Future Generation Computer Systems 80 (2018) 188–197

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Programmable architecture based on Software Defined Network for Internet of Things: Connected Dominated Sets approach

Djamila Bendouda *, Abderrezak Rachedi, Hafid Haffaf

Oran 1 Ahmed Benbella University, Faculty of sciences, Department of Computer Science Research Laboratory in Industrial Computing and Networks (LRIIR), P.OBox 1524 El M'Naouar, Oran, Algeria
University Paris-Est, LIGM (UMR8049), UPEM F-77454, Marne-la-Vallée, France

ARTICLE INFO

Article history:
Received 1 July 2017
Received in revised form 15 September 2017
Accepted 26 September 2017
Available online 10 November 2017

Keywords:
Internet of Things (IoT)
Software Defined Networks (SDN)
Centralized/distributed and semi-distributed
Connected Dominated Set (CDS)

ABSTRACT

In this paper, we propose a new programmable architecture based on Software Defined Network (SDN) paradigm for network control functionalities in Internet of Things (IoT) using the Connected Dominating Sets (CDS). In order to reduce the traffic load and to avoid a single point of failure at the controller node, we distribute the controllers role by introducing three levels of control: Principal Controller (PC), Secondary Controller (SC), the Local Controller (LC). The PC has a global view of the network infrastructure which is not the case of SC where it focuses only on one network technology. The LC acts locally by managing and relaying signaling messages from ordinary nodes to the SC. In order to select the LC nodes, we propose a Distributed Local Controller Connected Dominating Set algorithm (DLC-CDS). The DLC-CDS is a distributed algorithm with single phase and supports the dynamic network topology. The selection strategy of DLC-CDS is based on an important function named score, which is computed using the fuzzy logic and it depends on several parameters such as: the connectivity degree, the average link quality, and the rank. The performance of the proposed DLC-CDS are evaluated and compared with another solution named Distributed Single Phase-CDS (DSP-CDS) using many scenarios with different parameters: the node density and the radio range. The obtained results show that the DLC-CDS converges rapidly with a minimum CDS size compared to a DSP-CDS.

1. Introduction

The Internet of Things (IoT) is a new concept that is promptly gaining success in the field of modern wireless telecommunications. The basic concept idea of IoT is the pervasive presence around us of a variety of smart objects or things through the internet, which can be considered as Radio-Frequency IDentification (RFID) tags, sensors, actuators, mobile phones, machines, devices, vehicles, and people equipped with wired or wireless communication ability. These smart objects, through unique addressing schemes, are able to interact with each other and cooperate with their neighbors to reach common goals [1,2].

The application field is large such as: smart cities, health-care, intelligent transportation systems, future manufacturing, border monitoring, etc. [3–7]. The IoT paradigm needs a specialized software with networking functionality and IP support, it combines the dimensions of conceptual and technical orders. From a conceptual point of view, the IoT characterizes connected physical objects having their own digital identity and capable of communicating with each other [3]. This network creates a gateway between the physical world and the virtual world. From a technical point of view, the IoT consists of direct and standardized digital identification (IP address, SMTP protocol, http...) of a physical object by means of a wireless communication system which can be a smart thing such as RFID, Bluetooth, or WiFi [8]. Many communication and network technologies are proposed for IoT context without taking into account coexistence and interoperability.

We distinguish licensed and unlicensed technologies. In the case of unlicensed technologies and according to applications context an important number of standards can be used such as: IEEE802.11 (WiFi), IEEE802.15.1 (Bluetooth), IEEE803.15.3 (UWB), IEEE802.15.4 (Zigbee, 6lowpan), IEEE802.15.6 (WBAN), etc. In the case of pervasive communication, licensed technologies are well adapted such as: LTE-A with Machine-Type Communication (MTC) and Device-to-Devices (D2D) communication in 5G. That is why it is important to propose a hybrid and programmable architecture able to consider the different technologies' characteristics, and to ensure their coexistence.

In this paper, we tackle the heterogeneity of wireless network technologies in the IoT context by proposing a new hybrid and

* Corresponding author.
E-mail addresses: bendjamila39@gmail.com, bendjamila@univ-mascara.dz (D. Bendouda).

https://doi.org/10.1016/j.future.2017.09.070
0167-739X/

programmable architecture based on Software Defined Network (SDN) paradigm. The main aim is not only to consider the heterogeneity of network technologies, but also to reduce the overhead related to network control mechanisms. Unlike the classical SDN, we propose a semi-distributed approach with three control levels: Principal (PC), Secondary (SC), and Local (LC) controllers.

The PC is located in the network core and it has a global view of the network infrastructure. On the other hand, SCs are located in the edge network and focus only on one technology by acting as controller. Finally, LCs are acting locally in access network by managing and relaying signaling messages from ordinary nodes to the SC. In order to select LCs nodes, we propose a new Distributed Local Controller algorithm (DLC-CDS) based on Connected Dominating Sets (CDS) and Fuzzy set approaches. The DLC-CDS algorithm selects the LC nodes using scoring strategy based on fuzzy set approach considering different parameters such as: the connectivity degree, the average link quality, and the distance from the gateway (Secondary controller). Each LC node uses one-hop neighborhood information and makes a local decision on whether to join the dominating set, if its score is superior to the other nodes.

Finally, we evaluate the performance of the proposed algorithm (DLC-CDS) and the scoring function strategy using different scenarios and network parameters. The obtained results are compared to other existing algorithms like DSP-CDS algorithm [9,10].

The rest of the paper is organized as follows: we present a brief overview of the existing SDN architectures, and Connected Dominating Set (CDS) algorithms in Section 2. In Section 3, we describe the proposed hybrid and programmable architecture, its different modules and their interaction. In Section 4, we present the local controller selection mechanism used with the DLC-CDS algorithm, and the key score parameter based on fuzzy set approach. In Section 5, we discuss and analyze the obtained simulation results to evaluate the performance of the proposed DLC-CDS selection mechanism. Finally, the conclusion and future works are addressed in the last section.

2. Related work

In this section, we present a brief overview of SDN-based architectures for the Internet of Things (IoT), and then browse the works focused on Connected Dominating Sets (CDS) algorithms with different approaches.

2.1. SDN-based architectures for IoT

There is two work researches category for the using of the SDN-Controller position, which the one is the Unique Central SDN-Controller proposition class and the second class is for the Multi SDN-Controller proposition.

The multi SDN-controller proposition class

The SDN has been proposed in the literature as a promising solution to introduce a flexible solution for it. For instance, the standardization issues and the impact of SDN on Wireless Sensor Networks (WSNs) and IoT in terms of deployments is discussed in [11,12]. In [11], the author presents the introduction of SDN in WSN and particularly Tiny platform protocol named TinySDN. The communication protocol between controller and sensor nodes is described. In [12] TinySDN is proposed as a multiple controller TinyOS based architecture within the WSN. It defines two main components: (i) the SDN-enabled sensor node, and (ii) the SDN controller node. The experimental results of TinySDN are presented and discussed. However, the description of the communication between the controllers and the controller selection process is not present. In addition, TinySDN is based on centralized approach without any detail about the added value of the routing process like RPL protocol.

The unique central SDN-controller proposition class

Another work is proposed to extend SDN to WSNs named SDN-WISE [13–15]. In [13], the authors present a Stateful SDN solution for WSNs in order to reduce the exchanged information between the sensor nodes and the SDN controllers. Two kinds of nodes are defined in the SDN-WISE network: the sensor node running at the data plane and the sink node which is the gateways between the sensor nodes and the controller running at the control plane. The WISE-Visor is defined as the adaptation layer for the management control.

The improvement of SDN-WISE protocol is proposed in [15] to reduce overload related to the network control, and to support the big data processing using MapReduce operation. In [14], the Open Network Operating System (ONOS) is proposed as an enhancement of the NOS (Network Operating System) for IoT. The idea consists in distributing network OS to manage network operations using SDN approach. The ONOS solution has been suitably extended to further improve the recently proposed SDN-WISE platform to support SDN using the OpenFlow standard in WSN [16].

According to the SDN-WISE proposed architecture [13], it should be noted that the sensor nodes run operations at data plane, which exceed their capacity in memory and energy consumption.

On the other hand, knowing that the node sink has different from the sensor nodes in capacity of resources, it represents just a gateway between the data plane and the control plane. The SDN-WISE does not respect the characteristics of the WSN in terms of the limited resources (process, memory occupancy and consumption of energy). In addition, there is a lack of cooperation between the controllers in case they run either in same node hosting the TM layer or in remote servers. Finally, the main proposed SDN-based architecture is considered as a centralized approach with a single controller located at Sink or network infrastructure.

2.2. Connected dominating sets (CDS)

The Connected Dominating Set (CDS) or virtual backbone is part of the graph theory. The CDS of graph $G = (V, E)$ is a subgraph of G where nodes are connected and dominating others. The size of a CDS is defined as the number of dominating nodes.

In literature, we distinguish not only two main CDS approaches: centralized and distributed, but also algorithms with several phases or one single phase. In the case of a centralized approach, the network topology is assumed as available which is not suitable for some scenarios of mobile wireless networks [17]. However, in the case of the decentralized approach, the local network information is essential. The decision is made in a distributed manner at each node [9] and [10].

For more details on the classification and performance of the CDS construction algorithm, a comparison of the major works related to CDS construction has also been provided in [9]. In this paper, we focus on the distributed CDS algorithm particularly the Distributed Single-Phase algorithm (DSP-CDS) [9]. This approach is suitable for wireless networks.

For the LC node selection in our proposed architecture, we have used the CDS construction approach with the Distributed local controller (DLC-CDS) algorithm [18]. It is the efficiency algorithm for a connected dominating set for the LC node selection. Compared to the other distributed algorithms existing in literature, DLC-CDS is based on the score computation with fuzzy logic method, it is a main goal to generate a CDS with a minimum delay in different radio range and node density situations.

3. The proposed architecture

In this section, we describe in detail the proposed hybrid and pro active architecture based on SDN paradigm with semi-decentralized approach for the network control process. This architecture is hybrid because it is able to support centralized and semi-distribute approach for the control process. It is described as proactive because of the capability to make a dynamic decision about the control process strategy while taking into account the network characteristics.

In the Internet of Things (IoT) many communication and network technologies exist with different and heterogeneous characteristics: short and long range communication, static and dynamic topology, low-rate and high-rate communication, with and without energy constraint, licensed and unlicensed communication link (spectrum), etc. However the coexistence of these technologies and the network resources management are still challenging issues in IoT [19,20,5,21].

Using SDN paradigm allows to benefit from the main advantages that are the separation between control and data plans, and to make the network easy to configure and to deploy. The control plan has key role in the network, particularly in the resources management, the admission control, the routing and forwarding process, etc.

Regarding the data plan focuses only on the content of packets and their characteristics in terms of resource needs. That is why, the flexibility and re-configurability of the network are a real added value in heterogeneous networks with different characteristics and needs. Unlike the existing architectures based on SDN paradigm where the control node is centralized to have an overview of the network, in the proposed architecture we introduce different levels of the control process [22,11,14].

The main idea consists in selecting some particularly nodes to act as controller at certain parts of the network. We introduce three control levels: principal controller (PC), secondary controller (SC), and local controller (LC). The Fig. 1, illustrates an overview of the proposed architecture which is divided into five main modules: (1) Control module, (2) Data module, (3) Cloud and Fog module, (4) Security and privacy module, and (5) End-users module. For the existing interaction between these modules, is shown in Fig. 2.

3.1. Control module

In the control module, we distinguish three types of controllers: (i) the principal controller (PC), (ii) the secondary controllers (SC) and the local controllers (LC).

3.1.1. The principal controller (PC)

The PC is a centralized network controller, which presents the first level of the network control. PC is located in the network infrastructure and it has an overview of the global network in terms of network architecture/topology, and different network parameters. The main roles of the PC are as follows: coordinate the other controllers (SC, and LC), manage the network resources while considering the network heterogeneity, and configure the network. In Fig. 1, the PC is presented by the control server.

3.1.2. The secondary controller (SC)

The SC is the second level of the control which is located at the edge routers. The SC is acting as intermediate nodes between PC and Local Controllers (LC) that means that all the control communication between PC and LC go through SC. On account of their strategic location, SCs are entry points to address different constraints related to networks and wireless technologies. The SC can play a key role in the case of heterogeneous networks. For instance, SC can adapt resource sharing and management strategies (e.g. scheduling policies, routing process) according to different parameters such as: short and long range communication, with or without energy constraints, licensed or license-free communication, etc. In addition, as secondary controllers are close to the end-user or to sensor nodes this make them a good candidate to host the fog computing.

3.1.3. The local controller (LC)

The Local Controllers (LCs) nodes are specific nodes of access networks and their roles consist in relaying and managing the control messages. For instance, the control messages can be sent by nodes acting as router with or without mobility [23]. Using these nodes to send control packets and to communicate with the gateway (SC nodes), this contributes to reduce the overhead related to control and signaling messages. The selection of LC nodes (Fig. 1) is based on several parameters such as: connectivity degree, link quality with their neighboring nodes, and their position in the network topology. In the proposed architecture, we use Connected Dominating Set (CDS) algorithm to select and choose these nodes. The selection algorithm is detailed in the section below.

3.2. Data module

Unlike the control module, the data module is responsible for the management of data plane, according to the SDN paradigm without carrying about network control and management. However, the strong link and synergy between data and control (particularly Principal Controller) modules are necessary to efficiently manage services at the application layer. The main roles are data oriented such as: data aggregation, data collection, data analysis and fusion, decision making and notifications. All these roles can be ensured by the data server as illustrated in Fig. 1.

Data collection and aggregation

Many applications in IoT require the data collection and aggregation from nodes. In the proposed architecture, the data collection and aggregation strategy is based on the network and wireless communication constraints. For instance, it is important to consider the network lifetime by reducing the nodes' resources consumption. Other network characteristics need to be considered such as: bandwidth, overhead, latency, and fault-tolerance.

Data fusion and analysis

The data analysis are set of techniques and methods able to extract the relevant information from different source nodes. In the case of data fusion, the used techniques consist in combining data from different nodes to extract inferences, and to get more information than if they obtained from a single node (source). In the proposed architecture, the data analysis process selects the adapted technique to the application constraints in terms of Quality of Services (QoS) and energy consumption like delay sensitivity for real time application.

Decision-making process

This process is a key point in the proposed architecture. It is based on previous mechanisms (data aggregation/collection, and data fusion/analysis) and adapted tools to make a relevant decision. Among different and promising tools that can be used, we quote learning algorithms and models such as: Markov Decision Process (MDP), R-learning, and Q-learning [24].

In the proposed architecture, the controller module can directly or indirectly interact with this process to make relevant decisions like reprogram the network. The idea behind network reprogramming using SDN paradigm is to adapt the control processes to applications constraints.

3.3. Cloud and fog computing module

This module represents the integration of cloud and fog computing concepts in the proposed architecture. The cloud computing is a set of connected servers located at core network able to offer different services from Software as a Services (SaaS) to Infrastructure as a Services (IaaS).

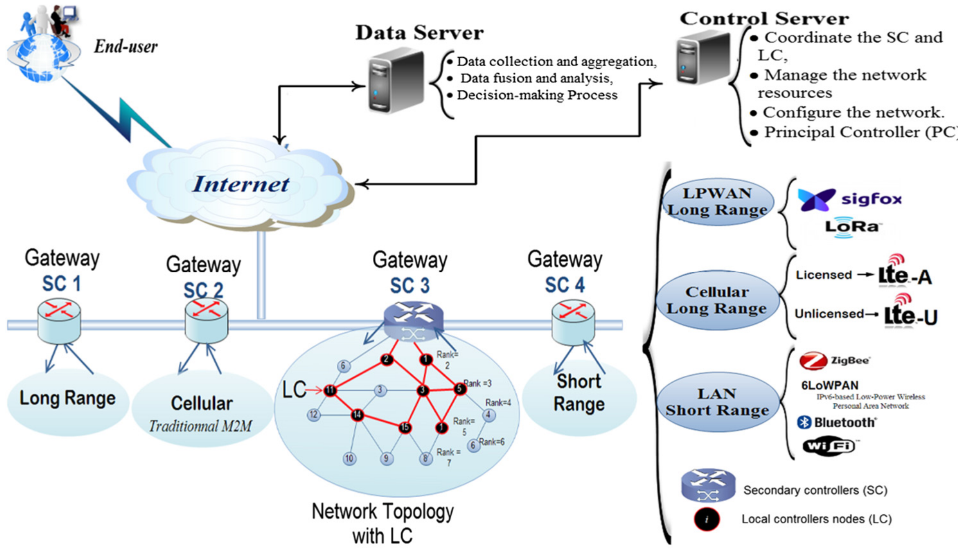


Fig. 1. The hybrid and proactive architecture based on SDN paradigm.

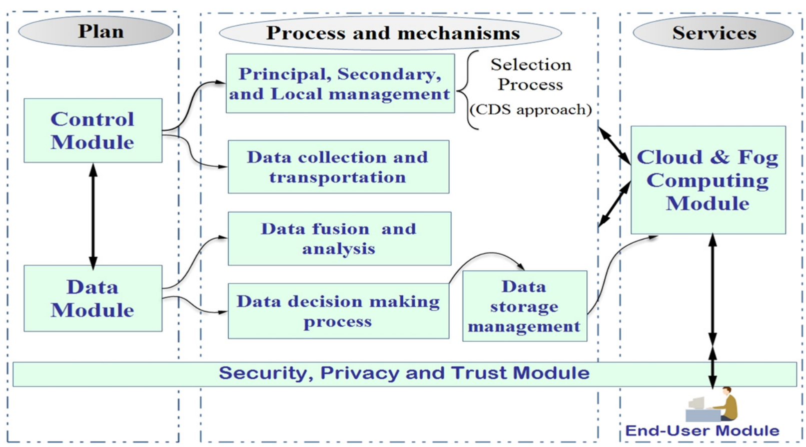


Fig. 2. The interaction between the main modules of the proposed architecture.

However, the fog computing is considered as an extension of cloud computing from the core network to the edge nodes. The fog computing is hosted at the edge nodes with secondary controllers (SC) in order to make services close to the end-users. As described above, the proposed architecture considers different characteristics: heterogeneity of network and communications technologies, non-centralized approach, flexible interoperability, and scalability. That is why, the fog computing is a good candidate to make services more reliable and efficient.

3.4. Security and privacy module

This module is responsible to dynamically manage security and privacy services. Unlike the existing architectures, the proposed architecture considers dynamic security services (Confidentiality, Integrity, Authentication) with different security levels able to automatically reprogram the network security policy. That is why the interaction with controller module is necessary to make the attacks detection more efficient and the reaction to potential attacks appropriate. For instance, firewalls at the edge nodes (SC controllers) can be dynamically tuned by adding or removing rules thanks to SDN paradigm. In addition, the trust model is required and it must be dynamic to track nodes (devices) behaviors [25,26]. For instance, introduce the Blockchain concept in the case of distributed trust management [27].

3.5. The end-user module

IoT applications have several and heterogeneous end-user profiles and requirements which must be considered in the design of any software architecture. For instance, the end-user can be a person, machine (hardware), or program (software) with different requirements and abilities.

In the case of the proposed architecture based on SDN paradigm, we not only distinguish between different end-user profiles, but also we consider the evolution of their requirements, and to make end-user active. This module introduces different methods, tools that empower users and to allow them to understand, configure, personalize, and control applications.

4. The local controllers selection using CDS algorithms

In this section, we focus on the part of the network behind gateways (Secondary controllers) and particularly the local controllers (LC) selection strategy. As described above, LC nodes have an important role to reduce the network overhead related to control messages and signaling packets.

We describe the used Connected Dominating Sets (CDS) algorithm to select subset of nodes to act as LC. In addition, we present the score computation model based on fuzzy set approach to make the strategy selection adaptable to different scenarios.

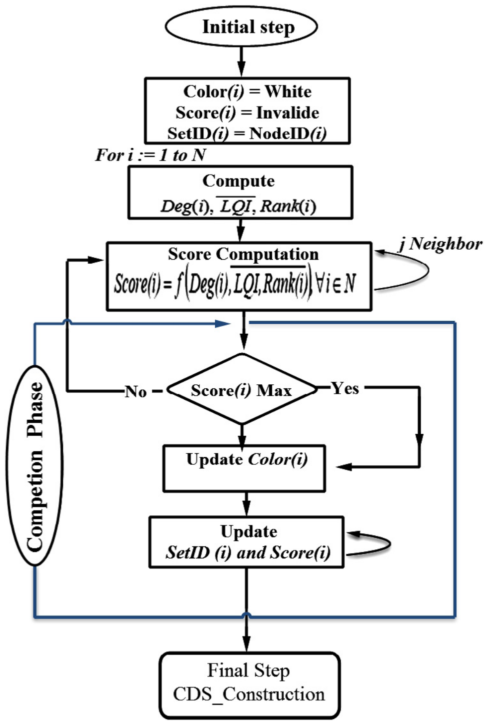


Fig. 3. The flowchart of the DLC-CDS process execution.

4.1. The CDS construction algorithm

We use Distributed Single-Phase (DSP) approach for CDS algorithm. This algorithm gives better performance in terms of delay, and CDS size [9]. The algorithm starts by giving white color to all nodes, and then according to the used strategy, some nodes change their color to black. Only the black nodes create CDS and act as local controller (LC).

The selection strategy for our proposed solution as called Distributed Local Controller Connected Dominating Set (DLC-CDS), is based on several parameters such as: the connectivity degree, the average neighboring link quality, and the rank with distance (in terms of hop number) from Secondary Controller (SC). These parameters are gathered into one parameter named "score" that is discussed in the next subsection. This algorithm suppose that each node has a unique ID (NodId) and a sub-set of connected nodes during the CDS construction has a unique ID (SetID). Three colors are used to define nodes state: white (for non-dominating nodes), gray (intermediate step), and black (for dominating node).

We distinguish two main steps: (i) the initial step consists in identifying and initialization of nodes state. In addition, this step starts the score computation process in order to give different weight and importance to nodes according to several parameters (see the next subsection). (ii) the competition step presents the decision process where nodes are colored white or black.

The flowchart in Fig. 3 represents the process execution of our proposed DLC-CDS algorithm, This diagrammatic representation illustrates a solution model of the selection strategy of the LCs nodes in the control module.

Initialization step

This step considers that all nodes are white and their score value is invalid (not available). Moreover, at this step the number of sub-set network is the same number of nodes in the network. Therefore, the identity of each sub-set (SetID) is the same on each node (NodID).

Algorithm 1: Initial step: All nodes are white

```
Input  : Score,i, N
Output: NodID, SetID, Color

N is a set of nodes
foreach i ∈ N do
    i.node ← NodID
    Net ← SetID
    if (i.Score ← InvalidScore) then
        i.Color ← white
        SetID ← NodID
    end
end
```

Network parameters assessment step

In this step, each node computes and assesses at least three important network parameters: (i) the connectivity degree (*Deg*) that presents the number of direct neighboring nodes, the average link quality index ($\overline{LQI}$) with these nodes, and the *Rank* that represents the distance (in terms of hop number) from the gateway (*Rank*). According to these network parameters the "Score" is computed using the proposed fuzzy model (detailed in the next subsection).

Competition and decision making step

In this step, each node computes its score according to the network parameters evaluated in the previous step. In order to make decisions about the node status, we distinguish two main algorithms: Algorithm 2 and Algorithm 3 running at sender and receiver nodes respectively.

In the case of the sender node (Algorithm 2), after it computes its score and compare it with non-black neighboring nodes then it makes decision about its color status. If the color becomes black then it broadcasts the message to its neighboring nodes with updated parameters: NodID, SetID, Color, and Score.

In the case of black or gray receiver node, if it receives packet from black nodes with greater *SetID* then receiver updates its *SetID* (belongs to the same sub-set with the receiver as master). However, in the case of white receiver node, it updates its *SetID* and it changes its color to gray as described in the Algorithm 3.

Algorithm 2: Competition step at sender node *i*

```
Input  : Score,NodID, SetID, i, neighbors(i)
Output: State decision of change Color nodes

foreach node i ∈ N do
    Score(i) ← ComputeScore()
    if (Score(i) ≠ 0) and (i.Color ≠ black) then
        foreach k ∈ non-black-neighbor(i) do
            S ← MaxScore(k)
            if (Score(i) > S) then
                i.Color ← black
                SetID = NodID
                Broadcast(SetID, NodID, Score)
            end
        end
    end
end
```

4.2. The score computation model

This subsection presents the proposed score computation model based on fuzzy set approach. As described above, the score

Algorithm 3: Competition step at receiver node *j*

Input : SetID, NodID, Color, j
Output: A connected dominated set with black node

foreach *Node j receives packet from black node i* **do**
  if $((j.Color = black) OR (j.Color = gray)) and (i.SetID > j.SetID)$ **then**
    j.SetID = i.SetID
  else
    if $(j.Color = white)$ **then**
      j.SetID = i.SetID
      $j.Color \leftarrow gray$
    end
  end
end

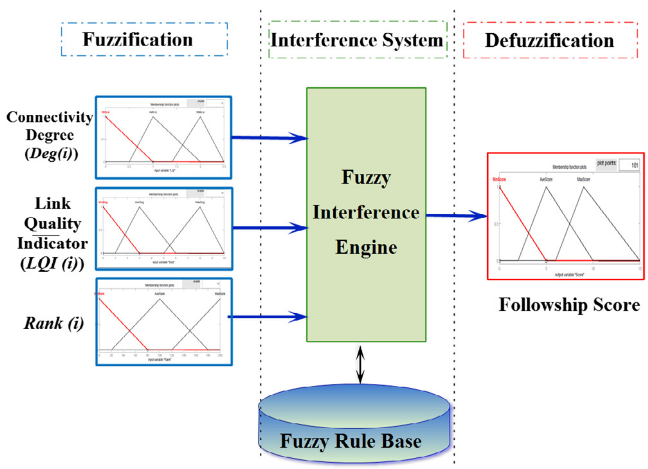


Fig. 4. The fuzzy logic approach for computation score: 3 inputs, 1 output, and 27 rules.

is an important parameter to create the CDS and to select the LC nodes. This depends on three key parameters: (i) $Deg(i)$, (ii) $\overline{LQI(i)}$, and (iii) the $Rank(i)$. However, the appropriate function (f) ability to combine these parameters is needed.

$$Score(i) = f(Deg(i), \overline{LQI(i)}, Rank(i)); \forall i \in N \quad (1)$$

We introduce the fuzzy logic controller to combine these parameters. Two major steps are needed to develop the fuzzy logic controller: (i) the one step defines membership functions for each input/output parameters, (ii) design the fuzzy rules.

We design a new system illustrated in Fig. 4 that has three input membership functions $Deg(i)$, $\overline{LQI(i)}$ and $Rank(i)$, and the one output membership function of $Score(i)$. The input and output membership functions take three linguistic values: *Minimum*, *Average* and *Maximum* as presented in Table 1.

In order to get maximum score of nodes, we consider the network topology with the *maximum* connectivity degree, the link performance with the *maximum* Link Quality Index ($\overline{LQI}$), and the distance from the gateway with the *minimum* rank. These membership functions (see below), define the fuzzy sets to input and output in the general inference rules (27 rules). These function is considered as weighting factors to determine their influence on the output sets.

4.2.1. The membership functions of fuzzification

In our proposed solution, the fuzzy logic controller uses three input parameters: $Deg(i)$, ($\overline{LQI}$) and $Rank(i)$ as described and

Table 1
Fl input and output membership values.

Input and output membership	Linguistic variables
Input 1: Connectivity degree	Maximum (*MaxDeg*) Average (*AvgDegr*) Minimum (*MinDegr*)
Input 2: Link Quality Indicator	Maximum (*MaxLQI*) Average (*AvgLQI*) Minimum (*MinLQI*)
Input 3: The Rank	Maximum (*MaxRank*) Average (*AvgRank*) Minimum (*MinRank*)
Output: The Score	Maximum (*MaxScore*) Average (*AvgScore*) Minimum (*MinScore*)

computed below. We used the trapeze trapmf membership function for the input parameters, which are defined as follow:

- The $Deg(i)$: It is the input 1 membership of FL, which is the number of neighbor nodes. It must be maximized to satisfy the choice of the LC node in the network topology.
- ($\overline{LQI(i)}$): The membership function of the link quality index, is a measurement of the quality of the received frame as defined by IEEE 802.15.4 standard and the relative calculate of the $\overline{LQI}$ value is the RSSI (received signal strength indicator) value.
- ($Rank(i)$): The membership function for Rank parameter, it is computed by the rank of the parent obtained as part of the neighbor node information.

4.2.2. The inference system

The fuzzy inference engine evaluates the control rules stored in the fuzzy rule base. We defined twenty seven rules using the centroid method [28] to process for the output membership function of $score(i)$. The principle of fuzzy rule is to express the knowledge with the conditional statements *If-Then* (*−Else*), for instance: *If* ($Deg(i)$) is maximum *AND* ($\overline{LQI(i)}$ is *maximum AND* ($Rank(i)$ is *minimum*) *Then* score of node ($Score(i)$ is *maximum*.

4.2.3. The output followship score-defuzzification

To get a finite number of the $Score(i)$ function, we need to go through the defuzzification process, there are many ways to do this as detailed in [28]. The most common is the Gaussian method, which is used in our case of score evaluation. In the final step of FL, the membership output with defuzzification results, we will have a maximum value of the $Score(i)$, which is used in the CDS construction algorithm (Section 4).

Remember that the maximum score indicates the better choice of the dominator node (black node).

5. Performance evaluation

In this section, we conduct numerical studies in two parts of the simulation for the LCs selection strategy with DLC-CDS algorithm. The first one, evaluates the score function with the fuzzy logic approach and the second one evaluates the DLC-CDS algorithm under different parameters.

5.1. The first part of the score evaluation

In this first part of simulation, we present the evaluation of the score computation with the fuzzy logic approach. In order to evaluate the impact of the input parameters on the score function, they are presented in different scenarios as depicted in Table 2 and described in the subsection below.

Table 2
Fuzzy scenarios for network score computation.

RulesInput/Output	Degree	LQI	Rank	Score
Scenario 1	Average	Minimum	Minimum	Minimum
Senario 2	Maximum	Minimum	Average	Average
Senario 3	Maximum	Maximum	Minimum	Maximum

5.1.1. Simulation setup of the fuzzy logic approach

To evaluate the score function as presented with Eq. (1), we have used the MATLAB simulator with the simulation parameters as follows: The nodes are randomly deployed in a network containing 49 nodes and one gateway node. All these nodes are gathered in a 100×100 network. The radio range $R = 10$ is used in all simulations for each node with a node density $p = 0.04$ for the same configuration of network topology.

In this simulation, we compute firstly the three parameters of $Deg(i)$, $\overline{LQI(i)}$ and $Rank(i)$. Which are exploited in the second step as an input parameters in the fuzzy system presented in the following subsection. We carried the obtained parameters of $Deg(i)$, $\overline{LQI(i)}$ and the $Rank(i)$, as a three input membership function to evaluate the $Score(i)$ function.

Both the input and the output variables take three linguistic values: maximum, Average, minimum. We used three scenarios that are summarized in Table 2, as an evaluation reference among the twenty seven rules implemented in the FL controller. We varied the $Deg(i)$ and $\overline{LQI(i)}$ parameters for each linguistic value (Maximum, Average) to respect in order the $Rank(i)$ for each scenario. The main objective of these scenarios is to evaluate and obtain a finite value to the $Score(i)$.

5.1.2. The score function performance

The obtained simulation results are plotted in Fig. 5 which shows the score function, behavior with different parameters such as:$Deg(i)$, $\overline{LQI(i)}$ and $Rank(i)$. We remark that the maximum score value is reached when both parameters: connectivity degree and link quality index are varied towards the maximum values, and with the fixed minimum rank, otherwise the minimum and the average score level depends on the minimum of Link quality, average connectivity degree and with the minimum rank. Despite that the rank is with the minimum value, it should be noted that there is an impact on the connectivity degree on the link quality index.

In this simulation results the score increase to take the maximum values in the set of {8,9,10} and the $MaxScore(i) = 10$. For the maximum value of the score, it is very interesting to choose an LC nodes with more neighbors nodes (Max$Deg(i)$) for the control process, with the best link of quality and near to the gateway node (SC). Then strongly, the results of this scenario illustrates the good results in terms of the local controller selection with the maximum score values obtained in the set range of 8, 9, 10.

In the final step we have introduced the maximum score value in the DLC-CDS algorithm indicated in the Section 4 and to make an extensive network simulation for the LC node creation, which is presented in the following subsection.

5.2. The second part of DLC-CDS evaluation

In this part, we have run as many simulations as required to evaluate the performance of DLC-CDS algorithm under different parameters. A comparison of the proposed solution of DLC-CDS with the DSP-CDS algorithm is also performed in this subsection using the Matlab simulator.

5.2.1. The simulation setup of DLC-CDS

The simulation models a network with N nodes randomly deployed in a square area with length L varied from 40 m to 120 m

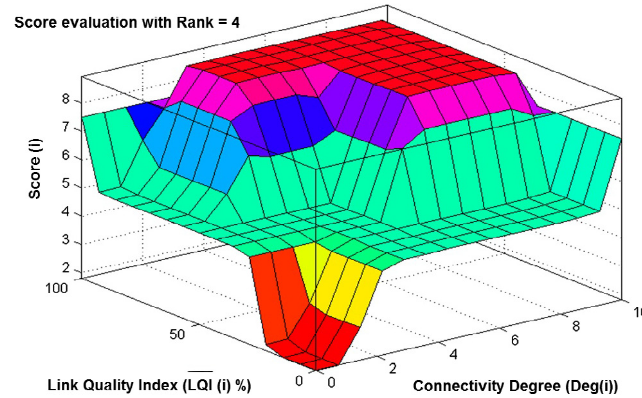


Fig. 5. The fuzzy simulation for score computation.

and the radio range R equally assumed to be $R = 10$ for each node and varied in the different simulation scenarios. For the number of nodes $N = L \times L \times \rho$, which ρ is the node density as defined the average number of nodes in the unit area. The detailed scenarios and simulation parameters are summarized in Table 3.

We considered four different scenarios (Table 3) with setting parameters as described below to evaluate the DLC-CDS algorithm. The comparison between the DSP-CDS algorithm and our proposed solution of DLC-CDS is introduced in this subsection.

Scenario 1

In this scenario, we carried out a comparison between the DLC-CDS and DSP-CDS methods to evaluate the CDS-Size performance metric. For this simulation, we considered the network with the variation of nodes from 48 to 576 and uses the fixed radio range $R = 10$. We varied the node density (p) with 0.03 and 0.04 values.

Scenario 2

In this scenario, we used the same network configuration as introduced in the scenario 1 in different to fix the node density $p = 0.04$ and to vary the radio range with $R = 10$ and $R = 18$. Another performance evaluation, a comparison of CDS-Size metric is studied in this scenario between the DLC-CDS and DSP-CDS methods.

Scenario 3

In this scenario, we evaluate our proposed solution of the DLC-CDS method to vary the node density p from 0.02 to 0.09 and with fixed the radio range $R = 10$.

Scenario 4

In this case of scenario simulation, the network configuration is fixed with $N = 864$ nodes and varied the radio range from 10 to 18. In this scenario, we evaluate the performance of DLC-CDS in terms of CDS-Size performance metric under the node density p variation from 0.02 to 0.06.

5.2.2. Simulation parameters of DLC-CDS

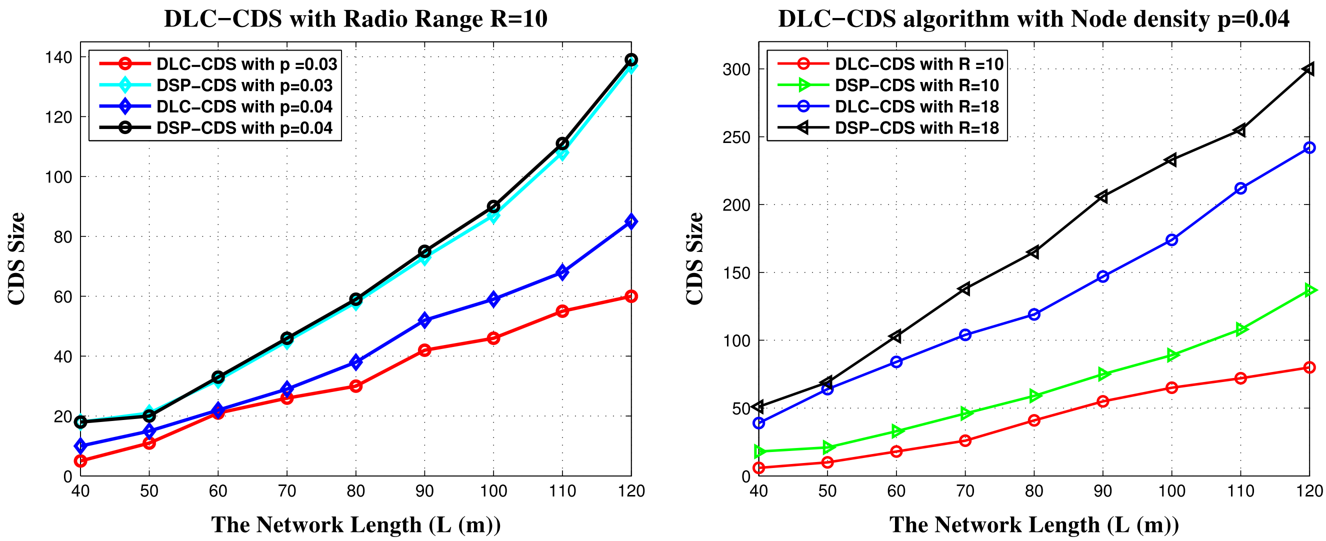
We have used two important parameters of the DLC-CDS simulation, the node density (ρ) and the Radio Range (R) as described below.

Node density (ρ)

This is an important parameter influencing the network connectivity, it determines the number of neighboring nodes within the node transmission range. In order to increase the node density in the network, we target the network density. Each node in the competition can decide to become a dominator (LC), if it has a

Table 3
Simulation parameters of DLC-CDS.

Network scenario	Number of node (N)	Node density (ρ)	Network length (L)	Radio range (R)
Scenario 1	48 to 576	0.03, 0.04	40 to 120	10
Senario 2	64 to 576	0.04	40 to 120	10, 18
Senario 3	32 to 1296	0.02 to 0.09	40 to 120	10
Senario 4	864	0.02 to 0.06	120	10 to 20



(a) Scenario 1: The impact of the nodes density p.

(b) Scenario 2: The impact of the radio range (R).

Fig. 6. The CDS size versus network length L (m).

maximum number of neighbors with the higher node density in the network area.

On the other hand, the node with the maximum score value has a greater chance to become a dominator to cover more neighboring nodes and share the same SetID. Each new dominator will try to cover a new area of the network with a fixed radio range R. For that reason, the CDS size should be mainly determined by the network size and depends on the node density.

Radio range (R)

In our proposed algorithm (DLC-CDS), we introduce the radio range as the second important parameter, it has an effect on the node density. Indeed, if the radio range for each node increases, the node can have more neighbors that may increase their degree of connectivity.

In our simulation, we change the radio range to evaluate the performance of the DLC-CDS algorithm. In this paper, we propose to change the radio range in simulation scenarios 2 and 4, to target the degree of connectivity of the node. The relative node density is $\pi \times R^2 \times \rho$, when the radio range is fixed, for example if it is at 18 and if the node density $\rho = 0.01$, the relative node density equals 10.17.

In this manner, the radio range and the node density have one to one correspondence and impact on the size of the CDS in the network area. By increasing the radio range, we have refined DLC-CDS performance evaluation compared to the DSP-CDS algorithm in terms of CDS-Size.

5.2.3. The performance metrics

In this subsection, we present the performance evaluation of the proposed DLC-CDS method with the performance metric. The comparison scenarios with the existing methods [9] are performed according to the CDS-size performance metric under different parameters as presented in Table 3.

All the simulations we carried out in this study (implementation and performance evaluation) show the effect of the radio range (R) and the node's density (p) parameters of the CDS construction on the network.

One metric of the CDS-size is used to evaluate the DLC-CDS performance in order to make a comparison with the DSP-CDS and the other distributed algorithms. The scenarios of simulation take place in several iterations (rounds), so at each iteration the node decides to change its color according to the value of the score and its neighboring nodes.

The CDS-size performance metric

Regarding all simulations, we used the evaluation metric of the CDS-Size for the performance evaluation of the DLC-CDS method. The CDS-Size is the number of the connected dominators nodes, which can be constructed the Connected Dominating Set (CDS) in the network.

5.2.4. The simulation results

In this section, we evaluate our proposed algorithm (DLC-CDS) through different simulation results and to compare the performance in terms CDS-Size of both algorithms: DLC-CDS, and the second algorithm [9] (DSP-CDS). The Fig. 6(a) and (b) present by order the results of the scenario 1 and scenario 2, to compare the performance of the DLC-CDS and DSP-CDS algorithms in terms of CDS-Size. For Figs. 7 and 8 present the results of the scenario 3 and scenario 4, which show the impact of the node density p and radio range R on the performance of DLC-CDS in terms of CDS Size.

The impact of the node density (ρ)

Fig. 6(a) illustrates the comparison between the DLC-CDS and the DSP-CDS algorithms. It shows that the DLC-CDS outperforms the DSP-CDS with a difference of 64% in terms of CDS size. This difference between both algorithms is caused by the selection

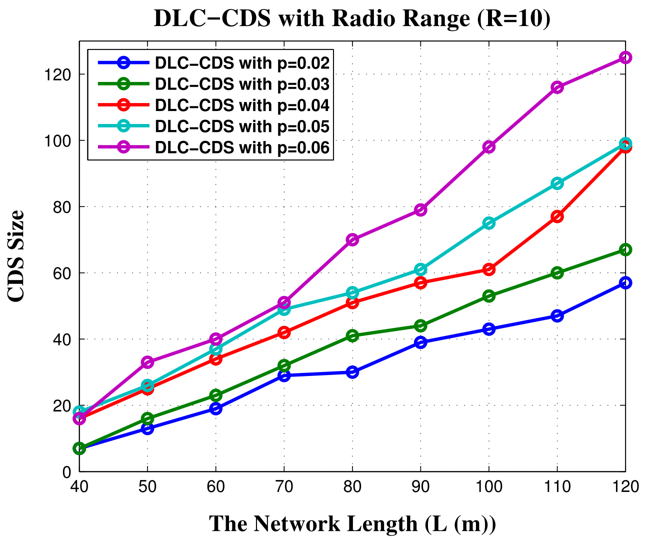


Fig. 7. Scenario 3: The impact of the nodes density *p* on the CDS-Size generated by DLC-CDS algorithm.

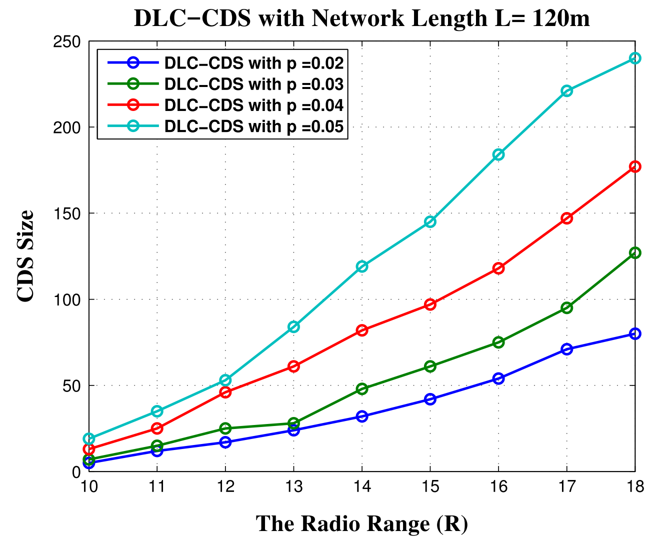


Fig. 8. Scenario 4: The impact of the nodes density on the CDS-Size generated by DLC-CDS algorithm.

strategy of the black nodes (Dominators) specified by each algorithm.

The selection strategy of DSP-CDS algorithm is based only on the connectivity degree parameter, which offers the chance to all nodes with a higher degree to become dominators (black nodes). For the DLC-CDS algorithm, an additional constraint is added fuzzy logic technique: the nodes with a mediocre *LQI* performance are eliminated. Thus, the *LQI* has an impact on the selection strategy of the dominator in the network area.

For the DLC-CDS algorithm, we remark that for the performance of DLC-CDS, there is a difference of 12% of CDS Size generated in the network with the variation of node's density (0.03 and 0.04). This is due to the impact of the node's density on the *LQI* constraint, when the node's density increases, the probability to get collision increase too and this can negatively impact the *LQI*.

The Fig. 7 really illustrates the impact of the node's density on the CDS Size, when the node's density increases the CDS Size increases. The result of this scenario presents the network's density when the neighboring nodes increase by the increasing of the node density.

The impact of the radio range (R)

In the radio range simulations, Figs. 6(b) and 8 show that the CDS-Size generated by the DLC-CDS are less than that generated by the DSP-CDS with the difference of 63%. When the radio range increases, there will be nodes with a higher connectivity degree (Deg), due to the neighborhood density. For this reason, the *LQI* presents the important parameter which decreases the CDS-Size with DLC-CDS algorithm, when some nodes with mediocre *LQI* are eliminated.

When the radio range is increased there are more neighbors, which implies that the *LQI* decreases and as a result to get a minimum CDS. The Fig. 6(b) confirms that, if the *LQI* decreases due to the increase of the radio range and if the communication resources can be shared with all the neighbor nodes, there are more collisions that will bring about a decrease in the performance link.

6. Conclusion

In this paper, we propose a new hybrid and programmable architecture based on Software Defined Network (SDN). The different modules and software component of this architecture are described and detailed. In order to make the proposed architecture adaptable to different network technologies, we introduce three levels of controls: principal, secondary, and local controllers. We focus on the local controllers (LCs) selection using two approaches: Connected Dominating Set (CDS) and fuzzy set. The distributed CDS algorithm with single phase (DLC-CDS) is used with an important function named score to select LCs nodes. The *Score* function depends on several parameters such as: the connectivity degree, the average link quality, and the rank which is the distance from the gateway (secondary controllers). In order to aggregate these parameters and to compute the *Score* function, we used a fuzzy logic approach. The performance evaluation of the *Score* function is proposed throughout network simulation. The DLC-CDS algorithm is implemented using the score function to construct the connected dominators acting as Local Controller (LC). The DLC-CDS outperforms the DSP-CDS by reducing the CDS size around 40% in the case of high nodes' density. In the case of different radio range, the simulation results show that the proposed algorithm improves by 36% of CDS size compared to DLC-CDS. The obtained results show the importance of the proposed model and its sensitivity to the different network. As future works, we plan to develop other modules of the proposed architecture and to make extensive network simulation with another parameters of security and quality of service (QoS).

References

[1] I. Lee, K. Lee, The Internet of Things (IoT): Applications, investments, and challenges for enterprises, Bus. Horizons 58 (4) (2015) 431–440.

[2] J. Gubbi, R. Buyya, S. Marusic, M. Palaniswami, Internet of Things (IoT): A vision, architectural elements, and future directions, Future Gener. Comput. Syst. 29 (7) (2013) 1645–1660.

[3] M. Younas, I. Awan, A. Pescape, Internet of things and cloud services, Future Gener. Comput. Syst. 56 (C) (2016) 605–606.

[4] A.I. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, M. Ayyash, Internet of Things: A survey on enabling technologies, protocols, and applications, IEEE Commun. Surv. Tutorials 17 (4) (2015) 2347–2376.

[5] A. Rachedi, M.H. Rehmani, S. Cherkaoui, J.J.P.C. Rodrigues, The plethora of research in Internet of Things (IoT), IEEE Access 4 (2016) 9575–9579. http://dx.doi.org/10.1109/ACCESS.2016.2647499.

[6] T. Mekki, I. Jabri, A. Rachedi, M. ben Jemaaa, Vehicular cloud networks: challenges, architectures, and future directions, Veh. Commun. (2017). http://dx.doi.org/10.1016/j.vehcom.2016.11.009.

[7] M. Bouaziz, A. Rachedi, B. Abdelfettah, EC-MRPL: An energy-efficient and mobility support routing protocol for Internet of Mobile Things, in: The 14th

Annual IEEE Consumer Communications & Networking Conference, CCNC, Las Vegas, US, 2017.
[8] M.T. Lazarescu, Design of a WSN platform for long-term environmental monitoring for IoT applications, IEEE J. Emerg. Sel. Top. Circuits Syst. 3 (1) (2013) 45–54.
[9] B. Yin, H. Shi, Y. Shang, An efficient algorithm for constructing a connected dominating set in mobile ad hoc networks, J. Parallel Distrib. Comput. 71 (1) (2011) 27–39.
[10] M.T. Thai, F. Wang, D. Liu, S. Zhu, D.-Z. Du, Connected dominating sets in wireless networks with different transmission ranges, IEEE Trans. Mobile comput. 6 (7) (2007).
[11] B.T. de Oliveira, R.C.A. Alves, C.B. Margi, Software-defined wireless sensor networks and Internet of Things standardization synergism, in: IEEE Conference on Standards for Communications and Networking, CSCN 2015, Tokyo, Japan, October 28-30, 2015, pp. 60–65.
[12] B.T. de Oliveira, C.B. Margi, L.B. Gabriel, TinySDN: Enabling multiple controllers for software-defined wireless sensor networks, in: IEEE Latin-America Conference on Communications, LATINCOM 2014, Cartagena de Indias, Colombia, November 5-7, 2014, pp. 1–6.
[13] L. Galluccio, S. Milardo, G. Morabito, S. Palazzo, SDN-WISE: Design, prototyping and experimentation of a stateful SDN solution for wireless sensor networks, in: 2015 IEEE Conference on Computer Communications, INFOCOM 2015, Kowloon, Hong Kong, April 26 - May 1, 2015, pp. 513–521.
[14] A. Anadiotis, L. Galluccio, S. Milardo, G. Morabito, S. Palazzo, Towards a software-defined network operating system for the IoT, in: WF-IoT, 2015, pp. 579–584.
[15] A.-C.G. Anadiotis, G. Morabito, S. Palazzo, An SDN-assisted framework for optimal deployment of mapreduce functions in WSNs, IEEE Trans. Mob. Comput. 15 (9) (2016) 2165–2178.
[16] T. Luo, H.-P. Tan, T.Q. Quek, Sensor OpenFlow: Enabling software-defined wireless sensor networks, IEEE Commun. Lett. 16 (11) (2012) 1896–1899.
[17] X. Chang, M. Ding, D. Chen, An approximation algorithm for connected dominating set in ad hoc networks, in: International Workshop on Theoretical Aspects of Wireless Ad Hoc, Sensor, and Peer-to-Peer Networks, 2004, pp. 376–380.
[18] D. Bendouda, A. Rachedi, H. Haffaf, An hybrid and proactive architecture based on SDN for Internet of Things, in: The 13th International Wireless Communications and Mobile Computing Conference, Valance, Spain, 2017. https://hal-upec-upem.archives-ouvertes.fr/hal-01510460.
[19] A.A. Khan, M.H. Rehmani, A. Rachedi, When cognitive radio meets the internet of things? in: 2016 International Wireless Communications and Mobile Computing Conference, IWCMC, Paphos, Cyprus, September 5-9, 2016, pp. 469–474.
[20] S. Hamdoun, A. Rachedi, H. Tembine, Y. Ghamri-Doudane, Efficient transmission strategy selection algorithm for M2M communications: An evolutionary game approach, in: 15th IEEE International Symposium on Network Computing and Applications, NCA 2016, Cambridge, Boston, MA, USA, October 31 - November 2, 2016, pp. 286–293.
[21] A.A. Khan, M.H. Rehmani, A. Rachedi, Cognitive-radio-based Internet of Things: applications, architectures, Spectrum related functionalities, and future research dIrections, IEEE Wirel. Commun. (2017) 3.
[22] S. Tomovic, K. Yoshigoe, I. Maljevic, I. Radusinovic, Software-Defined fog network architecture for IoT, Wirel. Pers. Commun. 92 (1) (2017) 181–196.
[23] M. Bouaziz, A. Rachedi, A survey on mobility management protocols in wireless sensor networks based on 6LoWPAN technology, Comput. Commun. 74 (2016) 3–15.
[24] A. Schwartz, A reinforcement learning method for maximizing undiscounted rewards, in: The 10th international conference on machine learning, Amherst, MA, 27–29 June, 1993, pp. 298–305.
[25] T. Gazdar, A. Rachedi, A. Benslimane, A. Belghith, A distributed advanced analytical trust model for VANETs, in: Global Communications Conference, GLOBECOM, 2012 IEEE, IEEE, 2012, pp. 201–206.
[26] G. Glissa, A. Rachedi, A. Meddeb, A secure routing protocol based on RPL for Internet of Things, in: Global Communications Conference, GLOBECOM 2016, IEEE, IEEE, 2016, pp. 1–7.
[27] A. Lei, H. Cruickshank, Y. Cao, P. Asuquo, C.P.A. Ogah, Z. Sun, Blockchain-based dynamic key management for heterogeneous intelligent transportation systems, IEEE Internet Things J. (2017).
[28] M. Works, Fuzzy logic toolbox users guide, Jan 1998.

Djamila Bendouda is a teacher at the Mustapha Stambouli university of Mascara and a researcher at Oran University, Algeria. She is a Ph.D. candidate at final stage. She received a M.S. degree (Magister) in computer science from Mascara University in 2010. Since 2008 she has been doing research in wireless ad hoc and sensor networks. She has published four papers in refereed conference proceedings. Her current research area focuses on control, Softawar Defined Network (SDN), IoT, energy consumption, quality of service, and security considerations in wireless sensor and actuator networks (WSANs).

Abderrezak Rachedi (S'05, M'08, SM'15) received the Engineering degree in computer science from the University of Technology and Science Houari Boumedienne, Algiers, Algeria, in 2002, the M.S. degree in computer science from the University of Savoie, Chambéry, France, in 2003, the Ph.D. degree in computer science from the University of Avignon, Avignon, France, in 2008, and the H.D.R. degree in computer science from Paris-Est University, Champs-sur-Marne, France, in 2015. He has been a member of the Gaspard Monge Computer Science Laboratory since 2008. He is currently an Associate Professor (maitre de conferences) with the University Paris-Est Marne-la-Valleée, Champs-sur-Marne, France. His research interests include the field of wireless networking, wireless multihop networks, wireless sensor networks, vehicular ad hoc networks, machine-type communication, Internet of Things, distributed algorithms, quality of services with security, trust models design, network performance analysis, and evaluation.

He advised multiple Ph.D. and masters students with Paris-Est University. His research efforts have culminated in more than 90 refereed journal, conference, and book publications in a wide variety of prestigious international conferences and journals, including the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, Elsevier Ad Hoc Networks, the IEEE ICC, and the IEEE Globecom.

He serves on the Editorial Board of the IEEE ACCESS Journal, Security & Privacy (SPY) journal, Wireless Communications and Mobile Computing (John Wiley) journal, and the International Journal of Communication Systems (John Wiley). He has served as a Technical Program Committee Member and Reviewer of many international conferences and journals.

Hafid Haffaf obtained his Doctor degree in Computer Science in 2000. He is Professor at the University of Oran 1 (Algeria). He actually heads the R.I.I.R. Laboratory at Computer Science Department–Oran University. His researches concern different domains as industrial diagnosis, optimization and reconfiguration systems. Using hypergraph and matroid theory, system of systems approaches, we can model many applications in monitoring domain. He has many collaborations projects with European laboratory: LAGIS in Polytech lille where he worked in intelligent transport systems infrastructures projects (Intrade, Weastflows) and where he had been Visiting Professor several times; and LIAUPA in PAU (France) in the domain of WSNs.