Sustainable Strategy for Recycling Edge Devices in Internet of Everything Networks

Weigang Hou
School of Computer Science and Engineering
Northeastern University
Shenyang, China
houweigang@cse.neu.edu.cn

Zhaolong Ning
School of Software
Dalian University of Technology
Dalian, China
zhaolongning@dlut.edu.cn

Lei Guo
School of Computer Science and Engineering
Northeastern University
Shenyang, China
guolei@cse.neu.edu.cn

ABSTRACT

Internet of Everything (IoE) devices have different operation principles, which weakens the network scalability and data interoperability. Virtualization is an economic way of solving this problem. The data—collected by different vendors' sensors—can share the same computing program encapsulated by the Virtual Machine (VM), thus neglecting the physical-layer difference. To eliminate the extreme cost and long delay of transferring VMs to the remote cloud, the Edge Device (ED) preliminary processes its running VMs. Currently, the recycling of IoE devices become a major dilemma for individuals, since it is not simply a matter of concern for environmental damage or a solution to an environmental problem. Therefore, the sustainable strategy for recycling EDs is an important way to safeguard the network sustainability. To improve the recycling efficiency, most of the EDs should be upgraded simultaneously during one batch by migrating their running VMs to others for the service continuity. We investigate the least upgrade batch for recycling EDs in IoE networks. A two-step algorithm called MSBP (Minimized upgrade batch VM Scheduling and Bandwidth Planning) is designed to minimize the number of upgrade batches. Because migrating VM brings the bandwidth consumption along trajectories, MSBP has two strategies—Shortest Trajectory First (STF) and Least Bandwidth Utilization First (LBUF)—of allocating bandwidth and trajectories. The simulation results show that: 1) MSBP has the optimal recycling efficiency (least number of upgrade batches) for EDs; 2) LBUF more effectively mitigates the phenomenon where VM migration trajectories compete for the common link bandwidth, thus achieving a lower negative impact of path contention level on the recycling efficiency; 3) the battery power is not exhausted for the ED functioned as the sensor head of data transferring, thus prolonging the network lifetime. In summary, our solution well improves the network, social, economic and ecological sustainability.

WWW 2017, April 3–7, 2017, Perth, Australia.
ACM 978-1-4503-4914-7/17/04.
http://dx.doi.org/10.1145/3041021.3054770

Keywords

IoE network sustainability; edge devices recycling

1. INTRODUCTION

The Internet of Everything (IoE) has a rapid development with the exponential growing requirement of connecting the real world to the Internet [1, 2]. The IoE network has two layers: sensor layer and Edge Device (ED) layer. In the sensor layer, the number of sensors is predicted to reach 50 billion in the next few years [3], and a vast number of sensors have different operation principles, thus resulting in challenging issues of network scalability and data interoperability. For example, the sensors—owned by different vendors—have various protocol stacks restricting the inter-sensor data operability. To solve the aforementioned problems, the virtualization is an economic way [4]. By virtualization, as shown in Fig. 1, the data—collected by different vendors' sensors—can share the same computing program encapsulated into one Virtual Machine (VM), provided that the data has a similar sensitivity level. In the ED layer, followed by edge computing [5], each ED preliminary processes VMs encapsulating computing programs tailored to the collected data of the same industry, which eliminates the extreme cost and long delay of transferring VMs to the remote cloud. Moreover, the ED also can dynamically schedule its running VMs [6, 7]through the consultation with other EDs. For example, in Fig. 1, the VMs 1-2 are migrated to ED 2 if ED 1 becomes inoperative later for one routine activity such as upgrade for recycling.

1.1 Motivations

Currently, the recycling of IoE devices become a major dilemma for individuals, since it is not simply a matter of concern for environmental damage or a solution to an environmental problem [8]. For IoE networks, the ED is a key device of processing VMs [9]. Therefore, the sustainable strategy for recycling EDs is an important way to safeguard the IoE network sustainability. To improve the recycling efficiency, multiple EDs should be upgraded simultaneously through one batch, but it disables the VM process provided by the EDs under upgrade. For the service continuity, some VMs may be migrated from the ED under upgrade to other

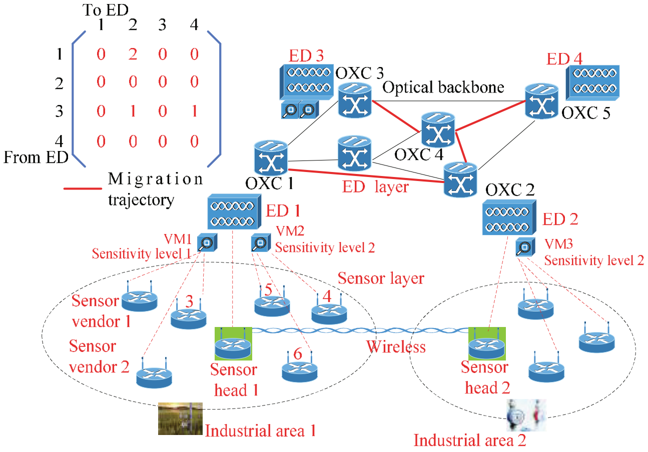

Figure 1: An instance of the IoE network.

EDs. In summary, the least upgrade batch VM scheduling is an important problem to solve.

On the other hand, the inter-ED VM migration brings the bandwidth consumption along migration trajectories. For example, in Fig. 1, during this upgrade batch, we assume that EDs 1 and 3 need to be upgraded simultaneously. The bandwidth—along the migration trajectory Optical Cross-Connect (OXC) 1→OXC 2—is consumed for the migration of VMs 1-2 from ED 1 to ED 2; the bandwidth—along the migration trajectories OXC 3→OXC 4→OXC 2 and OXC 3→OXC 4→OXC 5—is consumed for the migration of VMs from ED 3 to EDs 2 and 4, respectively. After VM migrations, EDs 1 and 3 are emptied for an upgrade without affecting the service continuity. Thus, in Fig. 1, we generate a 4×4 VM migration matrix where the element denotes the total bandwidth required by migrating the VM(s) from one ED to another. For instance, since we have migrated 2 VMs from ED 1 to ED 2, the corresponding matrix element value is 2. Here, each VM occupies one unit of the bandwidth. Intuitively, each batch generates one VM migration matrix. For the migration matrix, a blind bandwidth and migration trajectories assignment may result in the phenomenon where several trajectories compete for the common link bandwidth, thus worsening the upgrade efficiency. Also as an example of Fig. 1, the above migration trajectories OXC 3→OXC 4→OXC 2 and OXC 3→OXC 4→OXC 5 will compete for the bandwidth of their common link OXC 3→OXC 4. If the link OXC 3→OXC 4 has a very limited residual bandwidth, the VM migration will become fail due to this path contention. Thus, it is important to achieve an appropriate assignment of trajectories and bandwidth for each VM migration matrix.

Finally, functioned as the static sensor head, the ED should transmit the data—collected by ordinary sensors—to other specific EDs before migrating its VMs out. As in Fig. 1, before migrating VMs 1 and 2, the ED 1 should transmit the data collected by ordinary sensors 1-6 to ED 2 via wireless since the data computing programs are in the VMs 1 and 2 that will also be migrated to ED 2. As a result, we cannot prolong the network lifetime once the battery power of the sensor head ED 1 has been sharply exhausted. Therefore, as another important metric of evaluating the network sustainability, the expansion of network lifetime is also our focus. Note that, in this paper, the difference among the operation principles of various vendors' sensors merely restricts the inter-sensor data operability instead of exchange.

1.2 Contributions

This paper is the first work focusing on sustainable strategy for recycling EDs in IoE networks. The main contributions are summarized in the following.

We describe the least upgrade batch VM scheduling problem for recycling EDs in IoE networks, mainly including NP-hard proof, feasibility condition proposition, and lower bound analysis. Here, the lower bound denotes the minimum number of upgrade batches so that we can demonstrate the algorithm optimality.

A two-step algorithm is designed by us. In the first step, a greedy strategy is utilized for minimizing the number of batches required to finish the system-scale upgrade while guaranteeing the service continuity. After running this algorithm, we obtain the VM migration matrix during each batch. In the second step, two strategies—Shortest Trajectory First (STF) and Least Bandwidth Utilization First (LBUF)—are further proposed to assign trajectories and bandwidth for VM migration matrices. In addition, the transmitting power is evaluated for the ED functioned as the static sensor head.

The simulation results show that: 1) MSBP has the optimal recycling efficiency (least number of upgrade batches) for EDs; 2) LBUF more effectively mitigates the phenomenon where VM migration trajectories compete for the common link bandwidth, thus achieving a lower negative impact of path contention level on the recycling efficiency; 3) the battery power is not exhausted for the ED functioned as the sensor head of transferring data, thus prolonging the network lifetime. In summary, our solution well improves the network, social, economic and ecological sustainability.

2. SYSTEM MODEL

In the IoE network: 1) the ED layer can be represented by a graph $G(N, L, K)$. Here, N is a set of OXCs; L is a set of optical links; K is a set of EDs. Based on the interconnection of $|K|$ EDs, the VMs can be dynamically migrated between EDs. All the EDs have the same computing capacity C_0. The i^{th} VM—initially processed in the ED j ($j \in [1, |K|]$)—is represented as VM_i^j. VM_i^j has the computation requirement r_i^j, and $r_i^j < C_0$. Then, the amount of utilized and residual computing resources—of the ED j—can be denoted as $U_j = \sum_i r_i^j$ and $E_j = C_0 - U_j$, respectively. The bandwidth b_i^j is consumed on the optical link(s) along the migration trajectory of VM_i^j. We let $b_i^j = r_i^j$, which is a reasonable assumption because the highly computing-dependent VM is also the bandwidth-intensive one. The VM migration delay is much less than the time length of upgrading EDs so that it can be ignored, because the VM migration is performed by the high-speed optical transmission; 2) in the sensor layer, there are $|K|$ industrial areas deployed with the same number of sensors owned by different vendors. Functioned as the sensor head, the j^{th} ED manages the sensors of j^{th} industrial area. The VM_i^j encapsulates the computing program for the i^{th} sensitivity-level data items collected by the sensors deployed in the j^{th} industrial area. As mentioned in the 3^{rd} paragraph of subsection 1.1, the j^{th} ED should transmit the data—collected by the sensors in the j^{th} industrial area—to other specific EDs that will accept the VMs migrated out from the the j^{th} ED. We assume that the amount of transmitted data is equal to the size of frequently migrated VMs, and one unit of the VM size consumes one unit of transmitting power. In other words, the initial battery power of the j^{th} ED is the computing capacity C_0, and we have the following sustainability level of the j^{th} ED.

$$sl_j = \frac{\sum_i r_i^j \cdot \eta_i^j}{C_0}, \forall j. \tag{1}$$

Here, η_i^j is a boolean variable that is 1 if the VM VM_i^j has been migrated more than once during the entire least upgrade batch VM scheduling; otherwise $\eta_i^j = 0$. Intuitively, if $sl_j = 1$, the battery power of the j^{th} ED will be exhausted.

The network sustainability level is thus defined as follows.

$$nl = argmax\{sl_j, j \in [1, |K|]\}. \tag{2}$$

Here, the network sustainability level is the maximum ED sustainability level. If $nl = 1$, there is exactly one ED whose battery power has been exhausted. In other words, a lower n_l corresponds to a better network sustainability level.

3. PROBLEM STATEMENT

Based on the system model above, we describe the problem as follows. Considering a stable status of the IoE network where there are n VMs over $|K|$ EDs interconnected by $|L|$ optical links, we make the least batch VM scheduling to finish the system-scale upgrade through the appropriate assignment of trajectories and bandwidth for VM migration matrices. The following statements are given: 1) Each ED must be upgraded in just one of batches. In other words, each ED must be upgraded only once, and the system-scale upgrade is terminated once all EDs complete their upgrade activity; 2) EDs have the same upgrade time length. In one batch, the EDs start and end their upgrade simultaneously, i.e., batches have the same time length; 3) the computing capacity of one ED becomes zero during the upgrade activity, and it will recover to a full computing capacity after the upgrade; 4) the service continuity is guaranteed by: VM migrations performed on optical links in the ED layer, and the data transferring performed on wireless links in the sensor layer. This also avoids the conflict between VM migrations and data transferring; 5) inter-ED VM migrations occupy the link bandwidth, while the inter-ED data transferring consumes the transmitting power.

3.1 Problem complexity analysis

We have the following analysis of the problem complexity.

Proposition 1: Our problem is NP-hard.

Proof. To analyze our problem complexity, we should find an NP-hard sub-problem whose complexity is equal to that of ED upgrade during one batch. Each ED can be seen as one physical server assigned with a fixed computing capacity C_0. Our objective is to maximize the number of emptied EDs for upgrade using VM migrations during each batch, which is equivalent to the *server consolidation* problem [10] where the number of emptied servers is maximized by consolidating VMs into high-loaded servers. Clearly, our problem is at least as hard as an NP-hard server consolidation because we may conduct several batches to achieve a system-scale upgrade. Therefore, our problem is NP-hard. □

3.2 Problem feasibility conditions

In this subsection, we analyze the problem from the perspective of feasibility conditions.

Proposition 2: The sufficient and necessary condition of our problem feasibility will be: all n VMs—processed over the IoE network—can be migrated to any $(|K| - 1)$ EDs.

Proof. In terms of the sufficient condition: Index $|K|$ EDs as $\{ED_1, ED_2, ...ED_{|K|}\}$. Suppose there is a scheduling scheme Θ for migrating n VMs to the last $(|K|-1)$ EDs $\{ED_2, ED_3, ...ED_{|K|}\}$, we can upgrade ED_1 at the first batch. In the following $q^{th} (q \geq 2)$ batch, we can empty ED_q for upgrade through migrating the VMs—processed over ED_q—to ED_{q-1}. Therefore, we can use $|K|$ batches to finish the system-level upgrade while guaranteeing the service continuity. Θ is one feasible solution. In terms of the necessary condition: If n VMs cannot be migrated to any $(|K| - 1)$ EDs, we cannot empty any ED for an upgrade. It violates our statement that each ED must be upgraded only once, thus resulting in no feasible solution. □

Proposition 3: According to proposition 2, the sufficient and necessary condition of our problem feasibility will be:

$$\left(\sum_{i=1}^{n} \sum_{j=1}^{|K|} r_i^j \right) \leq (|K| - 1) \times C_0, \tag{3}$$

where $\sum_{i=1}^{n} \sum_{j=1}^{|K|} r_i^j$ represents the total computation requirement of all n VMs, and $(|K|-1) \times C_0$ denotes the total computing capacity of any $(|K| - 1)$ EDs.

Proof. Equation (3) ensures that all n VMs can be migrated to any $(|K| - 1)$ EDs. If such condition cannot be satisfied, it will be impossible to upgrade any ED without interrupting services, thus leading to no feasible solution. □

3.3 Lower bound

In this subsection, we mathematically derive the lower bound to demonstrate the optimality of our heuristic algorithm presented later. Here, the lower bound denotes the minimum number of upgrade batches.

Proposition 4: For our problem, the lower bound will be:

$$L_B = \lceil \frac{|K|}{|K| - K_0} \rceil. \tag{4}$$

During each batch, K_0 is the minimum number of EDs needed to process all n VMs, then we have:

$$K_0 = \frac{\sum_{i=1}^{n} \sum_{j=1}^{|K|} r_i^j}{C_0}. \tag{5}$$

Proof. During each batch, what will be the minimum number of EDs needed to process all n VMs? We have $K_0 = \frac{\sum_{i=1}^{n} \sum_{j=1}^{|K|} r_i^j}{C_0}$. In other words, at most $(|K| - K_0)$ EDs can be emptied for upgrade during each batch. Thus, the minimum number of upgrade batches $L_B = \lceil \frac{|K|}{|K|-K_0} \rceil$. □

4. HEURISTIC ALGORITHM

Since our problem is NP-hard, in this section, we design a heuristic algorithm called MSBP (Minimized upgrade batch VM Scheduling and Bandwidth Planning) to solve it. We greedily maximize the number of emptied EDs for an upgrade during each batch. As a result, the least upgrade batch VM scheduling can be achieved. After that, one VM migration matrix generates at each batch, and we assign appropriate migration trajectories and bandwidth for all the matrices. Note that, even with the minimum number of batches, an entire VM scheduling will fail once one matrix cannot be satisfied. More specifically, the path contention phenomenon—where several VM migration trajectories compete for the common link bandwidth—seriously worsens the upgrade efficiency. Consequently, we have the following definition for the path contention level.

$$pcl = argmax\{wb_l, l \in [1, |L|\} - aver\{wb_l, l \in [1, |L|\}. \quad (6)$$

Here, the path contention level is the difference between the maximum link bandwidth consumption and the average link bandwidth consumption. Obviously, a smaller pcl denotes a lower path contention, thus achieving a better effect of bandwidth and trajectories assignment.

Denote the set of EDs—which have been upgraded so far—as the target ED set (TDS), and denote the set of EDs—which have not been upgraded—as the existed ED set (EDS). Obviously, the union of sets TDS and EDS is the whole ED set K. Naturally, the cardinality of the TDS set will expand whereas the size of the EDS set will shrink along with the implementation of upgrade batches. To achieve the least upgrade batch, we maximize the cardinality of the TDS set at each batch, meanwhile, give the priority to the EDs of TDS for accommodating the VMs migrated from the ED of EDS in the following batches. Record the VM migration matrix Ω_q during the q^{th} batch. In Ω_q, the element ω_q^{ij} denotes the total bandwidth required by migrating the VM(s) from the ED ED_i to the ED ED_j during the q^{th} batch. For the corresponding bandwidth and trajectories assignment, we have two strategies: Shortest Trajectory First (STF) and Least Bandwidth Utilization First (LBUF). In STF, for ω_q^{ij}, we find the shortest distance migration trajectory—whose residual bandwidth is not smaller than ω_q^{ij}—from ED_i to ED_j during the q^{th} batch. In LBUF, for ω_q^{ij}, we find the migration trajectory—along the optical link(s) which have large available residual bandwidth—from ED_i to ED_j during the q^{th} batch. Finally, we evaluate the transmitting power for EDs according to VM migration results. The main steps of the MSBP are shown as follows.

Step 1: For the first upgrade batch, initialize $|TDS|$=0 and $|EDS|$=$|K|$. In EDS, sort the EDs by the increasing residual computing resources E_j. In EDS, empty the EDs—which currently have the least computation utilization—by migrating their VMs to other sorted EDs through the first-fit scheme. Basically, the first-fit scheme intends to move VMs to the first sorted ED that can accommodate those VMs. After VM migrations, denote the set of emptied EDs as TDS, and the set of remaining EDs as EDS. Implement the first upgrade batch for the EDs in TDS. Generate the corresponding VM migration matrix Ω_1.

Step 2: For the following batches, sort the EDs in TDS and EDS, respectively, by the increasing residual computing resources E_j. In EDS, empty the EDs—which currently have the least computation utilization—by migrating their VMs to other sorted EDs (EDs in TDS have priority) through the first-fit scheme. In this way, the computing resources—owned by the EDs in TDS—can be fully utilized to maximize the number of emptied EDs in EDS. Execute a new upgrade batch for those emptied EDs. Generate the corresponding VM migration matrix. Such procedure should be repeated until all $|K|$ EDs have been upgraded only once in batches.

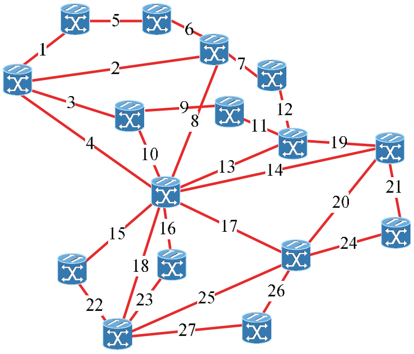

Figure 2: Simulation topology.

Step 3: After performing the aforementioned least upgrade batch VM scheduling, we assign the bandwidth and trajectories for VM migration matrices according to the batch order. More specifically, we first consider the VM migration matrix Ω_q before Ω_{q+1}. For the element ω_q^{ij} of the current VM migration matrix Ω_q, we can choose one of the following strategies to finish the corresponding bandwidth and trajectories assignment.

1) Followed by STF strategy (Algorithm 2), we utilize Dijkstra [11] to determined the shortest distance migration trajectory pd_q^{ij} from the ED ED_i to the ED ED_j during the q^{th} batch. If all the optical links—traversed by pd_q^{ij}—has the residual bandwidth which is not lower than ω_q^{ij}, we make the corresponding bandwidth assignment and update the residual bandwidth of relative optical links; otherwise, the VM scheduling fails.

2) Followed by LBUF strategy (Algorithm 3), we utilize the following equation to update the link weight as c_l' so that the optical link l—which has large available residual bandwidth F_l—tends to be selected by the migration trajectory. Note that $c_l' = \infty$ if $F_l = 0$. After updating the link weight, we still utilize Dijkstra, but this time, the least weight migration trajectory pw_q^{ij} is determined from ED_i to ED_j during the q^{th} batch. Similarly, if all the optical links—traversed by pw_q^{ij}—has the residual bandwidth which is not lower than ω_q^{ij}, we make the corresponding bandwidth assignment and update the residual bandwidth of relative optical links; otherwise, the scheduling fails.

$$c_l' = c_l - \alpha \cdot F_l. \quad (7)$$

Here, the value of α should comply with the condition $(c_l - \alpha \cdot F_l) > 0$, i.e., $\alpha < \frac{c_l}{F_l}$.

Step 4: If the scheduling succeeds, we return the total number of batches Q, and evaluate the network sustainability level as well as path contention level according to Eqs. (2) and (6), respectively.

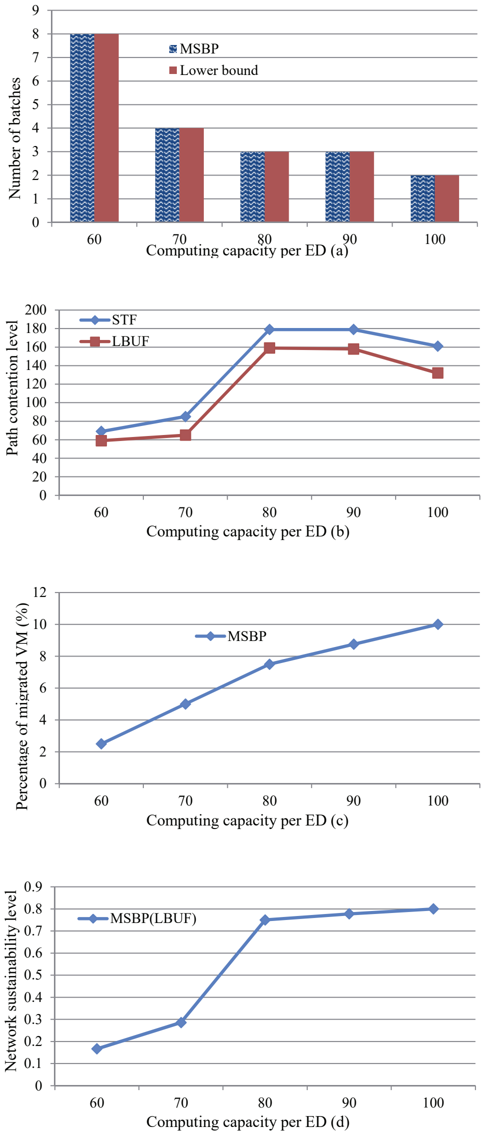

Figure 3: Simulation results when $r_i^j = 10$ and $n = 80$.

5. SIMULATION RESULTS

We first demonstrate the effectiveness of the least upgrade batch VM scheduling in our MSBP by checking whether the heuristic solution well matches the problem lower bound. We then compare the path contention level between STF and LBUF strategies. Finally, we check whether our MBSP can ensure a good network sustainability.

In Fig. 2, we utilize the real 16-node and 27-edge RedIRIS [12] as the test IoE network topology, where each OXC node is locally equipped with an ED, i.e., $|N| = |K| = 16$. The optical link supports the VM migrations with two contrary directions. In each optical link, the number represents the link index. We consider that the initial weight—owned by each optical link—is 1, i.e., $c_l = 1, l = 1, 2, ..., 27$. The ED is configured with the unified computing capacity C_0 measured as a number of resource units (e.g., CPU resource units). Initially, there are n VMs running over the system. Being defined as the number of resource units that one VM may request, the variable r_i^j is smaller than C_0. In our simulations, according to the Amazon EC2 trace file [13], we consider the following scenario: all the VMs have the same $r_i^j = 10$, which is the size of the $t2.large$ VM actually provided by Amazon EC2. Note that, following the trace file of r_i^j, we vary C_0 under the proposed problem feasibility condition in Eq. (3).

Given $r_i^j = 10$ and $n = 80$ in Fig. 3, to satisfy Eq. (3), we vary C_0 followed by the constraint $C_0 \geq \frac{\sum_{i=1}^{n}\sum_{j=1}^{|K|} r_i^j}{(|K|-1)} = \frac{80\times 10}{15} \approx 54$. Correspondingly, C_0 is increased from 60 to 100. In Fig. 3(a), we configure each optical link with a rich bandwidth provisioning $n \times r_i^j = 80 \times 10 = 800$, in order to demonstrate the effectiveness of the least upgrade batch VM scheduling in our MSBP. We observe that the number of batches drops significantly with the increasing computing capacity. It is due to the fact that with a larger computing capacity, more VMs can be migrated to a relatively smaller set of EDs so that a larger set of EDs can be emptied at each batch, thus reducing the total number of batches. Moreover, the heuristic solution well matches the lower bound, which actually demonstrates the effectiveness of the least upgrade batch VM scheduling in our MSBP. The reason for this is that due to the existence of the problem feasibility condition, the total computing capacity of all the EDs absolutely exceeds the total computing resource requirement of VMs, which makes the MSBP always achieve the optimal value, thus leading to the minimum number of batches.

The aforementioned least upgrade batch VM scheduling still fails if one VM migration matrix cannot be satisfied due to the path contention where multiple VM migration trajectories compete for the common link bandwidth. Thus, in Fig. 3(b), we compare the path contention level—measured by Eq. (6)—between LTF and LBUF strategies under the same parameter settings as Fig. 3(a). Here, $\alpha = 0.01$ for L-BUF, and a lower path contention level means a better effect of bandwidth and trajectories assignment. The simulation results show that the LBUF has the lower path contention level than that of LTF. The corresponding improvement ratio is 16%. This is because that the LBUF makes the optical link—which has large available residual bandwidth—tend to be selected by the migration trajectory, which mitigates the phenomenon where a common link is always occupied by a large number of trajectories.

As shown in Eq. (1), the frequently migrated VMs should be found in prior if we want to evaluate the ED sustainability level. Correspondingly, we analyze the percentage of VMs—which have been migrated more than once during the entire least upgrade batch VM scheduling—in Fig. 3(c). The parameter settings are the same as Fig. 3(b). We can see that the number of frequently migrated VMs follows a rising trend with the increasing computing capacity per ED. This tendency is rational because more VMs can be migrated under a larger computing capacity. More importantly, merely at most 10% VMs have been frequently migrated, thus resulting in a low ED sustainability level. Therefore, as shown in Fig. 3(d), the network sustainability level is always lower than 1, which means that there is no ED whose battery power has been exhausted. The network lifetime is thus well prolonged using our MSBP.

6. CONCLUSION AND FUTURE WORK

In this paper, we comprehensively studied sustainable strategy for recycling EDs in IoE networks. We formulated the problem based on the system model, and we mathematically derived the problem feasibility condition and the lower bound. Since the problem was NP-hard by nature, the heuristic MSBP algorithm was designed to minimize the number of batches while conserving the service continuity through VM migrations. Moreover, to guarantee the optimality of our MSBP, the LBUF strategy was utilized to satisfy VM migration matrices via obtaining a lower path contention level. The simulation results demonstrated the optimality of our MSBP because the heuristic solution always well matched the lower bound. In addition, the LBUF strategy actually had the better performance of allocating bandwidth and migration trajectories compared with STF. Finally, the proposed solution had the highest efficiency of recycling EDs while guaranteeing a long network lifetime. In summary, our solution well improves the network, social, economic and ecological sustainability.

In our work, the bandwidth and migration trajectory assignment strategy was utilized for all kinds of VMs. Once the differentiated sensitivity-level-aware environment—where different VMs correspond to various sensitivity-level data—was involved, the hybrid strategy combined with STF and LBUF would be designed in future, aiming at further reducing the path contention level.

7. ACKNOWLEDGEMENTS

This work was supported in part by the National Natural Science Foundation of China (Grant Nos. 61401082, 61471109, 61502075, 61672123, 91438110, U1301253), the Fundamental Research Funds for Central Universities (Grant Nos. N161604004, N161608001, N150401002), Liaoning Province Doctor Startup Fund under Grant No. 201501166, and China Post-Doctoral Science Foundation Project under Grant No. 2015M580224.

8. REFERENCES

[1] Dung Pham Van, Bhaskar Prasad Rimal, Jiajia Chen, Paolo Monti, Lena Wosinska, and Martin Maier. Power-saving methods for internet of things over converged fiber-wireless access networks. *IEEE Communications Magazine*, 54(11):166–175, 2016.

[2] Tie Qiu, Xize Liu, Lin Feng, Yu Zhou, and Kaiyu Zheng. An efficient tree-based self-organizing protocol for internet of things. *IEEE Access*, 4:3535–3546, 2016.

[3] Xiong Xiong, Lu Hou, Kan Zheng, Wei Xiang, M Shamim Hossain, and Sk Md Mizanur Rahman. Smdp-based radio resource allocation scheme in software-defined internet of things networks. *IEEE Sensors Journal*, 16(20):7304–7314, 2016.

[4] Yongan Guo, Hongbo Zhu, and Longxiang Yang. Service-oriented network virtualization architecture for internet of things. *China Communications*, 13(9):163–172, 2016.

[5] Peter Corcoran and Soumya Kanti Datta. Mobile-edge computing and the internet of things for consumers: Extending cloud computing and services to the edge of the network. *IEEE Consumer Electronics Magazine*, 5(4):73–74, 2016.

[6] Mohammad Aazam and Eui-Nam Huh. Fog computing: The cloud-iot\/ioe middleware paradigm. *IEEE Potentials*, 35(3):40–44, 2016.

[7] Subhadeep Sarkar and Sudip Misra. Theoretical modelling of fog computing: a green computing paradigm to support iot applications. *IET Networks*, 5(2):23–29, 2016.

[8] T Gomes, F Salgado, S Pinto, J Cabral, and A Tavares. Towards an fpga-based network layer filter for the internet of things edge devices. In *Emerging Technologies and Factory Automation (ETFA), 2016 IEEE 21st International Conference on*, pages 1–4. IEEE, 2016.

[9] Rudyar Cortés, Xavier Bonnaire, Olivier Marin, and Pierre Sens. Stream processing of healthcare sensor data: studying user traces to identify challenges from a big data perspective. *Procedia Computer Science*, 52:1004–1009, 2015.

[10] Nicolò Maria Calcavecchia, Ofer Biran, Erez Hadad, and Yosef Moatti. Vm placement strategies for cloud scenarios. In *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*, pages 852–859. IEEE, 2012.

[11] `http://en.wikipedia.org/wiki/Dijkstra`.

[12] `https://en.wikipedia.org/wiki/RedIRIS`.

[13] `https://aws.amazon.com/cn/ec2/instance-types/`.