101-$BPref_1$01111 by $x_1x_2x_3x_4x_5Suffix-x_2x_1x_3x_4x_5Suffix-x_4x_3x_1x_2x_5Suffix-x_5x_2x_1x_3x_4Suffix$- $x_2x_5x_1x_3x_4Suffix$.
$APref_1$01100-$BPref_1$01100 or $APref_1$01110-$BPref_1$01110 by $x_1x_2x_3x_4x_5Suffix-x_4x_3x_2x_1x_5Suffix-x_5x_1x_2x_3x_4Suffix-x_4x_3x_2x_1x_5Suffix$.
$APref_1$01011-$BPref_1$01001 or $APref_1$01111-$BPref_1$01101 by $x_1x_2x_3x_4x_5Suffix-x_2x_1x_3x_4x_5Suffix-x_4x_3x_1x_2x_5Suffix-x_5x_2x_1x_3x_4Suffix-x_3x_1x_2x_5x_4Suffix$.
$APref1_1$10000-$BPref_1$10000 or $APref_1$10010-$BPref_1$0010 by $x_1x_2x_3x_4x_5Suffix-x_3x_2x_1x_4x_5Suffix-x_5x_4x_1x_2x_3Suffix-x_1x_4x_5x_2x_3Suffix$.
$APref_1$10001-$BPref_1$10011 or $APref_1$10101-$BPref_1$10111 by $x_1x_2x_3x_4x_5Suffix-x_2x_1x_3x_4x_5Suffix-x_5x_4x_3x_1x_2Suffix-x_3x_4x_5x_1x_2Suffix$- $x_4x_3x_5x_1x_2Suffix-x_4x_3x_5x_1x_2Suffix$.
$APref_1$10100-$BPref_1$10100 or $APref_1$10110-$BPref_1$10110 by $x_1x_2x_3x_4x_5Suffix-x_5x_4x_3x_2x_1Suffix$.
$APref_1$10011-$BPref_1$10001 or $APref_1$10111-$BPref_1$10101 by $x_1x_2x_3x_4x_5Suffix-x_5x_4x_3x_2x_1Suffix-x_4x_5x_3x_2x_1Suffix-x_3x_5x_4x_2x_1Suffix-x_2x_4x_5x_3x_1Suffix-x_4x_2x_5x_3x_1Suffix$.
$APref_1$11000-$BPref_1$11000 or $APref$11010-$BPref_1$11010 by $x_1x_2x_3x_4x_5Suffix-x_4x_3x_2x_1x_5Suffix-x_2x_3x_4x_1x_5Suffix-x_5x_1x_4x_3x_2Suffix$- $x_4x_1x_5x_3x_2Suffix-x_3x_5x_1x_4x_2Suffix$.
$APref_1$11001-$BPref_1$11011 or $APref_1$11101-$BPref_1$11111 by $x_1x_2x_3x_4x_5Suffix-x_5x_4x_3x_2x_1Suffix-x_4x_5x_3x_2x_1Suffix-x_2x_3x_5x_4x_1Suffix$.
$APref_1$11100-$BPref_1$11100 or $APref_1$11110-$BPref_1$11110 by $x_1x_2x_3x_4x_5Suffix-x_4x_3x_2x_1x_5Suffix-x_2x_3x_4x_1x_5Suffix-x_5x_1x_4x_3x_2Suffix-x_4x_1x_5x_3x_2Suffix-x_3x_5x_1x_4x_2Suffix$.
$APref_1$11011-$BPref_1$11001 or $APref_1$11111-$BPref_1$1101 by $x_1x_2x_3x_4x_5Suffix-x_5x_4x_3x_2x_1Suffix-x_4x_5x_3x_2x_1Suffix-x_2x_3x_5x_4x_1Suffix$.

4. Dilations of Many-to-One Multiply-Twisted Hypercube Embedded Pancake

4.1 Lemma 1

The Multiply-Twisted Hypercube $TQ_n^{'}(V,U_1)$ has many-to-one dilation 3 embedding into $G_n^{'}$ = ($P_n^{'}$, $E_n^{'}$) ∀*n*>3.

Proof: We prove this lemma by deduction. ***Base***: For $n=3$, the results of the Embed1_edge(nodedep, nodearr) algorithm represent all paths between the embedded nodes of TQ_3 into G_3 with dilation 3.

Deduction hypothesis: Suppose that for $k \le n$-1, $TQ_{k-1}^{'}$ embedding many-to-one dilation 3 into $G^{'}_{k\text{-}1}$ is true. Let us now prove that is true for *k*=*n*. We have the following cases:

Case 1: k is pair

$TQ_n^{'}(V,U_1)$ is constructed by two copies of $TQ_{n-1}^{'}$, one copy is prefixed by 0, the second one is prefixed by 1. All nodes *A* Є *V*, such that A=0 $Pref^{'}$ $a_{n\text{-}3}a_{n\text{-}2}a_{n\text{-}1}$= $Pref_1$ $a_{k\text{-}3}a_{k\text{-}2}a_{k\text{-}1}$, where $Pref_1$=0 $Pref^{'}$ are embedded into the first super node $G_k^{'}[k,k]$. And all nodes *B* Є *V*, *B*=1 $Pref^{'}$ $b_{k\text{-}3}b_{k\text{-}2}b_{k\text{-}1}$ or *B*= $Pref_2$ $b_{k\text{-}3}b_{k\text{-}2}b_{k\text{-}1}$ where $Pref_2$= 1 $Pref^{'}$, are embedded into the second super node $G_k^{'}[k,l]$. That is to say, they are embedded into $G_{k-1}^{'}$. However, Embed1_edge(nodedep, nodearr) algorithm is used in this situation and the dilation of embedding into $G_{k-1}^{'}$ is 3 (hypothesis of deduction).

$Pref^{'}$ is a function which extracts sub-string of (*n*-4) characters starting from the first character of the most significant weight.

Case 2: k is odd

Let *k*=2*m*+1, where *m* Є □ , and TQ_n is obtained from two copies of $0TQ_{2m}^{'}$ and $1TQ_{2m}^{'}$, and suppose that for *N*=2*m* we have $0TQ_N^{'}$ and $1TQ_N^{'}$, that is to say, $00TQ_{N-1}^{'}$, $01TQ_{N-1}^{'}$ and $10TQ_{N-1}^{'}$, $11TQ_{N-1}^{'}$. The Embed-node(*A*) embed all nodes *A*=00 $Pref_2$ $a_{N\text{-}3}a_{N\text{-}2}a_{N\text{-}1}$ (*A* Є *V*) into the first super node $G_N^{'}[N,N]$, the nodes *A*=10 $Pref_2$ $a_{N\text{-}3}a_{N\text{-}2}a_{N\text{-}1}$ into $G_N^{'}[N,1]$, the nodes *A*=01 $Pref_2$ $a_{N\text{-}3}a_{N\text{-}2}a_{N\text{-}1}$ into $G_N^{'}[N,3]$, and the nodes *A*=11 $Pref_2$ $a_{N\text{-}3}a_{N\text{-}2}a_{N\text{-}1}$ into $G_N^{'}[N,2]$. In other words, we use only four super nodes among the *k* super nodes. $G_N^{'}$ is (*n*-1)-Pancake. Using Embed1_edge (nodedep, nodearr) algorithm the dilation of many-to-one embedding into $G_N^{'}$ *is* 3 (hypothesis of deduction).

4.2 Lemma 2

The Multiply-Twisted Hypercube $TQ_n^{''}(V,U_2)$ has many-to-one dilation 4 embedding into $G_n^{''}$=($P_n^{''}$, $E_n^{''}$) ∀*n* >4.

Proof: We use the same method to prove lemma 2.

4.2.1 Theorem

The Multiply-Twisted Hypercube $TQ_k(V,U)$ has many-to-one dilation 5 embedding into $G_n=(P_n,E_n)$ ∀*n* >5.

Proof: ***Base***: For *n*=6, the different actions given in the procedure Embed3_edge (nodedep, nodearr) algorithm represent the embedding of all edges of TQ_6 into G_6 with dilation 5. For *n*=7, the different actions given (same thing for case 2, case 3 and case 4 which are not represented in this paper) represent the embedding of all edges of TQ_7 into G_7 with dilation 5.

Deduction hypothesis: Assume that this lemma holds for *k* ≤ *n*-1. That is TQ_{k-1} embedding many-to-one dilation 5 into $G_{k\text{-}1}$ is true. Now we will prove that this is true for *k*=*n*.

Case 1: ***k*** is pair, there are two sub-cases:

Case a: Let *A* and *B* Є *V*, where the address of a given node is *A*=0 $Pref$ $a_{k\text{-}4}a_{k\text{-}3}a_{k\text{-}2}a_{k\text{-}1}$= $Pref_1^{'}$ $a_{k\text{-}4}a_{k\text{-}3}a_{k\text{-}2}a_{k\text{-}1}$ as $Pref_1^{'}$ =0 $Pref$ and B= $Pref_1^{'}$ $b_{k\text{-}4}b_{k\text{-}3}b_{k\text{-}2}b_{k\text{-}1}$.

The embedding of (*A, B*) Є *U* into the first super node $G_k[k, k]$ and all the edges (*A, B*) Є *U* such that, *A*=1 $Pref$ $a_{k\text{-}4}a_{k\text{-}3}a_{k\text{-}2}a_{k\text{-}1}$ or *A*= $Pref_2$ $a_{k\text{-}4}a_{k\text{-}3}a_{k\text{-}2}a_{k\text{-}1}$, where $Pref_2$ =1 $Pref$, and the node *B*= $Pref_2$ $b_{k\text{-}4}b_{k\text{-}3}b_{k\text{-}2}b_{k\text{-}1}$ are embedded into the second super node $G_k[k,1]$ in other words, into $G_{k\text{-}1}$. However, the dilation of embedding into $G_{k\text{-}1}$ is 5 hypothesis of deduction.

Case b: Let *A* and *B* Є *V*, *A*=0 $Pref$ $a_{k\text{-}4}a_{k\text{-}3}a_{k\text{-}2}a_{k\text{-}1}$ or *A*= $Pref_1$ $a_{k\text{-}4}a_{k\text{-}3}a_{k\text{-}2}a_{k\text{-}1}$ as $Pref_1$ =0 $Pref$ and *B*= $Pref_1$ $b_{k\text{-}4}b_{k\text{-}3}b_{k\text{-}2}b_{k\text{-}1}$. The Embed_node(A) algorithm embed all nodes *A* into a super node $G_k[k, k]$ and all nodes *B* into a super node $G_k[k,1]$. The different edges of TQ_k are embedded into different paths. The first node of every edge of TQ_k is embedded into the super node $G_k[k, k]$ and the last node of this edge is embedded into the super node $G_k[k, 1]$, that is to say, we use the results of the Embed3_edge(nodedep, nodearr) algorithm outlined in case 1.

Note that cases 2, 3 and 4 are not given in this paper. In all cases the dilation is 5. *Pref* is a function which extracts sub-string of (*n*-5) characters starting from the second character of highest weight.

Case 2: ***k*** is odd, there are two sub-cases:

Case a: Let k=2m+1, where m Є□ , TQ_k is produced by two copies of $0TQ'_{2k}$ and $1TQ'_{2k}$. Suppose that for N=2k we have $0TQ'_N$, $1TQ'_N$, i.e., $00TQ'_{N-1}$, $01TQ'_{N-1}$, $10TQ'_{N-1}$ and $11TQ'_{N-1}$. Let A and B Є V, where $A=A_1A_2$, such that, A_1=(00, 01, 10, 11), A_2= *Pref* $a_{N\text{-}4}a_{N\text{-}3}a_{N\text{-}2}a_{N\text{-}1}$ *as* $Pref_1 = A_1$ *Pref* , hence, A= $Pref_1$ $a_{N\text{-}4}a_{N\text{-}3}a_{N\text{-}2}a_{N\text{-}1}$ and B= $Pref_1$ $b_{N\text{-}3}b_{N\text{-}2}b_{N\text{-}1}b_N$. The embedding of ($A$, B) Є U is into the first super node $G_N[N, N]$ if A_1=(00), it is into the second super node $G_N[N, 1]$ if A_1=(10), it is in the third super node $G_N[N, 3]$ if A_1=(01), and it is in the fourth super nodes $G_N[N, 2]$ if A_1=(11). The dilation in all super nodes is 5 (hypothesis deduction).

Case b: Let A, B Є V and $A=A_1A_2$, $B=B_1B_2$ as (A_1, B_1)=(00, 01), (00, 10), (01, 11), (10, 11) and A_2= $Pref_1$ $a_{N\text{-}4}a_{N\text{-}3}a_{N\text{-}2}a_{N\text{-}1}$, B_2= $Pref_1$ $b_{N\text{-}3}b_{N\text{-}2}b_{N\text{-}1}b_N$. The embedding of ($A$, B) Є U is in a path between two super nodes ($G_N[N, N]$,$G_N[N, 3]$), ($G_N[N, N]$,$G_N[N, 1]$), ($G_N[N, 3]$,$G_N[N, 2]$), ($G_N[N, 1]$,$G_N[N, 2]$). Each given super node contains exactly 2^{l-1} G_4. In other words, case 1 or 2 is used because the first node of the different path is located in one node of G_4 of the super node $G_N[N,N]$, the last node is located in one node of G_4 of the super node $G_N[N,3]$. Or for all edges of TQ_k having at the first extremity a node prefixed by 00 *Pref* , and at the second extremity a node prefixed by 01 *Pref* , for instance, case 1 is used. Note that cases 2, 3 and 4 are not given in this paper. In all cases the dilation is 5.

5. Proposed New Approach

In this section, we propose a new approach that allows forming Cluster-Tree networks without regard to the density of the network. Our approach is not based on SuperFrames or even Beacon frame transmission scheduling.

Indeed, scheduling algorithms are complex to implement and to execute by sensor nodes. The main innovation of this paper is to introduce a scheduling free mechanism enabling the construction of beacon Cluster-Tree networks. Using this approach, all the nodes are sharing the same SuperFrame which avoids inter-beacon networks perturbation. Indirect beacon frame collision can be solved by the following two approaches:

The reactive approach: The network is started normally and the coordinators do not do much to prevent the beacon frames from colliding. Once a collision occurs, the node will start orphan scans to try to resynchronize with its coordinator.

However, if after a number of orphan scans, the node is still unable to correctly receive the beacon frame, it initiates a beacon conflict command; beacon coordinators receiving this command will adjust their beacon transmission time in order to solve the problem. This approach is simple, but, the recovery from a beacon conflict can take a long time.

The proactive approach: In the proactive approach, coordinators try to avoid beacon frame conflicts before starting their beacon frame transmissions. A beacon coordinator listens to the channel and collects its neighbors beacon frame transmission time.

However, if a beacon frame collision is reported, the network is able to solve the problem using the reactive approach. Notice that this approach is more complicated than the first one.

In our approach, the Cluster-Tree network is constructed based on the following: The *BO* is a parameter used by the associated nodes for calculating the beacon interval, which is computed using this formula (Benakila, et al., 2010),

$$BI = aBaseSuperFrameDuration \times 2^{BO} \tag{4}$$

with *aBaseSuperFrameDuration* = 60 symbols.

The *BO* value should be between 0 and 14. A *BO* with a value of 15 indicates that the device operates in non-beacon mode. The *SO* is a parameter used for calculating the active period duration,

$$SD = aBaseSuperFrameDuration \times 2^{SO} \tag{5}$$

with $0 \leq SO \leq BO \leq 14$

- The same *BO* and *SO* values is used for all the nodes of the network (Part 15.4, 2006).

- All the nodes are synchronized thanks to beacon frame transmissions.
- All the nodes transmit during the same SuperFrame.

To enable collision-free beacon transmissions, we are adopting a novel approach, described in this section.

The beacon frame is divided into two parts:

- The common part: It is the part that does not change from a beacon coordinator to another. It contains the *SO* and the *BO* parameters.
- The specific part: It is the part specific to a beacon coordinator, i.e., changes from a given beacon coordinator to another.

Each part is a separate frame. Node synchronization is achieved by the transmission of the common part. It gives the start signal to sensor nodes to begin the beacon mode SuperFrame. The common part contains only synchronization information (*BO* and *SO*). The specific part is transmitted during the CAP period and it contains the traditional beacon information (GTSs organization, Pending addresses, etc.). The PAN Coordinator broadcasts the beacon common's part. When the beacon's common part is received, a beacon coordinator begins its SuperFrame and forwards the same frame (i.e., the beacon's common part) to its neighbor nodes. Using this mechanism, beacon coordinators at the same level of the Cluster-Tree transmit the common part at the same time. This should not cause a reception problem if a node receives more than one frame at the same time.

Indeed, all the beacon routers are broadcasting the same frame, the same bit configuration which means that all the beacon routers are transmitting the same RF signal which can be assimilated to a multipath reception.

Multipath propagation occurs when RF signals take different paths from a source to a destination. A part of the signal goes to the destination while another part bounces off an obstruction, and then arrives at the destination. As a result, a part of the signal encounters delay and travels a longer path to the destination. Multipath can be defined as the combination of the original signal plus the duplicate wave fronts that result from the waves' reflection off obstacles between the transmitter and the receiver (CISCO, 2008).

Multipath propagation occurs even with only one transmitter and one receiver. Nowadays receivers are able to retrieve the information from a RF signal perturbed by multipath signals since interference, and not multipath fading, is the primary cause of unpredictable performance (Raman et al., 2009). Thus, a node receiving the beacon common part from more than one beacon coordinator (i.e., more than one RF signal) should be able to retrieve the common part information since the node is able to deal with multipath Radio Frequency (RF) signals (Benakila et al., 2009). The transmission mechanism is described by the use of the Pancake topology as explained in section 3. An example of the embedding Pancake graph of TQ_5 into G_5 is represented in Figure 6. The main advantage of this topology is the gain in dilation compared to the standard used topology. The transmission mechanism is described in section 3 by using the new topology of the embedding Twisted Hypercube into Pancake graphs. As an example refer to the Figure 6 where we represent the embedding graph of TQ_5 into G_5. The main advantages of this topology compared to the standard one is the dilation and the fault tolerance and we described these in the section 4.

6. Narrowband Multipath Fading Model

If a single pulse is transmitted over a multipath channel then the received signal will appear as a pulse train, with each pulse in the train corresponding to the line-of-sight (LOS) component or a distinct multipath component associated with a distinct scatter.

The time delay spread of a multipath channel can result in significant distortion of the received signal. We say that two multipath components with delay and τ_2 are resolvable if their delay difference significantly exceeds the inverse signal bandwidth: $|\tau_1 - \tau_2| \square 1/B_u$. Multipath components that do not satisfy this resolvability criterion cannot be separated out at the receiver, and thus these components are non-resolvable. These non-resolvable components are combined into a single multipath component with amplitude and phase corresponding to the sum of the different components (Goldsmith, 2005). Another characteristic of the multipath channel is the time varying nature. Time variation arises because either the transmitter or the receiver is moving and hence the location of reflectors in the transmitting path will change over time (Goldsmith, 2005). Let the transmitted signal be,

$$\begin{aligned} s(t) &= \mathrm{Re}\left\{u(t)e^{j2\pi f_c t}\right\} = \mathrm{Re}\left\{u(t)\right\}\cos(2\pi f_c t)\square \\ &\mathrm{Im}\left\{u(t)\right\}\sin(2\pi f_c t) \end{aligned} \tag{6}$$

where *u(t)* is the equivalent lowpass signal for *s(t)* with bandwidth B_u and where f_c is the carrier frequency. Neglecting noise, the received signal is the sum of the line-of-sight path and all resolvable multipath components (Goldsmith, 2005),

$$r(t) = Re\left\{\sum_{n=0}^{N(t)} \alpha_n(t)u(t-\tau_n(t))e^{j(2\pi f_c(t-\tau_n(t))+\phi_{D_n}(t)}\right\} \tag{7}$$

where: n = 0 is the LOS component.

N(*t*) is the number of the resolvable multipath components. $\tau_n(t)$ is the delay of the n^{th} multipath component. It depends on the path length $r_n(t)$ and could be written: $\tau_n(t) = r_n(t) \div c$ where the $\alpha_n(t)$ is the amplitude. $\phi_{D_n}(t)$ is the Doppler phase shift. We assume that all the nodes in the network are static, thus $\phi_{D_n}(t) = 0$. For simplification, let (since the IEEE 802.15.4 is a narrowband channel), the delay spread T_m satisfies $T_m << B^{-1}$. Thus $\tau_i \le T_m$ for all i, so $u(t-\tau_i(t)) \approx u(t)$.

Equation (7) could be written as:

$$r(t) = \mathrm{Re}\left\{u(t)e^{j2\pi f_c t}(\sum_{n=0}^{N(t)} \alpha_n(t)e^{-j\phi_n(t)})\right\} \tag{8}$$

The complex scale factor in large parentheses is independent of the transmitted signal and of the equivalent lowpass signal *u(t).* If we consider *s(t)* to be an unmodulated carrier with random phase offset ϕ_0 then s(t) could be written as:

$$s(t) = \mathrm{Re}\left\{e^{j(2\pi f_c t+\phi_0}\right\} \tag{9}$$

The received signal *r*(*t*) will be:

$$r(t) = \mathrm{Re}\left\{\left(\sum_{n=0}^{N(t)} \alpha_n(t)e^{-j\phi_n(t)}\right)e^{j2\pi f_c t}\right\} \tag{10}$$

$$r(t) = r_I(t)\cos 2\pi f_c t - r_Q(t)\sin 2\pi f_c t \tag{11}$$

where,

$$r_I(t) = \sum_{n=0}^{N(t)} \alpha_n(t)\cos\phi_n(t) \tag{12}$$

$$r_Q(t) = -\sum_{n=0}^{N(t)} \alpha_n(t)\sin\phi_n(t) \tag{13}$$

And the phase $\phi_n(t)$ as:

$$\phi_n(t) = 2\pi f_c \tau_n(t) - \phi_0 \tag{14}$$

Now, considering P transmitters transmitting the same RF signal, the received signal *r*(*t*) is,

$$r(t) = r_I(t)\cos 2\pi f_c t - r_Q(t)\sin 2\pi f_c t \tag{15}$$

Where,

$$r_I(t) = \sum_{l=0}^{P}\sum_{n=0}^{N(t)} \alpha_{\ln}(t)\cos\phi_{\ln} \tag{16}$$

$$r_Q(t) = \sum_{l=0}^{P}\sum_{n=0}^{N(t)} \alpha_{\ln}(t)\sin\phi_{\ln}(t) \tag{17}$$

$\alpha_{\ln}(t)$ is the amplitude of the n^{th} multipath component transmitted by the l^{th} transmitter. $\phi_n(t)$ is the phase of the n^{th} multipath component transmitted by the l^{th} transmitter.

7. Model Multipath Simulation

In this section, a Matlab/Simulink model is developed to test the mechanism and to measure the impact of a multipath reception on the BER value. The model is represented in Figure 7; it consists of a transmitter, a channel, and a receiver. The transmitter is sending an OQPSK modulated signal over a channel constituted of an AWGN channel and a collection of transmitting paths. Each path is emulating an independent transmitter and each transmitted signal is propagated through 3 paths with variable delays, thus, simulating multipath phenomena (see Figure 9). All transmitted signals and multipaths are combined at the receiver (see Figure 8), which filters the signal in order to retrieve the information. Figure 10 and Figure 11 show respectively the scatter plot and the eye diagram obtained from this simulation in the case of the presence of only a LOS signal. In order to measure the impact of receiving a signal from several transmitters on the BER value, we assume one LOS transmitter and we increase the number of non-LOS transmitters. For each number of non- LOS transmitters, we measure the BER value at several noise levels. Each transmitter line contains three delayed versions of the signal. Figure 12 shows the BER results for one transmitter (LOS transmitter) two transmitters (one LOS transmitter and a non- LOS transmitter), three transmitters (one LOS transmitter and two non-LOS transmitters), and four transmitters (one LOS transmitter and three transmitters). From Figure 12, we can clearly see that, in the presence of an LOS component, the number of non-LOS transmitters does not significantly affect the BER value.

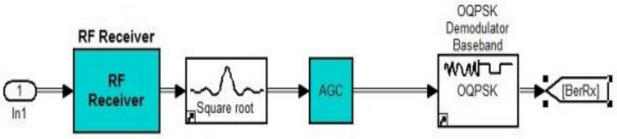


Figure 7. Simulink simulation model

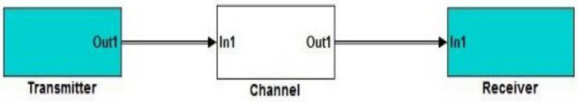


Figure 8. Receiver architecture

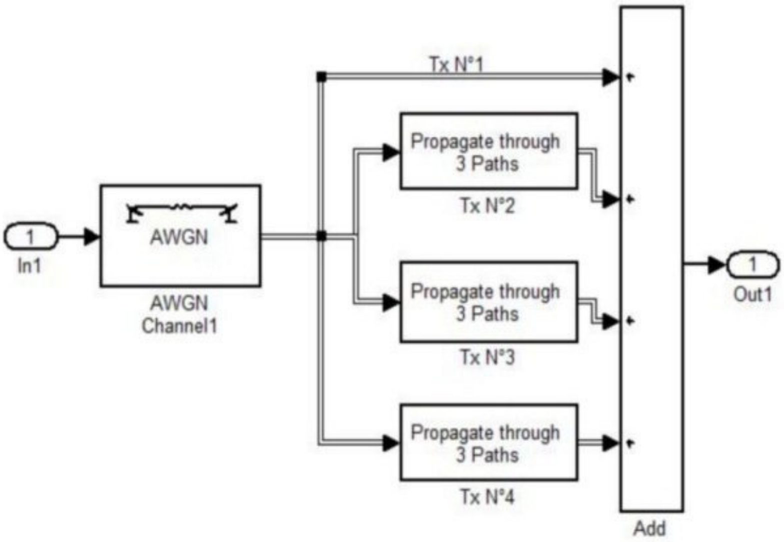

1
In1
AWGN
AWGN Channel1
Tx N°1
Propagate through 3 Paths
Tx N°2
Propagate through 3 Paths
Tx N°3
Propagate through 3 Paths
Tx N°4
Add
1
Out1

Figure 9. Channel architecture

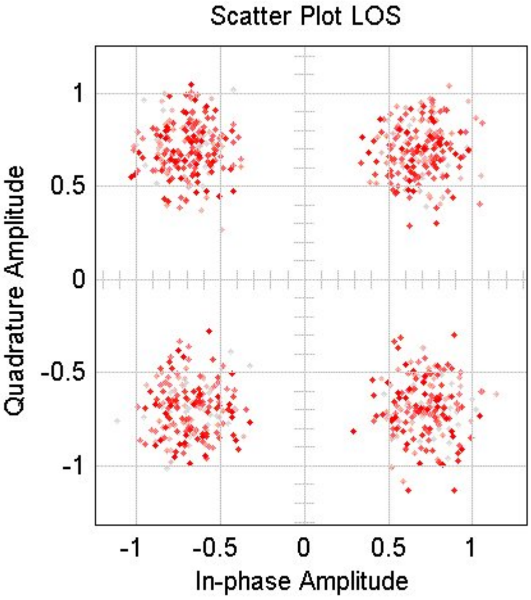


Figure 10. LOS transmission scatter plot

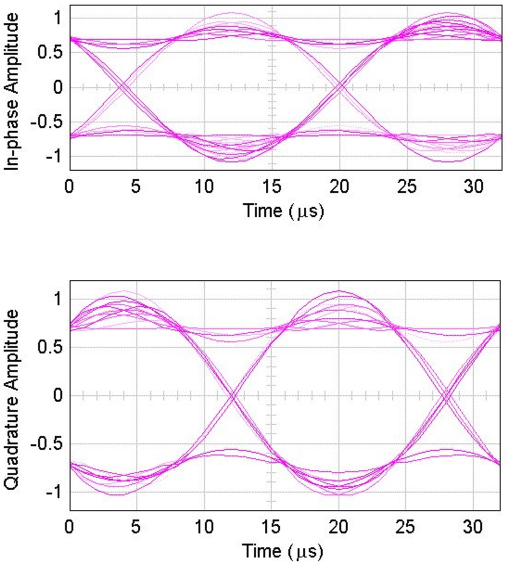


Figure 11. LOS eye diagram

As a second step, we are interested in measuring the BER value in the presence of several LOS transmitters.

In this simulation, we measure the BER value for each number of present transmitters and at several noise levels. Figure 13 shows the BER values obtained from this simulation, we can conclude that the presence of more than one LOS transmitter affects significantly the BER value which means that to guarantee an acceptable BER value, LOS transmitter should be present.

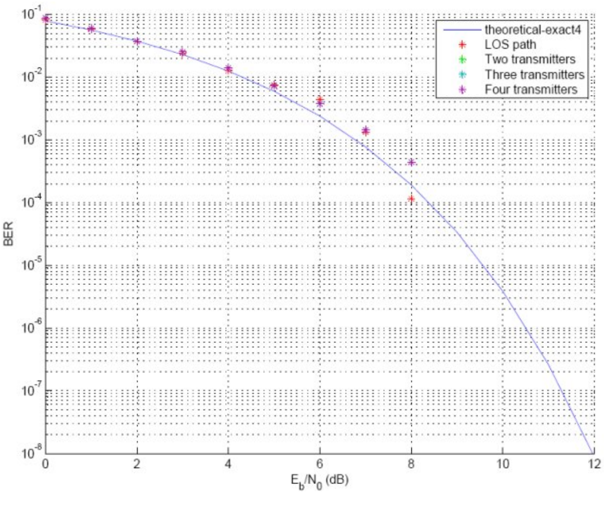


Figure 12. BER in presence of several transmitters but only one LOS transmitter

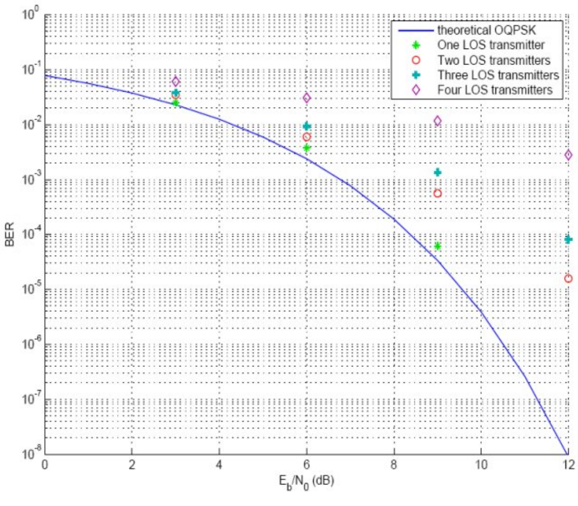


Figure 13. BER in presence of several LOS transmitters

8. Test Bed Bench

The core of the presented approach is the exploitation of multipath phenomena to avoid beacon transmission scheduling. Assuming that RF receivers are able to deal with multipath signals to retrieve the information, they can retrieve the information when receiving an RF signal transmitted by different nodes at the same time since each transmitted signal will be treated as a multipath signal by the receiver.

To stress this point, we conceived a real experimentation using the TelosB motes. Crossbow's TelosB mote (Figure 1) is an open source platform designed to enable cutting-edge experimentation for the research community. The principle is to send the same frame by several nodes to one receiver. For visual consideration, we chose a MicroChip sniffer as a second receiver. An example of the network architecture with 3 routers is presented in Figure 14.

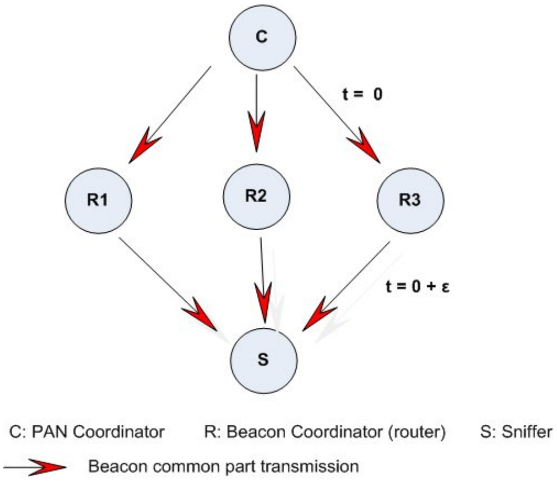
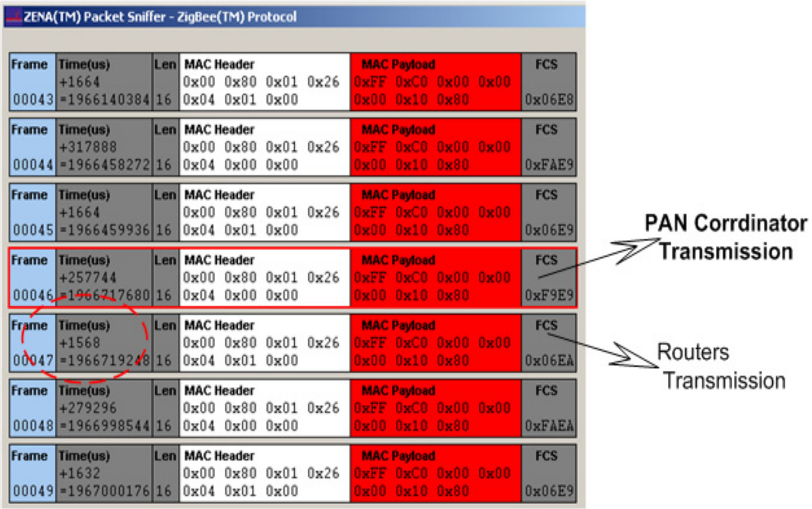

C
t = 0
R1
R2
R3
t = 0 + ε
S
C: PAN Coordinator
R: Beacon Coordinator (router)
S: Sniffer
Beacon common part transmission

Figure 14. A test network architecture with 3 routers

The receiver is in the transmission range of the PAN Coordinator and the three routers, i.e., the sniffer's software shows the frames transmitted by the coordinator and the routers. When a router receives the frame from the PAN Coordinator, it retransmits it immediately, i.e., all routers are transmitting the same frame at the same time. As is shown in Figure 14, the sniffer receives two frames: the first one is the frame transmitted by the PAN Coordinator and the second one is the frame transmitted by the routers. The sniffer receives only one frame from the routers even if there is more than one transmission. Consequently, the receiver considers all the transmissions as only one transmission (see Figure 15).


ZENA(TM) Packet Sniffer - ZigBee(TM) Protocol
PAN Corrdinator Transmission
Routers Transmission

Frame	Time(us)	Len	MAC Header	MAC Payload	FCS
00043	+1664 =1966140384	16	0x00 0x80 0x01 0x26 0x04 0x01 0x00	0xFF 0xC0 0x00 0x00 0x00 0x10 0x80	0x06E8
00044	+317888 =1966458272	16	0x00 0x80 0x01 0x26 0x04 0x00 0x00	0xFF 0xC0 0x00 0x00 0x00 0x10 0x80	0xFAE9
00045	+1664 =1966459936	16	0x00 0x80 0x01 0x26 0x04 0x01 0x00	0xFF 0xC0 0x00 0x00 0x00 0x10 0x80	0x06E9
00046	+257744 =1966717680	16	0x00 0x80 0x01 0x26 0x04 0x00 0x00	0xFF 0xC0 0x00 0x00 0x00 0x10 0x80	0xF9E9
00047	+1568 =1966719248	16	0x00 0x80 0x01 0x26 0x04 0x01 0x00	0xFF 0xC0 0x00 0x00 0x00 0x10 0x80	0x06EA
00048	+279296 =1966998544	16	0x00 0x80 0x01 0x26 0x04 0x00 0x00	0xFF 0xC0 0x00 0x00 0x00 0x10 0x80	0xFAEA
00049	+1632 =1967000176	16	0x00 0x80 0x01 0x26 0x04 0x01 0x00	0xFF 0xC0 0x00 0x00 0x00 0x10 0x80	0x06E9

Figure 15. The received packets from the sniffer

The receiver's capability to process multipaths avoids the introduction of beacon or SuperFrames scheduling mechanisms. In order to measure the impact of the distance and the number of transmitting nodes on the beacon common part frame receiving mechanism, a test was conducted and it consists of a receiver node and a variable number of nodes at variable distances in an indoor (office) environment and an outdoor environment. For each measure, transmitter nodes are transmitting 500 beacon common frames. Table 7 shows the amount of correctly received packets from the 500 transmitted packets.

Table 7. Beacon frame's common part reception error rate in indoor and outdoor environment

Indoor environment						**Outdoor environment**					
Meter/nodes	1	2	3	4	5	Meter/nodes	1	2	3	4	5
2	463	487	494	495	491	2	489	491	490	493	495
3	148	158	239	305	366	*3*	317	317	412	464	472
4	26	72	173	242	296	*4*	137	270	390	423	441

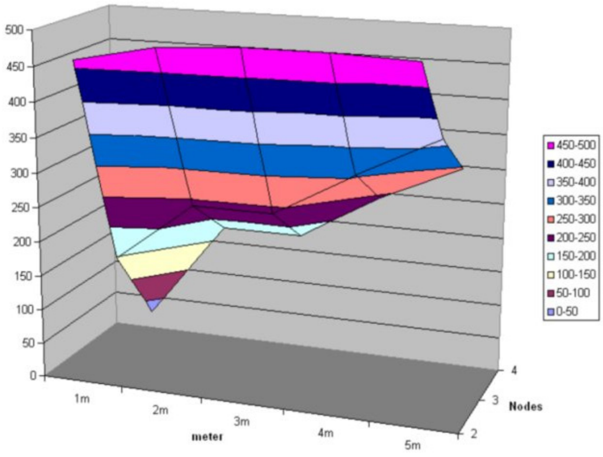


Figure 16. Received common part packets (Number of packets 0-500) versus distance (Meter) and number of transmitters in an indoor environement (Nodes)

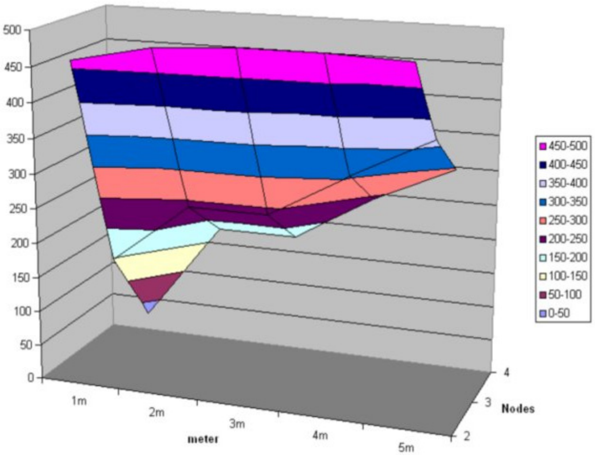


Figure 17. Received common part packets (Number of packets 0-500) versus distance (Meter) and number of transmitters in an outdoor environement (Nodes)

These results are plotted in Figure 16 for an indoor environment and in Figure 17 for an outdoor environment.

From Figure 16 and Figure 17, we can clearly see that the packet loss is greater in an indoor environment than in an outdoor environment. This is a consequence of the number of multipath components arriving at the receiver node. Indeed, generally, in outdoor environments, scattering objects are less present than in indoor environments. The second test is deployed inside an office area, we fix the receiver's position and we vary the number of beacon's common part frame transmitters.

Figure 18 is a part of the floor map and it shows node deployment (one receiver and 10 transmitters). Each beacon transmitter is transmitting 500 packets and we measure the number of correctly received packets by the receiver. The position of transmitter nodes is chosen randomly and for each number of transmitter nodes several measures are made with different transmitter node position.

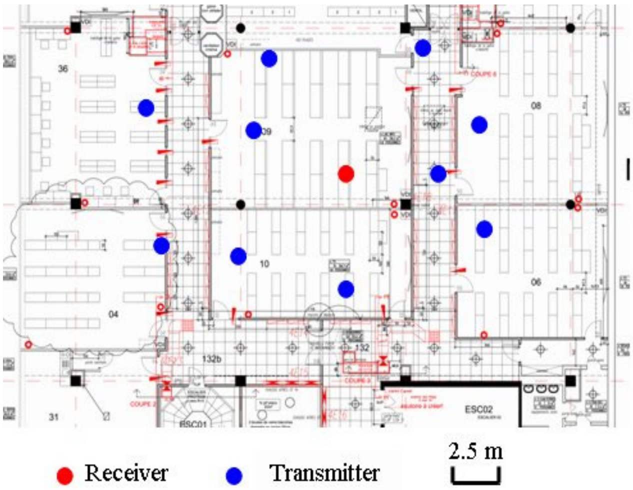


Figure 18. Node deployement on the floor

Figure 19 shows the number of received beacon frames versus the number of transmitters present in the receiver's range. We can see that, in a real case where beacon coordinators are deployed in an indoor environment, the receiver node is well tolerant to simultaneous transmissions which make this mechanism very interesting for constructing beacon Cluster-Tree topologies based on the Pancake graphs.

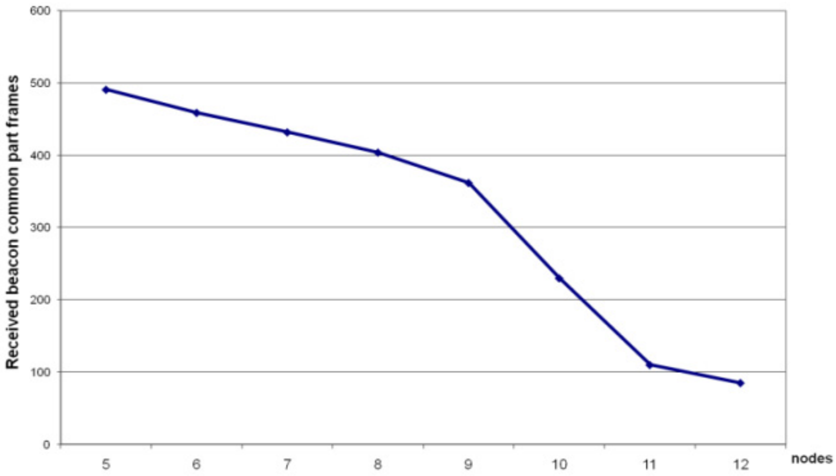


Figure 19. BER versus the number of transmitter nodes in the area of the receiver

9. Conclusion

In this paper, we present a new approach for the construction of ZigBee/IEEE 802.15.4 Cluster-Tree networks based on Pancake graphs. The presented approach tackles the problems of beacon frame and SuperFrame scheduling and allows the construction of a Cluster-Tree topology without introducing constraints on SuperFrame structure. We propose a collision-free beacon transmission approach that exploits nodes capabilities in extracting the information from a signal perturbed by simultaneous transmissions of several beacon coordinators. These transmitted RF signals are seen as reflected RF signals since all the beacon coordinators are broadcasting the same signal. We show from experiments and simulations that a receiver is able to retrieve the information from a combination of several transmitted signals. In this study we conclude that it is very significant to investigate the possibility of embedding a different architecture into a Pancake. It presents also a many-to-one embedding of Multiply-Twisted Hypercube into Pancake with dilation 5. The study of the dilation of this new function of many-to-one embedding is realized in three steps. The first step treated the edges of Twisted Hypercube embedded into the same 3-Pancake with a dilation 3. The second step is that all edges of Multiply-Twisted Hypercube which are embedded into the same 4-Pancake with a dilation 4. The latter step is embedding many-to-one dilation 5 of all edges of Multiply-Twisted Hypercube into the paths between two different super nodes of Pancake, and this latter step is used for the construction of Beacon mesh networks based on Pancake graphs for our WSN. Future works will deal with adapting the presented approach to enable the construction of Beacon mesh networks based on Pancake graphs. For time sensitive applications a GTS collision avoidance mechanism must be introduced to grant GTS traffic.

References

Abraham, S., & Padmanabhan, K. (1991). Twisted cube: a study in asymmetry. *In Parallel Distributions Computer, 13,* 104-110.

Akers, S.B., & Krishnamurthy, B. (1989). A group-theoretic model for symmetric interconnection networks. *In IEEE transactions on Computers, 38*(4), 555-566. http://dx.doi.org/10.1109/12.21148

Benakila, I., George, L., & Femmam, S. (2009). A Beacon-Aware Device for the Interconnection of ZigBee Networks. *Elsevier FET'2009, 8th IFAC Int. Conf. on Fieldbuses and networks in Industrial and Embedded Systems, 8,* 123-130, Ansan, Republic of Korea.

Benakila, I., George, L., & Femmam, S. (2010). A Beacon Cluster-Tree Construction Approach for ZigBee/IEEE802.15.4 Networks. *The 4th IEEE International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies UBICOMM*, 130-138, Florence, Italy. ISBN: 978-1-61208-100-7.

Chalhoub, G., & Misson, M. (2010). Cluster-tree based energy efficient protocol for wireless sensor networks. *In Proceed. ICNSC, International Conference on Networking, Sensing and Control*, 664-669.

CISCO. (2008). Multipath and diversity. Retrieved from http://www.cisco.com/application/pdf/paws/27147/multipath.pdf

Dondi, D., Bertacchini, A., Brunelli, D., Larcher, L., & Benini, L. (2008). Modeling and Optimization of a Solar Energy Harvester System for Self-Powered Wireless Sensor Networks. *IEEE Trans. on Industrial Electronics, 55*(7), 2759-2766. http://dx.doi.org/10.1109/TIE.2008.924449

Goldsmith, A. (2005). *Wireless communications*. Cambridge University Press.

Gungor, V. C., & Hancke, G. P. (2009). Industrial Wireless Sensor Networks: Challenges, Design Principles, and Technical Approaches. In *IEEE Transaction on Industrial Electronics, 56*(10), 4258-4265. http://dx.doi.org/10.1109/TIE.2009.2015754

Gupta, G., & Younis, M. (2003). Fault-tolerant clustering of wireless sensor networks. *In Proc. WCNC*, 1579-1584.

Heydari, M. H., & Sudborough, I. H. (1997). On the diameter of Pancake network. *In Algorithms Journal, 25*(1), 67-94.

Hsieh, S. Y., Chen, G. H., & Ho, C. W. (1998). Embed longest rings onto star graphs with vertex faults. *In Inter. Conference on Parallel Processing Proceeding,* 140-147.

Huan, W. T., Tan, J. M., Hung, C. N., & Hsu, L. H. (2002). Fault-tolerant Hamiltonocity of twisted cubes. *In Parallel Distributions and Computer, 62,* 582-604.

Hung, Hsu, H., Liang, K., & Hsu, L. (2003). Ring embedding in faulty Pancake graphs. In *Inf. Proc. Letters, 86,* 271-275.

IEEE standard. (2006). Part 15.4: Wireless medium access control (MAC) and physical layer (Phy) specifications for low-rate wireless personal area networks (WPANs).

Ilyas, M., & Hahgoub, I. (2004). *Handbook of sensor networks: Compact wireless and wired sensing systems*. CRC Press. ISBN:0849319684. http://dx.doi.org/10.1201/9780203489635

Junghee, H. (2009). Global Optimization of ZigBee Parameters for End-to-End Deadline Guarantee of Real-Time Data. *IEEE Sensors Journal, 9*(5), 512-514. http://dx.doi.org/10.1109/JSEN.2009.2016590

Kanevsky, C. A., & Feng, C. (1995). On the embedding of cycles in Pancake graphs. *In Parallel Computer Journal*, 923-926.

Koubaa, A., Alves, M., & Tovar, E. (2006). Modeling and worst-case dimen-sioning of cluster-tree wireless sensor networks. *In Proc. RTSS*, 412-421.

Kulasinghe, P., & Bettayeb, S. (1995). Multiply-Twisted Hypercube with Five or More Dimensions is not Vertex-Transitive. *In Information Proces Letters, 5,* 33-36.

Leighton, T. (1992). *Introduction to parallel algorithms and architectures: Arrays, Trees, Hypercubes*. Mogan Kaufman Publisher Inc.

Lin, C. K., Tan, J. J. M., M., Huang, H. M., Hsu, D., F., & Hsu, L. H. (2008). Mutually Independent Hamiltonianicity of Pancake Graphs and Star Graphs. *The Int. Sym. on Parallel Architectures, Algorithms, and Networks* (*i-span*), 151-158. http://doi.ieeecomputersociety.org/10.1109/I-SPAN.2008.41

Lin, J. C., Yang, J. S., Hsu, C. C., & Chang, J. M. (2010). Independent spanning trees VS edge-disjoint spanning trees in locally twisted cubes. *Information Processing Letters Journal, Elsevier, 110*(10), 414-419. http://dx.doi.org/10.1016/j.ipl.2010.03.012

Mhatre, V., & Rosenberg, C. (2004). Design guidelines for wireless sensor networks: communication, clustering and aggregation. *Ad Hoc Networks, 2*(1), 45-63. http://dx.doi.org/10.1016/S1570-8705(03)00047-7

Miller, Z., Pritikin, D., & Sudborough, I. H. (1994). Near embedding of Hypercubes into Cayley graphs on the symmetric group. *In IEEE Transactions on Computers, 43*(1), 13-22. http://dx.doi.org/10.1109/12.250605

Morales, L., & Sudborough, I. (1996). Comparing Star and Pancake Networks. *Symposium Parallel and Distributed Processing*. http://doi.ieeecomputersociety.org/10.1109/SPDP.1996.570392

Raman, B., Chebrolu, K., Gokhale, D., & Sen, S. (2009). On the Feasibility of the Link Abstraction in Wireless Mesh Networks. *In IEEE/ACM Trans. on Networking, 17*(2), 528-541. http://dx.doi.org/10.1109/TNET.2009.2013706

Senoussi, H., & Lavault, C. (1997). Embeddings into the Pancake interconnection network. High Performance Computing and Grid. In Asia Pacific Region. *International Conference on High-Performance Computing on the Information Superhighway, HPC-Asia'97*. http://doi.ieeecomputersociety.org/10.1109/HPC.1997.592125

Xu, M. (2010). Edge-pancyclicity and Hamiltonian connectivity of twisted cubes. *Springer Acta Mathematica Sinica Journal, 26*(14), 1315-1322. http://dx.doi.org/10.1007/s10114-010-8581-x

Yoo, S., Chong, P. K., Kim, D., Doh, Y., Pham, M. L., Choi, E., & Huh, J. (2010). Guaranteeing Real-Time Services for Industrial Wireless Sensor Networks with IEEE 802.15.4. *IEEE Trans on Industrial Electronics, 57*(11), 3868-3876. http://dx.doi.org/10.1109/TIE.2010.2040630