Latency Minimization for Energy Internet Communications with SDN Virtualization Infrastructure

Ardiansyah*, Yonghoon Choi†, Muhammad Reza Kahar Aziz‡, Deokjai Choi*
*Department of Electronics and Computer Engineering, Chonnam National University, South Korea
†Department of Electrical Engineering, Chonnam National University, South Korea
‡Department of Electrical Engineering, Institut Teknologi Sumatera, Indonesia
Email: ardi@ejnu.net, {yh.choi, dchoi}@jnu.ac.kr, reza.kahar@el.itera.ac.id

***Abstract*—Software-defined networking (SDN) technology is expected to be utilized to link energy stakeholders in a way that encourages active participation in building the energy internet (EI) ecosystem. However, EI is a very complex system with various production and non-production business applications that have specific and strict functional requirements. Hence, network service chaining is indispensable to be implemented. Currently, SDN can be combined with network function virtualization (NFV) technology, and they become SDN virtualization infrastructure. NFV has a role in providing service function chaining (SFC) on consolidated middleboxes, and SDN has a position as a glue between those functions. In this paper, we present our works to provide real-time capable EI communications based on SDN virtualization infrastructure. We address our problem as an NFV middleboxes placement strategy to minimizes flow-network latency subject to the middleboxes processing power capacity and the switch resources constraint. We investigate the existing middleboxes placement approaches and propose a flow-network partitioning algorithm as our heuristic solution. The result shows that our approach could improve latency minimization significantly. The average flow-latency can reach 20.19% and 7.10% lower than the baseline approach in two network topologies. We believe our works may further help on realizing flexible and real-time capable EI communication network infrastructure.**

***Index Terms*—Energy internet, latency minimization, middleboxes placement, NFV, SDN**

I. Introduction

Smart grid technological advancements bring opportunities to transform the current power system to energy internet (EI), an internet business model of energy utilization containing all novel phases of energy generation, transmission, storage, and distribution [1]–[3]. In the EI system, distributed renewable energy resources (DRER) and distributed energy storage devices (DESD) can be joined flexibly and seamlessly to the power network. It considerably improves the proportion of renewable energy usage and reduces greenhouse gas emissions. Furthermore, using a solid-state transformer or known as energy-router, the energy can be exchanged and transferred bidirectionally between one node to another node. It brings more effective energy utilization for both local microgrid and the global power grid. An energy sharing economy concept or energy e-commerce can be realized shortly, in which energy consumers can obtain the supplies directly from the nearest producers [4]. The cascading failure (blackout) could also be resisted by an extensive interconnection of the future renewable electric energy delivery and management (FREEDM) system, which improves the stability of power systems.

In recent years, EI has attracted increasing attention of government and institutions in many countries. The Chinese government and the state grid corporation of China (SGCC) launches the global energy internet project across the country power networks [5]. Moreover, as a response to the Fukushima crisis, a diverse group of large Japanese firms is starting to explore this internet-inspired electricity grid [6]. They believe, the energy internet can be an ultimate solution to reorganize the country's power system, which shifts from centralized to decentralized control with various energy resources. Moreover, the EI system is the most promising solution for rural electrification in a country with more than 10,000 islands like Indonesia [7]. However, due to the diverse and rigorous requirements of reliability, security, latency, and flexibility, many smart grid technologies and applications are needed to be developed for EI ecosystem.

The core task of EI construction is the building of open platforms for implementing end-to-end interactions across the entire value chain. Thus, software-defined networking (SDN) technology is expected to be utilized. The initial adoption of SDN in EI ecosystem has been presented in [4], [8]. As depicted in Fig. 1, the SDN-based EI communication network designed hierarchically from microgrid cluster zone level to the global grid network. By applying this hierarchical architecture, we can manage applications data-flow from a high-level view. However, the existing works only focus on the underlying network implementation, by merely comparing the traditional internet protocol (IP) network with the SDN-based EI communication network. Whereas, EI is a very complex system with various production and non-production business applications that have specific and strict communication network requirements [9]. Hence, network service chaining implementation such as application-level optimization, network security applications, and traffic analysis element is inevitable.

Most recently, network function virtualization (NFV) technology allows us to provide network service chaining in a consolidated middlebox [10]. In different with the traditional middlebox, the consolidated middlebox is a set of service function chaining (SFC) which is provided virtually at a single machine. Hence, SDN and NFV can be combined, and they become an SDN virtualization infrastructure. NFV has a role in providing SFC requested by applications, and SDN has a position as a glue between those functions provided by the NFV. Some existing works have proposed the use of NFV consolidated middlebox in SDN virtualization infrastructure [10], [11]. They present their work by placing consolidated middleboxes with several flow routing schemes. However, no matter what flow routing scheme is used, the performance of SFC always depends on where these middleboxes are placed [12].

In this paper, we present our works on latency minimization for real-time capable EI communications using SDN virtualization infrastructure. By addressing a middleboxes placement strategy, our goal is to find all middleboxes location that minimizes flow-network latency meanwhile optimally fulfilling the SFC requirements. Similar to the facilities layout problem, a middleboxes placement problem is known as an NP-hard, which is complex and time-consuming decision problem [12]–[14]. The main contribution of this paper is as follows:

1) At first, we investigate some existing NFV middleboxes placement approaches to find intuitive properties regarding objective functions and constraints.
2) We then formulate our objective function to realize real-time capable EI communications with SDN virtualization infrastructure, which is vital for the most EI applications. This problem has two constraints, i.e., the middlebox SFC processing power constraint and the OpenFlow-switch resource constraint.
3) We propose a flow-network partitioning algorithm as our heuristic solution. This solution is based on a graph clustering analysis approach, in which the observations start in one cluster, and splits are performed recursively until flow-network with a set of ingress-switches which are close to each other share the same middlebox in a cluster.
4) We evaluate our approach along with baseline methods on two communication network topologies, i.e., Abilene and FatTree topologies.

We believe that this works may further help on realizing flexible and real-time capable EI communication infrastructure.

The rest of this paper is organized as follows. In the next section, we provide a brief description of the energy internet system, a summary of EI applications, and their functional requirements. Section III presents our investigation on existing approaches to provide network service chaining in the SDN virtualization infrastructure. Section IV describes the system model and problem formulation. Section V presents our proposed method to solve the problem and preliminary analysis. Finally, section VI concludes this paper.

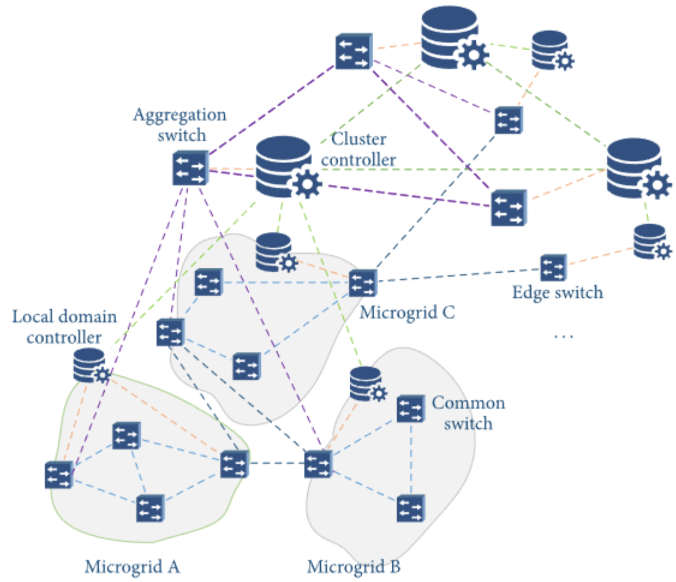

Fig. 1. SDN-based energy internet communication network [3].

II. BACKGROUND

A. Energy Internet System

Energy internet is expected to involve a large number of energy customers with locally operating DRER and DESD. This considerably can improve the efficiency of energy utilization with the combination of self-production power and utility-owned power distribution [3]. To accommodate the high penetration of DRER and DESD, there are some pivotal technologies in FREEDM system development [1], such as

- A plug and play interface: allows DRER and DESD to be inserted or removed at any time and anywhere seamlessly, similar to a USB port of a computer.
- An intelligent energy management software: helps to calculate and analyze the price information and the availability of DRER and DESD in real-time.
- A distributed grid intelligence software: helps to distribute control the migration of power between load demand, renewable energy generation, and energy storage.
- The solid-state transformer or energy router: this is the most critical component in the FREEDM system. Like the internet router, energy router is used to bidirectionally exchange the energy between one node to another node in the energy internet.
- An internet equipment management device: employed to provide real-time monitoring on the loads and other operating devices.
- A fault isolation device: intelligent fault management to maintain system stability.

B. Functional Requirements

W. Zang *et al.* [9] describes that in general, there are two types information-flow in energy internet, i.e., production and non-production business applications as depicted in Fig. 2.

The production business applications include various smart grid applications for energy internet operation control and information. Non-production business applications use for administration and management such include internet access, audio/video conference, office automation, and so on. It is also predicted that several new business application will be coming soon. For example, energy e-commerce in a residential distribution system [4], in which energy customers can get supplies from the closest producers in peer to peer manner. Blockchain technology gives a promising solution to enable a reliable transaction for this ICT based energy trading [15].

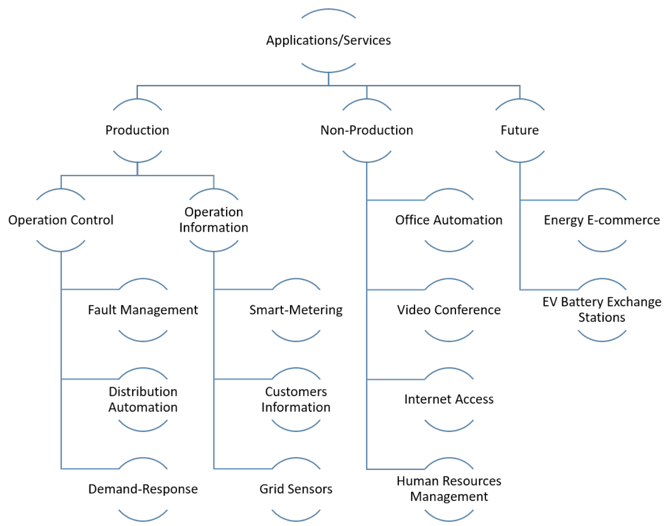

Fig. 2. Example applications in energy internet.

In recent years, the national institute of standards and technology (NIST) and the open smart grid (OpenSG) network task force has been analyzing functional requirements of various smart grid applications. Currently, more than 1400 application data flows have been specified with detail functional requirements such as the payload type, payload size, frequency of data transmission, security, latency, required reliability, daily clock periods, and so forth [16]. Some functional requirements of the essential applications are listed in Table I. Many applications, including distribution automation, load control signaling, and outage alarming, are described to require a real-time capable communication network, while other applications, such as smart metering, are more latency-tolerant [8], [17]. Moreover, the North American synchrophasor initiative network (NASPInet) has also contextualized synchrophasor application data quality in wide-area situational awareness systems (WASA) [18]. They determine the fitness of use for two types of data, i.e., static data and streams data. In terms of completeness, accuracy and logical consistency for static data and in terms of timeliness, availability, and origination for streams data. Low communication latency is required to avoid lousy data quality [19]. Furthermore, the latency minimization is an important factor for the reliability requirement, as in

$$P_f + P_s \int_L^{\infty} f_{D|s}(t)dt \leq 1 - R, \text{ or } P_s \int_0^L f_{D|s}(t)dt \geq R, \tag{1}$$

where P_f is the probability that the application payload is lost somewhere in transit, $P_s = 1 - P_f$ is the probability that the payload reaches the destination node, L is the application latency requirement, R is the reliability requirement, and $f_{D|s(t)}$ is the delay density of a payload, conditioned on successful delivery. The reliability requirement is defined so that the successful delivery of the application data flow with a delay that is greater than the threshold is still considered as a failure [20]. Hence, to support all the mentioned functional requirements above, a flexible and real-time capable communication network is indispensable.

TABLE I
FUNCTIONAL REQUIREMENTS OF PRODUCTION-BUSINESS APPLICATIONS.

Application Type	**Functional Requirements**		
	Bandwidth	***Reliability***	***Latency***
Substation Automation	9.6-56kbps	99.99%	15-200ms
WASA	600-1500kbps	99.99%	15-200ms
Smart Metering	100-500kbps	99.0%	2000ms
Demand Response	14-100kbps	99.0%	500ms
Outage Management	56kbps	99.0%	2000ms
Distribution Automation	1000kbps	99.99%	20-200ms
DRER and DESD	9.6-56kbps	99.0%	200-2000ms

III. SDN VIRTUALIZATION INFRASTRUCTURE

A. Network Service Chaining

Through a higher level abstraction, SDN can specify network service chaining without the need for reconfiguring low-level settings at each of the forwarding nodes [21]. The number of flow classes and service policies are unrestricted, allowing for fine-grained traffic control based on the user needs, enabling the intent-based or user-defined network infrastructure [22]. For SDN-enabled EI communications, the number of policies can be defined as much as the number of application data flows if necessary. An SDN-controller has the task to apply all service policies correctly to the underlying network resources. In this regard, two main aspects need to be considered: 1) providing service policies for specific stakeholder or business entity flow; 2) providing service policies for a particular application data flow.

Many approaches have been proposed to provide quality of services (QoS) guarantee, but they effectively devolve into two categories, i.e., resource reservation approach, and per-flow routing approach [10], [11]. Resource reservation approach is the most common solution. Typically, this approach consists of two modules: a flow classifier module and a rate shaper tool. The classifier read the network packet's fields and assign a specific priority to each flow based on policies defined in the controller. The rate shaper then set up resource reservation rules in an OpenFlow-enabled switch based on the classification. Similar to the resource reservation approach, per-flow routing approach also differentiate flows through the classifier.

However, instead of reserving resources for each flow slice at each forwarding device, the high priority flows will be put dynamically on QoS guaranteed routes by the controller. At the same time, the least priority flows remain on their usual routes. In this approach, to ensure that the routing path is optimal, a constraints shortest path (CSP) has been formulated [23], as

$$r^* = \arg\min_{r}\{f_C(r)|r \in R_{st}, f_D(r) \leq D_{\max}\}, \tag{2}$$

that is, finding a forwarding route r from set of all routes R_{st} that minimizes the objective function $f_c(r)$ such that the delay variation $f_D(r)$ to be less than or equal to the threshold value D_{max} [23], [24]. The constraints could be varied, ranging from bandwidth consumption, deployment cost, energy consumption, and so on [21].

B. Service Function Chaining using NFV Middleboxes

Some existing works have proposed flow routing schemes in the SDN virtualization infrastructure [10], [11]. By taking service policies as inputs, service function chaining in NFV middleboxes fulfilling the functional requirement, meanwhile the optimal paths of all policies are selected, taking account of the defined constraints. However, no matter what flow routing scheme is used, the middleboxes placement gives the most significant impact on the network service chaining performance [12].

By assuming that middleboxes can be moved easily, some middleboxes placement approaches with different goals has been proposed [12]–[14]. For example, [12] formulated two objective functions, i.e., delay minimization and bandwidth consumption minimization, as depicted in (3) and (7) respectively, as

$$\min \quad D^{tot}, \tag{3}$$

$$s.t. \quad \sum_{\forall s_l \in \mathcal{S}} x_{i,l} = 1, \forall q_i \in \mathcal{Q}, \tag{4}$$

$$\sum_{\forall q_i \in \mathcal{Q}} R(q_i)x_{i,l} \leq C(s_l), \forall s_l \in \mathcal{S}, \tag{5}$$

$$x_{i,l} = 0, \forall q_i \in \mathcal{Q}, \forall s_l \in \mathcal{S} \backslash S_i, \tag{6}$$

$$\min \quad B^{tot}, \tag{7}$$

$$s.t. \quad (4) - (6), \tag{8}$$

where, D^{tot} and B^{tot} are the total end-to-end delay and the total bandwidth consumption of all service policies, $x_{i,l}$ is the binary variables to represent middlebox placement scheme, $R(q_i)$ is the required resource to deploy middlebox $q_i \in Q$, $C(s_l)$ is resource capacity of each switch S to deploy middleboxes. For both delay and bandwidth consumption minimization problem we have three constraints. Constraint (4) to guarantee that each middlebox should be successfully deployed at one location, where $x_{i,l}$ = 1 denotes that middlebox q_i is connected to switch S_l, otherwise $x_{i,l}$ = 0. Constraint (5) to guarantee that the total resource demand for middleboxes deployment at one location should not exceed the resource capacity. Constraint (6) to accommodate for middleboxes that can only be placed in certain places. It is considering that a middlebox may require a power supply and acceleration by some dedicated platforms, which are available only at some locations.

Table II summarizes the objective function as well as the constraints of each existing approaches. As mentioned in the previous section, the real-time capable network is inevitable for EI communications. Thus, in the next chapter, we formulate a middleboxes placement strategy for latency minimization.

TABLE II
CHARACTERISTICS OF MIDDLEBOX PLACEMENT STRATEGY.

Approach	***Objective***	***Constraints***
[12]	Minimize delay & bandwidth	Network resources capacity
[13]	Minimize cost & bandwidth	Traffic-chaining ratio
[14]	Minimize energy consumption	Delay & bandwidth

IV. SYSTEM MODEL AND PROBLEM FORMULATION

A. System Model

Let the SDN virtualization infrastructure is represented as a simple directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of edges (also called links). The number of policies that are stored in a switch $\mathcal{S} \in \mathcal{V}$ is denoted as p_s, and the maximum number of policies can be stored in a switch is P_s. If the set of all service function chaining is denoted as $\mathcal{C}$, then a middlebox b which supplies those function as $\mathcal{C}_b \subset \mathcal{C}$. There may be $\mathcal{N}$ number of middlebox may be available in the network, thus let us denoted $\mathcal{N}_b \in \mathcal{V}$ as the number of middlebox type b. Each middlebox has a maximum processing power capacity $\mathcal{O}_b$ that can be provided. This processing power capacity depends on the available CPU in each middlebox, which is represented in Mbps unit [14].

If $source_k$, and $dest_k$ are the source and the destination node, and the c_k is the must be visited services of a flow's network traffic from source to destination. Then a flow f can be described as $f_k = \{source_k, dest_k, c_k, o_k\}$. Where o_k is the amount of middlebox processing power capacity occupies by a flow. With the knowledge of all information in advance, we can generate F_b, a set of flows which require services from a type b middlebox.

B. Problem Formulation

In this paper, we define our goal to minimize total flow-network latency; meanwhile, NFV middleboxes fulfilling the functional requirements. Let us define total flow-network latency, as

$$D_f^{tot} = \sum_{\forall o_k} \sum_{\forall v_{ij} \in \mathcal{V}} d_{if,bf} + \sum_{\forall o_k} \sum_{\forall v_{ij} \in \mathcal{V}} d_{bf,ef}, \tag{9}$$

where, $d_{if,bf}$ is the aggregate flow-network latency from source ingress-switch to the corresponding middlebox and

$d_{bf,ef}$ is the aggregate flow-network latency from the corresponding middlebox to the destination egress-switch. Flow-network latency depends on the packet delivery time and the service processing delay. The packet delivery time in each link between two nodes is depicted as

$$d_{v_i,v_j} = \frac{Z_{max}}{B_r} + \frac{Xv_i, v_j}{L_s}, \tag{10}$$

where, Z_{max} is the maximum packet size in bit, B_r is the transmission bit rate in bit/s, Xv_i, v_j is the distance or the length of transmission medium in meter, and L_s is the ratio of actual propagation speed to the speed of light of the medium in m/s. To the best of our knowledge, the propagation speed depends on the physical medium of the link, e.g., 2 x 10^8 m/s for copper wires and 3 x 10^8 for wireless communication.

We need to find all middleboxes location that minimizing the total latency of each flow $f \in Fb$ as follows,

$$\min \; D_f^{tot} \; \forall f \in F_b \;\; \text{according to (9)}, \tag{11}$$

$$s.t. \quad p_s \leq P_s, \; \forall \mathcal{S} \in \mathcal{V}, \tag{12}$$

$$\sum_{f_i \in F_b} o_{f_i} \leq \sum_{j=1}^{b} O_{b_j}, \; \forall N_b \in \mathcal{V}. \tag{13}$$

In this problem, we have two constraints, i.e., Constraint (12) and (13). Constraint (12) is the OpenFlow switch resource constraint, which used to confirm that a switch has available memory for storing new policy entries. Constraint (13) is the service chaining constraint, to ensure that a corresponding middlebox has enough processing power capacity to process the network service function requested by network-flows.

V. Proposed Method

A. Heuristic Solution

We design a heuristic solution based on two intuitive properties. First, inherent property derived from [12], that is, it is better to put a middlebox as near as possible to ingress-switch of its corresponding flows. Second, our intuitive belief, that is, most applications in EI running regionally to improve power grid operational efficiency and customers' satisfaction. Thus, it may better to divide our network such that the network-flows with a set of ingress-switches that are close to each other to share the same middlebox in a cluster. Therefore, we develop a flow-network partitioning method, as expressed in Algorithm 1. Adopting the basic idea of [25], we construct our method with some additional consideration as follows

1) Since the objective of the partition is to minimize the latency. The cluster center initialization method plays a significant role. Randomly choosing initial center will not guarantee that the packet delivery time between the cluster center and other nodes in the subnetwork to be shortened.
2) The recalculated cluster center should be selected from $\mathcal{S}$ ($K_c \in \mathcal{S}$) that guarantee there is a physical connection between the control center and other nodes.
3) The distance calculation method need to accommodate indirect connection between nodes which may not physically be connected.

Let a cluster K is composed of the ingress-switches of corresponding flows, that is $K = (s_1, s_2, ..., s_i)$, where s_f is the ingress-switch of a flow f. If $\mathcal{S}_b \in S$ is the set of ingress-switches of corresponding flows in F_b, then to determine the packet delivery time between each pair of switches, we calculate the shortest path (SP) delay time between them, as

$$d_{s_i,s_j} = d_{SP(s_i,s_j)}, \text{for } \forall \; s_i, s_j \in \mathcal{S}_b, \tag{14}$$

Using (14), we determine the packet delivery time thresholds Θ for each cluster as follows

$$\Theta_b = min \; d_{s_i,s_j} + \frac{max \; d_{s_i,s_j} - min \; d_{s_i,s_j}}{N_b} \; \forall \; s_i, s_j \in \mathcal{S}_b. \tag{15}$$

Algorithm 1 Flow-Network Partitioning Algorithm.

Input: $G = (S, E)$, F_b, $S_b \in S$
Output: Set of switches in K clusters

1: **Step 1**: Compute the shortest path $d_{SP(s_i,s_j)}$
2: **Step 2**: Initialize cluster $K_0 = s_0$ and select one node from $S_b \in S$ as the first initial center ($K_c(0)$) of G
3: **Step 3**: Calculate packet delivery time from s_i to existing cluster center as d_{s_i,K_c}.
    If $d_{s_i,K_c} \geq \Theta_b$ **then**
        create a new cluster to contain s_i,
    else $K_c = K_c \cup s_i$
4: **Step 4**: Update cluster centre $K_c^{'}$ to find the closest switch to each obtained cluster, where the sum of shortest path delay time to reach all ingress-switches in a cluster is minimized.
5: **Step 5**: Calculate p_s for each switch $s_i \in K$
    If $\exists s_i \in K$ *such that* $p_s \geq P_s$ **then**
        s_i unqualified, continue to s_{i+1},
6: **Step 6**: Repeat steps 3, 4, 5 until the flow-network is partitioned into optimal K subnetworks.

B. Preliminary Analysis

We implement the SDN virtualization infrastructure testbed in two separate machines, one for SDN controller, and the rest for a mininet emulator. Each machine has 3.40 GHz eight-core CPUs and 8192 MB RAM. We decide to deploy two network topologies, i.e., Abilene and FatTree topology, to represent two possible communication network topology in EI operation. The characteristics of these network topologies are summarized in Table III.

We set our testbed with several assumptions as follow. First, the links packet delivery time in each network topology is randomly assigned following the normal distribution. Second, there are five NFV middleboxes in each simulation. We assume

TABLE III
THE CHARACTERISTIC OF NETWORK TOPOLOGIES.

Topology	***FatTree***	***Abilene***
S	20	11
E	32	14

that each middlebox could provide five kinds of network services, and there are a different number of flows that randomly request a specific function. The summary of the parameters setting is described in Table IV.

TABLE IV
PARAMETER SETTING FOR THE PERFORMANCE EVALUATION.

Parameter	***Value/Range***
N_b	5
C_b	5
F_b	30, 40, 50
o_k (Mbps)	From 0.1 to 1
$P(s)$	25
Z_{max} (bytes)	1500
B_r (Mbps)	100
L_s ratio (m/s)	from 0.6 to 0.9 * 2 x 10^8
Xv_i, v_j (m)	from 0.4 to 0.8

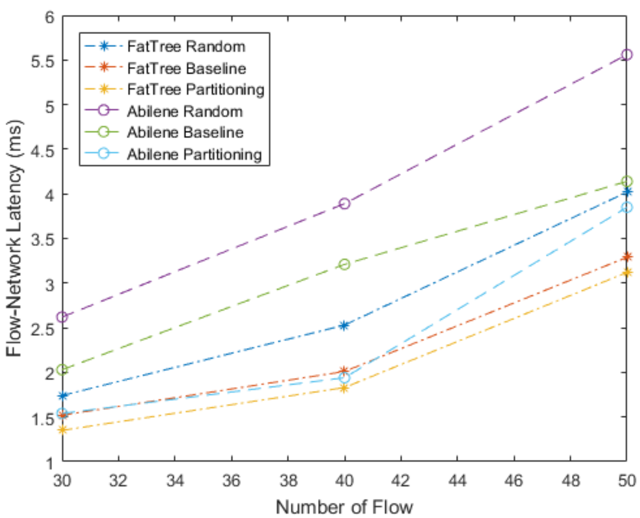

Fig. 3. Average of flow-network latency.

Fig. 3 depicts the average of flow-network latency in both simulated network topologies. The results show that in both network topologies, the flow-network partitioning gives an enormous impact on reducing flow-network latency compare to the baseline placement approach. Compared to the [12], our approach improves latency minimization around 20.19% and 7.10% in the Abilene and FatTree topology, respectively. These results prove our intuitive properties as mentioned in the previous subsection, that is, with the knowledge of flow information in advance, it is better to put a middlebox as near as possible to the set ingress-switches of its corresponding flows and group them into one cluster.

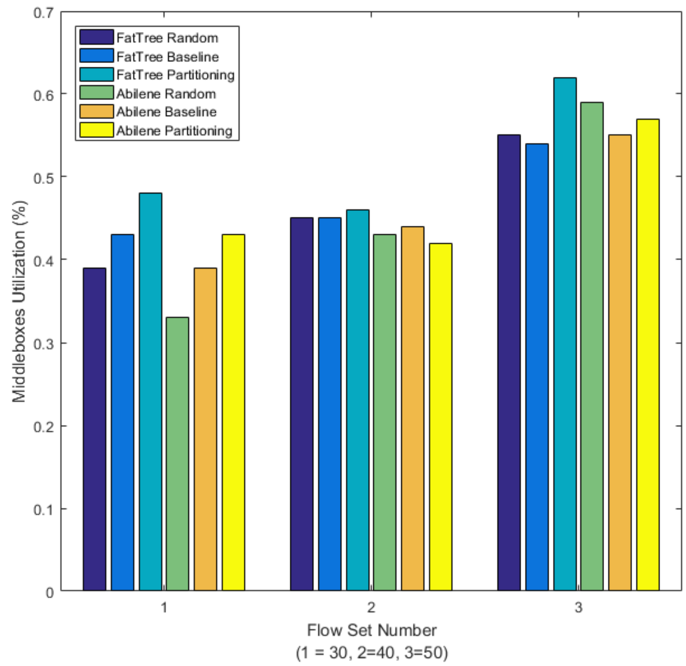

Fig. 4. Average middleboxes utilization.

Fig. 4 shows the average of middleboxes resource utilization. Based on these results, we could not find a direct relation between middleboxes placement strategy to resource utilization. We can only conclude that utilization increases along to the increasing number of flows. As mentioned before, the middlebox power capacity depends on the available CPU in each middlebox. Thus, considering our constraint in (13), we can adopt CPU bandwidth control as proposed in [26]. However, the utilization control will effect to the service processing latency.

VI. CONCLUSION

In this paper, we introduce SDN virtualization infrastructure to fulfill the latency requirement of various applications in energy internet. However, how to place NFV middleboxes into SDN-enabled network is a big challenge. By modeling the flow-network partition problem, we propose our heuristic solution to minimize total flow-network latency that is vital for the most applications. Our preliminary results have verified that flow-network partitioning could improve latency minimization significantly. The average flow-latency can reach 20.19% and 7.10% lower than the baseline approach in two network topologies. Even though the main objective of this paper is latency minimization, more targets such as security awareness and energy saving can be implemented fulfilling all functional requirements in the future.

REFERENCES

[1] K. Wang et al., "A survey on energy internet: architecture, approach, and emerging Technologies," IEEE Systems Journal, vol. 12, no. 3, pp. 2403-2416, Sept. 2018.

[2] J. C. Prado, W. Qiao, L. Qu, and J. R. Aguero, ”The next-generation retail electricity market in the context of distributed energy Resources: Vision and integrating framework,” Energies, vol. 12, no. 3, pp. 491, Feb. 2019.
[3] Z. Lu, C. Sun, J. Cheng, Y. Li, Y. Li, and X. Wen, SDN-enabled communication network framework for energy internet, Journal of Computer Networks and Communications, vol. 2017, 2017.
[4] F. Li, R. Li, Z. Zhang, M. Dale, D. Tolley and P. Ahokangas, ”Big data analytics for flexible energy sharing: accelerating a low-carbon future,” IEEE Power and Energy Magazine, vol. 16, no. 3, pp. 35-42, May-June 2018.
[5] Z. Liu, Global energy interconnection. Beijing: China Electric Power Press, 2015.
[6] J. Boyd, ”An internet-inspired electricity grid,” IEEE Spectrum, vol. 50, no. 1, pp. 12-14, Jan. 2013.
[7] M. E. Khodayar, ”Rural electrification and expansion planing of off-grid microgrid,” The Electricity Journal, vol. 30, no.4, pp. 68-74, 2017.
[8] W. Zhong, R. Yu, S. Xie, Y. Zhang and D. H. K. Tsang, ”Software defined networking for flexible and green energy internet,” IEEE Communications Magazine, vol. 54, no. 12, pp. 68-75, December 2016.
[9] W. Zhang, J. Li, J. Zhou, and Z. Hu, ”The requirement and the key technologies of communication network in internet of energy,” Lecture Notes in Computer Science, Vol 9567, pp. 842-848, 2016.
[10] Y. Li and M. Chen, ”Software-defined network function virtualization: a survey,” IEEE Access, vol. 3, pp. 2542-2553, 2015.
[11] A. Gushchin, A. Walid, and A. Tang, Scalable routing in SDN enabled networks with consolidated middleboxes, Proc. ACM SIGCOMM Workshop on Hot Topics in Middleboxes and Network Function Virtualization, HotMiddlebox 15, pp.5560, 2015.
[12] J. Liu, Y. Li, Y. Zhang, L. Su and D. Jin, ”Improve service chaining performance with optimized middlebox placement,” IEEE Transactions on Services Computing, vol. 10, no. 4, pp. 560-573, 1 July-Aug. 2017.
[13] Y. Chen and J. Wu, ”NFV middlebox placement with balanced set-up cost and bandwidth consumption,” Proceedings of the 47th International Conference on Parallel Processing, 2018.
[14] M. A. Raayatpanah and T. Weise. ”Virtual network function placement for service function chaining with minimum energy consumption,” Proceedings of the 2018 IEEE International Conference on Computer and Communication Engineering Technology (CCET’18), Beijing, China, August 18-20, 2018.
[15] A. Musa, G. D. Nugraha, D. Choi, ”A design of blockchain framework for the P2P energy system,” Proceedings of the KNOM Conference 2018, Jeju Island, South Korea, 2018.
[16] T. Sato, et al., ”Smart grid standards: specifications, requirements, and technologies, Singapore: John Wiley and Sons, 2015.
[17] K. Wang, X. Hu, H. Li, P. Li, D. Zeng and S. Guo, ”A survey on energy internet communications for sustainability,” IEEE Transactions on Sustainable Computing, vol. 2, no. 3, pp. 231-254, 1 July-Sept. 2017.
[18] A. Sundararajan, T. Khan, A. Moghadasi, A. I. Sarwat, ”Survey on synchrophasor data quality and cybersecurity challenges, and evaluation of their interdependencies,” Journal of Modern Power Systems and Clean Energy, pp. 1-19, 2018.
[19] B. Yang, K. V. Katsaros, W. K. Chai and G. Pavlou, ”Cost-efficient low latency communication infrastructure for synchrophasor applications in smart grids,” IEEE Systems Journal, vol. 12, no. 1, pp. 948-958, March 2018.
[20] D. Griffith, M. Souryal and N. Golmie, ”Wireless networks for smart grid applications,” Smart Grid Communications and Networking, Cambridge: Cambridge University Press, pp. 234-262, 2012.
[21] M. Karakus, A. Durresi, ”Quality of service (QoS) in software defined networking (SDN): a survey,” Journal of Network and Computer Applications, 2017.
[22] A. Musa et al., ”Aggregation management design for user-defined network infrastructure,” Proceedings of The 16th Asia-Pacific Network Operations and Management Symposium, Hsinchu, pp. 1-4, 2014.
[23] H. E. Egilmez, S. T. Dane, K. T. Bagci and A. M. Tekalp, ”OpenQoS: An OpenFlow controller design for multimedia delivery with end-to-end quality of service over software-defined networks,” Proceedings of The 2012 Asia Pacific Signal and Information Processing Association Annual Summit and Conference, Hollywood, CA, pp. 1-8, 2012.
[24] Z. Shu et al., ”Traffic engineering in software-defined networking: Measurement and management,” in IEEE Access, vol. 4, pp. 3246-3256, 2016.
[25] T. D. Vu and K. Kim, ”Flow clustering based efficient consolidated middlebox positioning approach for SDN/NFV-enabled network,” IEICE Transactions on Information and Systems, August 2016.
[26] P. Turner, B. R. Bharata and Nikhil Rao, ”CPU bandwidth control for CFS” Proceedings of Linux Symposium, p. 245. 2010.