An Energy-Aware SDN/NFV Architecture for the Internet of Things

Dipon Saha*, Meysam Shojaee*, Michael Baddeley‡ and Israat Haque*
* Department of Computer Science, Dalhousie University, Halifax, Canada
‡ Toshiba Research Europe Ltd., United Kingdom (Great Britain)
Email:dipon.saha@dal.ca, Meysam.Shojaee@dal.ca, Michael.Baddeley@toshiba-trel.com, israat@dal.ca

***Abstract*—The Internet of Things (IoT) is an essential component of emerging network applications like smart cities, where billions of IoT devices are connected to transport massive network traffic. A programmable network such as software-defined networking (SDN) can cope with such data explosion and constrained network resources. Furthermore, Network Function Virtualization (NFV) can enable on-demand network functions deployment. Thus, SDN and NFV can complement each other to empower an architecture for the IoT. In this paper, we first design an SDN/NFV enabled IoT node for dynamic deployment of network functions. The proposed IoT node is then used to develop an SDN/NFV architecture to realize on-demand network functions like data aggregation. Next, we define an Integer Linear programming (ILP) problem to optimize the energy consumption of the IoT nodes by activating an optimal number of NFV nodes and optimally assigning regular nodes to those activated NFV nodes. Finally, we design a heuristic and evaluate it in the Cooja simulator. Extensive evaluation confirms that the proposed solution outperforms its counterparts in communication energy consumption and packet delivery ratio.**

***Index Terms*—SDN, NFV, IoT, ILP, energy-aware routing.**

I. INTRODUCTION

The Internet of Things (IoT) is a system of connected computing devices (e.g., sensors) that is capable of sharing the sensed data over the Internet. In emerging applications like smart cities, billions of IoT devices are connected to transport massive network traffic. Thus, we need a new programmable architecture to dynamically monitor and configure IoT networks [1]. Software-Defined Networking (SDN) [2] decouples network control logic (control plane) from network elements (data plane) to enable programmability and smooth protocol evolution. The logically centralized controller dynamically configures network elements to meet application demand using its global network view. Network Function Virtualization (NFV) [3], on the other hand, virtualizes network functions (e.g., firewall, load balancer) by decoupling them from proprietary hardware appliances. Thus, the IoT resources can be abstracted as logical units, and different services can be deployed as virtualized network functions. We can implement these virtualized functions in an SDN architecture, where the controller can act as an orchestrator.

However, SDN and NFV are not intended for IoT and need revisiting their architecture to support IoT applications. Baddeley *et al.* [4] propose a lightweight SDN architecture, called *μSDN*, adopting and optimizing *RPL* (the IPv6 Routing Protocol for Low-Power and Lossy Networks) [5], which is built on the IEEE 802.15.4. Anadiotis *et al.* [6] propose, *SD-WISE*, for Wireless Sensor Networks (WSNs) to support NFV. The authors in [7], [8] focus on energy consumption in SDN-based wired networks. However, there is no work designing an energy-aware SDN/NFV architecture to support IoT applications. One important design aspect in the IoT environment is to reduce the control overhead. The flow rule installation approach of a new incoming flow in traditional SDN is not suitable for low-power lossy IoT networks. We can deploy source routing to reduce the control overhead [4]. We argue that *data aggregation* [9] can further reduce the network traffic and improve the overall resource utilization, e.g., energy consumption. The data aggregation can be defined as an NFV instance and dynamically deploy at the required IoT nodes. There are various data aggregation techniques [10] that we can choose as the virtual function to meet the application demand. In this work, we use energy and computational efficient hierarchical structured in-network aggregation that calculates mean of the sensed data [10].

Annex to ISBN 978-3-903176-28-7© 2020 IFIP

In this paper, we design energy-aware SDN/NFV architecture and protocol for IoT networks. We first propose a node architecture to support virtualized functions by adopting μSDN node architecture. We use the proposed node to implement data aggregation, but the initial investigation reveals that the number and the placement of the NFV nodes have an impact on the overall network throughput and energy consumption. Thus, we define an *Integer Linear Programming* (ILP) problem. The model optimizes the number of activated NFV nodes and the assignment of regular IoT nodes to them while minimizing the total energy consumption of the network. Then, we define a heuristic (*EA-SDN/NFV*) as the ILP problem is NP-complete. We implement and evaluate the proposed solution in the *Cooja* simulator for *Contiki* OS [11]. The results reveal that EA-SDN/NFV improves energy consumption around 1.4x compared to μSDN. Also, the NFV based protocols have a better packet delivery ratio.

The rest of the paper is organized as follows. Section II briefly compares and contrasts existing SDN/NFV IoT designs with the proposed solution. The proposed ILP model and corresponding heuristic are presented in Section III following the SDN/NFV architecture in Section IV. The last two sections provide the discussion on evaluation results and conclusions.

II. RELATED WORK

In this section, we present and discuss research work related to our proposed architecture and model. μSDN incorporates the RPL protocol for topology discovery and flow-based source routing for data traffic. The solution makes appropriate changes and optimization to different network functions to cope with IoT networks and implements a controller assisted interference mitigation as a use case. SDN-WISE [12] is the first stateful software-defined architecture for WSN. An extension of SDN-WISE is proposed in [6] to accommodate NFV (e.g., geographic routing) provisioning. SDSense [13] is another SDN-based WSN design, where it decomposes network functions based on their scope of agility. However, none of the above solutions is compatible with IPv6 and RPL except μSDN - both of which are useful for low power lossy IoT. Furthermore, none of them offers any energy optimization. Also, SDN-WISE and SDSense do not provide any NFV provisioning. The authors in [7], [8], [14]–[18] optimize the energy consumption in wired and wireless networks. The authors in [14] optimize energy in a wireless environment without SDN/NFV support. Ding *et al.* [19] design an interference-aware energy-efficient packet routing algorithm for software-defined WSNs. However, the design does not consider reducing the control traffic overhead. Also, the solution does not have any NFV provisioning. In summary, none of the existing designs considers the IoT domain-specific challenges like the compatibility with IPv6 and RPL protocol, the energy awareness, and the ability to dynamically deploying network functions. In this paper, we fill this gap and propose a software-defined architecture, an ILP model, and a heuristic to dynamically deploy essential network functions in desired network elements to improve the overall network performance.

III. THE OPTIMIZATION MODEL

In this section, we first define the proposed ILP problem and each constraint. The objectives of our optimization formulation include: finding the optimal number of NFV nodes that must be activated to support all the source IoT nodes and assigning these nodes to the activated NFV nodes to minimize the total activation cost of NFV nodes and the energy consumption of the network. Then, we develop a heuristic as the model is NP-complete.

Let $G = (V, L)$ be a graph representing a network topology, where $V = \{1, 2, 3, \ldots, n\}$ be the network nodes (IoT devices) and L be the set of links among these nodes. We consider four types of nodes: the sink node $S \in V$, the NFV nodes $N \subset V$, the source nodes $R \subset V$, and the intermediate relay nodes $I \subset V$, where $(R \cap N \cap I) = 0$ and $(R \cup N \cup I) = V - S$. The sources generate traffic for the sink, where remaining nodes transport that either as regular relays or NFV nodes (see Figure 1b). We assume that the NFV nodes are already placed at optimal locations in the network, where the activation cost of an NFV node, n, is c_n. Recall that in-network data aggregation requires certain memory and CPU at the NFV nodes; thus, we define their activation cost, c_n, as the operational resource utilization (CPU and memory).

TABLE I: The list of notations used in the formulation.

Notation	**Description**
V	Set of all nodes.
N	Set of NFV nodes.
R	Set of source nodes.
S	Sink node.
I	Set of intermediate nodes.
L	Set of all links.
(i, j)	A link $(i, j) \in L$.
c_n	Cost of activating an NFV node n.
e_j	Energy level of a node j.
CP_n	Capacity of an NFV node n.
TH_j	Energy threshold for a node j.
a_n	Decision variable to activate an NFV node n.
$w1, w2$	Normalization or scaling factor.
x_{rn}	Decision variable for the assignment of a node r.
y_{ijrn}	Decision variable for a link (i, j) selection.
z_{ijrn}	Combined decision variable for assignment and link selection.

The capacity of an NFV node, n, is referred to as CP_n that denotes the maximum source nodes

$$\min \quad \sum_{n \in N} w_1 c_n \, a_n + \sum_{n \in N} \sum_{r \in R} \sum_{(i,j) \in L} x_{rn} \, y_{ijrn} \, (1 - w_2 e_j) \tag{1}$$

$$\text{s.t.} \quad \sum_{n \in N} a_n \geq 1 \tag{2}$$

$$\sum_{n \in N} x_{rn} = 1, \quad \forall r \in R \tag{3}$$

$$x_{rn} \leq a_n, \quad \forall r \in R, \forall n \in N \tag{4}$$

$$\sum_{r \in R} x_{rn} \ \leq \ CP_n, \quad \forall n \in N \tag{5}$$

$$y_{ijrn} \, e_j \ \geq \ TH_j, \quad \forall (i, j) \in L, \forall n \in N \tag{6}$$

$$\sum_{\substack{i \in V \\ (i,j) \in L \\ j = n}} y_{ijrn} = x_{rn} \quad \forall r \in R, \forall n \in N \tag{7}$$

$$\sum_{\substack{j \in V \\ (i,j) \in L \\ i = r}} y_{ijrn} = x_{rn} \quad \forall r \in R, \forall n \in N \tag{8}$$

$$\sum_{r \in R} \sum_{\substack{(i,j) \in L \\ j \in (S \cup I) \\ i = n}} y_{ijrn} = \sum_{r \in R} x_{rn} \quad \forall n \in N \tag{9}$$

$$\sum_{r \in R} \sum_{\substack{(i,j) \in L \\ i \in (N \cup I) \\ j = S}} y_{ijrn} = \sum_{r \in R} x_{rn} \quad \forall n \in N \tag{10}$$

$$\sum_{\substack{j \in I \\ (i,j) \in L}} y_{jirn} = \sum_{\substack{j \in I \\ (i,j) \in L}} y_{ijrn} \quad \forall r \in R, \forall n \in N \tag{11}$$

$$\sum_{\substack{j \in V \\ (i,j) \in L}} y_{ijrn} \ \leq \ 1 \quad \forall i \in V, \forall r \in R, n \in N \tag{12}$$

it can serve. The residual energy of a node, $j \in V$, is defined as e_j, which is proportional to the number of successful transmissions. The energy threshold for a node $j \in V$ is TH_j, i.e., below that threshold a node cannot operate. We also use the following three binary decision variables in the model. $a_n \in [1, 0]$ decides if an NFV node $n \in N$, is activated or not while $x_{rn} \in [1, 0]$ takes care if a node $r \in R$ is assigned to $n \in N$, and y_{ijrn} decides if a link $(i, j) \in L$ is selected for the assignment of r to n.

The first part of the objective function (1) minimizes the number of activated NFV nodes. The second part calculates the optimal assignment of source nodes to the activated NFV nodes by determining the most energy-efficient routes. The constraints in (2), (3), and (4) are for the NFV node activation and assignment, where (2) ensures that at least one NFV node is activated, whereas constraint (3) guarantees that each source node is assigned to exactly one NFV node. The constraints in (4) and (5) prevent assigning a source node to a non-activated NFV node and guarantee that service capacity of an NFV node does not exceed during an assignment, respectively. The constraints (6) ensure that if the energy-level of a node falls below a given threshold, the associated link is not selected. The constraints (7) to (11) and (12) are for route selection, where (7) to (11) are flow conservation constraints and (12) is for the cycle avoidance.

$$\min \quad \sum_{n \in N} w_1 c_n \, a_n + \sum_{n \in N} \sum_{r \in R} \sum_{i,j \in L} z_{ijrn} \, (1 - w_2 e_j) \tag{13}$$

$$\text{s.t.} \quad Constraints\ (2)\ to\ (12)$$

$$z_{ijrn} \leq x_{rn} \quad \forall (i, j) \in L, \forall r \in R, \forall n \in N \tag{14}$$

$$z_{ijrn} \leq y_{ijrn} \quad \forall (i, j) \in L, \forall r \in R, \forall n \in N \tag{15}$$

$$z_{ijrn} \geq x_{rn} + y_{ijrn} - 1 \; \forall (i, j) \in L, \forall r \in R, \forall n \in N \tag{16}$$

It is evident that the optimization problem (1) to (12) is non-linear as the objective function (1) contains the multiplication of two decision variables. Thus, we replace the variables x_{rn} and y_{ijrn} with a new one, z_{ijrn}. Furthermore, we need to impose a set of constraints on variable z_{ijrn} to validate this replacement. Thus, the new linearized optimization problem is presented from (13) to (16). In this new formulation, the constraints 14 to 16 satisfies $z_{ijrn} = x_{rn} \times y_{ijrn}$.

The heuristic: The ILP problems are NP-complete; therefore, we cannot find the solutions quickly, especially in large IoT networks. Thus, we build our heuristic Assignment and Path Selector (APS) in Algorithm 1. In APS, we assign source nodes to the available NFV nodes in such a way that the number of activated NFV nodes and total energy consumption of the network is minimized. First, we calculate the two shortest routes from each source node, r, to the available NFV nodes (line 3). Each of these routes has the associated energy-cost (the total communication energy). We sort these two routes based on that energy-cost (line 4). Then, we assign the source node to one of the available NFV nodes based on the energy-cost and activation cost (operational resource utilization) of an NFV node such that both costs are minimized while the capacity of that chosen NFV node does not exceed (line 5 to line 16). We furthermore update that NFV node's capacity (line 12) and repeat the entire process for all source nodes.

Algorithm 1 Assignment and Path Selector Algorithm (APS)

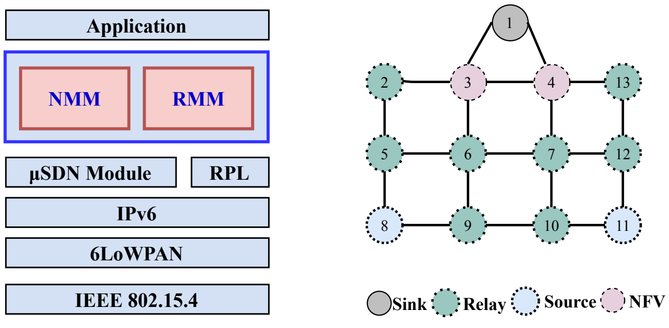
```
Input
    NFV nodes: $N$
    Source nodes: $R$
Output
    NFV nodes map: $X_{RN}$
    All routes map: $Y_{RN}$

1:  for all $r \in R$ do
2:      for all $n \in N$ do
3:          $routes_r$ ← get two shortest routes from r to n
4:          $primary_r[n], secondary_r[n]$ ← sort $routes_r$ with (energy-cost)
5:          $EnergyCost_r$ ← get total Energy ($primary_r[n]$, $secondary_r[n]$)
6:          $c_n$ ← get the activation cost of $n$
7:          $Cost_r[n]$ ← totalCost($EnergyCost_r, c_n$)
8:      end for
9:      $N_{sorted}$ ← sort the set of NFV($Cost_r, N$)
10:     for all $nfv_r \in N_{sorted}$ do
11:         if $nfv_r$ has capacity then
12:             $nfv_r.capacity \leftarrow nfv_r.capacity + 1$
13:             break
14:         end if
15:     end for
16:     $X_{RN}.append(r, nfv_r)$
17:     $Y_{RN}.append(r, primary_r[nfv_r], secondary_r[nfv_r])$
18: end for
19: return  $X_{RN}$, $Y_{RN}$
```

IV. ARCHITECTURE

In this section, we provide the SDN/NFV based node and network architectures that we use to implement the proposed heuristic. Recall that we have four types of nodes: sink, source, intermediate relay, and NFV, where the source nodes require only the SDN capability, while the others need both the SDN and NFV capabilities.

SDN/NFV Node Architecture: The proposed node architecture is presented in Fig. 1a, which is an extension of μSDN node. We introduce two modules: *NFV Management Module (NMM)* and *Route Management Module (RMM)*. NMM consists of *Virtual Network Function (VNF) container* and *VNF manager* to keep the available network function definitions (e.g., aggregation) and provide an API to an NFV enabled node, respectively. RMM module of a node maintains the energy-state information of its neighbors and shares it with the controller. The controller uses the energy-state information to activate an optimal number of NFV nodes and constructs corresponding energy-aware routes, which are also maintained at RMM.



(a)

(b)

Fig. 1: The SDN/NFV based node architecture (a) and an example grid topology (b).

The Functional Modules of the SDN Controller: We deploy the controller at the sink. The controller uses five different control messages: *Node-state Update (NSU)*, *Flow table query (FTQ)*, *Flow table set (FTS)*, *Configuration (CONF)*, and *NFV configuration (NFV-CONF)* as well as RPL control messages. The *topology discovery* module uses RPL messages to construct and maintain the routing topology. In particular, RPL DAO and CONF messages update the network state (e.g., energy) at the controller and configuration metrics at remaining nodes, respectively. The *NFV node activation and route construction* module runs the heuristic that we proposed in Section III on the constructed routing topology to determine the set of activated NFV nodes and corresponding route assignment. The controller then initiates the NFV-CONF message to activate the VNF (data aggregation) at the chosen NFV nodes and notifies corresponding source nodes. The sources then generate the FTQ message to the controller to get the primary and secondary flow table entries towards the assigned NFV node, where the controller uses FTS message. A source node switches between these two routes after a pre-determined number of transmissions.

The sources forward packets as per the configured flow entries, whereas intermediate relays use *Source-Routing Header* (SRH) of a packet (source-routing) to forward it to the next-hop to avoid additional control traffic. NFV nodes aggregate the received packets once their buffer gets full. The *network state collection* module periodically collects state information from the data plane nodes using NSU messages to maintain and update the routing topology. Also, a node sends state traffic over the NSU message if its energy-level approaches a minimal operational threshold or its local topology changes. The controller maintains this state information to decide on the topology and route reconstruction.

V. EVALUATION

In this section, we present the evaluation setup and results of the energy-aware protocol in the proposed SDN/NFV architecture.

Evaluation Setup: We consider *transmission energy consumption (mJ)*, which is the total energy that successful transmissions of application data traffic consume, and *packet delivery ratio (PDR)*, which is the ratio of the total number of packets successfully reached a destination over the total number of packets sent to that destination. Each IoT node consists of TI's MSP430F5438 CPU and CC2420 radio [4]. For the power supply, we consider a coin-type lithium-ion battery with 3V and 150 mA-h power rating. We create different topologies by changing the location of the NFV, sink, and other nodes in a 40 node grid topology (similar to Fig.1b) in each run. There are a sink, five NFV, and ten source nodes. The remaining are the relay nodes. We evaluate the proposed solutions in the Cooja simulator running on Contiki OS and compare the outcome with μSDN. In the SDN/NFV design without energy optimization, we assign the sources to an NFV node over the shortest route. The sources alternate between the primary and secondary routes towards the NFV node to distribute the load and energy utilization. The controller is deployed at the sink, which invokes the NFV assignment and route constructions if the energy level of a node along the routes between a source and the sink reaches 20% of its initial energy and impact the current routes and topology. We run the evaluation 50 times to get the average with a 95% confidence interval.

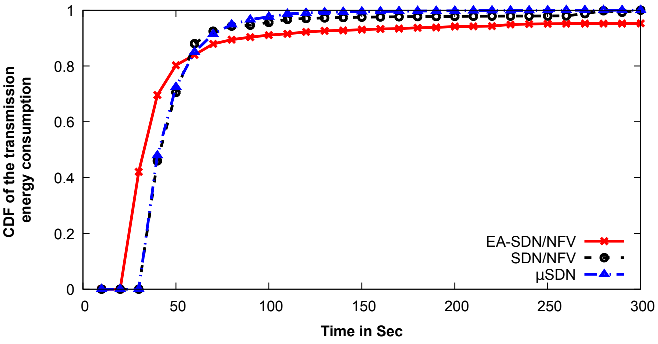
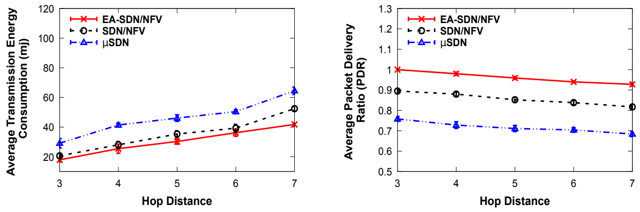


Fig. 2: The CDF of the transmission energy consumption over time.

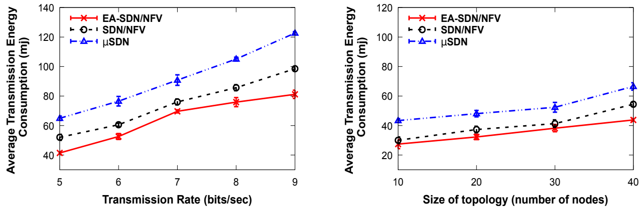
Discussion on Results: The CDF of transmission energy consumption of EA-SDN/NFV and the other two schemes is presented in Fig. 2. Out of the three schemes, EA-SDN/NFV slowly gets close to the saturation, i.e., the situation where nodes have utilized most of their energy. We also observe a clear gap between EA-SDN/NFV and the other two schemes that do not consider energy optimization. Thus, EA-SDN/NFV improves the overall network lifetime. Also, the relay nodes have the most energy savings; therefore, as the majority node type, their savings improves on the overall network performance.

Fig. 3a shows the average transmission energy consumption of the application data. Overall, EA-SDN/NFV has the best energy consumption, which increases with the increasing number of hops as expected. The data aggregation and energy distribution over the two disjoint routes help EA-SDN/NFV to utilize the resources better. μSDN has the worst performance because



(a) (b)

Fig. 3: The average transmission energy consumption (a) and packet delivery ratio (b).



(a) (b)

Fig. 4: The average transmission energy consumption with different traffic loads (a) and topology sizes (b).

of not having any energy optimization or data aggregators. EA-SDN/NFV improves around 1.35x transmission energy consumption compared to μSDN, where relay nodes are the main contributor to that improvement. If we do not distribute the energy consumption among relays, some may carry more traffic compared to others, which has an impact on the overall energy consumption. The comparison of the average packet delivery ratio (PDR) is shown in Fig. 3b. EA-SDN/NFV and SDN/NFV both have the higher PDR compared to μSDN. We suspect that the interference has a roll on this performance. In the case of μSDN, packets close to the sink has a higher chance of experiencing interference due to the small number of alternatives routes. However, in the other two schemes, NFV nodes reduce the number of transmissions towards the sink as well as the chance of interference. Thus, the deployment of NFV nodes shows a clear benefit for achieving high PDR and better energy utilization.

We present the energy consumption in Fig. 4a with different packet generation rates (loads) over the maximum hop distance between sources and sink. The results again confirm that EA-SDN/NFV has the best performance with 1.38x improvement over μSDN. We speculate that with the increased traffic, the relay nodes need to carry a large amount of traffic compared to others. As EA-SDN/NFV significantly improves the energy utilization in those relay nodes, the overall improvement is prominent at high load. The NFV nodes further reduce the amount of traffic towards the sink as well as the chance of interference. The energy consumption with varying topology sizes (Fig. 4b) also offers similar performance trend.

VI. CONCLUSION

In this paper, we have proposed an energy-aware SDN/NFV framework for IoT networks. We have formulated an ILP problem for an optimal number of NFV nodes activation and the assignment of sources to those activated NFV nodes over energy-efficient routes. We have then developed a heuristic to implement the model as the proposed ILP problem is NP-complete. We have furthermore designed a network architecture to dynamically deploy the proposed heuristic in an IoT environment. We have evaluated the heuristic in the Cooja simulator. The evaluation results have confirmed that the proposed energy-aware SDN/NFV based protocol improved around 1.4x energy utilization compared to its counterpart. It furthermore has improved the packet delivery ratio and network lifetime.

REFERENCES

[1] I. Haque and N. Abu-Ghazaleh, “Wireless software defined networking: a survey and taxonomy,” *IEEE Comm. Surveys and Tutorials*, vol. 18, no. 4, pp. 2713–2737, May 2016.
[2] D. Kreutz *et al.*, “Software-defined networking: A comprehensive survey,” *IEEE*, 2015.
[3] B. Han *et al.*, “Network function virtualization: Challenges and opportunities for innovations,” *IEEE Comm. Magazine*, vol. 53, no. 2, pp. 90–97, 2015.
[4] M. Baddeley *et al.*, “Evolving SDN for low-power IoT networks,” in *IEEE NetSoft*, 2018, pp. 71–79.
[5] T. Winter *et al.*, “RPL: IPv6 routing protocol for low-power and lossy networks,” Tech. Rep., 2012.
[6] A.-C. Anadiotis *et al.*, “SD-WISE: a software-defined wireless sensor network,” *arXiv preprint arXiv:1710.09147*, 2017.
[7] A. Markiewicz, P. N. Tran, and A. Timm-Giel, “Energy consumption optimization for software defined networks considering dynamic traffic,” in *IEEE CloudNet*, 2014, pp. 155–160.
[8] A. Fernandez-Fernandez, C. Cervello-Pastor, and L. Ochoa-Aday, “Achieving energy efficiency: An energy-aware approach in SDN,” in *IEEE GLOBECOM*, 2016, pp. 1–7.
[9] M. Khan *et al.*, “Big data processing using internet of software defined things in smart cities,” *International Journal of Parallel Programming*, pp. 1–14, 2018.
[10] S. A. Dehkordi *et al.*, “A survey on data aggregation techniques in iot sensor networks,” *Wireless Networks*, vol. 26, no. 2, pp. 1243–1263, 2020.
[11] A. Dunkels, B. Gronvall, and T. Voigt, “Contiki-a lightweight and flexible operating system for tiny networked sensors,” in *IEEE international conference on local computer networks*, 2004, pp. 455–462.
[12] L. Galluccio *et al.*, “SDN-WISE: Design, prototyping and experimentation of a stateful SDN solution for WIreless SEnsor networks,” in *IEEE INFOCOM*, 2015, pp. 513–521.
[13] I. Haque *et al.*, “SDSense: An agile and flexible SDN-based framework for wireless sensor networks,” *The IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1866 – 1876, February 2019.
[14] Y. Yao, Q. Cao, and A. V. Vasilakos, “EDAL: An energy-efficient, delay-aware, and lifetime-balancing data collection protocol for heterogeneous wireless sensor networks,” *IEEE/ACM Transactions on Networking (TON)*, vol. 23, no. 3, pp. 810–823, 2015.
[15] K. Kaur *et al.*, “An energy-driven network function virtualization for multi-domain software defined networks,” *arXiv preprint arXiv:1903.09924*, 2019.
[16] I. Haque and C. Assi, “Localized energy efficient routing in mobile ad hoc networks,” *The Willey Journal of Wireless and Mobile Computing*, vol. 7, no. 6, pp. 781–793, August 2007.
[17] I. T. Haque and C. Assi, “OLEAR: Optimal localized energy aware routing in mobile ad hoc networks,” in *IEEE International Conference on Communications*, ser. ICC ’05, 2005.
[18] I. Haque, S. Islam, and J. Harms, “On selecting a reliable topology in wireless sensor networks,” in *IEEE International Conference on Communications*, ser. ICC ’15, 2015.
[19] Z. Ding *et al.*, “An interference-aware energy-efficient routing algorithm with quality of service requirements for software-defined wsns,” *IET Comm.*, vol. 13, no. 18, pp. 3105–3116, 2019.