Routing Policy Verification for Enhanced Energy Quality of Service and Security Monitoring in IoT Networks

Konstantinos Papachristou, Traianos-Ioannis Theodorou, Stavros Papadopoulos,
Aikaterini Protogerou, Anastasios Drosou, Dimitrios Tzovaras
Centre for Research & Technology Hellas, Thessaloniki, Greece
{*kostas.papachristou, theodorou, spap, k.protogerou, drosou, dimitrios.tzovaras*}*@iti.gr*

***Abstract*—The Internet of Things (IoT) is growing rapidly controlling and connecting thousands of devices every day. Software Defined Networking (SDN) simplifies network management tasks by separating the control plane. However, the increased network traffic results in energy and Quality of Service (QoS) efficiency issues, whereas IoT devices are susceptible to failures and attacks that have serious security consequences. In this regard, providing a guarantee that SDN routing satisfies energy, QoS and security related policies is crucial for the network management. In this paper, we propose a policy-based framework aiming to verify that SDN routing decisions are optimal regarding energy, QoS and security properties. The proposed framework will enable the IoT network operator to adjust the policy constraints according to the demands of each use case (e.g., aiming at more secure or faster network). Finally, our framework is illustrated using a representative evaluation scenario.**

***Keywords*-policy-based network management, software-defined networking, energy, quality of service, security, internet of things**

I. Introduction

The recent technological development of Internet of Things (IoT) allows different devices and systems with computing and sensorial functionalities to collect and transfer data [1]. The IoT tecnologies and applications enable easy access and interaction between a wide variety of devices (e.g., surveillance cameras, monitoring sensors, actuators, vehicles, etc) and effective processing of large amount and variety of generated data. These services can be utilized by different users (for example citizens and companies) in many different domains, such as smart health-care, intelligent transportation systems, industrial automation and smart energy systems [2], [3], [4], [5].

Such a heterogeneous field of devices makes their communication a formidable challenge. In the traditional networks, each forwarder uses static routing tables (control plane) that are locally maintained in order to learn where the data packets (data plane) will be directed. The data routing rules cannot be modified in real-time, resulting in a static, decentralized and very complex networking infrastructure. On the other hand, the software-defined networking (SDN) technology [6] separates the control plane from the data plane, leaving the transportation of data to the forwarder and using the SDN controller to route the data packets. Therefore, the SDN technology facilitates dynamic network configuration and central control of the network, thus achieving higher network performance and monitoring capabilities.

A. Motivation

Monitoring of SDN routing is essential to network management to prevent it from cyber or physical attacks that affect the Quality of Service (QoS) and security of oT network, while the increased network traffic leads to energy efficiency problems [7]. Provided that time-varying communication links (i.e. flow routing paths) are defined by the SDN infrastructure to enable the communication between devices, the selected flow routing path leverages the performance of the network. This dynamic character of the IoT network allows the operator to define different policies based on the demands of the use cases of network. For instance, in some cases the main priority of the operator is the data security routing over the network, while , in other cases, the operator aims to achieve a very fast network, without extra delays.

B. Contribution

Recent advances of SDN technology have presented frameworks for policy-based network management that monitoring the compliance of routing decisions. Such policies are based on network data that may related to energy [8], QoS [9] and security [10] characteristics. However, a unified policy-based framework that takes into account energy, QoS and security information has not yet been analyzed, to the best of our knowledge. Toward this end, we therefore make the following technical contributions:

- A complete policy framework for verification that the SDN routing decisions ensure energy, QoS and security efficiency.
- A number of routing policies are formulated that incorporate security, QoS and energy information.
- A methodology that estimates the optimal flow rules with or without knowledge about the significance of each policy.

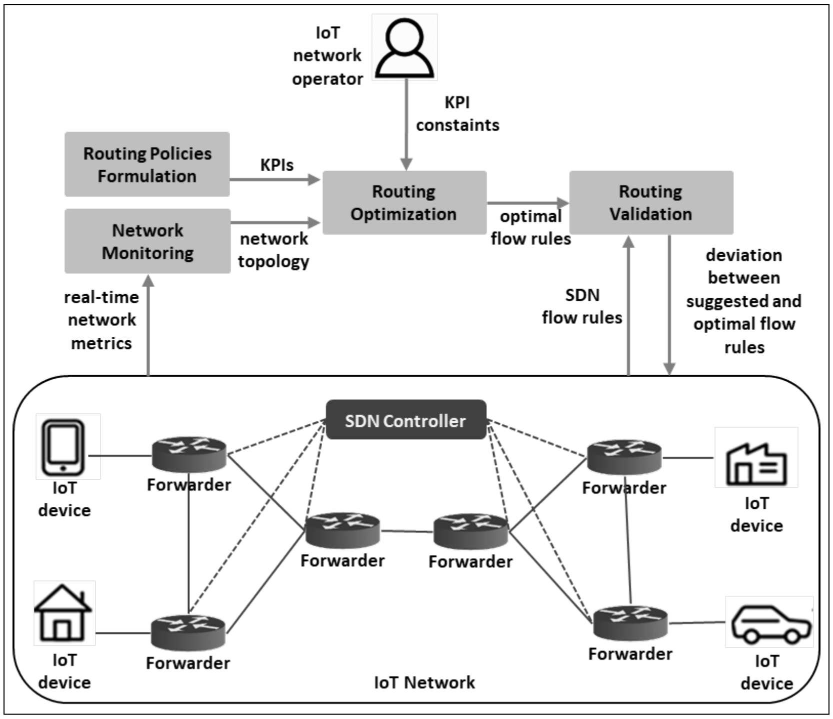


Figure 1. The workflow of the proposed flow routing policy verification framework: An IoT network enables different devices to collect and exchange data. To achieve this, the SDN controller generates flow rules that send the IoT data to the forwarders. The proposed policy-based framework verifies that SDN routing decisions are optimal regarding energy, QoS and security properties. More specifically, in the *Network Monitoring* step, real-time network metrics that concern energy, QoS and security information are collected. This information is used for the calculation of routing policies formulated as Key Performance Indicators (KPIs) in the *Routing Policies Formulation* step and they estimate how good a routing decision is regarding energy, QoS and security. By employing multi-objective optimization, a set of the best solutions (i.e. flow rules) is identified in the *Routing Optimization* step. The set of optimal flow rules is then compared with the flow rules created by the SDN, in order to compare them and provide a deviation metric that is the actual verification result (*Routing Validation* step).

C. Structure

The rest of this paper is organized as follows. First we present related work on policy-based network management in Section II. Section III presents the proposed routing policy verification framework, while in Section IV, an evaluation scenario is described that highlights the efficiency of the proposed framework. Finally, conclusion remarks with some future research directions are given in Section V.

II. Related Work

The dynamic character of SDN technology enables the network operators to define their own policies on the network management. In [9] a QoS policy enforcement framework for SDN has been proposed. This framework specifies policies to enforce QoS in an OpenFlow-based network. It checks the compliance of the policies and autonomically adapts the flow rules to satisfy the policies if needed.

A security-related policy framework was presented in [10]. In this framework, a network operator is able to create and implement security policies using human-readable language. the security policies define which security services (e.g., deep packet inspection, intrusion detection, spam detection, etc.) must be applied and specify how SDN system reacts if malicious traffic is detected. It also provides a detailed explanation of how these policies are converted into a series of OpenFlow messages to implement such a policy.

[11] presented a framework for security policy management in OpenFlow-based SDN networks. It enables the network operator to define and manage security policies, including for example access control rules, detecting conflicting policies, defining priorities, delegating rights, etc. It translates the high-level security policies into OpenFlow messages and monitors the network to check the security policies.

An OpenFlow-based SDN architecture to enforce security policies was proposed in [12]. The policy implementation includes the installiation of the flow entries directly in OpenFlow switches/routers and minimizes the flow table size to avoid risks regard the configuration and updating of these switches.

Finally, a theoretical analysis of policy-based network management frameworks was provided in [13]. It examines whether and how the application of such policies on the SDN infrastructure affects the performance of the network in terms of QoS and security.

III. Flow Routing Policy Verification

In this section, we present the flow routing policy methodology that is able to verify that the routing decisions taken by the SDN infrastructure of the IoT network ensure energy, QoS and security efficiency. The workflow of the verification methodology is illustrated in Figure 1. In more detail, we collect real-time metrics from the network of which the topology is formulated as an undirected graph. This information is used for the calculation of routing policies formulated as Key Performance Indicators (KPIs) which estimate how good a routing decision is and they concern energy, QoS and security information. By employing multi-objective optimization, a set of the best solutions (i.e. flow rules) is identified. The set of optimal flow rules is then compared with the flow rules created by the SDN, in order to compare them and provide a deviation metric that is the actual verification result. The verification is considered successful if the deviation similarity metric is within a pre-defined range of values.

A. Network Monitoring

This subsection briefly presents the information collected from the SDN infrastructure which concerns security, QoS and energy consumption information related to the forwarders and the flows of the IoT network.

Every communication and packet transmission in the network consumes some amount of electrical energy, measured in watts. Since each watt consumed costs money, ideally the traffic in the network should be optimized in order to minimize the total energy consumption, and thus, have lower operational costs. In our approach, the energy usage within forwarder per packet (watts/packet) is collected.

The QoS is a quantitative description of the overall performance of the services in a network. It depends in several aspects in the network, such as packet loss, bit rate, transmission delay, etc. In our approach, we use the delays between forwarders when sending cognitive packets from node to node.

Ideally the forwarders should be malware free and the flows should not contain data related to attacks to ensure security. Since this is not a realistic assumption, the flows that are sensitive must be protected from alterations and eavesdropping by avoiding paths that contain low confidence forwarders, while sensitive forwarders must be protected from potential attacks that may disrupt their normal operation. In our approach, we assume that each forwarder has a confidence and a sensitivity value. Confidence represents the (inverse) probability that the forwarder is infected by a malware which affects the integrity and the confidentiality of the flows. High confidence corresponds to low probability of malware (maximum value corresponds to 1.0), while low confidence corresponds to high probability of malware (0.0 is the minimum value). On the other hand, sensitivity measures how sensitive a forwarder is with respect to different aspects of importance. For example, sensitive forwarders include those forwarders that are central in the network that process large amounts of flows, or forwarders which are used to route sensitive (e.g. private/confidential) information.

Concluding, the SDN topology of an IoT network is modeled as an undirected graph, $G = (N, E)$, where N indicates the set of the nodes (n) that represent the forwarders of the SDN and the E is the set of edges (e) that refer to the communication links between two forwarders. Each node n of the graph has the following attributes:

- e_n corresponds to the energy consumption of the node n (i.e. forwarder) in order to process a packet.
- s_n is the the sensitivity of the node n which characterizes how vulnerable a node is. The higher the levels of sensitivity, the lower the selectivity of this node has to be.
- c_n represents the probability of a node n to keep the security properties unviolated. The lower the node confidence is, the more flows should be avoided to travel through this node.

Finally, each edge e of the graph has the d_e attribute which is the delay of the communication between two forwarders.

B. Routing Policies Formulation

The communication between two network connected devices can be done through various alternative paths of forwarders of the SDN infrastructure of an IoT network. A path p corresponds to a flow rule and consists of a number of forwarders n and links between forwarders e, while the set of all the alternative paths is devoted by P, where $p \in P$. This subsection presents four flow routing policies expressed as KPIs that verify that flow rules generated by the SDN concern security, QoS and energy consumption information. In more detail, the following policies must be followed:

1) **Minimize energy consumption**: The energy KPI is defined as
$$J_1(p) = \sum_{n \in p} e_n, \tag{1}$$
where e_n is the energy usage per packet of the forwarder n belonging to the path p. The energy KPI $J_1(p)$ should be as small as possible.

2) **Maximize the network QoS**: The QoS KPI is defined as
$$J_2(p) = \sum_{e \in p} d_e, \tag{2}$$
where d_e is the connection delay of a the communication link e belonging to the path p . The QoS KPI $J_2(p)$ should be as small as possible.

3) **Protect sensitive flows along the paths they follow by avoiding low confidence forwarders**: The flow security violation KPI is defined as
$$J_3(p) = -\sum_{n \in p} c_n, \tag{3}$$

where c_n is the confidence of the forwarder n belonging to the path p. The flow security violation KPI $J_3(p)$ should be as big as possible. Here, the minus is used, since the confidence is inversely proportional with the probability that a forwarder has affected by a malware.

4) **Protect sensitive forwarders from possible malware infections, by avoiding sending low confidence flows through them**: The forwarder security violation KPI is defined as

$$J_4(p) = \sum_{n \in p} s_n, \tag{4}$$

where s_n is the sensitivity of the forwarder n belonging to the path p. The forwarder security violation KPI $J_4(p)$ should be as small as possible.

C. Routing Optimization

In order to optimize the above KPIs simultaneously, our methodology is based on multi-objective optimization approaches. The optimization objectives may be contradicting, i.e. optimizing the value of one objective may be affecting negatively the values of other objectives. In such cases, there are not exist a single trivial solution, but instead the multi-objective optimization identifies a set of optimal solutions. The solutions included in this set are called Pareto optimal [14]. Without additional subjective preferences regarding the significance of the optimization objectives, all the solutions within the Pareto front are considered as equally good, and there is no way to pick a single solution as the best overall one. An example of Pareto optimal solutions is illustrated in 2. In this example, two objectives must be minimized simultaneously, J_1 and J_2. S_F represents the set of all possible solutions and S_P the set of Pareto optimal solutions. Two solutions selected from the Pareto are p_1 and p_2. Solution p_1 has larger J_2 value than p_2, but also smaller J_1 value than p_2.

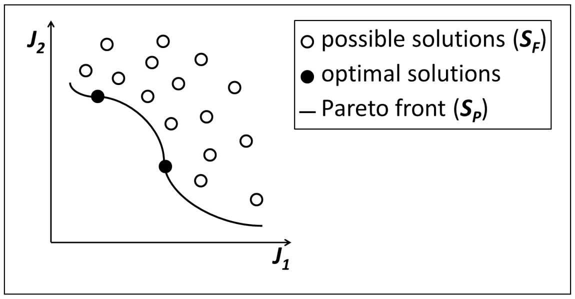


Figure 2. An example of Pareto optimal solutions [14]: Two objectives, J_1 and J_2, must be simultaneously minimized. The set of optimal solutions is represented as S_P. Two solutions from the Pareto set are p_1 and p_2. S_F represents the set of possible solutions.

In the general case (i.e. without knowledge about the significance of the optimization objectives (KPIs)), the multi-objective optimization problem is formulated as follows:

$$\begin{aligned} &\arg\min_{p} && (J_1(p), J_2(p), J_3(p), J_4(p)) \\ &\text{subject to} && J_i^{min} \leq J_i(p) \leq J_i^{max}, \forall i \in [1,4], p \in P, \end{aligned} \tag{5}$$

where P is the possible set of solutions, $J_i, \forall i \in [1,4]$ are the objective functions (equations (1), (2), (3) and (4), respectively) that must be minimized simultaneously and $J_i^{min} \leq J_i(p) \leq J_i^{max}$ are optional constraints that the objectives might have.

In case where the network operator has references regarding the importance of each objective, the unique optimal solution can by found by minimizing the following objective function

$$\arg\min_{p} \alpha\, J_1(p) + \beta\, J_2(p) + \gamma\, J_3(p) + \delta\, J_4(p), \tag{6}$$

where $\alpha + \beta + \gamma + \delta = 1$. The values of $\alpha, \beta, \gamma, \delta$ define the user preference for each policy.

D. Routing Validation

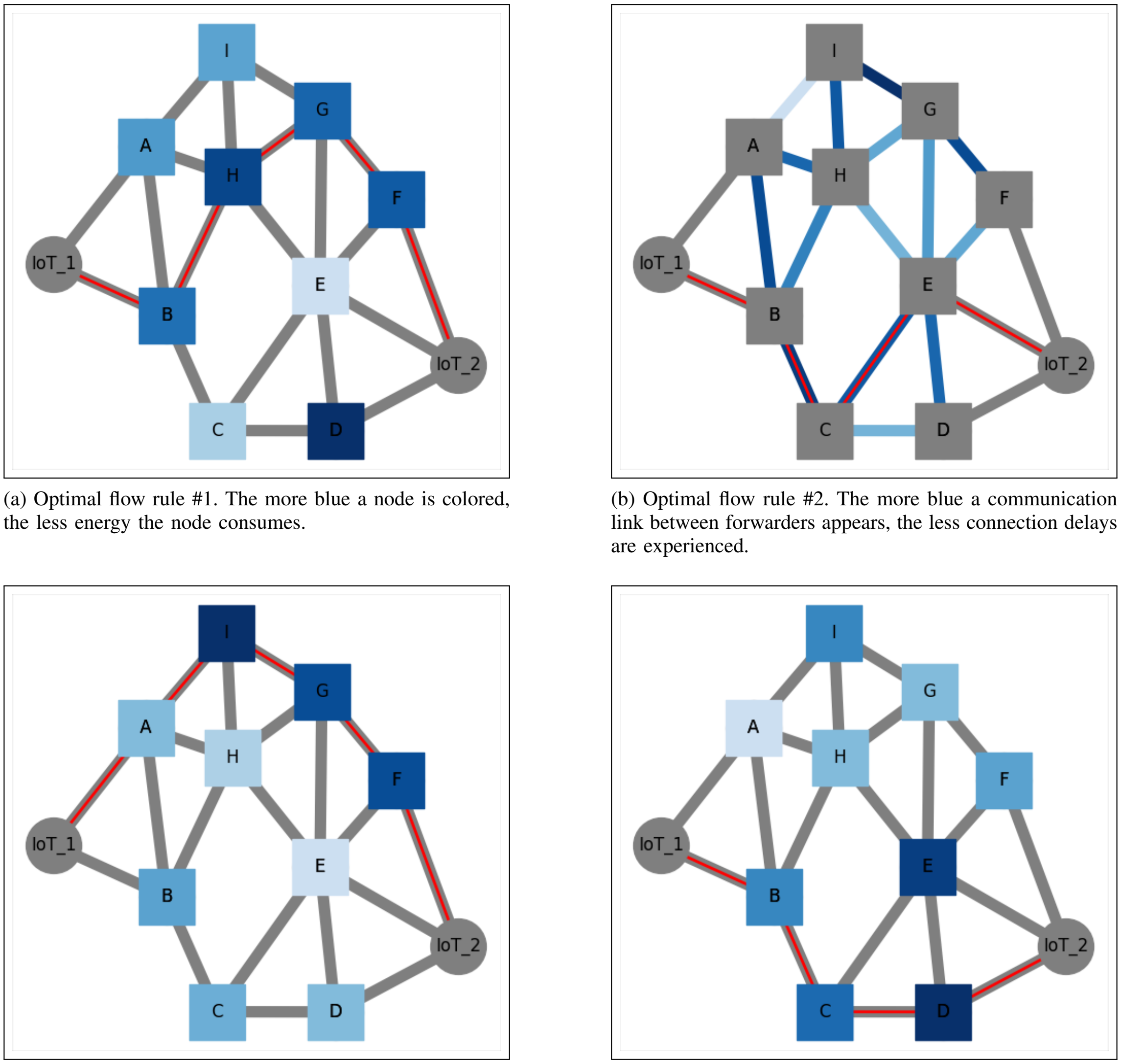
The set of optimal flow rules is compared with the flow rule created by the SDN, in order to verify whether the SDN flow rule is in the optimal solutions set. In case that the solution proposed by the SDN is not in the set of optimal solutions, we conclude that the policy verification failed, and the degree of deviation from the optimal solution is defined as follows:

$$\text{Deviation degree} = \arg\min_{p \in P_{optimal}} \frac{1}{4} \sum_{i=1}^{4} \theta\left((J_i(p) - J_i(p_{SDN})\right)^2 \tag{7}$$

Where J_i is the ith KPI, $P_{optimal}$ is the set of optimal solutions, p_{SDN} is the solution proposed by the SDN, and θ is a normalizing constant utilized in order to make the KPIs have the same scaling. More specifically, the deviation degree is the mean square distance of the SDN solution from the closest solution within the Pareto set. In case that the SDN solution is within the optimal set, then this distance is zero.

IV. Evaluation Scenario

To illustrate the proposed routing policy verification methodology, we applied it in a synthetic representative example. In more detail, we assume that there is a request for communication between two network-connected IoT devices, (i.e. IoT_1 and IoT_2) and the data flow can be routed from a set of forwarder elements. For each forwarder and communication link between forwarders, we randomly assigned values that correspond to network metrics, namely energy consumption (e_n) each forwarder, connection delays (d_e) between forwarders and forwarder sensitivity (s_n) and confidentiality (c_n) values. Using these values, we computed the four KPIs (i.e. equations (1), (2), (3) and (4), respectively) for all the possible paths (i.e. flow rules) between



(a) Optimal flow rule #1. The more blue a node is colored, the less energy the node consumes.

(b) Optimal flow rule #2. The more blue a communication link between forwarders appears, the less connection delays are experienced.

(c) Optimal flow rule #3. The more sensitivity a node has, the more blue the node is colored.

(d) Optimal flow rule #4. The more blue a node appears, the higher confidence it has.

Figure 3. The estimated Pareto optimal solution set for the data flow IoT_1 and IoT_2 consists of 4 flow rules (i.e. paths BHGF, BCE, AIGF and BCD) that are red highlighted, as computed by simultaneously optimizing the four KPIs (i.e. equations (1), (2), (3) and (4), respectively). As can be seen, the proposed routing policy verification methodology is able to define a set with optimal flow rules corresponding to optimal QoS (flow rule #2) or energy usage (flow rule #1), while the flow rules avoid to include forwarder with low confidence (flow rule #4) or high sensitivity (flow rule #3).

IoT_1 and IoT_2 and, then, the Pareto optimal solution set was calculated by equation (5).

The estimated Pareto optimal solution set for the data flow IoT_1 and IoT_2 consists of 4 flow rules that red highlighted in Figure 3, as computed by simultaneously optimizing the four KPIs. Additionally, the forwarder nodes and links are colored according to their efficiency in energy, QoS and flow and forwarder security. More specifically, the sub-figure 3a depicts the optimal flow rule #1 (path BHGF), while the forwarder nodes are colored based on the energy usage within the forwarder. The more blue a node is colored, the less energy this node consumes and thus more energy efficiency is achieved. As can be seen, the path BHGF is energy efficient, since it consists of forwarders with low energy usage. In the sub-figure 3b, the more blue a communication link between forwarders appears, the less connection delays are experienced and, thus, it corresponds to more QoS efficient option. Indeed, the depicted optimal flow rule #2 (path BCE) corresponds to the optimal flow rule from a QoS perspective. In the sub-figure 3c, the more sensitivity a node has, the more blue the node is colored. The optimal flow rule #3 (path AIGF) depicted in this sub-figure indeed corresponds to the optimal path, since it avoids forwarders with high sensitivity. Finally, the sub-figure 3d depicts the last estimated optimal flow rule #4 (path BCD). In this sub-figure, the more blue a node appears, the higher confidence the node has. As can be seen, the path BCD consists of forwarders with high confidence. It should be noted that the last flow rule also corresponds to the case where all the four KPIs have the same weight (equation (6)). Whether the SDN flow rule corresponds to these four solutions, the deviation degree (equation (7)) is zero and we can conclude that the policy verification succeeded.

In conclusion, the proposed routing policy verification methodology is able to define a set with optimal flow rules corresponding to optimal QoS or energy usage, while the flow rules avoid to include forwarder with low confidence or high sensitivity. According to the special demands of the network, the operator can define its own routing policy by accordingly weighting the importance of each KPI by equation (6).

V. Conclusion

In this paper we presented a policy framework for verification of SDN routing decisions of an IoT network, involving security, QoS and energy information to ensure confidentiality, integrity and availability security properties. The policy constraints can be defined by the IoT network operator depending on the demands of the network and the use cases. In future work, evolutionary algorithms will be employed in multi-objective optimization for quick verification of routing policies, while the presented verification frameworks will be evaluated using real scenarios with heterogeneous IoT platforms and devices including domains such as surveillance, intelligent transport systems and flexible manufacturing. Finally, we will study how the objective functions of routing policies can be enriched using more real-time network metrics in order to better represent security, QoS and energy information.

VI. Acknowledgments

This work has been partially supported by the European Commission through project SerIoT funded by the European Union H2020-IOT-2016-2017 (H2020-IOT-2017) Program under Grant Agreement no. 780139. The opinions expressed in this paper are those of the authors and do not necessarily reflect the views of the European Commission.

References

[1] A. Whitmore, A. Agarwal, and L. Da Xu, “The internet of things—a survey of topics and trends,” *Information Systems Frontiers*, vol. 17, no. 2, pp. 261–274, Apr 2015.

[2] L. D. Xu, W. He, and S. Li, “Internet of things in industries: A survey,” *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2233–2243, Nov 2014.

[3] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, “Internet of things for smart cities,” *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, Feb 2014.

[4] S. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K.-S. Kwak, “The internet of things for health care: a comprehensive survey,” *IEEE Access*, vol. 3, pp. 678–708, 2015.

[5] J. Sherly and D. Somasundareswari, “Internet of things based smart transportation systems,” *International Research Journal of Engineering and Technology*, vol. 2, no. 7, pp. 1207–1210, 2015.

[6] S. Bera, S. Misra, and A. V. Vasilakos, “Software-defined networking for internet of things: A survey,” *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1994–2008, Dec 2017.

[7] S. Scott-Hayward, S. Natarajan, and S. Sezer, “A survey of security in software defined networks,” *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 623–654, 2016.

[8] B. B. Rodrigues, A. C. Riekstin, G. C. Januário, V. T. Nascimento, T. C. Carvalho, and C. Meirosu, “Greensdn: Bringing energy efficiency to an sdn emulation environment,” in *IFIP/IEEE International Symposium on Integrated Network Management (IM)*. IEEE, 2015, pp. 948–953.

[9] M. F. Bari, S. R. Chowdhury, R. Ahmed, and R. Boutaba, “Policycop: An autonomic qos policy enforcement framework for software defined networks,” in *IEEE SDN for Future Networks and Services (SDN4FNS)*. IEEE, 2013, pp. 1–7.

[10] A. Lara and B. Ramamurthy, “Opensec: Policy-based security using software-defined networking,” *IEEE transactions on network and service management*, vol. 13, no. 1, pp. 30–42, 2016.

[11] D. Rosendo, P. T. Endo, D. Sadok, and J. Kelner, “An autonomic and policy-based authorization framework for openflow networks,” in *2017 13th International Conference on Network and Service Management (CNSM)*. IEEE, 2017, pp. 1–5.

[12] J. Liu, Y. Li, H. Wang, D. Jin, L. Su, L. Zeng, and T. Vasilakos, “Leveraging software-defined networking for security policy enforcement,” *Information Sciences*, vol. 327, pp. 288 – 299, 2016.

[13] K. Sood, K. K. Karmakar, V. Varadharajan, U. Tupakula, and S. Yu, “Analysis of policy-based security management system in software-defined networks,” *IEEE Communications Letters*, 2019.

[14] K. Deb and D. Kalyanmoy, *Multi-Objective Optimization Using Evolutionary Algorithms*. New York, NY, USA: John Wiley & Sons, Inc., 2001.