Generation of broadcasting for fractal adaptive Internet of things reconfiguration under the swarm intelligence paradigm

Jaime Moreno¹, Oswaldo Morales¹, Ricardo Tejeida², Hugo Quintana¹ and Grigori Sidorov³

Abstract
Recently, a wide range of small devices, such as Wi-Fi Internet of things development boards, which are a kind of the microcontroller units in a general purpose board, are interrelated throughout the planet. In addition, certain microcontroller units interact inside our homes when turning lights on or detecting movements, measuring various parameters, such as gas concentrations, CO₂, humidity, and the temperature inside a room, or adjusting the intensity of the lights inside and outside of the house. Likewise, there is a great diversity of microcontroller units, ranging from smart cellular telephones or reduced general purpose devices, ESP8266 or Raspberry Pi3 to any type of Internet of things devices. Therefore, the general way of connecting the microcontroller units to the Internet is through hub nodes, so that the information can be propagated and shared among them. The main purpose of this article is to yield an adaptive reconfiguration algorithm to link all the sensor nodes (microcontroller units) of a network based on the fractal topology, avoiding the use of hub nodes, in order for the microcontroller units to share all the parameters established in the Internet of things network only through two adjacent sensor nodes as long as any sensor node in the network knows all the parameters of the other ones, even if the sensor nodes are not adjacent. To achieve the above, in this work, an Internet of things network was built based on the Hilbert fractal for being a filling-space curve yielded from the L-systems paradigm, so this fractal Hilbert topology allows access to the entire Internet of things network in a dynamic way, and it is possible to reconfigure the network topology when a new sensor node is attached by applying artificial intelligence to intelligent and dynamic environments.

Keywords
Wireless sensor network, Internet of things network, Hilbert fractal topology, L-system, adaptive algorithm, machine learning, intelligent systems

Introduction
The diffusion and sharing node sensor parameters in wireless sensor networks (WSNs), specifically Internet of things (IoT) networks, for yielding energy efficiency have generated greater interest in academia and industry. IoT networks seem to be the WSN new generation because they combine the monitoring of the

¹Escuela Superior de Ingeniería Mecánica y Eléctrica, Unidad Zacatenco, Instituto Politécnico Nacional, Ciudad de México, México
²Escuela Superior de Turismo, Instituto Politécnico Nacional, Ciudad de México, México
³Centro de Investigación en Computación, Instituto Politécnico Nacional, Ciudad de México, México

Corresponding author:
Jaime Moreno, Departamento de Ingeniería en Comunicaciones y Electrónica, Escuela Superior de Ingeniería Mecánica y Eléctrica, Unidad Zacatenco, Instituto Politécnico Nacional, Av. Luis Enrique Erro S/N, Unidad Profesional Adolfo López Mateos, Zacatenco, Alcaldía Gustavo A. Madero, C.P. 07738, Ciudad de México, México.
Email: jemoreno@esimez.mx

Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License (https://creativecommons.org/licenses/by/4.0/) which permits any use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (https://us.sagepub.com/en-us/nam/open-access-at-sage).
environment, temperature, humidity, CO2, as well as the environmental lights tuning with transmission and processing of data through WSN. IoT networks are an option to transmit and share sensor parameters, since the nodes perform specific and simple tasks when acquiring data from the environment. Several sensors are added wirelessly in homes to obtain data on the physical or environmental states of the parameters of the network being monitored. After collecting sensor node data, transmitting them over the network IoT, interchanging parameters among nodes, processing and analyzing structural problems such as attaching new sensor nodes to the network, and communicating these parameters centrally as a star configuration, can be anticipated and successfully adapted, thus future requirements of an IoT network can be addressed in a timely manner. However, it is believed by 2020 the IoT will have reached critical mass. By connecting the large amount of intelligence in the cloud to billions of mobile devices and having very low-cost node sensors and network identifiers microcontroller units (MCU) integrated throughout the IoT network, it will generate benefits to almost all human beings, in terms of health, safety, and convenience. Also by the year 2020, there will be four billion people connected to the cloud with an income of four billion dollars when downloading more than 25 million applications from the main application stores, so that approximately more than 25,000 need millions of MCUs that will generate 50 trillion Gigabytes of data within the worldwide IoT network.

As the number of IoT devices increases in a conventional house, wirelessly connected sensors that monitor the IoT network parameters are required to ensure the proper performance of the IoT network. In an IoT network, there is a wireless access point (WAP) connecting all their devices to an Internet or intranet network as the hub node. If a node sensor will share parameters with other MCUs, all these MCUs must be linked to the WAP. Even if two MCUs are the physically closest, if they are not linked to the WAP, they cannot be linked and exchange parameters.

The rest of the article is organized as follows. In section “Related work,” four works are explained whose goal is to propose algorithms for making energy efficiency in WSN. Section “Environment of SI in an IoT network” offers a general description of our proposal. Section “Wi-Fi IoT development boards and Hilbert fractal curve” describes the integrated systems in the IoT network as well as the Hilbert filling-space curve. Section “Fractal adaptive reconfiguration in an IoT network” developed the main scheme. In section “Experimentation,” simulations of different protocols are carried out for verifying the correction and feasibility of our algorithm. Finally, we conclude the article in section “Conclusion.”

Related work

Mostafaei and Menth proposed the software-defined (SD)-WSN algorithm for saving energy in WSN, since the software-defined networking (SDN), the forwarding elements are not configured by means of distributed control protocols but remotely using centralized controllers. Several complex challenges arise from the configuration WSN, namely, in a distributed manner, thus, from a theoretical point of view, this situation can be better solved by knowing the entire network, since the solution is only configured for distributed sensor nodes. In their work, the authors explain the concepts of WSN and SDN, describe the fundamentals of SD-WSNs (such as different topologies and routing decisions), and explain how the SDN can improve the operation of WSN, through a comparison between the different routing protocols in SD-WSNs and among the works related to quality of service (QoS) in SD-WSNs. They found that distributed node coordination and energy efficiency are the most important challenges at WSNs. For non-SDN-supported WSNs, these are mainly resolved in a distributed manner, since the SD-WSN prefers central control. This implies energy savings because redundant communication can be avoided, power-restricting nodes can be removed from power-efficient tasks and moved to the controller, so specific tasks are accomplished with fewer active sensor nodes, that is, by intelligent operation.

Javadi et al. generated a topology control algorithm from a learning automaton, called LBLATC, to reduce the power consumption of the sensor nodes and increase the capacity of the WSN for a longer period of time. They assumed that topology designed to transmit in a lower range allowed the sensor nodes exploit efficiently their resources and mitigate their power consumption. Therefore, in the LBLATC, the learning automaton of each sensor node chooses the lower range transmission. The authors simulated parameters such as the network area, the number of nodes, the transmission range, finding that the LBLATC performed 15% better than other algorithms reported in the reviewed literature, in terms of selecting an appropriate transmission range, which is reflected in energy saving in the WSN. However, the LBLATC did not take into account the mobility that is a static protocol.

The QoS routing in WSN is critical for surveillance systems. As a consequence of high packet loss and power efficiency, it is the necessary reliable links in the streaming multihop data from WSNs for the end-to-end data delivery. The studies consulted by Mostafaei proposed a solution based on an
optimal multi-constrained route, yielding the distributed learning automaton (DLA) algorithm. This author found the smallest amount of sensor nodes to keep the better QoS requirements. To do this, Mostafaie considered various QoS routing restrictions such as end-to-end reliability and delay, depending on the chosen route. He simulated the DLA and the results obtained confirmed the effectiveness of his solution, that is, his algorithm performed better with respect to the other algorithms reported in the literature regarding end-to-end delay (E2ED) and energy efficiency.

Haqbeen et al.\textsuperscript{8} yielded the adaptive intelligent hybrid medium access control (AI-HMAC) traffic protocol, resulting from combining the functionalities of the time division multiple access (TDMA) and carrier sense multiple access (CSMA) configurations by programming transmissions and links to improve the performance and power consumption of the protocol. It is possible to operate this protocol within a wide range of data traffic and choose the send/receive mechanism according to the rate of packaging and data loss, reducing energy consumption. However, Zedadra et al.\textsuperscript{9} employed multi-agent system (MAS) technology, the swarm intelligence (SI) paradigm, and frontier computing to model an IoT system as a simple swarm of intelligent objects with limited computational and storage capabilities that use stigmergic communication to generate collective and complex behavior of the system; smart objects employ self-organizing and self-adaptive behaviors to meet IoT system requirements. Savaglio et al.\textsuperscript{10} designed a Q-learning MAC (QL-MAC) energy preservation protocol to extend the life of the WSN. QL-MAC is the result of Q-learning, which iteratively adjusts the MAC parameters through a trial-and-error process to converge to a low-energy stadium, by allowing each node to independently predict an efficient schedule to wake up or activate to save energy by limiting the period in which it is in active mode. This protocol minimizes energy consumption without the need to predetermine the system model and also provides a self-adaptive protocol for topological and other external changes. Finally, Kwon et al.\textsuperscript{11} developed a multi-hop ad hoc IoT network where IoT mobile devices strategically set their wireless transmission ranges based on a reinforcement learning approach deep. An IoT device is a decision-maker that strategically establishes its transmission range to maximize network performance and minimize power consumption during transmission, thus only the nodes receiving and transmitting information are activated with the maximum transmission power, and the other nodes are deactivated to minimize the consumption of energy.

We proposed an adaptive reconfiguration algorithm that links all the sensor nodes of an IoT or WSN network using the Hilbert fractal for being a filling-space curve, in order to avoid the use of hub nodes and that the sensor or MCUs of the IoT network share all the parameters only through adjacent nodes. With the proposed algorithm, it is possible to have access to the entire IoT network in a dynamic way, allowing to reconfigure the topology of the IoT network when a new node is added. This work has its background from a paper presented by Moreno et al.\textsuperscript{12} in the volume 11288 of the Lecture Notes in artificial intelligence.

Environment of SI in an IoT network

From an anthropomorphic view of point, intelligence was defined as something belonging to a single entity, but with a huge complexity. Nonetheless, if we are looking for a super intelligent entity, we could talk about a collective intelligence, such as a distributed processing entity, which is one of the most important basis of the artificial intelligence (AI).

By looking at nature, it is possible to learn from it (this is what we call bio-inspired computer models). In large colonies of social animals, such as bees, termites, or ants, any member by itself is so intelligent due to their members follow very simple algorithm patterns. However, as a group or colony, their collective behavior displays a huge intelligence. For example, an ant by itself cannot defend the colony or feed the queen, but when it interacts with the other members of the colony, it is possible to do a lot of tasks to guarantee to preserve the colony. In this way, our human societies achieve great feats by collaborating and working as a team, where would be the novelty that our way of managing groups is usually hierarchical: there are leaders, supervisors who check the functioning of the whole, who direct it and re-direct it toward its own objectives. In insect colonies, on the other hand, there are no supervisors, there is no one to govern the system. The achievement of the objectives lies in the synergy of the independent functioning of each of the parties. In addition, interestingly, the goals achieved are spectacularly intelligent as an entity.

Furthermore, the comparison with the human brain and consciousness seems compelling. A neuron has a very simple behavior pattern, but millions of them working simultaneously can generate intelligent behavior and even consciousness. In addition, introducing the concept of simultaneity in IoT networks or unmanned vehicles is overcoming one of the fundamental differences between machines and brains. Our computers, despite their incredible speed, only perform one action at a time, which is a great loss of power. Millions of small processors make more powerful things than just a very fast one as the human brain does, since it is much slower than a processor but it processes a lot more information.

Accordingly, the main goal of this research would be to adapt which emergent properties are obtained...
from combinations of simple action patterns, to understand the collective behavior of the swarms or masses of individual MCUs as such from the simple behavior of their individuals to the associate. In fact, this model is already being used to study the origin of culture. And in robotics, that is what they are doing in the Symbion Project by Schlachter et al.\textsuperscript{13} which makes use of microrobots that form various macrorobots without losing their micro-identity. This also gives them an enormous versatility that would not have been static pieces of a macrorobot. We entered to the era of social devices and networks.

It can also be very interesting to apply this model to explain social events. Our societies are sets of many individuals operating, which create, in the same way, systemic or emergent properties not foreseen when analyzing individual behavior in an isolated way. Perhaps one of the great problems of our world is that these properties occur without being foreseen or controlled by the authorities (environmental deterioration, economic crisis, etc.). In this way, perhaps it is that the model of government should not be channeled to control all sectors of the system (absolutely immeasurable), but only to coordinate them to achieve the desirable emerging consequences. More than directing part by part, it is better to study the communication between them and their forms of interaction. So we can say that swarm or collaborative intelligence leads us to a distributed social and IoT networks.

In general, collective behavior or SI has the following features:

- Self-organization.
- Stigma: collaboration through the physical medium.
- Emergent behavior: distributed intelligence or sturdiness.
- Strong local interaction.
- Highly structured social organization.
- Collaboration versus competition.
- Stochastic components.
- Bio-inspiration.

We can highlight that a SI system could normally seem chaotic and meaningless, speaking of the behavior of a flock, a school, a herd, or specifically a swarm, when transformed into a group of individuals that communicate and cooperate in unison to guarantee social welfare, to be cataloged as superorganisms, especially referring to the case of termites, ants, and bees, for instance.

Moreover, Beni and Wang\textsuperscript{14} defined the SI as the collective and decentralized behavior of self-organized systems in a natural or artificial way, allowing the intelligence of simple systems be transformed into the intelligence of complex systems. Something that is remembered in this sense is that mold that behaves like a multicellular organism, when environmental conditions are propitious, but it becomes unicellular when it encounters inhospitable situations and latent waiting, the propitious moment, to return as mold.

We are observing intelligent systems that self-organize and work cooperatively for the benefit of their society, whether lasting or transient, following a series of binary instructions, which are reinforced with the response they get from their environment, their mistakes, their interaction with other individuals, and their iterations in carrying them out. Emphasizing that there is no leader, but the main task to strictly follow the instructions, learning from their failures.

Every SI system is made up of a group of individuals, being their organisms or subsystems, in general. We adapt an IoT network with SI, so the MCUs learn simple tasks and interact without having a central command system. Thus, the interrelationships between MCUs generate enormous advantages with respect to the search and execution time, which produces an intelligence behavior by the whole and not by the individual performance.

### Wi-Fi IoT development boards

#### Based on ESP8266

ESP8266 is a low-cost Wi-Fi chip that works through the transmission control protocol/internet protocol (TCP/IP) protocol. It includes a microcontroller (Tensilica Xtensa LX106) to manage this protocol and the necessary software for the 802.11 connection. In addition, most models have general purpose digital inputs/outputs (GPIO), as well as an analog input with 10 bit analog–digital converter.

By the price of 3 dollars, it is feasible to program directly the Wi-Fi connection in an Arduino microcontroller because it is a perfect chip to develop some IoT applications.

For carrying out this work, we use the ESP-12-E model for being the ESP8266 most widely used over the world. However, the software and hardware of the NodeMCU allow one to consider them as an open-source project for integrating the IoT from the development of a simple model based on Wi-Fi. For example, Jabbar et al.\textsuperscript{15} proposed a system which uses a Wi-Fi-based gateway to connect different sensors and updates its data to Adafruit IO cloud server.

When using the ESP-12-E model, programming is entirely transparent, since it does not requires any change in its pins for programming, and having a USB connection like Arduino. The board by itself has a power regulator, as well as a USB-serial chip for the ESP8266 communication directly from the computer’s USB, in addition to an easy access to the GPIO pins, a
1 Mb flash memory for the storage of several programs, two Rx/Tx ports, etc. Briefly, it is the easiest and cheapest way to develop IoT applications due to the large community and the large number of compatible software and existing libraries for programming the ESP8266.

**Based on the Raspberry Pi 3 B+**

Raspberry is a low-cost computer board developed in the United Kingdom by the Raspberry Pi Foundation, with the aim of stimulating the teaching of computers in schools. The Raspberry Pi 3 model B is the third generation of Raspberry Pi. It can be used as a low-power web server and build your own Smart-TV, as a print server and make your own cloud, as a global positioning system (GPS) locator. There are thousands of projects on the Internet to get the most out of it.

Many Linux operating systems have been optimized for the Raspberry Pi and also to give any use to this board you have a very versatile operating system is the RASPBIAN OS that you can download from the official website.

In addition, Raspberry Pi offers the Raspberry Pi 3 B+ model, which is useful for learning, coding, and creating projects. The Pi B+ model is the latest product in the Raspberry Pi 3 range and features a quad-core 64-bit processor running at 1.4 GHz, dual-band 2.4 GHz and 5 GHz wireless LAN, Bluetooth 4.2/ BLE, faster Ethernet, and power over Ethernet (PoE) capability through separate PoE hardware attached on top (HAT). The Raspberry Pi 3 model B+ has achieved the certification of modular compliance, which allows to design the plate in final products with proof of compliance of wireless LAN significantly reduced, improving the cost and time of commercialization. All this happens while maintaining the same mechanical footprint as the Raspberry Pi 2 model B and the Raspberry Pi 3 model B.

**Wi-Fi P2P and direct**

Wi-Fi direct is a form of Wi-Fi connectivity, which allows connecting two certified devices such as Wi-Fi direct, such as tablets, telephones, printers, or MCUs, directly. Unlike a normal Wi-Fi connection, Wi-Fi direct does not need an access point to manage the connection, such as a router or hotspot, that is, without that access point, the devices cannot be seen; Wi-Fi direct can send files directly and at a speed and reach far superior to Bluetooth, approximately ten times faster and almost four times more range.

These devices must meet certain requirements:

- Be compatible to use that protocol, here we must point out that only with one of the devices integrated with Wi-Fi direct, the connection would be possible.
- Be in the same physical space, since the distance, although greater than by Bluetooth, must be adequate for the devices to detect each other.

The process of sharing files using Wi-Fi direct is similar to how it is done through Bluetooth. Once activated on both devices, each one identifies the other and once the connection is accepted, we can send the files to us. In MCU devices, generally, if we go to settings and then to Wi-Fi, we can activate the Wi-Fi direct system so that this device is visible by others who are also using Wi-Fi direct. Once the devices detect each other from one, the other is invited to establish a connection, and when it accepts, the two can exchange files. The way to send files is similar to how we do when we share something on the network. Thus, some devices that although detect other connected devices and are detected in turn, they do not have the options to share the one of Wi-Fi direct. The solution for these would be to install an application that incorporates that option between the typical connection.

On one hand, Bluetooth is better known and supported by more variety of devices. However, Wi-Fi direct has a higher transfer rate, advanced encryption standard (AES) of 256 bits, while Bluetooth of 128 bits, and greater range, being 100 m the maximum distance to which the devices that are connected and 50 m can be in the Bluetooth case. On the other hand, among the disadvantages of Wi-Fi direct, we can find by requiring greater energy consumption, when we have devices equipped with a battery, it can be used up more quickly. In reference to the number of devices that support the Wi-Fi direct protocol, although more and more manufacturers choose to incorporate it, which incorporates in its products another similar technology called AirPlay. With this we leave out the possibility of exchanging files using Wi-Fi direct among a big amount of devices even MCUs. Although Wi-Fi direct is a very useful and easy-to-use technology, and which has been present in our devices for several years, this is why Demir states that there is still a lot of ignorance, and consequently it is not used as much as one might expect.

**Hilbert fractal curve**

Recursive geometric mathematical objects are mathematical monsters and fractals. Some fractals, such as the Peano and Hilbert fractals, are curves that travel through all the points of a square, that is, they are filling-space curves which are frequently found in nature. For example, there are three systems with
access to each part of the kidney, namely: arterial, venous, and urinary systems, whose dynamics can be modeled as a filling-space curve to organize efficiently their tangled structures.\(^1\)

To facilitate the construction of fractals due to their recursivity, the L-systems paradigm was proposed by Lindenmayer\(^1\) as a language of fractals and created to describe the processes of growth in nature. In addition, L-systems are constituted by formal descriptions known as parallel rewriting systems.

A L-system is made up by the rules of production, which refers to the mechanism by which the growth process is generated, and by the axiom, which is the initial state in which the L-system is located. A L-system can be understood as a feedback machine that operates in chains, where the rules of production (or rewriting) point out the way a given input chain, through iterations, is transformed into an output chain. Once the production rules have been established, the dynamics of the same can be simulated in a computer to generate a long chain of symbols translated into a graphic output, which allows to compare the image with the behavior of the real object.

The production rules in L-systems require a graphic validation, thus Papert\(^2\) proposed a graphical interpretation algorithm to construct fractals, including the filling-space curves. This algorithm allows to develop L-systems with very few production rules, starting to take an object moving through the system, in order to trace a path or curve based on its movements. Immediately, several commands are taught to the object in the form of symbols used in the L-system. For this work, we were interested specifically in three commands:

1. \(\Phi\), move forward by a certain fixed step length \(\eta\).
2. \(\odot\), turn left (counterclockwise) by a fixed angle \(\psi\).
3. \(\odot\), turn right (clockwise) by the angle \(\psi\).

Accordingly, our object must decode symbols based on sequences of the above three commands:

\begin{enumerate}
\item \(\Phi\), move forward by a certain fixed step length \(\eta\).
\item \(\odot\), turn left (counterclockwise) by a fixed angle \(\psi\).
\item \(\odot\), turn right (clockwise) by the angle \(\psi\).
\end{enumerate}

Fractal adaptive reconfiguration in an IoT network

Our proposed adaptive reconfiguration algorithm is made up of three steps:

- Identifying the main MCU or hub node in the IoT network.
- Designing the topology or structure of the IoT network.
- Indexing, connecting, or establishing of links between the nodes of the IoT network.

However, any configuration of our IoT network consists of a WAP and an \(\alpha\)-number of elements, namely, embedded systems or MCUs, sharing parameters among them inside a given room or floor plant (see Figure 2). In this case, \(\alpha\) is a range that meets the following expression

\[ 4^{n-1} \geq \alpha > 4^n - 1 \quad (1) \]
We also define $\alpha$ by means of equation (2) as

$$\alpha = \sum_{i=1}^{4^n} MCU_i$$

where $n$ is the Hilbert fractal level for reaching the last $MCU_i$ from the first one, according to equation (3). Here, $i$ denotes the index inside the IoT network of a given MCU

$$n = \lceil \log_4(\alpha) \rceil$$

In this case, $MCU_1$ is the first MCU, $MCU_2$ the second one, and so on.

The IoT network has $\alpha$ MCUs at specific time, reaching up to $4^n - 1$ sensor nodes. To design a new topology when $\alpha \geq 4^n$, the $n$ value must be recalculated using equation (3). Otherwise, the proposed algorithm just attaches a new device in the network, in order to link or index every sensor node along of the network by means of the Hilbert fractal curves generated previously. Immediately, it must measure the effectiveness of our algorithm by exchanging the parameters throughout the network. Finally, we configure the point-to-point algorithms of Wi-Fi (P2P) and SoftAP.

Step 1: identifying the main MCU

First, we assumed that $\alpha$ IoT Development Boards are randomly distributed along a certain floor plant what makes up a matrix $\gamma$ with dimension $2^n \times 2^n$, so that the main WAP is located in the upper left corner, although it can be placed anywhere within the room, to have connections only in the southeast of it and a positional configuration of the topology can be achieved, that is, all connections to the WAP will be below and to the right of it. According to the above, all sensor nodes try to be linked to the nearest WAP by means of Institute of Electrical and Electronics Engineers (IEEE) 802.11 b/g/n protocol, but some will find it and others will not. All the sensor nodes connected to the WAP are completely disabled from the Wi-Fi direct mode, meantime the WAP sends a broadcasting to the entire network to measure the linking-speed of every sensor node communicated to it. However, we denoted MNI to label each microcontroller network identifier with 8 bits, since our system is constrained to link until $2^8$ development boards. Thus, the Hilbert filled-space curve will only grow up to the fourth iteration or level. In the first iteration, a consecutive number of MNIs will be assigned based on their linking-speed to the WAP. Therefore, the $MCU_i$ with the best linking-speed will be labeled as $i = 1$ or $MCU_1$. Since the walls attenuating signals, there are certain devices or sensor nodes with shorter linear distance having a slower maximum speed than others that are farther from the WAP.

Step 2: designing the topology

When the $MCU_{i-1}$ is identified, it is assigned to it as the main mode. Then, all users or $\alpha$-embedded systems with enough bandwidth are capable to reach a link enable Wi-Fi direct mode. With this, each device, except the $MCU_{i-1}$, fill a table with the respective bandwidths of all the sensor nodes to which it has access, it is listed as a list of accessible nodes or LRN. Each device estimates an LNR, so within a network it is possible to generate up to $4^n$ LCRs.

Thus, each LRN $i$ has the linking-speed of all the $MCU_i$, with which it is linked. Something very important to note is that in the IoT network all $MCU_i$ are connected at least with other. When the LRN $i$ are shared all the $\sum_{i=1}^{4^n} MCU_i$ sensor nodes of the network know what is the path to reach any sensor node, that is, everyone knows the connection topology. So far, no AI tool is used; on the contrary, it is a filling of tables or lists to optimize the connection of each sensor node of the IoT network.

Step 3: indexing of links

To generate the Hilbert fractal curve, we applied the L-system paradigm and the Papert's graphic interpretation, in order to the smallest elements (symbols) be replaced. In our case, the Hilbert curve are made up of copies of the $H_0$ in $H_1$. Subsequently, each of these copies is replaced by a copy of $H_1$ to yield $H_2$. Later, we applied the L-systems to carried out formal substitutions for adjusting automatically the appropriate size of the line segments or MCUs used to connect.

After choosing the angle $\eta = 90^\circ$, we encoded $H_0$ by the symbol $\rho$ (interpreted by the parameter to share as $\Phi$ and $\Phi$) and $H_1$ by $\rho$ (see Figure 3). The parameter to be shared leads to the right in both the start and end positions. Since $H_1$ is yielded from $H_0$, the latter determines the first production rule.
of the L-system. For yielding $H_2$, we rewrote the rule by $\rho$. The symbols reflected from $\lambda$ and $\rho$ are only different by their orientation. Moreover, the $\lambda$-curve is traced clockwise $\circ$ by the parameter to be shared starting with a left turn $\circ$, whereas the $\rho$-curve is traced counterclockwise by the same parameter starting with a right turn. Thus, the rewriting rule of $\rho$ is a mirror of the rule for $\lambda$, that is, a right turn yields a left turn and vice versa. Finally, it is necessary to exchange the $\rho$ and $\lambda$ symbols.

Since the goal to yield the topology of the IoT network is to enable the parameter sharing among their sensor nodes, and based on the Hilbert fractal curve, we proposed the following L-system:

\[
\begin{align*}
\text{Axiom} & \quad \rho \\
\text{Production rules} & \quad \rho \rightarrow \circ \rho \Phi \circ \lambda \Phi \lambda \circ \Phi \rho \\
& \quad \lambda \rightarrow \circ \lambda \Phi \circ \rho \Phi \rho \circ \lambda \Phi \\
& \quad \Phi \rightarrow \Phi \\
& \quad \circ \rightarrow \circ \\
& \quad \circ \rightarrow \circ \\
\text{Parameter:} & \quad \eta = 90^\circ
\end{align*}
\]

Figure 3. Symbols $\lambda$ and $\rho$ encode the geometric interpretation of the curve on the left side. On the right side, axioms $\lambda$ and $\rho$ generate unfolded representations of the Hilbert fractal curve.

For the first three stages, namely, we can scan $4^3 = 64$ MCUs or $\sum_{i=1}^{4^3} MCU_i$, this L-system gives

\[
\begin{align*}
\text{Stage 1} & \quad \rho \\
\text{Stage 2} & \quad \rho \Phi \rho \Phi \rho \Phi \circ \lambda \Phi \lambda \circ \Phi \rho \\
& \quad \rho \Phi \rho \Phi \rho \Phi \Phi \circ \lambda \Phi \lambda \circ \Phi \rho \\
& \quad \rho \Phi \rho \Phi \rho \Phi \Phi \circ \lambda \Phi \lambda \circ \Phi \rho \\
& \quad \rho \Phi \rho \Phi \rho \Phi \Phi \circ \lambda \Phi \lambda \circ \Phi \rho \\
& \quad \rho \Phi \rho \Phi \rho \Phi \Phi \circ \lambda \Phi \lambda \circ \Phi \rho \\
& \quad \rho \Phi \rho \Phi \rho \Phi \Phi \circ \lambda \Phi \lambda \circ \Phi \rho \\
& \quad \rho \Phi \rho \Phi \rho \Phi \Phi \circ \lambda \Phi \lambda \circ \Phi \rho \\
& \quad \rho \Phi \rho \Phi \rho \Phi \Phi \circ \lambda \Phi \lambda \circ \Phi \rho \\
& \quad \rho \Phi \rho \Phi \rho \Phi \Phi \circ \lambda \Phi \lambda \circ \Phi \rho \\
& \quad \rho \Phi \rho \Phi \rho \Phi \Phi \circ \lambda \Phi \lambda \circ \Phi \rho \\
& \quad \rho \Phi \rho \Phi \rho \Phi \Phi \circ \lambda \Phi \lambda \circ \Phi \rho \\
& \quad \rho \Phi \rho \Phi \rho \Phi \Phi \circ \lambda \Phi \lambda \circ \Phi \rho \\
& \quad \rho \Phi \rho \Phi \rho \Phi \Phi \circ \lambda \Phi \lambda \circ \Phi \rho \\
& \quad \rho \Phi \rho \Phi \rho \Phi \Phi \circ \lambda \Phi \lambda \circ \Phi \rho \\
\end{align*}
\]

Table 1 shows all $\frac{64^n}{n}$ $MCU_i$ distributed randomly in a conventional room. It is important to mention that reordered as vector is $\vec{k}$. Furthermore, Figure 4(c) shows the order of the Hilbert fractal curve of the $MCU_i$, giving as a result the indexed vector $\vec{k}$.

The possible parameters are expressed in Table 1, and these parameters are shared along the IoT network, thus all the nodes have all the types of sensors that exist in the network. We proposed an 8-bit marker since it would give us as a result that we can differentiate and shared 28 or 256 types of sensor nodes, thus this marker could identify a temperature sensor or simply a smart phone, for instance.

Equation (4) shows the estimation of the $n$ stages of the an optimized path, using the signal strength correlation or link bandwidth

\[
\phi_{kl} = \frac{\sum_{i=1}^{n} (k_i - \bar{k})(l_i - \bar{l})}{\sqrt{\sum_{i=1}^{n} (k_i - \bar{k})^2} \sqrt{\sum_{i=1}^{n} (l_i - \bar{l})^2}}
\]

where the coefficient of correlation of bandwidth sample is expressed by $\phi_{kl_i}$, level of the Hilbert fractal curve (Figure 4(a)) or $n = 3$, in a $MCU_i$, $k_i$ and $l_i$ represent the best two results in terms of bandwidth, while $ith$ denotes the index of a certain $MCU_i$ of the IoT network and $\bar{k}$ and $\bar{l}$ are the average of the two best results along the whole network. In this way, a particular $MCU_i$ best linked can be estimated.

Figure 4(b) depicts all $\sum_{i=1}^{64} MCU_i$ distributed randomly in a conventional room.
LSNi(t) is estimated using the methodology exposed in Algorithm 1, where i is the number of reachable nodes and LRNi(t) is the list of them. In order to consider a significant sensor node, a threshold bandwidth $B_{thr}$ and an average bandwidth $m_i(t)$ are defined. In addition, $\sigma_i(t)$ is the variance of $LRN_i(t)$.

**Experimentation**

This section is divided as follows:

1. Experiment Setup,
2. Network topology estimation,
3. Discussion, and
4. Impact of node density and QoS

The basis of experimentation are defined, the estimation of the IoT Network topology is made, and an analysis of the obtained results are made. Then, the discussion of the results of finding are exposed. Finally, the impact of node density and QoS is exposed.

**Experiment setup**

A specific topology in a certain room is generated in order to propose an experimental system that is divided mainly as follows:

1. WAP.
2. IoT Development Boards.
3. The place where the Development Boards are placed.

We defined as the main sensor node of the IoT network a Router Tenda-AC1200. This WAP has a Broadcom 900MHz Processor and it can handle a total available bandwidth of 1.2 Gbps due to two Wi-Fi simultaneous 2.4 GHz 300 Mbps and 5 GHz 867 Mbps connections.

Moreover, we used 18 boards Raspberry Pi 3 B + and 46 boards ModeMCU-ESP8266, giving as a result 64 IoT Development Boards for conforming the simulated Network.

Both MCU-RPi3 and MCU-ESP8266 support the IEEE 802.11n standard, so we configured an IoT network with up to 300 Mbps at 2.4 GHz as maximum bandwidth.

Finally, the 64 IoT Development Boards were randomly distributed along an area (Figure 2). The connection depicted by Figure 5(a) shows the reachable sensor nodes by the main router, giving an idea of how these boards are located before estimating the connection among them and the ways to connect and share their data obtained by their different sensors.

An initial configuration can be defined as follows:

| Algorithm 1: Estimation of significant nodes ($\rho_i(t)$) and the list of significant nodes (LSNs) |
|---|---|
| **Input:** & $i$, LRNi(t) |
| **Output:** & $\rho_i(t)$, LSNi(t) |
| 1 & $m = LRNi(t)$ size for the MCUi(t) |
| 2 & LRNi(t) is sorted, depending on bandwidth link in $S_{LRNi}(t)$ |
| 3 & $B_{thr} = \mu_i(t) - \sigma_i(t)$ |
| 4 & $l = 1$ and $LSN = \{0, 0, 0\}$ |
| 5 for $l = 1$ to $m$ do |
| 6 & Estimation of $B_{LRNi}(t)$ from $SLR_{LRNi}(t)$ in Kbps |
| 7 & if $B_{LRNi}(t) > B_{thr}$ AND $n < 3$ then |
| 8 & $LSN(n) = MCU_{label}(t)$ |
| 9 & $n = n + 1$ |
| 10 & $LSN(n) = \{\text{filter the best 3 links in } B_{LRNi}\}$ |
| 11 & All the elements $\neq 0$ are stored in $\rho_i(t)$ |

| Table 1. Examples of types of sensors in an IoT network or list of sensor network, LSNi(t). |
|---|---|
| 0000 0000 & GPS |
| 0000 0001 & Passive infrared sensor |
| 0000 0100 & Door control |
| 0000 0110 & RFID |
| 0000 1000 & Biosensors |
| 0000 2001 & Webcam |
| 0000 1111 & Temperature control |
| 0000 1000 & Temperature sensor |
| 0000 1000 & CO2 |
| 0000 1000 & Biorhythm |
| 0000 1000 & Wi-Fi door bell |
| 0000 1000 & General purpose MCU |
| 0000 1000 & Gas sensor |
| 0000 1100 & Smart phone |
| 0000 1110 & Humidity |
| 0000 1111 & Pressure sensor |

MCU: microcontroller unit; GPS: global positioning system; RFID: radio-frequency identification.

$LSN_i(t)$ is estimated using the methodology exposed in Algorithm 1, where $i$ is the number of reachable nodes and $LRN_i(t)$ is the list of them. In order to consider a significant sensor node, a threshold bandwidth $B_{thr}$ and an average bandwidth $\mu_i(t)$ are defined. In addition, $\sigma_i(t)$ is the variance of $LRN_i(t)$.
The axiom chosen by the Hilbert fractal curve indicates that the main WAP must be placed in the upper left quadrant. The IEEE 802.11 b/g/n standard is configured for connecting all 64 devices to the WAP. The connection to the WAP is established by some sensor nodes in the network. Figure 5(a) shows those who do not achieve this connection with the WAP, then enable Wi-Fi P2P mode in listen or wait mode. All the sensor nodes connected to the Tenda-AC1200 are completely disabled from the Wi-Fi direct mode. To measure the linking-speed of each node, the WAP broadcasts. With the response of each MCU, the bandwidth of each one is estimated. It is with the above, that the 64 MNI tags or labels that this network contains can be identified. For the first time, these labels are identified and numbered consecutively. This assignment is made according to the linking-speed to the Tenda-AC1200.

With the help of Table 2 and Figure 5(a), a NodeMCU-ESP8266 development board can be identified that maintains the best average link with 222.74 Mbps, which is why it is identified with a label 1, assuming the name MCU-ESP8266-1, although within the designed network it is not important what type of IoT device is identified.

The room where the experimentation was carried out, as shown in Figure 5(a), contains some walls attenuating the signals from both the WAP and the IoT devices. It is because there are cases where devices that are closer to the WAP but with a greater obstruction, they experience a decrease in the linking-speed, and there are cases where there is no obstruction, greater bandwidths are achieved despite their greater distance to the WAP. For example, the MCU-ESP8266-16 reaches a linking-speed of 76.51 Mbps and is barely 8.5 m from the WAP since a wall divides the room obstructs it, while without obstruction the MCU-ESP8266-8 reaches a linking-speed of 145.06 Mbps, being 9 m from the WAP.

Table 2 depicts a list of devices according the average bandwidth in the proposed IoT network. The MCU-ESP8266-1 and MCU-RPi3-2 obtained the two best linking-speeds with 222.74 and 218.17 Mbps, respectively. This list or table is considered full until the maximum speed of 10 links is measured.

**Estimation of the network topology**

It is important to determine the initial topology of the network and Table 2 shows the spatial position of each sensor node. The first sensor node, in terms of higher linking-speed, will be known as Node 1. For this example, this is the MCU-ESP8266-1, and the other nodes are numbered based on their linking-speed. Thus, this Node 1 is the only one that establishes a link with the main WAP, while the MNI, with \( i \geq 2 \) and \( i = 2, 3, \ldots, 2^n \), deactivates the IEEE 802.11 b/g/n standard and activates the Wi-Fi Direct mode.

Once the initial sensor node or MNI is known, the rest of the MCUs try to connect to as many sensor nodes as possible, thereby obtaining a list of reachable nodes or LRNi for this MNI, that is why the IoT network contains as many LRNi as there are in MNI.

Thus, it can be mentioned that any IoT network configuration has up to \( 2^n \) LNRs, Figure 5(b) shows...
the connections that the MNI has, which can be connected to 19 other MCUs, in order to fill the LNR. The maximum average bandwidth of each MNI is shown in Table 3, where the bold indicates the two best results for the MNI, while the results in italics highlight the two worst links of MNI.

From Table 3, it can be highlighted that the worst linking-speed is obtained by connecting MNI with 126.78 Mbps, while the best linking-speed is obtained by connecting MNI with 227.31 Mbps.

Table 3. LRN for the MCU-RPi3-7. Bold results indicate the two best links in terms of bandwidth for MNI, while italics highlight the worst bandwidth connections.

| MNI          | Linear Distance (m) | Actual Distance (m) | Bandwidth (Mbps) |
|--------------|---------------------|---------------------|------------------|
| MCU-RPi3-11  | 6.23                | 6.23                | 195.32           |
| MCU-RPi3-15  | 6.53                | 11.48               | 147.34           |
| MCU-RPi3-2   | 5.53                | 8.98                | 170.19           |
| MCU-RPi3-27  | 9.53                | 9.93                | 161.51           |
| MCU-ESP8266-1| 5.73                | 10.63               | 155.11           |
| MCU-ESP8266-3| 7.48                | 7.48                | 183.90           |
| MCU-ESP8266-4| 4.73                | 4.73                | 209.03           |
| MCU-ESP8266-5| 4.68                | 4.68                | 209.49           |
| MCU-ESP8266-6| 2.73                | 2.73                | 227.31           |
| MCU-ESP8266-8| 3.03                | 3.03                | 224.57           |
| MCU-ESP8266-9| 5.23                | 5.23                | 204.46           |
| MCU-ESP8266-10| 4.73            | 4.73                | 209.03           |
| MCU-ESP8266-12| 8.23            | 8.23                | 177.04           |
| MCU-ESP8266-13| 7.73            | 7.73                | 181.61           |
| MCU-ESP8266-14| 10.73           | 10.73               | 154.20           |
| MCU-ESP8266-16| 8.63            | 13.73               | 126.78           |
| MCU-ESP8266-22| 7.43             | 10.03               | 160.59           |
| MCU-ESP8266-23| 6.23             | 6.23                | 195.32           |
| MCU-ESP8266-28| 8.43             | 10.63               | 155.11           |

LRN: list of accessible nodes; MNI: microcontroller network identifier; MCU: microcontroller unit.

Discussion

The similarity or Hausdorff–Besicovitch dimension D of a self-similar fractal formed by n copies of itself at scale r is defined as $D = \log(n)/\log(r)$. This is why the Hilbert fractal curve with $n=4$ and $r=0.5$ approximates its dimension D practically to 2. This feature helps to perfectly order a bi-dimensional matrix into a vector, increasing redundancies in the neighborhood. AI relies on the optimal choice of connection, and thus reconfigure it, depending on how many MCUs are connected or disconnected to the IoT network.

The position within the IoT network of each device is shown in Figure 4(b). From Figure 6, the way to communicate to MNI with MNI is first to communicate to MNI with MNI, then with MNI, to finally arrive at MNI. This path seems to be longer in distance but it is always the highest in linking-speed, so packets arrive faster on this path than if they were connected directly.

In addition, each MCU has a certain position within the room or floor layout, which is optimized according to the highest linking-speed (Figure 4(b)). If the Hilbert vector indexing is done, it is found that the best way to send an information packet will always be MNI or MNI, that is with its closest neighbor. Therefore, a general increase is made in the average bandwidth of the entire IoT network due to the synergy generated by all the devices that operate as if they were a single organized entity. Also, the IoT network now behaves as a device where any valid parameter can be sensed in it. To exemplify a fractal of $n=3$, that is, handle up to 43 devices, Figure 6 is presented, highlighting the vector indexing in this order.

There are devices that are as far as possible from each other, these cases are considered critical. With them, the worst connectivity scenarios are measured. In this case, the bandwidth between MNI and MNI is found. Thus, 218.06 Mbps average connection MNI can be established when sending and receiving 1000 information packets, while the link MNI averages 215.72 Mbps. There are average links of 223.56 Mbps, which are closer to the maximum bandwidth established by the IEEE 802.11 b/g/n standard, that is, 300 Mbps. It is difficult to compare the proposed topology with the traditional central topology, since the very distant devices such as MNI or MNI have no connection or communication. With this mainstream topology, there would be a link only with 10 of the devices closest to the WAP, that is, the devices that appear in Table 2, while the remaining 54 sensor nodes could not communicate with the WAP, much less share their parameters between the sensors scattered in the room. This conventional topology has an average linking-speed of 157.39 Mbps, which would mean 28% less of linking-speed on the 10 reachable devices without having communication throughout the IoT network.
**QoS and impact of node density**

The performance of this proposal is evaluated in this section. This is done by measuring the impact of the density of the nodes (ND), in the first instance to then measure the QoS. Then, the performance of the distributed learning fractal algorithm (DLFA) algorithm by means of the packet delivery ratio (PDR) and the E2ED are estimated. Thus, the performance of this proposal is compared with similar algorithms such as

1. Mostafaei: Reliable routing with distributed learning automaton (RRDLA).
2. Liu et al.: Delay-energy tradeoff in WSNs with reliable communication (DETR).
3. Li et al.: Reliable and energy-efficient routing (REER).

The same network parameters, proposed by Zorzi and Rao and by Karp and Kung are employed in order to make a real comparison to the state-of-the-art algorithms.

To understand the experiments carried out, it is important to first define what the E2ED is, which is the time it takes for a packet to reach the receiver from the moment it is emitted until it is received, being then the average sum of the differences between the delays sending and receiving. Whereas PDR is the probability of recovery from the original packets by the receiver before their lifetime expires.

From these definitions, we find equation (5) which defines the rate ($\beta$) of change between the lowest and the highest delay as follows

$$\beta(\tau) = \frac{E2ED(\tau)_{\text{max}} - E2ED(\tau)_{\text{min}}}{PDR(\tau)_{\text{max}} - PDR(\tau)_{\text{min}}}$$

where $\beta \to 0$ when E2ED is constant but $\beta \to \infty$ when PDR is constant, regardless E2ED. In addition, reliability requirement is an important term in QoS, since it is necessary to measure the impact of node density or $\tau$. The same experimental design was taken from Mostafaei and we compared this proposal with the algorithms developed by Niu et al. and Zeng et al.

The comparison between E2ED and PDR is made in two ways. The first in terms of reliability requirements and node density whose results are shown in Figure 7(a) and (b), respectively. The lower the PDR, the better the performance, while the E2ED works otherwise, when it increases the performance of the algorithm, it also increases this metric. Thus, the best results are obtained by our proposal as the RRDLA algorithm, this happens because both minimize the delay for data transfer from the node sensors when the packet receiver retrieves it. This proposal is slightly better than other algorithms and exceeds them by at least 2 %. Therefore, it can be highlighted the propagation delay in the IoT network, both this proposal and RRDLA, remains stable regardless of the ND, that is, the QoS. Other algorithms such as DETR and REER obtain much larger delays than this proposal.

The impact of node density and QoS in $\beta(\tau)$ is shown in Table 4. The best result obtained by this proposed method is highlighted in **bold**, while the second best result obtained by RRDLA algorithm is shown in *italics*.

![Figure 7. QoS and ND impact: (a) reliability requirements and (b) node density.](image)

![Table 4. ND and QoS impact for $\beta(\tau)$](table)

| Algorithm | $\beta$ (ND) | $\beta$ (QoS) |
|-----------|--------------|---------------|
| RRDLA     | 0.0833       | 0.0909        |
| DETR      | 1.1667       | 0.2143        |
| REER      | 3.000        | 0.4000        |
| This proposal | **0.0417** | **0.0783** |

ND: density of the nodes; QoS: quality of service; RRDLA: Reliable routing with distributed learning automaton; DETR: delay-energy tradeoff; REER: reliable and energy-efficient routing.

The best result obtained by this proposed method is highlighted in **bold**, while the second best result obtained by RRDLA algorithm is shown in *italics*.

Finally, when a sensor shares data in the IoT network, it takes less time to reach the receiver when the proposed algorithm is used, thus, the recovery propagation of the original packet before its lifetime expires is significantly increased.
Conclusion

In past centuries, global changes in the world were gradual, but in the most recent decades, these changes have accelerated and have had a direct impact on the climate, a way to encourage the negative effects is to save as much energy as possible. This work proposed a topology applied to a specific IoT network, which is based on the fractal or David Hilbert fractal curve. A Hilbert fractal curve is a fractal with a Hausdorff–Besicovitch dimension equals to 2, which means that this type of fractal tries to space-fill an entire two-dimensional area under the paradigm of L-systems, which requires three main characteristics: an axiom, production rules, and an alphabet. Thus, this fractal was used to optimize the linking-speed and thereby reduce energy consumption.

Much of the energy wasted is due to the need of connecting the sensors to a central sensor node. If the neighboring sensor nodes are used, the average linking-speed of the network is increased. In addition, the complexity of connections is reduced and generates an entity that works as a single node sensor full of parameters that can be measured. There are various algorithms that try to make the connection between nodes of an IoT network more efficient and share their parameters more efficiently. But these algorithms have not completely solved the high costs of interconnection. Given these initial conditions, this work proposed a simple development algorithm for WSNs, using the IEEE 802.11 b/g/n Wi-Fi standard.

The analysis carried out to measure the performance indicated that the reconfiguration time of the IoT network was reduced to a few milliseconds, which increased the lifetime of the network. And the conjunction of these factors increases the efficiency of the network and thus improves the way to share the parameters in the IoT network. In other words, the IoT network was fully filled with all the sensor parameters. In the topology example that we have presented, 122.78 μs were required for the complete reconfiguration of the IoT network, when more than one sensor node and up to 4^n − 1 devices were added. The algorithm can obtain all the parameters of connected sensor nodes of the IoT network in any MCU, in a few microseconds.

Finally, the proposal was based on the use of SI, since all the sensor nodes know the parameters of the IoT network. And when there is a reconfiguration, all the sensor nodes help to get this purpose, knowing how the sensor nodes behave in terms of their linking-speed. All these changes, even the physical ones, can be shared by the same IoT network, and with it, all the sensor nodes propose the most optimal reconfiguration of the current IoT network.

Acknowledgements

The research described in this work was mainly carried out at Escuela Superior de Ingeniería Mecánica y Eléctrica, Campus Zacatenco. In addition, the authors thank Juan Posadas and Liliana Chanona for their technical contributions.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by Instituto Politécnico Nacional, Mexico by means of Projects SIP20200638 and SIP20200324 granted by Secretaría de Investigación y Posgrado and National Council of Science and Technology of Mexico (CONACyT).

ORCID iD

Jaime Moreno https://orcid.org/0000-0003-4032-0511

References

1. Hassan T, Aslam S and Jang JW. Fully automated multi-resolution channels and multithreaded spectrum allocation protocol for IoT based sensor nets. IEEE Access 2018; 6: 22545–22556.
2. Jo O, Kim YK and Kim J. Internet of things for smart railway: feasibility and applications. IEEE IoT J 2018; 5(2): 482–490.
3. Sandoval RM, Garcia-Sanchez AJ and Garcia-Haro J. Improving RSSI-based path-loss models accuracy for critical infrastructures: a smart grid substation case-study. IEEE Trans Ind Inform 2018; 14(5): 2230–2240.
4. Zhang C, Ge J, Pan M, et al. One stone two birds: a joint thing and relay selection for diverse IoT networks. IEEE Trans Veh Technol 2018; 67(6): 5424–5434.
5. Mostafaei H and Menth M. Software-defined wireless sensor networks: a survey. J Netw Comput Appl 2018; 119: 42–56.
6. Javadi M, Mostafaei H, Chowdhurry MU, et al. Learning automaton based topology control protocol for extending wireless sensor networks lifetime. J Netw Comput Appl 2018; 122: 128–136.
7. Mostafaei H. Energy-efficient algorithm for reliable routing of wireless sensor networks. IEEE Trans Ind Electron 2019; 66(7): 5567–5575.
8. Haqbeen JA, Ito T, Otsuka T, et al. Traffic adaptive hybrid MAC with QoS driven energy efficiency for WSNs through joint dynamic scheduling mode. In: Proceedings of the 2018 19th IEEE/ACIS international conference on software engineering, artificial intelligence, networking and parallel/distributed computing (SNPD).
9. Zedadra O, Savaglio C, Jouandeau N, et al. Towards a reference architecture for swarm intelligence-based internet of things. In: Proceedings of the international conference on internet and distributed computing systems, Mana Island, Fiji, 11–13 December 2017.
10. Savaglio C, Pace P, Aloí G, et al. Lightweight reinforcement learning for energy efficient communications in wireless sensor networks. *IEEE Access* 2019; 7: 29355–29364.
11. Kwon M, Lee J and Park H. Intelligent IoT connectivity: deep reinforcement learning approach. *IEEE Sensor J* 2020; 20(5): 2782–2791.
12. Moreno J, Morales O, Tejeida R, et al. Broadcasting and sharing of parameters in an IoT network by means of a fractal of Hilbert using Swarm Intelligence. In: Batyrshin I, Martínez-Villaseñor L and Espinosa H (eds) Advances in soft computing: evolutionary and nature-inspired intelligence (Lecture Notes in Artificial Intelligence), vol. 11288. New York: Springer, 2018, pp.62–75.
13. Schlachter F, Meister E, Kernbach S, et al. Evolve-ability of the robot platform in the Symbrion project. In: Proceedings of the 2008 2nd IEEE international conference on self-adaptive and self, 2008, pp.144–149, http://www.cybertronica.co/sites/default/files/publications/Evolveability-Symbrion.pdf
14. Beni G and Wang J. Swarm intelligence in cellular robotic systems. In: Dario P, Sandini G and Aebischer P (eds) Robots biological systems: towards a new bionics? New York: Springer, 1993, pp.703–712.
15. Jabbar WA, Kian TK, Ramli RM, et al. Design and fabrication of smart home with internet of things enabled automation system. *IEEE Access* 2019; 7: 144059–144074.
16. Li Z, Liang Q, Cheng X, et al. Outage performance evaluation over Rayleigh fading channels of WiFi direct networks in smart grid application. In: Proceedings of the IEEE PES innovative smart grid technologies, Tianjin, China, 21–24 May 2012, pp.1–5. New York: IEEE.
17. Demir U, Tapparello C and Heinzelman W. Maintaining connectivity in ad hoc networks through WiFi direct. In: Proceedings of the 2017 IEEE 14th international conference on mobile ad hoc and sensor systems (MASS), Orlando, FL, 22–25 October 2017, pp.308–312. New York: IEEE.
18. Peitgen HO, Jürgens H and Saupe D. Chaos and fractals: new frontiers of science. New York: Springer, 2004.
19. Lindenmayer A. Mathematical models for cellular interaction in development, parts I and II. *J Theor Biol* 1968; 18: 280–315.
20. Papert S. *Mindstorms: children, computers, and powerful ideas*. New York: Basic Books, 1980.
21. Liu Z, Dai L, Xue L, et al. Reliability considered routing protocol in wireless sensor networks. In: Proceedings of the 30th Chinese control conference, Yantai, China, 22–24 July 2011, pp.5011–5016. New York: IEEE.
22. Li X, Wang Y, Chen H, et al. Reliable and energy-efficient routing for static wireless ad hoc networks with unreliable links. *IEEE Trans Parall Distr Syst* 2009; 20(10): 1408–1421.
23. Zorzi M and Rao RR. Geographic random forwarding (GeRaF) for ad hoc and sensor networks: energy and latency performance. *IEEE Trans Mob Comput* 2003; 2(4): 349–365.
24. Karp B and Kung HT. GPSR: greedy perimeter stateless routing for wireless networks. In: Proceedings of the 6th annual international conference on mobile computing and networking, MobiCom’00, Boston, MA, 6–11 August 2000, pp.243–254. New York: ACM.
25. Liu J, Sheng M, Xu Y, et al. End-to-end delay modeling in buffer-limited MANETs: a general theoretical framework. *IEEE Trans Wirel Commun* 2016; 15(1): 498–511.
26. Yang B, Chen Y, Cai Y, et al. Packet delivery ratio/cost in MANETs with erasure coding and packet replication. *IEEE Trans Veh Technol* 2015; 64(5): 2062–2070.
27. Niu J, Cheng L, Gu Y, et al. R3e: reliable reactive routing enhancement for wireless sensor networks. *IEEE Trans Ind Inform* 2014; 10(1): 784–794.
28. Zeng K, Lou W, Yang J, et al. On throughput efficiency of geographic opportunistic routing in multihop wireless networks. *Mob New Appl* 2007; 12(5): 347–357.