International Journal of Internet of Things 2016, 5(1): 17-28
DOI: 10.5923/j.ijit.20160501.03

A Geometric Structure for Internet of Things Indexing

Telesphore Tiendrebeogo[1,*], Oumarou Si é[2]

[1]Department of Mathematics and Computer Science, Polytechnic University of Bobo Dioulasso, Bobo Dioulasso, Burkina Faso
[2]Department of Computer Science, University of Ouagadougou, Ouagadougou, Burkina Faso

Abstract As the number of wireless devices increases and their size becomes smaller, there can be more interaction between everyday objects of our life. The proliferation of these devices in a communicating-actuating network creates the Internet of Things (IoT). We currently attend the deployment of a new generation of interconnected objects equipped with capacities of communication, detection and activation (wireless transport networks of information, RFID (Radio Frequency Identification System), WSAN (Wireless Sensor and Actuator Network), etc) for many applications. Thus, the interconnection of objects equipped with advanced capacities of treatment will lead to a revolution in term of creation and availability of service and deeply will change our way of interacting with our environment. In short, we attend an increasing mixture between the physical world and the virtual world. This paper presents a vision of cloud centric for worldwide implementation of Internet of Things. We propose a structure of tree hyperbolic, using the properties of the hyperbolic geometry and which allows the interconnection then the indexation of the various objects associated with this standard system. These mechanisms are based on greedy routing algorithm in the Poincar é disk model. We show how this virtualization of the objects in our structure which is not other than a Distributed Hash Table (DHT), allows to make scalable, reliable and resisting the communication between the entities even in a dynamic context of networks.

Keywords Internet of Thing, DHT, Hyperbolic Tree, Indexing, Poincar é Disk, Greedy Routing, Routing by holes bypass

1. Introduction

The internet of things [1] refers to the only addressable objects and their virtual representations in a structure similar to the Internet. Such objects can be bound with information on them, or can transmit data of real time sensor of their state or other useful properties associated with the object. RFID Technology [2-4] sees generally as a key activator of Internet of things, because of its capacity to track a large number of only recognizable objects with the use of Electronic Product Code (EPC). However, other sorts of omnipresent devices of sensor, bar codes, or 2D codes can be also used to allow Internet of Things (IoT). The vision of the Internet inclusion of the things of simple more complex and more intelligent, objects of communication, involves to implement multi-field approach of new technologies, concepts and models (Development of integrated circuits, management of energy, systems and principles of communication, incorporated systems and packaging, acquisition and data processing, experiment on the ground) and supposes to take of many scientists, commercial and challenges of techniques.

Besides, a radical evolution of current Internet in a Network of the connected objects which collects not only information of the environment and interacts with the physical world (the activation/command/control), but also uses existing standards of Internet to supply services for the transfer of the information, the analytics, the applications and the communications. Fuelled by the frequency of devices allowed by the wireless technology opened as Bluetooth, RFID, services of WiFi and phone data as well as the incorporated sensor and actuator nodes, IoT went out of its early childhood and is about to transform current static Internet in Future completely integrated Internet [5]. Internet revolution led to the interconnection between people on an unprecedented scale and to the speed.

Figure 1. Internet of Things Schematic showing the end users and application areas based on data

* Corresponding author:
tetiendreb@gmail.com (Telesphore Tiendrebeogo)

Published online at http://journal.sapub.org/ijit

The following revolution will be the interconnection between objects and the creation of an intelligent environment. Only in 2011, the number of devices connected on the planet caught up the real number of people living there. At present there is 9 billion connected devices and we wait in the fact that it reaches 32 billion devices before 2020 [6].

Figure 1 presents architecture of the Internet of things.

In this paper, we propose a structure of tree hyperbolic using the properties of the geometry hyperbolic and embedded in the Poincaré disc model, allowing to ensure the scalability, reliability and resistance in the communication between entities. Our work is organized as follows. In Section II, we present related work. In Section III, we present some properties of the hyperbolic geometry. Next, Section IV shows the practical use of the hyperbolic geometry associated to the Poincaré disc model. Section V, presents its mechanism of naming and binding of the IoT devices in our system. In section VI, we show the management of our IoT system essentially in term of storage et lookup process. Next, Section VII presents experimental study of reliability and scalability properties. Finally, section VII comes to conclude our work and to propose the future work.

2. Related Work

The discovery of Virtual Entities and the Services IoT is an essential aspect of Internet of Things. Contrary in small-scale silos of application making up an Intranet of Things, the applications and the services cannot be configured as regards fixed set services. At first, this is due to the underlying dynamics of the system IoT resulting from the mobility of physical entities and devices IoT, as well as the changeable availability of services because of the constraints of the underlying resources and from devices. Secondly, new sorts of applications and the services can be allowed, if the application can be written for a generic scenario, for example, an intelligent environment of the city, which should then work the same way to Berlin and Madrid without vast configurations.

The proposed approach [7] is based on the creation and the preservation of a topology of cover which connects logically all the participating nodes/objects in the physical network (see Figure 2). Any node which connects to the physical network has to join the virtual infrastructure. In the virtual infrastructure, multiple networks of cover can be created with everything or a subset of the nodes of network. During the initiating of the physical network, the connected devices must be able to become identified. On the establishment of connectivity and the capacity of communication of multi-hop among participant's nodes, overlay networks must be formulated. Overlay networks can summarize the topology of physical network and hide any details of the underlying physical infrastructure, bind for example the establishment or torn downward, from failures of node, the mobility of node, etc. Given the existence of an overlay network, participating in nodes has to store and retrieve data using a protocol P2P. Information retrieval can be realized with the storage and the recovery of pairs of key value. Every node which wishes to store a pair, of key-value, or to question a value based on a key, can realize it by using a DHT that operates on-top of the overlay topology. In a similar way, several applications can be built taking under consideration the existence of a high level Generic Application Programming Interface (API) that will provide them access to distributed (or centralized) repositories of data. Specific functions, as put (key, value) and get (key), can be displayed for the interaction of licence with a protocol DHT which works over a non-reliable IoT environment.

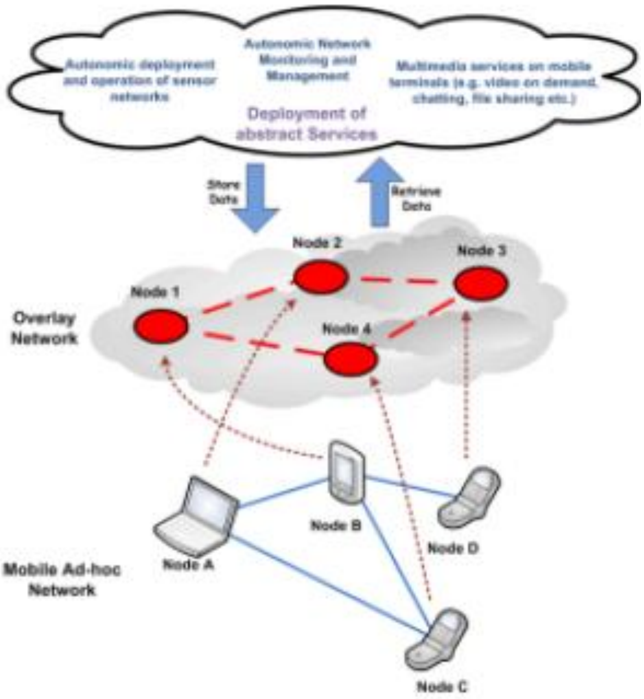

Autonomic deployment and operation of sensor networks
Autonomic Network Monitoring and Management
Multimedia services on mobile terminals (e.g. video on demand, chatting, file sharing etc.)
Deployment of abstract Services
Store Data
Retrieve Data
Node 2
Node 3
Node 1
Node 4
Overlay Network
Node B
Node D
Node A
Mobile Ad-hoc Network
Node C

Figure 2. Proposed approach for autonomic and decentralized services in the IoT

In the literature, already exist there a kind of implementation of mechanisms of Service Discovery (SD). Most of them, however, were designed for Local Area Networks (LANs) and, then, have been extended for Constrained IPV6 over Low-power Wireless Personal Area Network (6LoWPAN) network. One of these mechanisms is Plug and Play Universal (UPnP) [8], a protocol which allows to create automatically a network of device-to-device. However, as UPnP uses TCP as a protocol of transport and XML as the exchange format of message, it was not advisable for forced devices.

Another proposed mechanism is based on the Protocol of Location of Service (SLP) [9] [10] by which computers and devices can find services in local networks without previous configuration. Devices use SLP to announce in the local area network their available services, which are grouped **i** reaches scopes, i.e., simple strings that allow to classify services. The use of SLP can be important in large-scale scenarios IoT, to make automatic SD. However, SLP does not aim at forced devices, as those used in IoT. Furthermore, it matters on centralized approaches, which can be inclined to the failures.

An alternative to UPNP is Zeroconf [11] the protocol of network, which allows to create automatically data networks based on the TCP/IP Internet stack and it does not require external configuration. Zeroconf implements three main features:

1) automatic network address assignment;
2) automatic distribution and resolution of host names;
3) automatic location of network services.

The mission of automatic network intervenes when a node connects at first to the network. The distribution of name host and the resolution are implemented using multicast DNS (mDNS) [12], a service which has the same interfaces, packet formats and semantics of the standard Domain Name System (DNS) of messages to solve names hosts in the networks which do not include a server of local name. As regards the phase SD, Zeroconf implements the Service Discovery of base DNS (DNS-SD) [13]. By using questions of standard DNS, a customer can discover, in a given domain, the cases named by the service of interest.

More recently, some works examined the use of systems Peer-to-Peer (P2P) to implement evolutionary and strong distributed services of discovery. Schoenemann et al. [14] Proposed architecture is Based on the P2P allows the information exchange among the participants of a supply chain.

In a similar way, Shrestha et al. [15] have proposed a Peer-to- Peer network, where every participant of the supply chain executes a node of the network. These nodes form an overlay network associated to the structured P2P with every node having a partial view of the other nodes. Manzanares-Lopez et al. [16] have proposed a service of discovery distributed for the network EPCglobal based on the architecture P2P, which offers the level track of article and the capacities of track along the whole supply chain, also when articles are not directly visible, because they is responsible in the systems different from storage (i.e., packages, boxes, containers, etc.).

These approaches Peer-to-Peer adopt typically the techniques of Distributed Hash Table (DHT). Distributed Hash

Tables are structures of distributed data where the objects of the information are placed in a determinist way, of the peer whose identifier corresponds to the unique key of the object of the information, according to a metric given distance [17].

We propose in this paper a structure of distribution of services and connection of devices via hyperbolic tree associated with the Poincaré disc model. This structure allows an autonomous management of device connections (with names-paces) and distribution of communication with the various entities in exchange. Our work exploits the properties of the hyperbolic geometry to achieve its aims.

3. Hyperbolic Geometry

In this section, we interest particularly in the various elements allowing to define the hyperbolic structure of the geometry which we use in our model. Indeed, we focus on the notion of points, lines and circles who allow to characterize the metrics us concerning. One of the properties which characterizes the hyperbolic geometry is the fact that the Euclide's fifth postulate is not anymore true [18] [19].

3.1. Notion of Hyperbolic Line and Unit Circle

We try to present briefly the notion of line in the hyperbolic plan together with the notion of unit circle represented in the spherical geometry. We present as follows these concepts:

3.1.1. The Unit Circle (Represented in Figure 3)

$$S^1 = \left\{ (x, y) \middle| \; x^2 + y^2 = 1 \right\} \tag{1}$$

3.1.2. The hyperbolic Line (Represented in Figure 4)

$$H^1 = \left\{ (x, t) \middle| \, x^2 - y^2 = -1, t > 0 \right\} \tag{2}$$

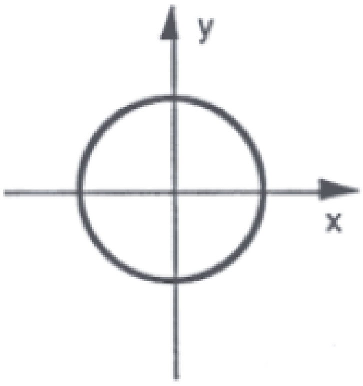


Figure 3. The circle

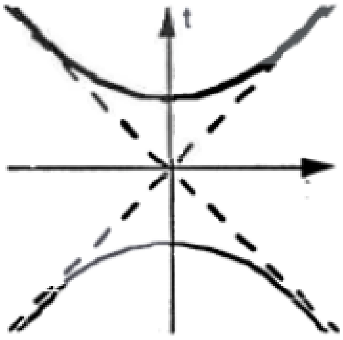


Figure 4. The hyperbola

3.2. Metric of Hyperbolic Distance and Computing Principle

Given two pairs of points of the hyperbolic plan how to compare their distances? To do it, it is necessary to calculate individually their distances before comparing the obtained values. From a theoretical point of view, we could by means of a rule connect the points two-two before comparing them. But this link between points has to suit to the group of geometrical transformation of the hyperbolic context. So, so that it is for the case of the line or the circle, we choose the group of hyperbolic transformation which suits us. Our point of view concerning the comparison of two line segments is illustrated by the following principle.

Principle of Superposition: Two line segments have the same length if and only if they can be superimposed by an element of the group of geometrical transformations.

These positive transformations [20] have as well the property as there is a unique (positive) transformation which takes into account one suggestion to another one. Indeed, we can use our principle of superimposing by returning all our segments of line managed in a fixed canonical position, which we call Paris, to compare lengths. For a while, Paris had kept a metrics fixed length which was used for the comparison all over the world. So there is a unique positive geometrical transformation which takes into account one suggestion in Paris. Otherwise, we can think in Paris being transformed in a given point θ By a positive transformation T_θ. We present T_θ explicitly and define Paris.

1) The circle Let Paris = $\begin{pmatrix}1\\0\end{pmatrix}$ Then

$$T_\theta = \begin{pmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{pmatrix} \tag{3}$$

So T_θ on S^1 is identified with

$$\begin{pmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix} = \begin{pmatrix}\cos\theta\\ \sin\theta\end{pmatrix} \tag{4}$$

2) The hyperbola Let Paris = $\begin{pmatrix}1\\0\end{pmatrix}$ Then

$$T_\theta = \begin{pmatrix}\cosh\theta & \sinh\theta\\ \sinh\theta & \cosh\theta\end{pmatrix} \tag{5}$$

Where the hyperbolic functions are defined by

$$\begin{cases}\cosh\theta = \left(e^{\theta} + e^{-\theta}\right)/2\\ \sinh\theta = \left(e^{\theta} - e^{-\theta}\right)/2\end{cases} \tag{6}$$

So T_θ on H^1 is defined with

$$\begin{pmatrix}\cosh\theta & \sinh\theta\\ \sinh\theta & \cosh\theta\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix} = \begin{pmatrix}\sinh\theta\\ \cosh\theta\end{pmatrix} \tag{7}$$

3.3. The Poincar éDisk Model Presentation

The model of Klein of the hyperbolic plan shows that the hyperbolic lines are "straight", but we could also consider other properties. We describe here a model, in plan H. Poincar é, which is another another equivalent description of the hyperbolic plan, but he has the interesting property which is that the hyperbolic circles are Euclidian circles. For the Klein-Beltrami model we used central projection of H^2 into the plane t = 1. Remind that for the sphere S^2 stereographic projection took circles to circles. We can apply the same idea for H^2 in our Minkowski geometry [21]. A natural place to choose from the projection point is the antipode of Paris, namely

$$\begin{pmatrix}0\\0\\-1\end{pmatrix}$$

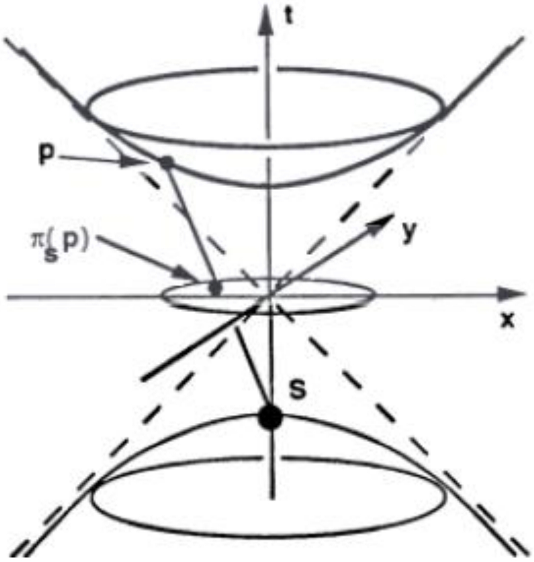


Figure 5. The Klein model

The projection from S into the plane t = 0 (Figure 5) is defined by $\pi_s : H^2 \longrightarrow R^2$ Besides, the image of H^2 under π_s is the unit disk again, but in the plane t = 0. But what represents the image of a hyperbolic line? One easily to notice that it is about circles situated in the unit disk and which are orthogonal on the border of the disk in question or are associated to diameters in the latter. Poincar é disk model possesses an important property which is that the angular gap between two circles images of two lines is the same measure that the angle describes by two lines in the hyperbolic plan H^2. We say that the model is conformal.

The continuation of our work will present the practical use of the properties of the hyperbolic geometry such we present it in the current section.

4. Practical Use of the Hyperbolic Geometry

Appeared around of 19th century, the hyperbolic geometry verifies four first ones Euclid’s postulate, but not the fifth. The Models the most popular is the model of the record of Poincar éwhich we use in this paper. This model is represented by Figure 6.

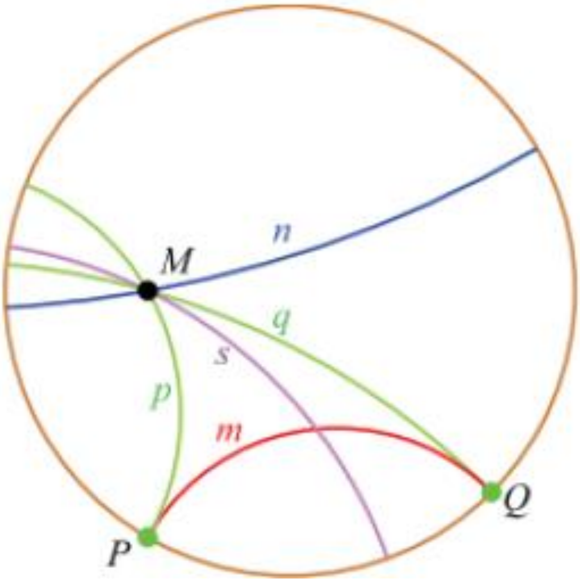


Figure 6. Poincaré’s disk model: inside the open disc, the points of the hyperbolic plane

Lines are drawn by diameters or orthogonal circles on the border of the disk, for example the line m. By the point A we can see a line which cuts itself the line m, two lines which are parallel in line m: p and q, line m touching in the model in P and Q respectively which are the points of the border and which are called points to the infinity. Finally and not the slightest: the line n also takes place by A without cutting the line m, neither inside the disk nor outside of it.

4.1. Tiling of the Hyperbolic Plane

From a famous theorem established by Poincar éat the end of the 19th century, we know that there are infinitely many tilings of the hyperbolic plan, every product by the reflection of a polygon P in the sides and, in a recurring way, in the reflection of the images on their sides, provided that the number p sides of P and the number q copies of P which can be put around a point A and exactly the cover of neighbourhood of A without satisfied overlapping the relation:

$$\frac{1}{p} + \frac{1}{q} < \frac{1}{2} \quad (9)$$

The values p and q characterize the tiling which is denoted {p, q} and the condition says that polygons are considered defined in the hyperbolic plan. We notice that three tilings of the Euclidian plan can be characterized by the obtained relation the by replacing < with = in the expression above.

We arrive, in this way, {4, 4} for the square, {3, 6} for the equilateral triangle and {6, 3} for the regular hexagon. In the paper, we focus on a simple system of tiling. We show how our hyperbolic tree is built at the beginning in this mechanism. We present in the Figure 7, a tree of q-regular of degree q = 3. Every node of the tree which we build in the Poincar édisk represents an entity of Internet of things.

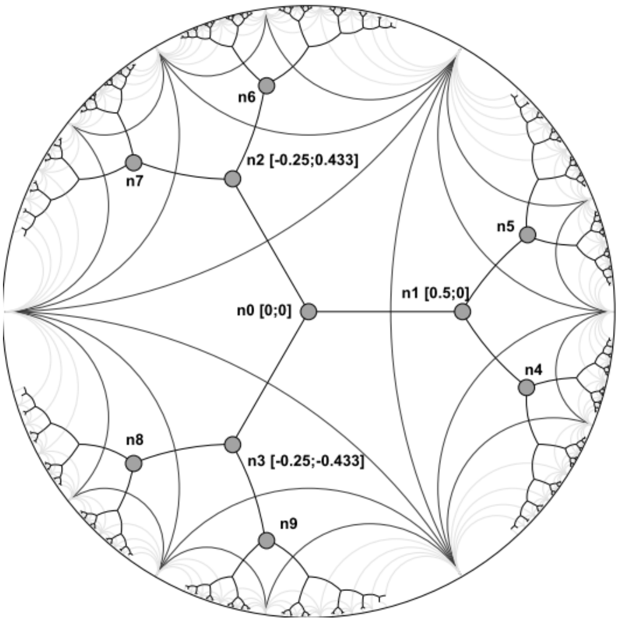


Figure 7. Tiling of the hyperbolic plane with 3-regular tree

In the Poincar édisk model, the distance between any two points z and w is given by curves that minimizes the distance between these two points and is called geodesic of the hyperbolic plan. To calculate the length of a geodesic between two points z and w allows to obtain the hyperbolic distance d_H, thus, we call on to a metrics of Poincar éwhich is an isometry invariant given by the formula:

$$d_H(z, w) = \arg\cosh(1 + 2 \times \frac{|z-w|^2}{\left(1-|z|^2\right)\left(1-|w|^2\right)} \quad (10)$$

This formula is used by the indexing process (greedy routing) shown in Section 4.2.

4.2. Information’s Routing Process in Hyperbolic Tree

The management of the communication between devices on the Internet of things is made by the use of the specific mechanism of indexing. This process combines one greedy routing [22] and a routing by holes bypass [23]. It concerns a kind of without knowledge of the neighbourhood, which remains very effective in a context of dynamic system Dynamic context characterized by the connection and the voluntary disconnection of nodes or by failure of the system.

By default a message is sent into eager’s mode via the calculation of the closest node to the destination. Thus, the following relay runner is chosen in a way distributed among nodes situated in the zone of relayage (immediate

Neighbourhood of the current node) in the time a phase of the application with a period of additional withdrawal.

This delay is a function of the hyperbolic distance of the relay runner in the node of the destination. However, in certain cases, when this zone is deprived of candidate node to relieve the message, the progress by the holes bypass is used. It consists initially in the discovery of the neighbourhood, then to cross in the second time with the planarization of the graph resulting from the neighbourhood before the choice of the relay runner among nodes resulting from the planar graph according to the rule “right hand” [24].

In the geographical routing, the purpose of the planarization is to avoid buckles during the routing process and require to have a knowledge of the 2-neighbourhood to have a planar graph in an environment of realistic radio operator for Internet of things with the precise hyperbolic contact.

Figure 8 illustrates the routing of message of connection or search for information in the disc of Poincar é that which we propose.

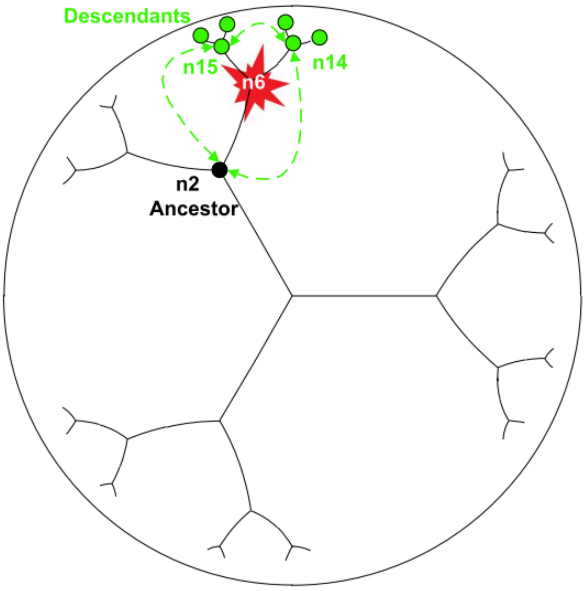


Figure 8. Routing process with Internet of things system

In this Figure, the device n6 failed, thus in the mechanism of routing of the downward nodes (n14, n15) towards the ancestor of node (n2) have to make a border of the dropped node (n6).

To make this kind of routing, we use the Algorithm 1 which associates the mechanism of greedy routing combined in a routing by holes bypass in case of the failure of a device in the course of path from the source to the destination.

5. Building and Function of Our IoT System

In order to show in a precise way, how our system of cloud for the Internet of things run, it is important to show the various algorithms which take part in the good management of devices. These algorithms are in particular that of the creation of our structure of hyperbolic tree, thus that of the inter-devices connections between devices.

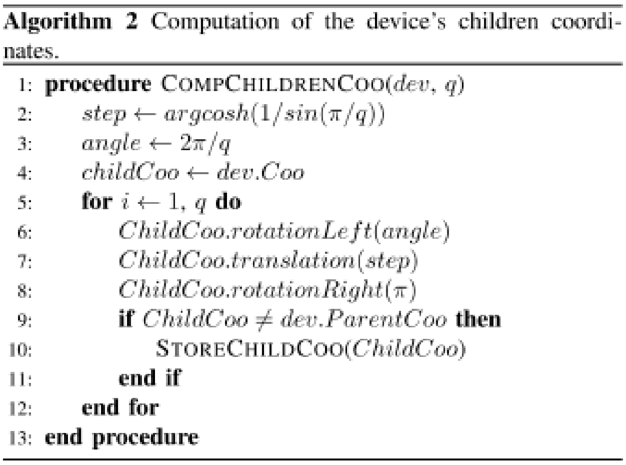
Algorithm 1 Routing a query in the IoT system.

```
function GETNEXTHOP($dev$, $msg$) return dev
    $w = msg.destinationDevCoo$
    $m = dev.Coo$
    $d_{min} = argcosh\left(1 + 2\frac{|m-w|^2}{(1-|m|^2)(1-|w|^2)}\right)$
    $p_{min} = dev$
    for all $neighbor \in dev.Neighbors$ do
        if !(neighbor.failed) then
            $n = neighbor.Coo$
            $d = argcosh\left(1 + 2\frac{|n-w|^2}{(1-|n|^2)(1-|w|^2)}\right)$
            if $d < d_{min}$ then
                $d_{min} = d$
                $p_{min} = neighbor$
            end if
        else
            $neighbor = neighbor.shortcut$
            $n = neighbor.Coo$
            $d = argcosh\left(1 + 2\frac{|n-w|^2}{(1-|n|^2)(1-|w|^2)}\right)$
            if $d < d_{min}$ then
                $d_{min} = d$
                $p_{min} = neighbor$
            end if
        end if
    end for
    return $p_{min}$
end function
```

5.1. Building of the Addressing’s tree

According to Algorithm 1, the first step in the creation of our hyperbolic tree is to start the first device and to choose the degree of the addressing’s tree associated with the hyperbolic tree. We recall that the hyperbolic coordinates (i.e. a complex number) of a node of the addressing’s tree are used as the address of the corresponding device in the system. A node of the tree can give the addresses corresponding to its children in the tree. The degree determines how many addresses each device will be able to give. The degree of the tree is defined at the beginning for all the lifetime of the Internet of things system. The latter is then built incrementally, with each new device joining one or more existing device. Over time, the devices will leave the system until there is no peer left, which is the end of the system. This method is scalable because unlike [25] [26], we do not have to make a two-pass algorithm over the whole network to find its highest degree. Also in our system, a device can connect to any other device at any time in order to obtain an address. It is about a system of structured Peer-to-Peer based on a Distributed Hash Table (DHT).

In the process of building of addressing’s tree, the first step is to define the degree of the tree because he allows to build the dual, namely the regular q-gon. We fix the root of the tree originally and we begin the tiling at the origin of the disk in the function of q. Every splitting of the space to create disjoint subspaces is assured when half spaces are the tangent; where from the primal is an infinite q-regular tree. We use the tree of theoretical infinity q-regular to build the fixation greedy for our tree of q-regular. Thus, the degree of the regular tree is the number of the sides of the polygon was used to building the dual (see Figure 7). In other words, the space is assigned for q devices children. Every device repeats the calculation for its own half of the space. In half of the space, the space is again assigned for q − 1 children. Every child can distribute its addresses in its half space. Algorithm 2 shows how to calculate the addresses that can be given to the children of a device. The first device takes the hyperbolic address (0; 0) and is the root of the tree. The root can assign q addresses. q is the number of half space created in the hyperbolic plane for each one.



Algorithm 2 Computation of the device's children coordinates.

```
procedure COMPCHILDRENCOO($dev$, $q$)
    $step \leftarrow argcosh(1/sin(\pi/q))$
    $angle \leftarrow 2\pi/q$
    $childCoo \leftarrow dev.Coo$
    for $i \leftarrow 1, q$ do
        $ChildCoo.rotationLeft(angle)$
        $ChildCoo.translation(step)$
        $ChildCoo.rotationRight(\pi)$
        if $ChildCoo \neq dev.ParentCoo$ then
            STORECHILDCOO($ChildCoo$)
        end if
    end for
end procedure
```

This distributed algorithm assures that devices are contained in different spaces and have unique address and coordinates. All the steps of the presented algorithm are suited for the distributed and asynchronous calculation. This algorithm allows the allocation of addresses as coordinates in the dynamic topology. As the global knowledge of the system is not necessary, a new device can obtain its coordinates (address) by asking an existing device to be its parent and to give its address for him. If the wanted device has already given all its addresses, the new device has to ask an address in another existing device. When the last one obtains an address, it calculates the addresses (i.e. hyperbolic coordinates) of its future devices that can potentially connect itself to the system. The tree of addressing builds itself by iteration at the same time as the system.

5.2. Referencing and Binding in Poincar éDisk Model

In this section, we present our structure of IoT based on the hyperbolic algorithm of greedy routing presented in the Section 4.2. We explain how our system stores and retrieves information in the peers of the network. With the departure, every new member of the system chooses a name which identifies the device where it is made. This name is protected by the device as long as the system exists.

Then, the device obtains an hyperbolic address and becomes an entity of the system while connecting itself to the addressing's tree. With each entity of the system is associated a pair (name, addresses). This pair makes it possible to create a link between the name and the address called binding because it is stored in the DHT by using the name as key. If the name of the entity is already a key of our system, then an error message is sent in answer to the entity so that it chooses another name, thus the structure of our IoT system itself guarantees that the names are single. To store the pair (name, addresses) associated to the device in the system, this latter creates a key by hashing its name with the algorithm SHA-2. The key of 512 bits is arbitrarily divided into 16 sub-keys of 32 bits. The system selects the first sub-key then and computes an angle by a linear transformation. The angle is given by equation 2. This angle gives us a virtual point on the circle.

$$\alpha = 2\pi \times \frac{32 - bit\ subkey}{0 \times FFFFFFFF} \tag{11}$$

Device can compute a virtual item v on the unit circle by using this angle:

$$v(x, y)\ with \begin{cases} x = \cos(\alpha) \\ y = \sin(\alpha) \end{cases} \tag{12}$$

Thereafter, the entity determines the contact of the nearest connection at the virtual point computed by the use of a tree of connection of depth given. The entity then sends a request blind to nearest even. This request is routed in the system by using the greedy routing algorithm in Section 4.2. If the request fails because the binder does not exist, or because of the failure of the node link, it is directed to next the nearer binder which is the father of the computed binder. This process continues until the request meets a father existing binder who is the nearest possible virtual point or center. This process guaranteed that the entities are always stored firstly in the binder nearest to the circle of unit and lately in the binder nearest to the center of the disk. If the tree of addressing is unbalanced, several entities can be stored in the entities close to the center for thus overloading them. With an aim of solving this problem, the even binder will be able to bind to a maximum number of entities beyond whose each new entity will be refused and the request transmitted in addition. To guarantee the redundancy, the entity will have to be able to use the previous process to store the 15 sub-key. Thus sixteen rays binders will be used and will improve the distribution of the entities. Besides this and within the framework of the redundancy, an entity can store more than one entity of the ray binder. A binder will be able to store an entity and to still redirect its request of storage towards other ancestors binders. The number of copies stored of a entity, according to the ray binder can be an arbitrary value given to the creation of the system. Same manner, the division of a key in 16 sub-keys is arbitrary and could be increased or reduced according to the needs for redundancy. To conclude, we can define two mechanisms for the storage of the copies of a given binder:

1) we can use more than one ray binders to create several Sub-keys according to a uniform distribution;
2) we can store more than one even binder in the same ray

binder.

Figure 9 illustrates the mode of naming and binding of our IoT system in the Poincar é disk model. In this illustration, device i try to connect at the IoT system by beginning with entity n2 before to find final entity n9. Furthermore, shortcut link allows is used as solution for routing by holes bypass presented in Section 4.2.

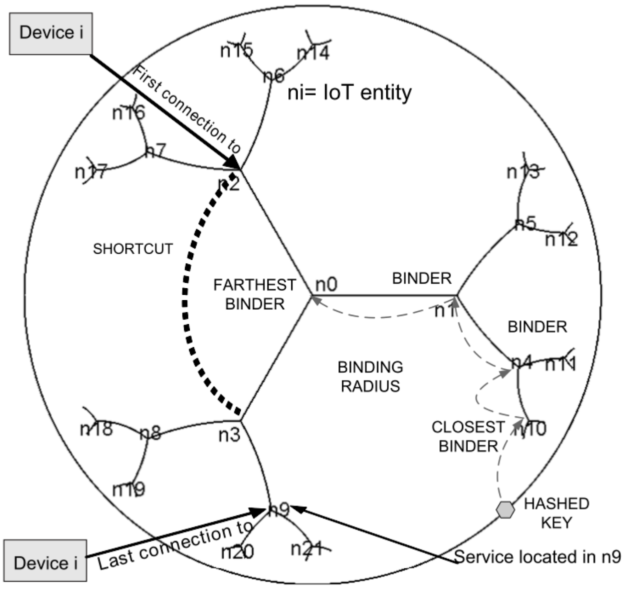


Figure 9. Hyperbolic IoT system

These mechanisms make it possible our IoT system to cope with the growth of the network and it guarantees that the par will be stored in a redundant way what will maximize the success rate of lookup. The number of sub-keys and the number of copies in a ray are parameters which can be given to the creation of the IoT system. Their increase causes compromise between the improvement of their effectiveness and the loss of storage space in the binder. Our solution has the property of consistent hashing: if an entity fails, only its keys are lost, the other binders are not impacted and the whole of the system remains coherent. As in several existing systems, the entities will be stored by using a hybrid strategy: software and material. We analyse the practical point of view, the influence of the replication strategy on the success rate, as well as the property of scalability in the continuation of our study.

6. Management of our IoT System

6.1. How to Cope with Dynamic Topologies

The routing in the hyperbolic plan is robust as long as the integrity of the tree is maintained. In effect, the embedded greedy of [27] depends on the subjacent connectivity of the spanning tree (or planar graph) embedded. In a real IoT environment, it is expected that the failures of the links or the entities often arrive. Clearly, the disadvantage of this method is resistance to breakdowns [28]. This is why, we consider two levels of breakdowns:

- the first level corresponds to the breakdowns of q-regular addressing's tree.
- the second level corresponds to the breakdowns in the graph of the IoT system.

On the first level, if a link in the tree is then broken the hyperbolic greedy routing will fail for the ways crossing this link. Worse, if an entity other than a leaf breaks down, the tree is broken in a forest of with more q trees and thus disturbs the connectivity of the tree. We can then use techniques of repair or maintenance of drill of tree such as the random walk which is a natural approach for the exploration of the graphs. The idea is attractive, but in our case we cannot guarantee the consistency of the addresses when the tree amalgamates, in other words, that could create buildings minimum. Indeed, fusion without an update of the addresses of the nodes of the sub-tree is not acceptable. Moreover, the random walk is not efficient in hyperbolic space because it aims towards the infinity. But in every case, the Achilles' heel remains the root. Therefore, to mitigate these breakdowns, it is necessary to use an alternative method of routing. If the tree of addressing is broken, two approaches can be used to restore connectivity:

- method of purging: consist in purging the addresses of the nodes starting from the link or device in breakdown and to reallocate the addresses with all these nodes.
- method of restoration: consist in restoring the tree in substitute the defective link by one new identical link or broken down device by a new device with the same connections.

The first solution that we call the method of purging can be expensive if the size of the IoT system is very great and/or if the nodes beyond the breakdown are numerous because that can carry out to readdress a vast part of entities. The second solution that we call the method of restoration is less expensive because the addresses are kept but the implementation is more complex in the case of the breakdown of an entity. Indeed, it is difficult for a new entity to be configured with same connections as those of the failing entity that it replaces.

On the second level, if the link of the IoT system is not touched by the breakdown in the addressing's tree or if it is an entity breaks into leaf the hyperbolic greedy routing then is carried out without error through the ways being able to be longer. The addressing's tree is built on the overlay network, which means that there exist subjacent links, potentially short cuts which can become the only possible road. In the course of time, when the solutions above are used to restore the addressing's tree, other techniques can be used to guarantee the success of the routing. A first technique, for the downward losers their ancestor, consists in trying to establish a connection with their first ancestor (i.e. grandparent) as well as with their brothers. If they make a success of the hyperbolic greedy routing will be guaranteed, if not the addressing's tree remains broken and the greedy algorithm fails because of the problem of "local minimum". To overcome this problem, heuristics called "Gravity-Presses (GP)" is presented in [28] and also used in

[29]. When a message arrives in an entity which is a "local minimum" it enters in mode "presses". In this mode, the message maintains a list of the nodes which it crosses and the number of times that it crosses them. This continuous process until the message finds an entity whose distance with the destination is smaller than the distance with the "local minimum". The solution put forward requires the storage of much information in the heading of the message which can be difficult and heavy to establish.

6.2. Storage by Replication Strategy in the IoT System

As described in Section 4.2., the system guarantees that the names given are single and makes it possible by linear transformation via algorithm SHA-512 to compute 16 different points on the Poincar édisc model.

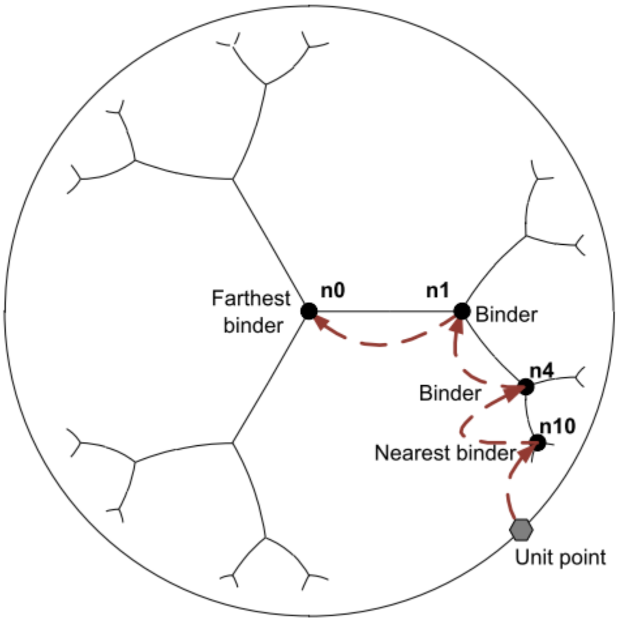


Figure 10. Binder use for storage process

Now, the entity determines the hyperbolic coordinates of the node nearest to this virtual point; this node is a binder of the couple (name, addresses). An example of the procedure of storage is shown in Figure 10). Thus to store in the entity n10, a request is sent to storage computed. This request is routed by using the hyperbolic greedy routing described in Section 4.2. If the request fails because the storage cannot be reached, because of a node or a link failed, then the request is redirected towards the storage one nearest which is the father of storage computed.

This process continues as long as the request did not reach storage valid. The request can thus go up in the addressing's tree towards the root which is storage the most distant. Nevertheless, we guarantee that the entities are stored in priority in the storage one nearest to the unit circle. The depth of an entity in the addressing's tree is defined like the number of entities separating it from the root (root included). We define, at the creation, the maximum value depth of storage in the addressing's tree. We speak then about the tree of the connection. All the entities which have a depth lower or equal to the depth of the tree of the connection in the tree of addressing can become the storage ones. Figure 10, where the depth of the tree of the connection is fixed to 3 not to overload Figure, watch how and where a couple is stored in the overlay network. In the case of a tree of unbalanced addressing, several entities could be stored near the root and thus overload the nodes. To cure this problem, the storage ones will be able to contain only one limited number of couples, the requests of additional storages will be refused and redirected. An increase amongst copies leads to a compromise between reliability and loss of storage space, but these mechanisms make it possible the IoT system to cope with at a growth not uniform of the overlay network. To ensure the consistency of IoT system, we use two mechanisms of replication:

- a replication known as "radial",
- another, known as "circular".

In the radial replication, storage is carried out on in most **S** first knots the closest to the unit circle in the direction of the root to avoid overloading the root in requests to be treated, the value of S is given by the following equation:

$$S \leq \left\lfloor \frac{1}{2} \times \frac{\log(N)}{\log(q)} \right\rfloor \tag{13}$$

Where N equal to number of entities in the IoT system and q equal to the degree of hyperbolic tree.

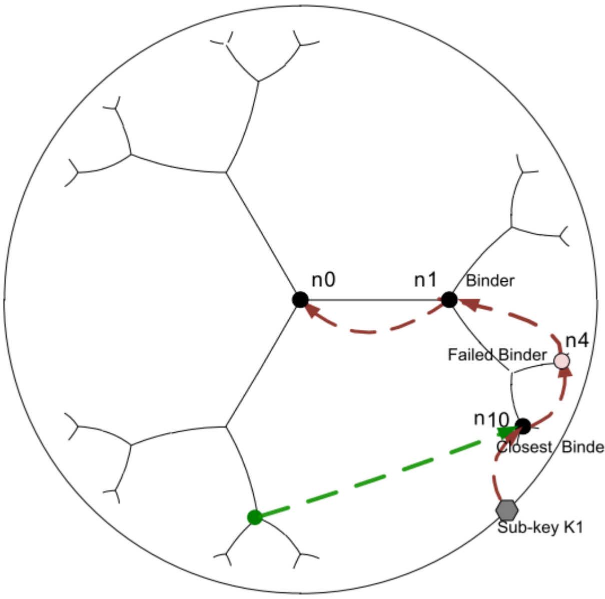


Figure 11. Radial replication process

Figure 11 shows the process of the radial replication explained higher. For which relates to the circular replication, we use each under key obtained with the principle explained higher to compute each first storage nearest of the unit circle. Thus, we can carry out to the maximum 16 circular replications in our case. Figure 12 shows the principle previously explained.

Moreover, these two mechanisms of replication thus make it possible to our IoT system to be consistent. Indeed, if storage only fails its keys are lost, other storages are not impacted and the whole of the system remains coherent. A couple is stored by its creator all the periods of time T if not it is removed from the storage ones. Also, a message of suppression can be sent by the creator to remove the couple before the end of the period. It is very time important to note that the circular choice of many replications is arbitrary. The number of sub-keys and the number of the copies are parameters which can be fixed with the creation of the overlay network.

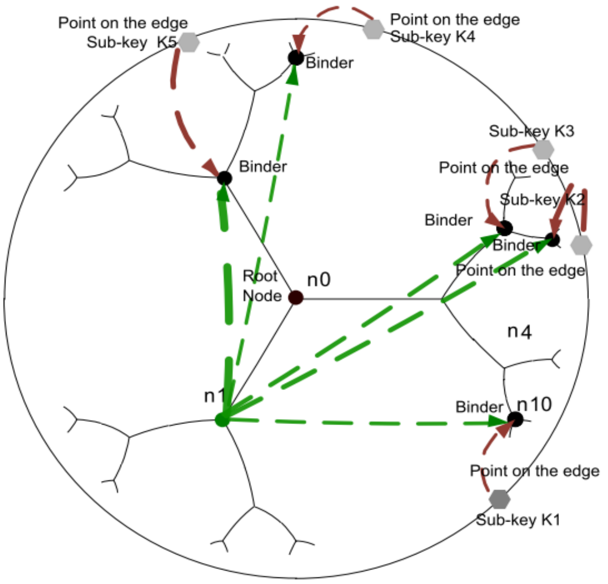


Figure 12. Circular replication process

7. Experimental Study of Reliability and Scalability Properties

In this section, we present the results of the simulations that we have run and we have evaluated the performances in term of scalability and reliability, of our devices addressing system. Our IoT system is considered as dynamic (the phenomenon of churn is applied). We use the Peersim [30] simulator for IoT system simulation and it allows to service name following the uniform distribution. Excepted, the study on the floating point precision issue, our simulation involves the following parameters:

- number of devices connected and used to store different kind of services is equal to 10000 in the starting up;
- we consider that our IoT system is dynamic and the rate of churn varying from 10% to 60%;
- we consider a simulation performed during 2 hours;
- the leaving and joining of the devices follows an exponential distribution as well as the inserting and the deleting of the services of the system associated.

7.1. Analysis of the Number of Sub-keys

Figure 13 presents the evolution of the number of primary binders according to the number of sub-keys chooses as the simulation. This plot shows a continuous growth of the number of binders according to the number of sub-keys. Furthermore, we can observe that this plot is below the ideal case or the number of binders always corresponds among sub-key. The feigned plot is besides very close to the ideal plot, what shows that we have a satisfactory situation for all the levels of replication.

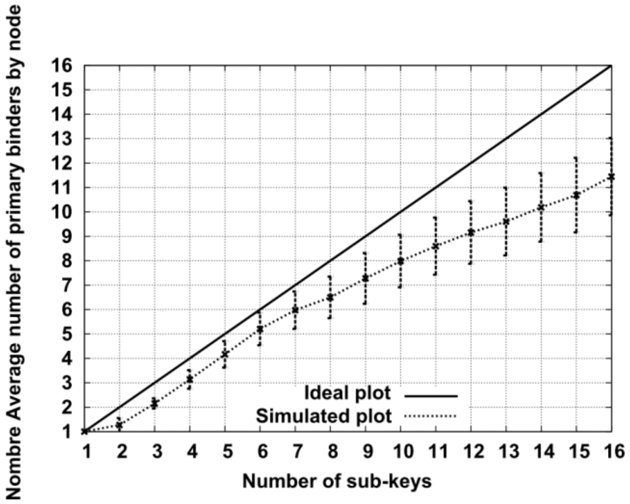


Figure 13. Variation of binder number depending of the number of sub-keys

7.2. Load Balancing in the IoT System

Figure 14 shows an experimental distribution of points corresponding to the scatter plot of the distribution of devices in our IoT system. Thus, we can mark that hyperbolic-tree of our IoT system is balanced. Indeed, we can noticed by part and others around the unit circle which we have providers of services.

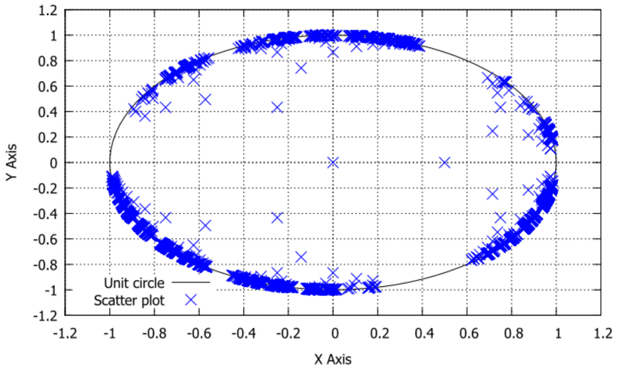


Figure 14. Scatter plot corresponding to the distributed devices

We can notice that this representation indicates a nearly uniform distribution of the various devices around of the root. Thus, we can assert that our system favorites a load balancing and then increases, its stability and its robustness

Figure 15 shows a correspondingly Poincaré disk mode that no address of resource device belongs on the edge of the unit circle. Indeed, the addresses of resource device were obtained by projection of the tree of the hyperbolic plane in a circle of the Euclidean plane of radius 1 and of centre with coordinates (0; 0).

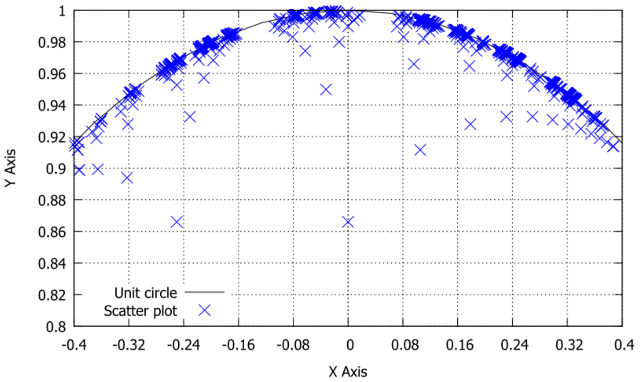


Figure 15. Distribution of nodes in the neighbourhood of the edge of the unit circle

8. Conclusion and Futur Work

This paper provides the foundation of a fully distributedcontrol plane for the Internet of Things. Thus, we propose a system which uses a virtual hyperbolic frame of referenc based on the Poincaré disc model. This system has on the one hand a certain number of hyperbolic geometry properties. Furthermore, a property of robustness and reliability expressed by a strategy of replication (radial and circular) then of a scalability owing to the fact that the device has a total independence in the choice of their point among in the system.

Through this study, we can realize through bound work that there is still much work to do with the domain of IoT, that is from a theoretical point of view that applied. But for the moment, no research approached the use of the Poincaré disc model in the field of IoT.

This paper can be considered as a contribution brought in the context of the interconnection and the exchange between devices of the domain of the Internet of Things. We hope that it will encourage people to venture along the tracks opened in this paper and, it would be the best, to go further towards new perspectives. Us concerning, we count following this work, to suggest some solutions of repairing of the addressing's tree in situation of flash crown.

REFERENCES

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," Future Gener. Comput. Syst., vol. 29, no. 7, pp. 1645–1660, Sep. 2013. [Online] http:dx.doi.org/10.1016/j.future.2013.01.010.

[2] C. Bornhovd, T. Lin, S. Haller, and J. Schaper, "Integrating automatic data acquisition with business processes experiences with sap's auto-id infrastructure," in Proceedings of the Thirtieth International Conference on Very Large Data Bases - Volume 30, VLDB '04.VLDB Endowment, 2004, pp.1182 –1188.

[3] F. Wang, S. Liu, and P. Liu, "Complex rfid event processing," The VLDB Journal, vol. 18, no. 4, pp. 913–931, Aug. 2009. [Online]. http://dx.doi.org/10. 1007/ s00778-009-0139-0.

[4] A. T. Karygiannis, B. Eydt, G. Barber, L. Bunn, and T. Phillips, "Sp 80098. guidelines for securing radio frequency identification (rfid) systems," Gaithers burg, MD, United States, Tech. Rep., 2007.

[5] R. Khan, S. U. Khan, R. Zaheer, and S. Khan, "Future internet: The internet of things architecture, possible applications and key challenges," in Proceedings of the 2012 10th International Conference on Frontiers of Information Technology, ser. FIT '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 257–260. http://dx.doi.org/10.1109/FIT.2012.53.

[6] V. Turner, D. Reinsel, J. F. Gantz, and S. Minton, "The Digital universe of opportunities: Rich data and the increasing value of the internet of things," in WHITE PAPER of IDC iview. IDC Analyze the Future, 2014. http://www.emc.com/leadership/digital-universe/index.htm.

[7] P. Gouvas, A. Zafeiropoulos, A. Liakopoulos, G. Mentzas, and N. Mitrou, "Integrating overlay protocols for providing autonomic services in mobile ad-hoc networks," IEICE Transactions, vol. 93-B, no. 8, pp. 2022–2034, 2010. [Online]. http://search. ieice.org/ bin/summary.php?id=e93-b 8 2022.

[8] "Upnp forums," accessed: 1999-09-30. [Online]. Available: http://www.upnp.org/

[9] E. Guttman, C. Perkins, and J. Veizades. (1999) Service location protocol, version 2. [Online]. Available: http://tools.ietf.org/html/rfc2608.

[10] E. Guttman. (2002) Vendor extensions for service location protocol, version 2. [Online]. Available: http://tools.ietf.org/html/rfc3224.

[11] "Zeroconf website," accessed: 1999-07-17. [Online]. Available: http://www.zeroconf.org/.

[12] S. Cheshire and M. Krochmal. (2013) Multicast dns. Available: http://tools.ietf.org/html/rfc6762.

[13] (2013) Dns-based service discovery. [Online]. Available: http://tools.ietf.org/html/rfc6763.

[14] N. Yulin, S. Huayou, L. Weiping, and C. Zhong, "Pdus: P2p-based distributed uddi service discovery approach," in Proceedings of the 2010 International Conference on Service Sciences, ser. ICSS '10. Washington, DC, USA: IEEE Computer Society, 2010, pp.3–8. http://dx.doi.org/10.1109/ICSS.2010.48.

[15] S. Kaffille, K. Loesing, and G. Wirtz, "Distributed ser vice discovery with guarantees in peer-to-peer networks using distributed hashtables." in PDPTA, H. R.Arabnia, Ed. CSREAPress, 2005, pp. 578–584. http://dblp.uni-trier.de/db/conf/pdpta/pdpta2005-2. html#KaffilleLW05

[16] M. Cai, A. Chervenak, and M. Frank, "A peer-to-peer replica location service based on a distributed hash table," in Proceedings of the 2004 ACM/IEEE Confe rence on Supercomputing, ser. SC '04. Washington, DC, USA: IEEE Computer Society, 2004, pp. 56–. [Online]. http://dx.doi.org/10.1109/SC.2004.7.

[17] M. Picone, M. Amoretti, and F. Zanichelli, "Geokad: A p2p distributed localization protocol." in PerCom Workshops. IEEE, 2010, pp. 800–803. [Online]. Available:

http://dblp.uni-trier.de/db/conf/percom/percomw2010.html# PiconeAZ10.

[18] D. Krioukov, F. Papadopoulos, M. Kitsak, A. Vahdat, and M. Bogu, "Hyperbolic Geometry of Complex Networks," Physical Review E, vol. 82, no. 036106, Oct 2010.

[19] T. Tiendrebeogo, D. Ahmat, and D. Magoni, "Relia ble and scalable distributed hash tables harnessing hyperbolic coordinates," in New Technologies, Mobility and Security (NTMS), 2012 5th International Conference on, 2012, pp. 1–6.

[20] T. Munzner, "H3: Laying out large directed graphs in 3d Hyperbolic space," in Proceedings of the 1997 IEEE Symposium on Information Visualization (Info Vis '97), Washington, DC, USA: IEEE Computer Society, 1997, pp. 2–. [Online]. Available: http://dl.acm.org/citation.cfm?id=85718 8.857627

[21] M. Zabka, "The minkowski geometry of numbers applied to the theory of tone systems." in MCM, ser. Lecture Notes in Computer Science, J. Yust, J. Wild, and J. A. Burgoyne, Eds., vol. 7937. Springer, 2013, pp. 226–240. [Online].http://dblp. uni-trier.de/db/conf/mcm2/mcm2013.html#Za.

[22] P. Fraigniaud and G. Giakkoupis, "Greedy routing in small-world networks with power-law degrees," Distributed Computing, vol. 27, no. 4, pp. 231–253, 2014.

[23] T. Lee, C. Qiao, M. Demirbas, and J. Xu, "Abc: A simple geographic forwarding scheme capable of by passing routing holes in sensor networks," Ad Hoc Netw., vol. 8, no. 4, pp. 361–377, Jun. 2010. http://dx.doi.org/10.1016/j.adhoc.2009. 08.004.

[24] B. Karp and H. T. Kung, "Gpsr: Greedy perimeter stateless routing for wireless networks," in Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, ser. MobiCom '00. New York, NY, USA: ACM, 2000, pp. 243–254. [Online]. Available:http://doi.acm.org/10. 11 45/345910.345953.

[25] H. Yang, K. Tang, J. Yu, L. Zhu, H. Xu, and Y. Cao, "Virtual coordinates in hyperbolic space based on ricci flow for wlans," Appl. Math. Comput., vol. 243, pp. 537–545, Sep. 2014. [Online]. Available: http://dx.doi.org/10.1016/j.amc.2014.05 .023.

[26] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica, "Geographic routing without location information," in Proceedings of the 9th Annual International Conference on Mobile Computing and Networking, ser. MobiCom '03. New York, NY, USA: ACM, 2003, pp.96–108. [Online]. Available: http://doi.acm.org/10.1145/938985.938996.

[27] R. Kleinberg, "Geographic routing using hyperbolic space," in proceedings of the 26th IEEE International Conference on Computer Communications, May 2007, pp. 1902–1909.

[28] Cvetkovski and M. Crovella, "Hyperbolic embedding and routing for dynamic graphs," in proceedings of the 28th IEEE International Conference on Computer Communications, April 2009.

[29] D. Krioukov, F. Papadopoulos, M. Bogu˜n´a, and A. Vahdat, "Greedy forwarding in scale-free networks embedded in hyperbolic metric spaces," SIGMETRICS Perform. Eval. Rev., vol. 37, no. 2, pp. 15–17, 2009.

[30] I. Kazmi and S. F. Y. Bukhari, "Peersim: An efficient & scalable testbed for heterogeneous cluster-based p2p network protocols." Washington, DC, USA: IEEE Computer Society, 2011, pp. 420–425. [Online]. http://dx.doi.org/10.1109/UKSI M.2011.86.