future internet

MDPI

Article

Dynamic Control Architecture Based on Software Defined Networking for the Internet of Things

Michele Bonanni, Francesco Chiti *, Romano Fantacci and Laura Pierucci

Department of Information Engineering, University of Florence, 50121 Firenze, Italy;
michele.bonanni@unifi.it (M.B.); romano.fantacci@unifi.it (R.F.); laura.pierucci@unifi.it (L.P.)
* Correspondence: francesco.chiti@unifi.it; Tel.: +39-055-275-8588

Abstract: Software Defined Networking (SDN) provides a new perspective for the Internet of Things (IoT), since, with the separation of the control from the data planes, it is viable to optimise the traditional networks operation management. In particular, the SDN Controller has a global vision of the network of sensors/actuators domain, allowing real-time network nodes and data flows reconfiguration. As a consequence, devices, usually facing limited communications and computing resources, are relieved of the route selection task in a distributed and, thus, suboptimal way. This paper proposes a SDN-IoT architecture, specifically focusing on the Controller design, which dynamically optimises in real time the end-to-end flows delivery. In particular, the dynamic routing policy adaptation is based on the real-time estimation of the network status and it allows jointly minimising the end-to-end latency and energy consumption and, consequently, to improve the network life time. The performance of the proposed approach is analysed in terms of the average latency, energy consumption and overhead, pointing out a better behaviour in comparison with the existing distributed approaches.

Keywords: software defined networking; green internet of things; routing strategies; energy awareness; sensors and actuators domain

check for updates

Citation: Bonanni, M.; Chiti, F.; Fantacci, R.; Pierucci, L. Dynamic Control Architecture Based on Software Defined Networking for the Internet of Things. *Future Internet* **2021**, *13*, 113. https://doi.org/10.3390/fi13050113

Academic Editors: Jacopo Soldani and Stefano Forti

Received: 2 April 2021
Accepted: 27 April 2021
Published: 28 April 2021

1. Introduction

The Internet of Things (IoT) represents a unifying paradigm that aims at automating business processes, since smart things are connected at any place/time via a *network of networks* and empowered with data analytics features [1–3]. This concept could be actualised to several areas, including smart homes/buildings, smart cities, intelligent healthcare, smart traffic and smart environments to name a few [4]. To this end, many enterprises leverage on IoT devices to collect relevant and heterogeneous data and better manage their processes, however two major challenges need to be faced for the expansion of the IoT.

First, security issues become more complex in the IoT use case, because: (i) usually unattended and constrained devices are extremely vulnerable; (ii) access networks can be violated to enter the system; and (iii) an attack to the Cloud services can alter the end-to-end (e2e) application logic. The second main challenge is represented by the network connectivity, since it is required for each of the devices to be able to connect in an efficient way.

The above interrelated challenges can be answered by applying the software-defined networking (SDN) approach [5,6]. The core idea of SDN is the separation of data payloads from the controlling data that, in turns, determines the messages path. This is usually achieved by separating Data Plane (DP) and Control Plane (CP) in order to simplify traditional networks operation management.

Specifically, the SDN Controller(s) within CP generates the rules for data forwarding, while devices only forward traffic executing the aforementioned rules. The advantages achievable by the SDN paradigm rely on the fact that the Controller maintains a *global view* of the underlying network, so it can dynamically optimise e2e data flows according to the state of the network, integrating both Cloud and Fog computing domains [7,8].

The SDN principle could be perfectly paired with the IoT as an effective way to solve its actual limitations [9,10]. In particular, IoT devices could be secured when they are operated on a separate network. This is because it makes more sense to secure the devices at the network level, rather than at the individual device level. Specifically, the application of the SDN principle allows to dynamically partition a *physical* network into different *virtual* networks, as it has been previously achieved through Virtual Local Area Network (VLAN) protocols to manage broadcast domains in an efficient and flexible way. Moreover, SDN represents a more general model for separating at a more higher level, i.e., to translate service requirements into an actual network configuration [11]. In addition to this, many IoT applications are time-sensitive, and, consequently, the optimisation of connectivity with the storing/processing resources is a priority performed while achieving the goal of load balancing. This is effective especially when it is expected to have the computing nodes *near* where the data are being used, due to severe jitter or latency requirements [12,13]. In this perspective, the combined use of IoT with SDN is one of the main catalysts for pushing resources to the *edge* of the network and simplifying the interaction with a distributed and dynamic Application Plane (AP).

However, the main constraint of a typical IoT domain is concerned with energy limitation, as smart devices are requested to perform battery intensive sensing/communications tasks, likely to greatly increase, due to the wide and pervasive adoption of this paradigm [14]. It compels the adoption of efficient energy management policies, which is commonly addressed as *green* IoT, where this means a reduction of devices energy consumption of IoT in order to fulfil a smart world with sustainability [15,16].

To this purpose, the proposed approach effectively introduces the SDN concept into a typical IoT ecosystem in order to optimise the e2e flows delivery by means of a centralised and agile multifaceted forwarding strategy. In particular, in formulating our objective function, we trade off flows delivery latency and overall network survivability, as we focused on typical IoT constraints, mainly in terms of residual battery available at each device. As a consequence, the SDN Controller, upon continuously monitoring the underlying network status (mainly in terms of both available communications links and remaining battery level) and QoS requirements, is able to evaluate the best path by executing the Dynamic Routing Module (DRM) and to update the forwarding tables of the involved devices with a limited overhead, when compared with traditional distributed approaches, regardless the nodes deployment.

The rest of the paper is organised as follows. In Section 2, the most relevant approaches already proposed in the literature to address SDN design for IoT domain are briefly presented. The proposed architecture and forwarding strategies are characterised in Section 3, whereas its performance is validated in Section 4 by means of numerical results. Finally, in Section 5, conclusions are drawn.

2. Related Works

The IoT context provides the use of hundreds and thousands of interconnected nodes for sensing and monitoring services. Usually, these devices have limited resources to carry out proactive routing protocols which require to store routing table entries, or cannot afford the complexity and overhead of the network topology discovery, due to the elevate number of neighbours, as is required in reactive routing protocols. Recently, the advantages of SDN architecture have been applied even in IoT domain, allowing a dynamic separation of the CP from DP. SDN is envisioned to make dynamic, robust and scalable both the design and operation of IoT services. In particular, the utility and the challenges of SDN-based technologies when applied to IoT can be found in the targeted surveys [9,17–19]. According to a basic SDN architecture, a Controller allows real-time network reconfigurations of sensors and actuators to optimise data flows according to the available resources, without depending from specific underlying network features. In addition, distributed SDN CP architectures, involving multiple SDN Controllers, have been proposed for achieving a global view of wide area network. Hybrid SDN-based architectures are also considered with some SDN Controllers dedicated to nodes afar and a centralised Controller to detect failures on low level local SDN Controllers.

In any case, advanced efficient routing strategies need to be identified to support SDN-IoT efficient management, matching QoS requirements or reconfiguring devices through the selection of the best nodes and links, which is effectively performed by adopting the OpenFlow-enabled Wireless Sensor Networks (WSN) [20,21], a popular SouthBound SDN protocol.

In the following, we consider mainly papers which detail energy-aware routing proposals. The authors of [22] presented a software-defined clustered sensor networks (SD-CSN) arranged into clusters managed by one SDN Controller. About 1000 nodes divided in five clusters exchange messages and forward packets among themselves according to the policies of the SDN Controller. This solution provides a scalable architecture for IoT, since it can route data packets flows among the devices. The same model, i.e., a centralised control and hierarchical cluster structure, is proposed in [23]. The Lion Optimization (LO) algorithm creates both the clusters of sensor nodes and the routes to forward data. A hierarchical structure allows a global control of the WSN, making easier the deployment of network-wide management protocols and on demand applications.

Software defined energy aware routing (SD-EAR) proposed in [24] divides the sensor network into clusters with a SDN Controller associated to each cluster, where Controllers know the topologies of the associated cluster as well as of the border nodes. Each Controller evaluates the best path from the source to destination if the destination falls in its area or, alternatively, it can send a request to discover the route to destination, taking into account the residual energy of nodes. Moreover, SD-EAR adopts sleeping strategy to reduce energy consumption in the network.

The contribution in [25] analyses the application of the SDN architecture to improve security and energy efficiency in the WSN domain. In particular, the SDN Controller executes the Sensor Protocol for Information via Negotiation (SPIN) protocol to generate neighbours table and, then, to select the best next hop, according to its energy level and link quality. Then, the selected node is used by sensor node to forward the information.

Shafique et al. [26] considered a reconfigurable distributed protocol, called Application-aware Distributed adaptive Flow Iterative Reconfiguring (SADFIR). It uses multiple distributed SDN Controllers selected among IoT nodes as cluster heads (CH) and computes the residual resources of the IoT devices. Then, SADFIR distributes the selected CHs to the IoT infrastructure allowing distributed cluster formation of heterogeneous network.

Beside, a modified low-energy adaptive clustering hierarchy (LEACH) clustering protocol is proposed in [27]. In particular, the same concept of the LEACH scheme [28], i.e., sensor nodes organised into clusters, is adopted with the selection of a cluster head. The clustering mechanism is applied to distribute supply energy among sensors, according to the time division multiple access (TDMA) based MAC protocol, which is further modified by introducing a threshold value for cluster head selection with the simultaneous switching of different power levels among nodes.

Several papers in the literature consider bio-inspired clustering algorithms. Ant Colony Optimization (ACO) is presented in [29], where the cluster heads selection is implemented as it happens with the ant pheromone level. The Artificial Bee Colony algorithm is proposed in [30] to monitor the state of the sensor nodes in a typical SDN architecture through the Controller which selects the best path. The wide use of software-defined WSNs (SDWSNs) to reconfigure the networks, even after their displacement, led the authors of [31] to consider an efficient Particle Swarm Optimization (PSO) algorithm based on the residual energy of the nodes and the transmission distance. A green routing algorithm using fork and join adaptive PSO (FJAPSO) is addressed in [32] to operate in SDWSN. The considered scenario presents a classical topology, where sensor nodes are divided into clusters and each cluster has its own CH. However, to maximise the lifetime of sensor nodes and allow the adaptation to scalable networking scenarios, the FJAPSO acts on two levels by optimising jointly the number of control nodes and the best suited position of all the control nodes with respect to the Control Server (CS) and between CHs and sensor nodes.

The Whale Optimization Algorithm, which considers the residual energy, communication cost and nodes density, is proposed in [33] to enhance the network lifetime.

The aforementioned works typically consider a well-known hierarchical routing algorithms relying on the partition in clusters and the selections of the proper CHs. Moreover, CHs are highly energy consuming nodes, since they are requested to transmit to the Sink over a single hop, especially if they are located far from the Sink. Conversely, this paper proposes a routing approach which allows dynamically optimising in real time the end-to-end flows delivery through a real time-estimation of network status providing an effective load balancing of the network. It easily adapts to the conditions of flexibility and scalability required by a SDN-IoT network and helps to achieve better results, as shown in Section 4.

3. System Model

The proposed SD-WSN architecture, combining the SDN paradigm with the main traditional WSN elements, is shown in Figure 1, where it is possible to notice the classical three-layers arrangement (i.e., DP, CP and AP). In more details, DP is responsible to process incoming data packets executing the CP directives. Differently from *wired* SDNs, in which the traffic paths can be dynamically and easily managed by modifying the flow entries of the OpenFlow Switches and selecting the appropriate output port, in the wireless domain, this approach is usually not straightforward, because packets are always *broadcasted*. Therefore, an e2e wireless path needs to be controlled in order to avoid unintended loops, due to spurious *re*transmissions. Depending on its different capabilities in handling data flows, a WSN node can be logically classified in:

- *Root* (or *Sink*) node h represents the unique interface between the local and an external IP network in order to collect or disseminate data.
- *Non-Root* node could be a:
  - *Leaf* node, which cannot support downstream connections, since it is not connected to any *child* node; or
  - *Relay* (or *Intermediate Parent*) node, which is acting neither as a Root node or Leaf, and can transmit, receive and forward packets flowing in its upstream and downstream connections.

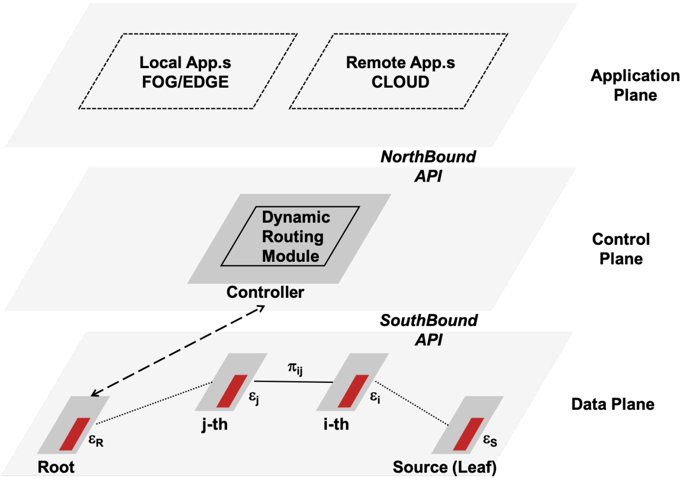

Figure 1. Reference SDN-IoT architecture, which shows the presence of Application, Control and Data Planes. AP is comprised of both local and remote applications, while CP is composed of a Controller with endowed the routing module (DRM). Finally, generic sensor nodes are within DP, where only Root is physically connected to the Controller.

The introduction of the SDN principle does not alter the node roles, but it shifts the routing task to the SDN Controller, which operates within the SDN CP and relies on a comprehensive and up-to-date view of the underlying WSN. Additionally, the SDN Controller can modify each node flows forwarding entries and dynamically change the data traffic path based on real-time requirements. The last feature is usually exploited by applications residing in the AP (though physically running in CP local devices or in remote Servers). It allows the interoperability of Cloud and Fog/Edge computing architecture in a dynamic way, depending on the evolution of the information flows which interest a large-scale distributed and heterogeneous systems, as depicted in Figure 1.

Differently from traditional WSNs, in which nodes resort to distributed routing protocols, the SD-WSN architecture can dynamically adopt centralised and specific forwarding strategies by simply allowing the SDN Controller to *annotate* the *network graph* for improving the overall system flexibility, reliability, and lifetime. In particular, the proposed approach is focused on dynamically adjusting the routing strategy based on the real-time network state for enhancing the network *responsiveness* and its *energetic footprint*. It is worth noticing that this policy being application-oriented is able to support time varying and heterogeneous QoS requirements usually originating when the Cloud and Fog/Edge architectures are effectively integrated.

To take into account the above-mentioned goals, we combined the typical one way latency requirements of time sensitive applications, together with the energy constraints of a green IoT domain, by focusing on the following optimisation problem:

$$\begin{aligned} \underset{w_{i,j}}{\text{minimise}} \quad & \frac{\lambda(s,d)}{\theta(\pi_{s,d})} \\ \text{subject to:} \quad & s,d \in \mathcal{G} \\ & i,j \in \pi_{s,d} \\ & \phi_{s,out} = \phi_{d,in} \end{aligned} \tag{1}$$

where $\mathcal{G}$ represents the overall network graph, s and d are a generic couple of source-destination nodes, $\pi_{s,d}$ is a possible path connecting s to d and i and j ate two generic adjacent nodes along $\pi_{s,d}$. Moreover, $\lambda(s,d)$ represents the *one-way* latency needed to deliver a messages *flow* from s to d nodes, while $\theta(\pi_{s,d})$ is the minimum time interval in which all the nodes along the path $\pi_{s,d}$ are jointly available. Finally, the last constraint is imposed to preserve the end-to-end flow from the source ($\phi_{s,out}$) toward the destination ($\phi_{d,in}$).

It is worth recalling that in principle the solution of the optimisation problem (1) can be easily obtained by applying the Djikstra algorithm, once a proper set of weights $\mathbf{w}$ is applied to every link. To this purpose, a generalised time varying weight $\mathbf{w}(t) \doteq \{w_{i,j}(t)\}$, associated to the link connecting the ith and jth nodes, is introduced as:

$$w_{i,j}(t) = w_\tau(t)\frac{\tau}{\tau_{\max}} + w_\epsilon(t)\frac{\epsilon}{\epsilon_{\max}} \tag{2}$$

where τ is the number of hops associated to the e2e path, ϵ is the residual battery level of the destination node, $\tau_{\max}$ is the maximum number of hops, $\epsilon_{\max}$ is the maximum battery level and $w_\tau(t)$ and $w_\epsilon(t)$ represent the weighting factors to each terms, such that $\forall t \ , \ w_\tau(t) + w_\epsilon(t) = 1$. In particular, $w_\tau(t)$ and $w_\epsilon(t)$ could be dynamically updated according to two different schemes:

- *Reactive* mode: An application can directly communicate with the SDN Controller and asynchronously compel the adoption of a shortest path, due to real time constraints. In this case, it is accomplished with $w_\tau(t) \gg w_\epsilon(t)$, whereas the condition $w_\tau(t) \ll w_\epsilon(t)$ is usually verified.
- *Proactive* mode: The SDN Controller autonomously modifies the weighting factors based on its own dataset of network related information.

It is worth noticing that, according to our unifying formulation, the hop-based mode and the energy-forwarding strategies represent two special cases that can be easily implemented setting $w_\tau(t) = 1$ and $w_\epsilon(t) = 0$ or $w_\tau(t) = 0$ and $w_\epsilon(t) = 1$, respectively.

Since battery consumption is mainly affected by packets transmission and reception, the SDN takes into account this drawback by adapting the *control* messages rate, typically reducing it when the residual battery is below a certain critical level. Specifically, this is gradually accomplished using two thresholds:

- *Soft*: It indicates that the node is no longer in a full energy state and, consequently, a moderate control message reduction can be applied.
- *Critical*: In this case, the node has a limited residual battery and a more stringent energy management is necessary to avoid its shut down.

When T_s is reached, the sending periods of `Beacon` and `Report` messages are exponentially increased. Further, when T_c is overcome, the above-mentioned periods are again increased according to an exponential rule. This approach allows the nodes to maintain an updated *upstream* path to the Sink node (and to the Controller) and publish their data to an external data center (e.g., a Cloud), without significant performance degradation. The network view achieved by the SDN Controller depends mainly on the sending period of the `Report` messages. A high sending rate reduction can cause an update delay of the Controller's view and remarkable differences from the actual WSN. This situation may involve a considerable Packet Loss Ratio (PLR) and a greater Round Trip Time (RTT), especially in a mobile scenario.

Additionally, the SDN Controller can also manage the node *Duty Cycle* (DC) by varying the duration of its *active* and *sleep* states, where the latter usually is increased as the residual battery decreases. This procedure is usually adopted when the critical threshold is approached.

To enable the DC procedure, the SDN Controller sends a specific message, called `Config` packet, which contains the sleeping period. Upon receiving this message, a node replies with an `ACK`. After the `ACK` reception, the Controller supposes that the specific node is going to sleep mode. If, after the sleeping period, the Controller does not receive a `Hello` packet from the node, it will consider the node disconnected from the network and it will remove him from the network view. In our implementation, the sleep interval of DC was set equal to the transmission period of `Data` messages; thus, nodes will be active only for the few seconds needed for data transmission.

There are several routing policies that an SDN Controller is able to adopt, depending upon the choice of the time varying weights $w_{i,j}(t)$ associated to each link, as expressed in the optimisation problem (1). An affordable solution could consider those weights as *constant* over time, resulting in a *static* strategy. If we specifically focus on the QoS constraints expressed in the optimisation problem (2), two opposite routing schemes are possible:

- MinHop: This algorithm minimises only the number of hops between source and destination nodes (i.e., $w_\tau(t) = 1\ , w_\epsilon(t) = 0\ ,\ \forall t$), thus being efficient especially in terms of latencies and overhead, but without taking into account the effect of energy consumption.
- MaxBattery: This approach merely monitors each node residual battery level and minimises its consumption in order to optimise the overall network lifetime, that is $w_\tau(t) = 0\ , w_\epsilon(t) = 1\ ,\ \forall t$. It also includes power-saving additional techniques, such as the packet rate reduction and the duty cycling.

In general, when taking into account both the energy consumption and delivery latency, a hybrid strategy is to be adopted, but always following a *dynamic* approach. Specifically, we integrated this scheme on board of the SDN Controller within the so-called Dynamic Routing Module (DRM), which performs a real-time estimation of the network status, integrating the local information received by each node (i.e., neighbours list, link quality indicator and battery level), with a sampling frequency which depends on the e2e control signalling delay. Besides, the DRM is capable of updating the global weights and to apply the correct strategy, as again detailed along Algorithm 1. Initially, the SDN Controller modifies the traffic flows minimising the path length and updates the network status according to the control messages sent by the nodes. Specifically, it changes the status of the nodes following the T_s and T_c thresholds. If the sum of nodes in soft or critical status is greater than the remaining ones, the DRM deactivates the default strategy and starts to consider the battery levels in the traffic path evaluation. Finally, when the soft nodes are lower than the number of nodes in critical status, the algorithm mostly optimises the residual energy and accordingly performs traffic redirection. It could be noticed that the proposed approach is close to the MinHop one if the number of nodes in soft or critical state is lower than the remaining ones, while it is similar to the MaxBattery strategy whenever the number of nodes with a critical battery level is greater than the ones with soft energetic level. In conclusion, the proposed DRM aims at jointly increasing the network lifetime, achieving a more uniform battery consumption and keeping as low as possible the communication latencies between any couple of nodes.

Algorithm 1: Dynamic Routing Module.

```
Require: Sensor nodes and soft (Ts) and critical (Tc) thresholds
Ensure: Tradeoff between latency and energy consumption
Initialisation of the MinHop Strategy
Setting $w_\tau$ = 1 and $w_\epsilon$ = 0
Initialisation of a Network Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.
A Battery state vector B is set as an empty array. Each next element
$b_i$ in B is a list of two elements (address a, battery state s).
while TRUE do
    update $\mathcal{G}$ and $B$ with the information received by the $i$-th node (neighbours list, link quality
    indicator and its battery level $b_l$)
    if $b_l > Ts$ then
        $b_{is} = 0$
    else if $b_l \leq Ts$ and $b_l > Tc$ then
        $b_{is} = 1$
    else
        $b_{is} = 2$
    end if
    if $\sum_{b_i \in B}^{N} b_{is=1} + \sum_{b_i \in B}^{N} b_{is=2} < \sum_{b_i \in B}^{N} b_{is=0}$ then
        Continue with the MinHop Strategy
    else if $\sum_{b_i \in B}^{N} b_{is=1} + \sum_{b_i \in B}^{N} b_{is=2} \geq \sum_{b_i \in B}^{N} b_{is=0}$ and $\sum_{b_i \in B}^{N} b_{is=2} < \sum_{b_i \in B}^{N} b_{is=1}$ then
        Deactivate the MinHop Optimisation Strategy
        Update $w_\tau$ and $w_\epsilon$ values
        Update node's flow entry of active flows
    else if $\sum_{b_i \in B}^{N} b_{is=1} + \sum_{b_i \in B}^{N} b_{is=2} \geq \sum_{b_i \in B}^{N} b_{is=0}$ and $\sum_{b_i \in B}^{N} b_{is=2} \geq \sum_{b_i \in B}^{N} b_{is=1}$ then
        Setting $w_\tau$ = 0 and $w_\epsilon$ = 1
        Activate the MaxBattery Strategy
        Update node's flow entry of active flows
    end if
end while
```

4. Results Analysis

The performance of the proposed DRM strategy is evaluated by means of numerical simulation resorting to Cooja simulator and SDN-WISE framework [34], the latter being a specific SDN solution targeted to WSN domain. It is based on IEEE 802.15.4 Physical and MAC layers, while the network elements can be distinguished into Sinks and Nodes, where only the former can directly communicate to the SDN Controller.

It is worth noting that the SDN Controller usually represents a single point of failure, in the sense that its temporary (*outage*) or permanent (*failure*) damage could compromise the correct forwarding of data flows. However, to partially mitigate this problem, we adopted an SDN-WISE based control messaging in background, comprised of `Beacon`, `Report`, `Request` and `Response` packets [35]. This allows the WSN to survive to the loss of its Controller, as far as the *upstream* flows are concerned. Particularly, network's nodes periodically send, with the characteristics summarised in Table 1, `Report` messages to the

SDN Controller, which contain information about the battery level of the source node, the distance from the Sink node, the neighbour list and some other important features (battery level and RSSI) of the neighbour nodes. These messages are collected by the Controller, which accordingly updates its network view and modifies the traffic flows between nodes. To change the network paths, the Controller sends specific flow rules to the nodes. This is performed using specific messages called `Openpath` packets as specified in SDN-WISE framework. Nodes will use these messages to update their flow entries, which are mainly composed of the following fields: (i) Matching rules; (ii) Action; and (iii) Statistics.

The nodes are spatially distributed according to a *bottleneck* topology which represents the worst case scenario for network lifetime. Specifically, only few nodes fall inside the Sink coverage area, thus they are supposed to forward to it the messages originated by other devices, which, instead, resort to multi-hop communications. As a consequence, the nodes belonging to the *first* tier are prone to quickly depleting their batteries. The simulation playground is 200 m^2 and the transmission range of sensor nodes is assumed to be 100 m. An example of bottleneck topology with 50 nodes in the playground is shown in Figure 2. All nodes are characterised by an identifier (ID) and a colour. The green node represents the Sink while purple and yellow elements identify all the other devices. Specifically, purple nodes are the only ones inside the Sink coverage area and subject to a higher packet retransmission. In particular, we refer to the MICAz mote model in terms of CPU states, transmit and receive power levels, as reported in Table 2.

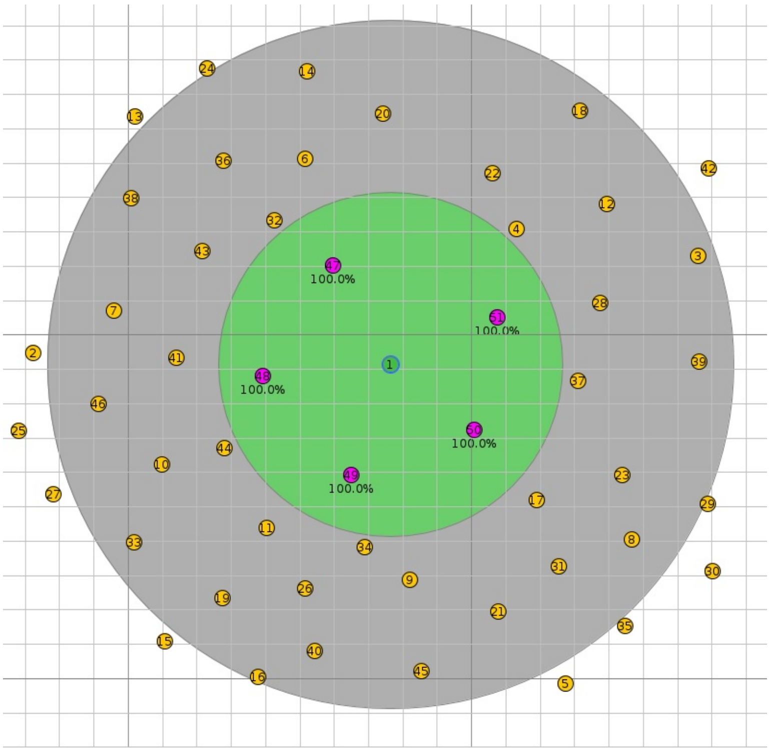

Figure 2. Bottleneck topology with 50 randomly deployed nodes.

In addition to this, we assumed that the `Beacon` and `Report` messages rates are increased respectively to 20 and 40 s, as soon as the T_s threshold is matched, while, when T_c is overcome, the above-mentioned rates are, respectively, updated to 30 and 60 s. Moreover, after an exhaustive measurement campaign, the obtained results have shown that, in static/quasi-static scenarios, the maximum applicable sending period of `Report` messages, which does not entail a considerable degradation of PLR and RTT, is about 60 s.

We start our evaluation by focusing on the average residual energy that the proposed routing policies running on SDN Controller can achieve. In deriving our results, we assumed that all nodes present the same initial energy value except for the Sink node, which is not battery powered and, therefore, is not considered in the performance evaluation. In Figure 3, the residual energy is presented with respect to the simulation time ranging from 100 to 900 s for the MaxBattery, DRM and MinHop strategies. The better behaviour of the MaxBattery and DRM approaches is shown; in particular, the battery saving achieved by these two alternatives positively affects the network life time. In Figure 3, it is also evident that the performance gap between DRM and MaxBattery keeps constant, while the other strategy gets worst with time. For instance, after 800 s, the gain between DRM and the MinHop strategy reaches a value nearly equal to 55%.

Table 1. Packet features.

Packet	Type	Sending Period (s)	Lenght (Bytes)	Description
Beacon	Control	10	2	It is sent by every nodes to inform its neighbours about the distance to the Sink node and its battery level.
Report	Control	20	[3, 108]	It is sent to the Controller to report information about battery level, distance to the Sink node and the neighbours list associated with their link quality indicator.
Data	Data	50	[1, 116]	It can be sent by every device including the Controller. Its size cannot exceed 116 Bytes.

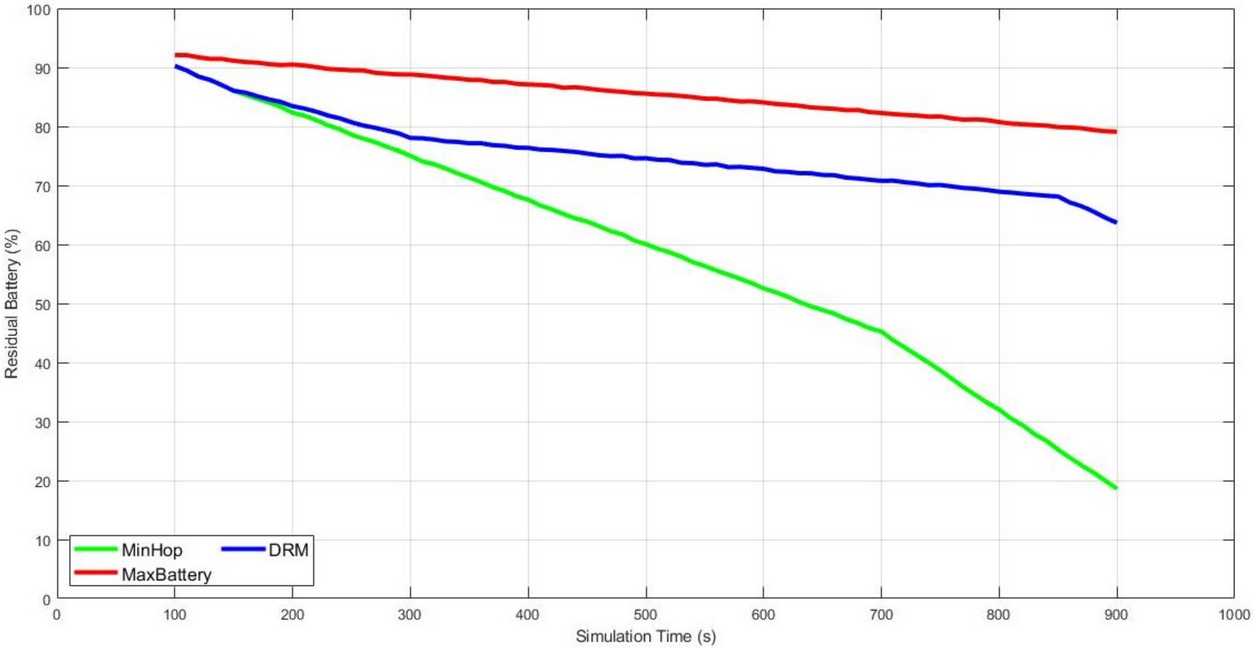

Figure 3. Average Residual Energy achieved by the proposed strategies MaxBattery, DRM and MinHop.

In addition, it is important to evaluate the fairness of the proposed DRM strategy in uniforming the energy depletion of the nodes, since this property affects the network partitions. To this regard, the so-called Jain's Fairness Index is considered in our validation campaign, defined as [36]:

$$J(\epsilon) \doteq \frac{\left(\sum_{i=1}^{N} \epsilon_i\right)^2}{N \sum_{i=1}^{N} \epsilon_i^2} \tag{3}$$

where N is the number of nodes and ϵ_i denotes again the battery level of node i. The Jain's Fairness Index approaches the unitary value when the residual battery level values of the nodes are extremely close. In Figure 4, this parameter is depicted as a function of time to compare the proposed different routing strategies in the case of bottleneck topology. As shown in Figure 4, the energy-based strategies, i.e., MaxBattery and DRM, both achieve Jain's Fairness Index values closest to the unit value; consequently, they are able to enhance the network lifetime by minimising the variance of battery consumption among nodes. Besides, the MaxBattery strategy shows an asymptotically better behaviour followed by the DRM strategy, while MinHop strategy presents a remarkable performance worsening, since it does not consider any energy consumption related information.

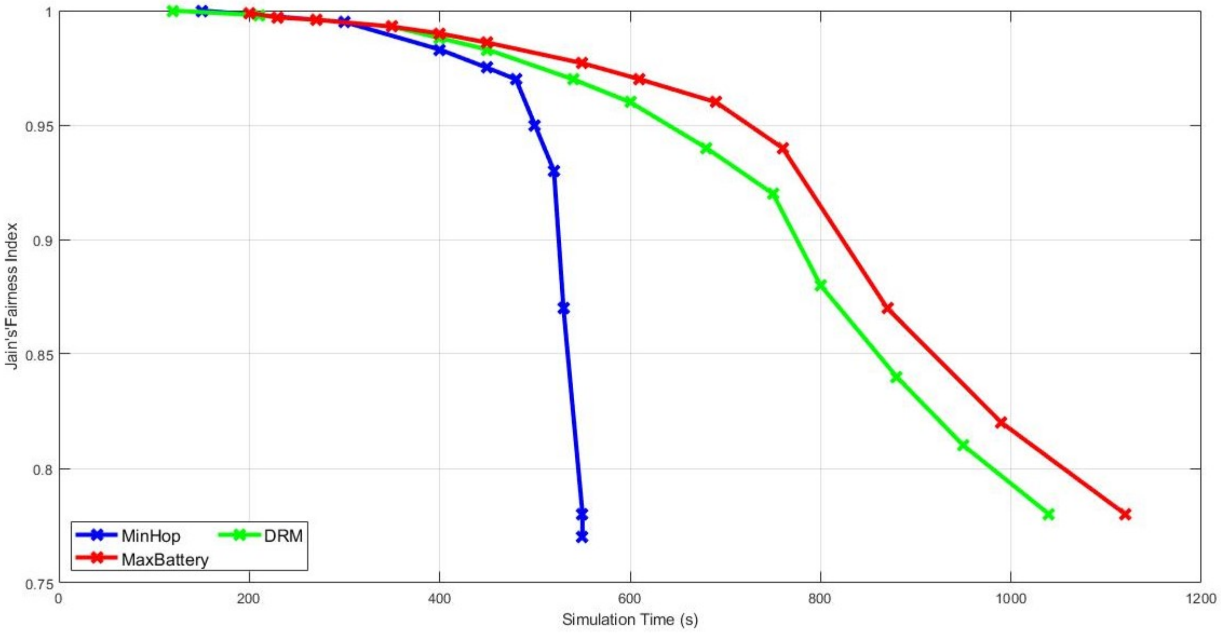

Figure 4. Jain's Fairness Index for the proposed strategies: MaxBattery, DRM and MinHop.

Another relevant issue to be addressed is the SDN Controller WSN nodes management. Specifically, the overall latency needed to monitor, update, evaluate and notify the selection of the best path represents a key parameter to be evaluated. Moreover, the process efficiency results in a better energy utilisation and, consequently, in a greater network lifetime.

Table 2. MICAz mote current consumption.

Microcontroller		Transceiver	
Active	1.8 mA	RX	19.7 mA
Idle	20 uA	TX (0 dBm)	17.4 mA
Sleep	5 uA	TX (−5 dBm)	13.9 mA
		TX (−10 dBm)	11.2 mA
		TX (−15 dBm)	9.9 mA
		TX (−25 dBm)	8.5 mA

To this end, in Figure 5, the average latency performance of the proposed strategies is shown, together with the related range of measured values. In more details, the average latency was evaluated considering N unicast communications between the Sink node and the farthest node from it. It is easy to point out that the MaxBattery approach presents a remarkably higher latency with respect to the other ones, since it only takes into account the battery nodes level, while DRM dynamically adapts its strategy by considering more the multi hop distance (in terms of w_τ) as long as the battery level of the majority of nodes is not critical.

In Figure 6, the overhead introduced by the various strategies is shown. Specifically, the results were obtained using the three different packets shown in Table 1.

It can be noticed that the DRM approach is slightly higher than the ones requested by the HopMin and MaxBattery solutions. This is due to the more complex control operated by the SDN Controller in order to modify the node flow table and, hence, increasing the overall network lifetime.

From the previous analysis, it can be intuitively argued that the proposed DRM approach achieves a better trade-off between lifetime and latency. Specifically, DRM can reach a network lifetime comparable with the MaxBattery approach, by maintaining low latency and overhead, which represent two of the most important QoS parameters for WSNs in IoT domains. Therefore, its performance is further compared against existing techniques (ACO [29], CSO [37], cooperative PSO [31,38] and LOA [23]) in terms of average latency and average energy consumption, as shown in Figures 7 and 8.

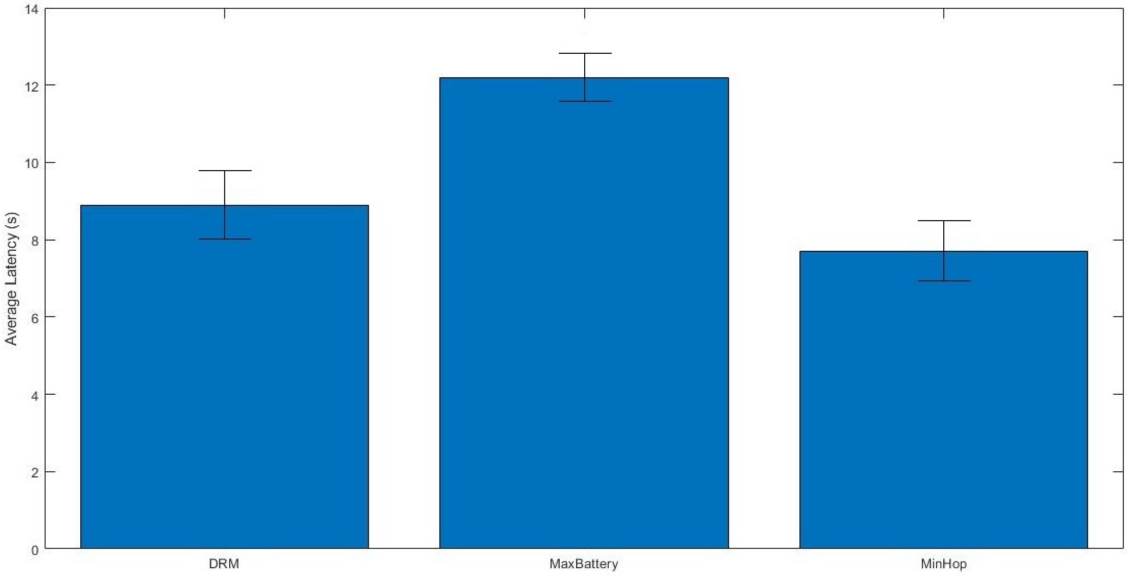

Figure 5. Comparison of proposed strategies MaxBattery, DRM and MinHop in terms of Average Latency.

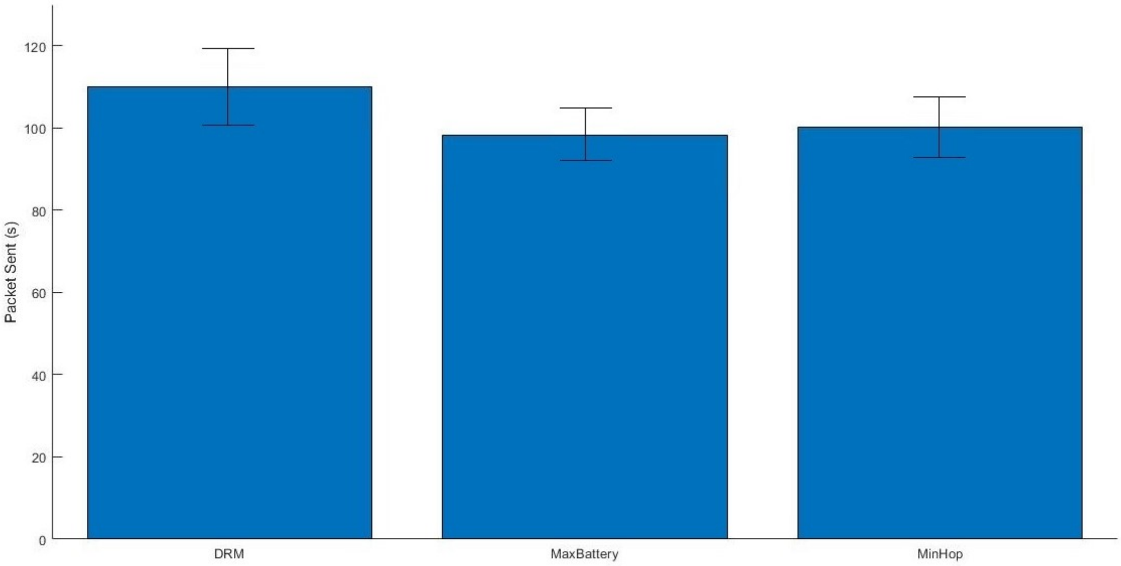

Figure 6. Overhead comparison for the proposed strategies: MaxBattery, DRM and MinHop.

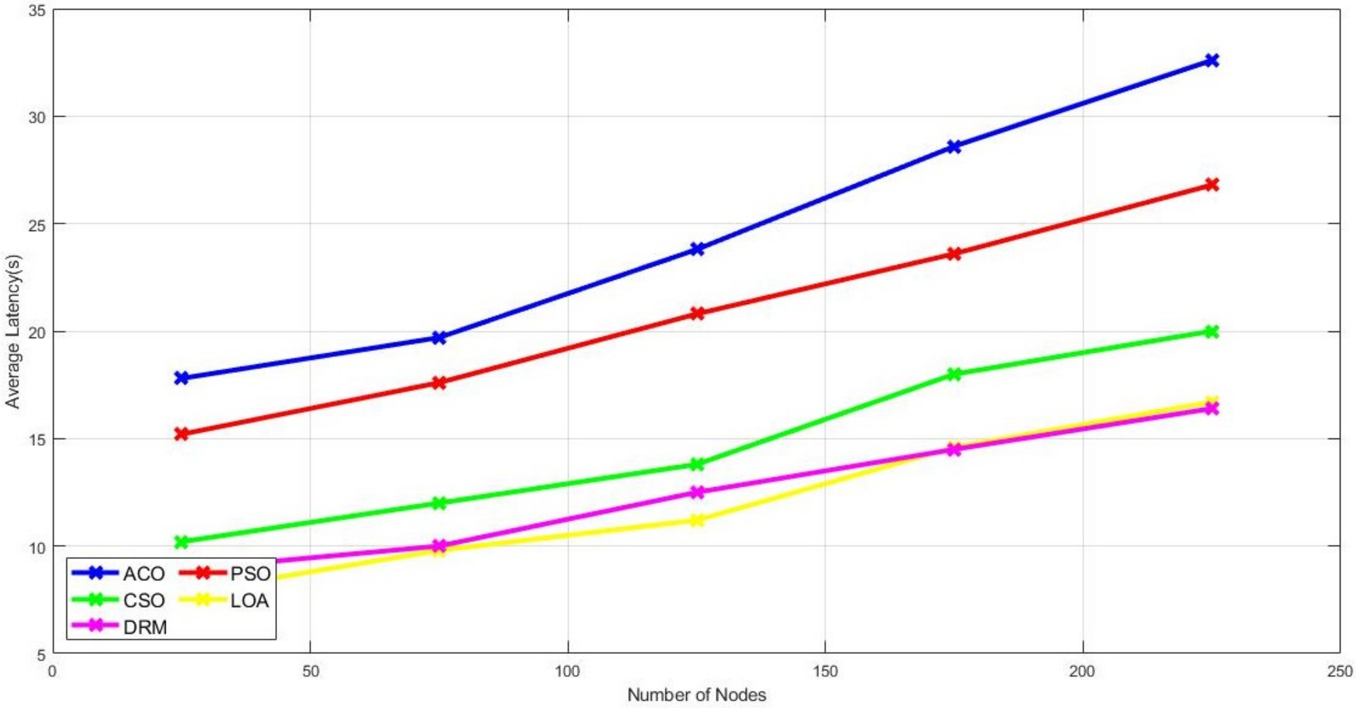

Figure 7. Average latency comparison for different network size.

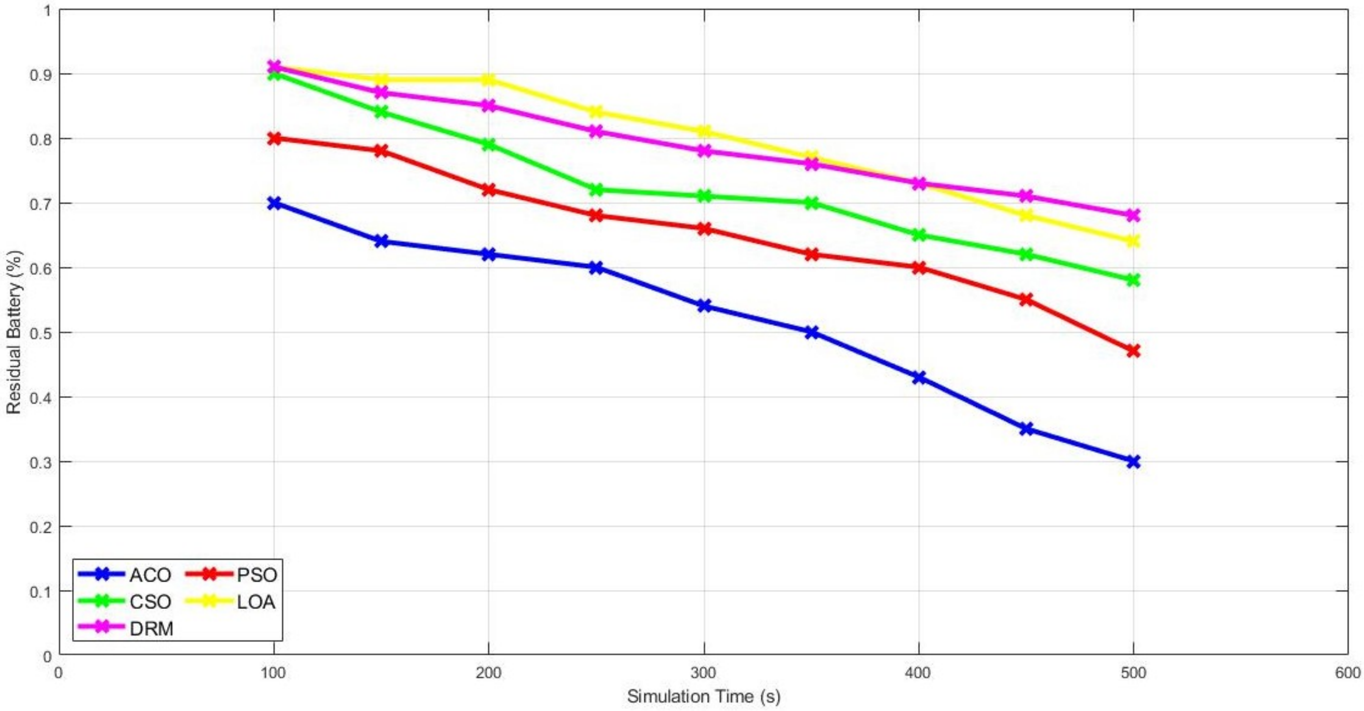

Figure 8. Average energy consumption comparison.

In particular, in Figure 7, the average latency values achieved by DRM approach and the above strategies are compared varying the network size from 25 to 225 nodes. The results show that the DRM approach outperforms all other strategies when the number of nodes is greater than 175. This minimal latency value is obtained by an adaptive use of the weight w_τ within the DRM, as shown in Algorithm 1.

In Figure 8, the comparison of DRM approach against the existing techniques in terms of average energy consumption is depicted for the worst case scenario of a network comprised of 225 nodes. It is possible to notice that the residual energy of all approaches monotonically decreases. The LOA algorithm outperforms all other solution most of the time, but the best asymptotic performance is achieved by the DRM solution. It mainly depends on the activation of energy saving procedures operated by the DRM approach, after which it is more focused on the network lifetime rather than the hops optimisation, thus reducing its decreasing slope.

Finally, Figure 9 highlights the objective function values for the optimisation problem (1) achieved by all approaches. Specifically, the results were obtained with a network topology of 225 nodes. Although the LOA strategy initially presents the better performance, DRM is able to attain the best asymptotic performance.

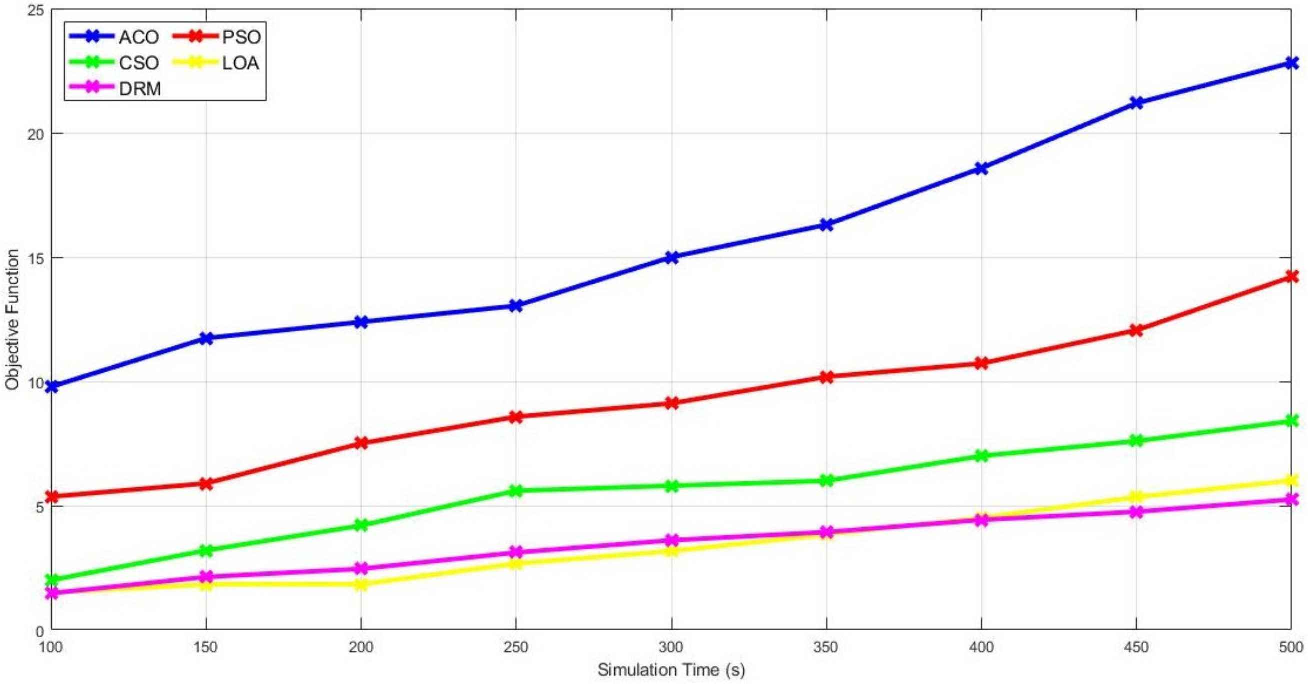

Figure 9. Objective function analysis.

5. Conclusions

This paper provides an SDN-oriented framework to improve the green management of an IoT domain. In particular, we address a general architecture design with a specific focus on the forwarding strategy operated by the SDN Controller, which dynamically adapts it to the network status in order to optimise the network responsiveness and energetic footprint. To this purpose, we characterise an optimisation problem, jointly based on nodes energy consumption and e2e flow delivery, and we address a practical solution implement in the SDN Controller in the DRM, with a limited control message overhead. We conducted a comprehensive and realistic simulation campaign, pointing out that the performance achieved by the proposed approach is better than SDN alternatives based on a hierarchical CP, as in LOA. Specifically, the SDN-DRM Controller presents the best trade-off between delay and battery consumption, without excessive overhead for comparable network sizes. Therefore, the proposed framework introduces a more effective and more flexible CP for a reference SDN IoT architecture. Future developments of the present work are planned mainly in terms of introduction of network-related machine learning techniques to predict future operative conditions to optimise in advance the weighting factors of the DRM approach.

Author Contributions: Conceptualisation, M.B. and F.C. and L.P.; methodology, M.B. and F.C.; software, M.B.; validation, M.B. and F.C. and L.P.; formal analysis, F.C.; investigation, M.B. and F.C. and L.P.; resources, F.C. and L.P.; data curation, M.B.; writing—original draft preparation, M.B. and F.C. and L.P.; writing—review and editing, M.B. and F.C. and L.P.; visualisation, M.B.; supervision, R.F.; and project administration, R.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not Applicable, the study does not report any data.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Weyrich, M.; Ebert, C. Reference Architectures for the Internet of Things. *IEEE Softw.* **2016**, *33*, 112–116. [CrossRef]
2. Barnaghi, P.; Sheth, A. On Searching the Internet of Things: Requirements and Challenges. *IEEE Intell. Syst.* **2016**, *31*, 71–75. [CrossRef]
3. Sriram, R.D.; Sheth, A. Internet of Things Perspectives. *IT Prof.* **2015**, *17*, 60–63. [CrossRef]
4. Sun, X.; Ansari, N. EdgeIoT: Mobile Edge Computing for the Internet of Things. *IEEE Commun. Mag.* **2016**, *54*, 22–29. [CrossRef]
5. Mishra, P.; Puthal, D.; Tiwary, M.; Mohanty, S.P. Software Defined IoT Systems: Properties, State of the Art, and Future Research. *IEEE Wirel. Commun.* **2019**, *26*, 64–71. [CrossRef]
6. Kalkan, K.; Zeadally, S. Securing Internet of Things with Software Defined Networking. *IEEE Commun. Mag.* **2018**, *56*, 186–192. [CrossRef]
7. Linthicum, D.S. Connecting Fog and Cloud Computing. *IEEE Cloud Comput.* **2017**, *4*, 18–20. [CrossRef]
8. Alippi, C.; Fantacci, R.; Marabissi, D.; Roveri, M. A Cloud to the Ground: The New Frontier of Intelligent and Autonomous Networks of Things. *IEEE Commun. Mag.* **2016**, *54*, 14–20. [CrossRef]
9. Rafique, W.; Qi, L.; Yaqoob, I.; Imran, M.; Rasool, R.U.; Dou, W. Complementing IoT Services Through Software Defined Networking and Edge Computing: A Comprehensive Survey. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1761–1804. [CrossRef]
10. Mavromatis, A.; Colman-Meixner, C.; Silva, A.P.; Vasilakos, X.; Nejabati, R.; Simeonidou, D. A Software-Defined IoT Device Management Framework for Edge and Cloud Computing. *IEEE Internet Things J.* **2020**, *7*, 1718–1735. [CrossRef]
11. Rauf, B.; Abbas, H.; Sheri, A.M.; Iqbal, W.; Khan, A.W. Enterprise Integration Patterns in SDN: A Reliable, Fault-Tolerant Communication Framework. *IEEE Internet Things J.* **2021**, *8*, 6359–6371. [CrossRef]
12. Pan, J.; McElhannon, J. Future Edge Cloud and Edge Computing for Internet of Things Applications. *IEEE Internet Things J.* **2018**, *5*, 439–449. [CrossRef]
13. Bonanni, M.; Chiti, F.; Fantacci, R. Mobile Mist Computing for the Internet of Vehicles. *Internet Technol. Lett.* **2020**, *3*, e176. [CrossRef]
14. Zhu, C.; Leung, V.C.M.; Shu, L.; Ngai, E.C.-H. Green Internet of Things for Smart World. *IEEE Access* **2015**, *3*, 2151–2162. [CrossRef]
15. Baccarelli, E.; Chiti, F.; Cordeschi, N.; Fantacci, R.; Marabissi, D.; Parisi, R.; Uncini, A. Green multimedia wireless sensor networks: Distributed intelligent data fusion, in-network processing, and optimized resource management. *IEEE Wirel. Commun.* **2014**, *21*, 20–26. [CrossRef]
16. Chiti, F.; Fantacci, R.; Pierucci, L. Energy Efficient Communications for Reliable IoT Multicast 5G/Satellite Services. *Future Internet* **2019**, *11*, 164. [CrossRef]
17. Bekri, W.; Jmal, R.; Chaari Fourati, L. Internet of Things Management Based on Software Defined Networking: A Survey. *Int. J. Wirel. Inf. Netw.* **2020**, *27*, 385–410. [CrossRef]
18. Bera, S.; Misra, S.; Vasilakos, A.V. Software-Defined Networking for Internet of Things: A Survey. *IEEE Internet Things J.* **2017**, *4*, 1994–2008. [CrossRef]
19. Tayyaba, S.K.; Shah, M.A.; Khan, O.A.; Ahmed, A.W. Software Defined Network (SDN) Based Internet of Things (IoT): A Road Ahead. In Proceedings of the International Conference on Future Networks and Distributed Systems, Cambridge, UK, 19–20 July 2017.
20. Li, H.; Shimin, S.; Bokgyu, J.; Xianshu, J.; Sunyoung, H. QoS-Aware Routing Mechanism in OpenFlow-Enabled Wireless Multimedia Sensor Networks. *Int. J. Distrib. Sens. Netw.* **2016**, *12*, 9378120.
21. Hu, F.; Hao, Q.; Bao, K. A Survey on Software-Defined Network and OpenFlow: From Concept to Implementation. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 2181–2206. [CrossRef]
22. Flauzac, O.; Santamaria, C.; Nolot, F.; Woungang, I. An SDN approach to route massive data flows of sensor networks. *Int. J. Commun. Syst.* **2020**, *33*, e4309. [CrossRef]
23. Srinivasa Ragavan, P.; Ramasamy, K. Software defined networking approach based efficient routing in multihop and relay surveillance using Lion Optimization algorithm. *Comput. Commun.* **2020**, *150*, 764–770. [CrossRef]
24. Banerjee, A.; Hussain, D. SD-EAR: Energy Aware Routing in Software Defined Wireless Sensor Networks. *Appl. Sci.* **2018**, *8*, 1013. [CrossRef]

25. Pradeepa, R.; Pushpalatha, M. SDN enabled SPIN routing protocol for wireless sensor networks. In Proceedings of the 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, India, 23–25 March 2016; IEEE: New York, NY, USA, 2016; pp. 639–643.
26. Shafique, A.; Cao, G.; Aslam, M.; Asad, M.; Ye, D. Application-Aware SDN-Based Iterative Reconfigurable Routing Protocol for Internet of Things (IoT). *Sensors* **2020**, *12*, 3521. [CrossRef]
27. Behera, T.M.; Samal, U.C.; Mohapatra, S.K. Energy-efficient modified LEACH protocol for IoT application. *IET Wirel. Sens. Syst.* **2018**, *8*, 223–228. [CrossRef]
28. Heinzelman, W.R.; Chandrakasan, A.; Balakrishnan, H. Energy-efficient communication protocol for wireless microsensor networks. In Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, Maui, HI, USA, 4–7 January 2000; Volume 2, p. 10.
29. Selvakennedy, S.; Sinnappan, S.; Shang, Y. A biologically-inspired clustering protocol for wireless sensor networks. *Comput. Commun.* **2007**, *30*, 2786–2801. [CrossRef]
30. Ke, C.; Wu, M.; Hsu, W.; Chen, C. Discover the Optimal IoT Packets Routing Path of Software-Defined Network via Artificial Bee Colony Algorithm. In *Wireless Internet. WiCON 2019. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*; Springer: Cham, Switzerland, 2020; Volume 317.
31. Xiang, W.; Wang, N.; Zhou, Y. An Energy-Efficient Routing Algorithm for Software-Defined Wireless Sensor Networks. *IEEE Sens. J.* **2016**, *16*, 7393–7400. [CrossRef]
32. Kumar, N.; Vidyarthi, D.P. A Green Routing Algorithm for IoT-Enabled Software Defined Wireless Sensor Network. *IEEE Sens. J.* **2018**, *18*, 9449–9460. [CrossRef]
33. Al-Janabi, T.A.; Al-Raweshidy, H.S. Efficient whale optimisation algorithm-based SDN clustering for IoT focused on node density. In Proceedings of the 2017 16th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net), Budva, Montenegro, 28–30 June 2017; pp. 1–6.
34. Galluccio, L.; Milardo, S.; Morabito, G.; Palazzo, S. SDN-WISE: Design, prototyping and experimentation of a stateful SDN solution for WIreless SEnsor networks. In Proceedings of the 2015 IEEE Conference on Computer Communications (INFOCOM), Hong Kong, China, 26 April–1 May 2015; IEEE: New York, NY, USA, 2015; pp. 513–521. [CrossRef]
35. Anadiotis, A.G.; Milardo, S.; Morabito, G.; Palazzo, S. Toward Unified Control of Networks of Switches and Sensors Through a Network Operating System. *IEEE Internet Things J.* **2018**, *5*, 895–904. [CrossRef]
36. Jain, R.; Chiu, D.; Hawe, W. A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer System. Available online: https://www.cs.wustl.edu/~jain/papers/ftp/fairness.pdf (accessed on 27 April 2021).
37. Kong, L.; Pan, J.S.; Tsai, P.W.; Vaclav, S.; Ho, J.H. A balanced power consumption algorithm based on enhanced parallel cat swarm optimization for wireless sensor network. *Int. J. Distrib. Sens. Netw.* **2015**, *11*, 729680. [CrossRef]
38. Hu, Y.F.; Ding, Y.S.; Ren, L.H.; Hao, K.R.; Han, H. An endocrine cooperative particle swarm optimization algorithm for routing recovery problem of wireless sensor networks with multiple mobile sinks. *Inf. Sci.* **2015**, *300*, 100–113. [CrossRef]