Modification of an energy-efficient virtual network mapping method for a load-dependent power consumption model

HIROMICHI SUGIYAMA YUKINOBU FUKUSHIMA TOKUMI YOKOHIRA
The Graduate School of Natural Science and Technology
Okayama University
Tsushimanaka 3-1-1, Kitaku, Okayama-city, Okayama, 700-8530
JAPAN
{fukusima, yokohira}@okayama-u.ac.jp http://www.ec.okayama-u.ac.jp/~net/index-en.html

Abstract: - This paper tackles an energy efficient virtual network mapping problem where virtual nodes and links in a given virtual network have to be mapped to physical nodes and paths in a physical network so that the total power consumption associated with the mapping is minimized. The conventional method assumes that power consumption of a physical node is constant regardless of its load (constant power consumption model), and successfully reduces the total power consumption by preferentially mapping virtual nodes and links to active (used) physical nodes and paths passing only active physical nodes. However, power consumption of a physical node will become variable dependent on its load (variable power consumption model) in the near future, and the conventional method may not reduce the total power consumption because its active-node-first policy can cause large additional power consumption under the variable power consumption model. In this paper, we try to minimize the total power consumption under the variable power consumption model. In order to achieve this, we modify the conventional method so that it adopts the minimum-additional-power-consumption-first policy. The modified method calculates the actual additional power consumption associated with node and link mapping, and preferentially assigns virtual nodes and links to physical nodes and paths so that the actual additional power consumption is minimized. Simulation results clarify that the modified method can 4-40% lower total power consumption than the conventional method under the variable power consumption model.

Key-Words: - Network virtualization, Virtual network, Virtual network mapping problem, Energy efficiency, load-dependent power consumption model, heuristic algorithm

1 Introduction

With the appearance of SDN (Software Defined Networking) [1] and NFV (Network Functions Virtualization) and the improvement of the performance of general-purpose servers, it is feasible to realize the network functions and control the operation of network devices with software. With these technologies, it is discussed that multiple virtual networks are constructed and operated on a single physical network [2, 3]. As a result, we can share and effectively use the resources of the physical network among the multiple network services and we can deploy a new network service rapidly using the virtual networks.

In order to construct a virtual network on a physical network, each virtual node in the virtual network has to be assigned (mapped) to a physical node in the physical network and each virtual link in the virtual network has to be assigned to a physical path in the physical network so that we can optimize a focused performance measure (e.g., QoS of the virtual network, income of virtual network mapping operator and power consumption associated with virtual network mapping). In addition, constraints as to node and link resources have to be satisfied. The above problem is known as virtual network mapping problems [4-7].

In this paper, we tackle an energy-efficient virtual network mapping problem where the objective is to minimize the power consumption associated with virtual network mapping. This is because we have to reduce the power consumption of the Internet that rapidly increases with the increase of the volume of communication traffic every year. For example, it is reported that the total power consumption of all routers in Japan in 2020s will reach the annual energy production in Japan in 2012 [8].

In [9], an energy efficient virtual network mapping method is proposed under a load-independent power consumption model where 1) an inactive (unused) physical node does not cause any power consumption and 2) an active physical node causes constant power consumption regardless its load. In order to minimize power consumption, the conventional method adopts an “active-node-first” policy. The method preferentially assigns a virtual node to an active physical node and also preferentially assigns a virtual link to a physical path that passes active physical nodes because such assignment do not cause additional power consumption. The conventional method successfully reduces the power consumption under the load-independent power consumption model.

However, network equipments whose power consumption follow a load-dependent power consumption model have been developed [10], and such physical nodes will be dominant in the near future. In the model, power consumption of an active physical node varies depending on its load. Under the model, the “active-node-first” policy in the conventional method causes an additional power consumption, and consequently the conventional method may not successfully reduce the power consumption.

In this paper, we try to minimize the power consumption associated with virtual network mapping under the load-dependent power consumption model. In order to achieve this, we modify the conventional method so that it adopts the “minimum-additional-power-consumption-first” policy. The modified method calculates the actual additional power consumption associated with node and link mapping, and preferentially assigns virtual nodes and links to physical nodes and paths so that the actual additional power consumption is minimized. We evaluate the performance of the modified method by simulation.

The remainder of this paper is organized as follows. Section 2 introduces the network model, the problem formulation, and the conventional method. Section 3 shows the load-dependent power consumption model and the modifications to the conventional method. Section 4 shows evaluation results of the modified method. Section 5 concludes the paper.

2 Energy Efficient Virtual Network Mapping

2.1 Network model

Fig. 1 depicts an example of physical networks. A physical network is given by a directed graph. In the graph, vertexes and edges correspond to physical nodes and physical links, respectively. A physical node is equipped with a limited amount of node resources (e.g., CPU) that are used by the assigned virtual nodes. In addition, for each physical node, the power consumption (maximum power consumption) when its load is 1.0 is determined in advance. In Fig. 1, two numbers on each physical node show 1) the remaining amount of node resources (residual node resources) and 2) the maximum power consumption. For example, physical node A has 45 remaining node resources and its maximum power consumption is 272W. Similarly, a physical link is equipped with a limited amount of link resources (e.g., bandwidth) that are used by the assigned virtual links. In Fig. 1, the number on a physical link shows the remaining amount of link resources (residual link resources). For example, physical link (A, B) has 70 remaining link resources. We ignore power consumption of physical links because power consumption of physical links are much smaller than that of physical nodes.

Fig. 2 shows an example of virtual networks. A virtual network is also given by a directed graph. In the graph, vertexes and edges correspond to virtual nodes and virtual links, respectively. A virtual node requires a predetermined amount (node demand) of node resources on the corresponding physical node. A virtual link also requires a predetermined amount (link demand) of link resources on each physical link of the corresponding physical path. In Fig. 2, the numbers on a virtual node and a virtual link show the node demand and the link demand, respectively. For example, virtual node w demands 45 node resources of the corresponding physical node and virtual link (w, y) demands 24 link resources to each physical link of the corresponding physical path.

2.2 Energy efficient virtual network mapping problem

In the energy efficient virtual network mapping problem we tackle in this paper, we have to decide 1) to which physical node each virtual node is assigned (node mapping) and 2) to which physical path each virtual link is assigned (link mapping) so that the total power consumption associated with mapping a given virtual network is minimized. We assume that a request for a virtual network mapping is generated one by one, and we map it in an on-line manner. The problem formulation is as follows.

Input: A single request for a virtual network mapping

Output: The mapped virtual network or a mapping-failure notification

Objective function: To minimize the total power consumption associated with mapping the given virtual network

Constraints:

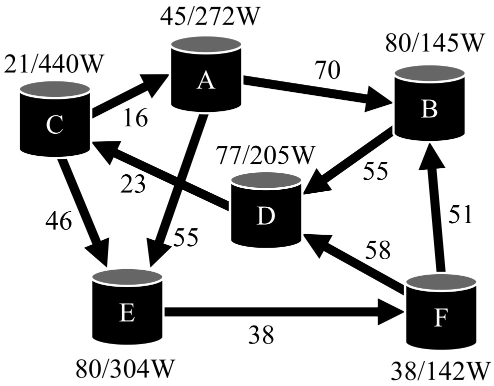


Fig. 1 Physical Network

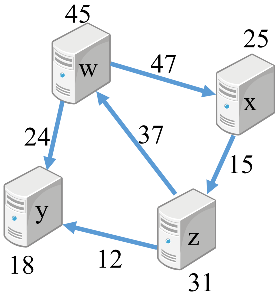


Fig. 2 Virtual Network

1. For any physical node, the sum of node demands required by the assigned virtual nodes have to be smaller than or equal to its residual node resources.
2. For any physical link, the sum of link demands required by the assigned virtual links have to be smaller than or equal to its residual link resources.

2.3 The conventional energy efficient virtual network mapping method

In this section, we introduce AdvSubgraph-MM-EE-Link [9], the conventional energy efficient virtual network mapping method. The method tries to minimized the total power consumption under the assumption that power consumption of physical nodes follow *a load-independent power consumption model (constant power consumption model)*.

In the model, a physical node consumes power only when it is *active* (i.e., load of the node is higher than zero) and its power consumption is constant regardless of its load. The power consumption of a physical node is shown in Fig. 3. A physical node becomes active if one or more virtual nodes are assigned to it or one or more virtual links pass it.

The basic idea of the conventional method is to give higher priority to assigning virtual nodes/links to active physical nodes/links than to assigning virtual nodes/links to inactive physical nodes/links.

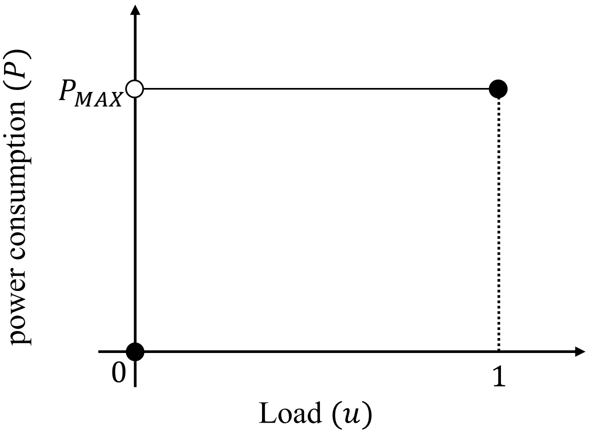


Fig. 3 The load-independent power consumption model

This is because such assignments do not cause any additional power consumption. The conventional method consists of two algorithms; Algorithm 1 and Algorithm 2, both of which are based on vnmFlib [11]. Table 1 describes the parameters used in the algorithms.

The detailed explanation of the functions used in the algorithm are as follows.

valid($M(G^V_{sub})$, (n^V, n^P), G^P)

This function returns true when mapping candidate (n^V, n^P) fulfills all of the following three conditions.

1. n^P's residual node resources is larger than or equal to n^V's node demand.
2. For any n^V_p in G^V_{sub}, there exists the path that connects n^P_p to n^P in G^P, the path length is within ε, and each of the physical links included in the path has residual link resources not less than the link demand of virtual link from n^V_p to n^V.
3. For any n^V_s in G^V_{sub}, there exists the path that connects n^P to n^P_s in G^P, the path length is within ε, and each of the physical links included in the path has residual link resources not less than the link demand of virtual link from n^V to n^V_s.

optimize(C)

This function removes infeasible mapping candidate (n^V, n^P) from C that does not satisfy both of the following two conditions.

1. n^P's residual node resources is larger than or equal to n^V's node demand.
2. For any virtual link between n^V and the virtual nodes in G^V_{sub}, the length of the corresponding physical path is within ε.

sort(C)

```
Algorithm 1 AdvSubgraph($G^V_{sub}$, $M(G^V_{sub})$, $G^V$, $G^P$)
Require: $G^V$, $G^P$, $G^V_{sub}$, $M(G^V_{sub})$
$C \leftarrow$ genneigh($G^P, G^V, G^V_{sub}, M(G^V_{sub})$)
for each($n^V$, $n^P$) in $C$ do
  if valid($M(G^V_{sub})$, ($n^V$, $n^P$), $G^P$)
    update $G^V_{sub}$ and $M(G^V_{sub})$ by adding ($n^V$, $n^P$)
    update $G^P$ by reducing node/link resources used by ($n^V$, $n^P$)
    AdvSubgraph($G^V_{sub}$, $M(G^V_{sub})$, $G^V$, $G^P$)
  end if
  if $G^V_{sub} == G^V$ then
    return $M(G^V_{sub})$
  end if
end for
```

```
Algorithm 2 genneigh($G^P$, $G^V$, $G^V_{sub}$, $M(G^V_{sub})$)
Require: $G^V$, $G^P$, $G^V_{sub}$, $M(G^V_{sub})$
if $F_{G^V_{sub}}(G^V) = \emptyset$ then
  $C \leftarrow N^V \times N^P$
else
  $C \leftarrow F_{G^V_{sub}}(G^V) \times N^P$
end if
optimize($C$)
sort($C$)
return $C$
```

This function sort the mapping candidates in C with the sort keys as follows.

1. n^P is active or not (active is preferred)
2. The maximum power consumption P_{MAX} of n^P (in ascending order)
3. n^V's node demand (in descending order)

The conventional method has two parameters. Parameter ε limits physical path length among the virtual nodes to realize link mapping with shorter path. This algorithm starts with $\varepsilon = 1$. If it cannot find the path until the number of mapping trials reaches ω, it resets $\omega = 0$ and increases ε by one and the algorithm continues. If ε is over the predetermined upper bound, the algorithm stops.

Parameter ω limits the number of mapping trials. Without this parameter, the conventional method performs the exhaustive search, and consequently the worst case complexity becomes $O(|N^P|!|N^V|)$. To avoid this, the method judges that there is no answer if the number of mapping trials reaches ω for the current value of ε.

Every time Algorithm 1 adopts mapping candidate (n^V, n^P) and assigns virtual node n^V to physical node n^P, it also maps all of the virtual links between n^V and the virtual nodes in G^V_{sub}. The conventional algorithm uses Dijkstra's algorithm for obtaining the physical paths assigned to the virtual links. The weight of physical link is set as follows.

$$1 + \alpha \cdot \left(1 - \mathrm{active}(n_i^P)\right) + \beta \cdot \frac{P_{MAX}^{n_i^P}}{\max_{u \in N^P} P_{MAX}^{u}} \quad (1)$$

where n^P_i is the physical node pointed by the physical link, active(n^P_i) returns one if n^P_i is active, and zero otherwise, α adjusts the weight of the cost for passing an inactive physical node and β adjusts to which extend the maximum power consumption of n^P_i should be considered.

Table 1 Parameters

Parameter	Description
G^V	Virtual network
G^P	Physical network
G^V_{sub}	The part of G^V consisting of the virtual nodes and links that have been already assigned
$M(G^V_{sub})$	The current mapping of the virtual nodes and links in G^V_{sub}
$F_{G^V_{sub}}(G^V)$	The set of virtual nodes that have not been assigned yet and are connected to a virtual node in G^V_{sub} with a virtual link
N^V	The set of virtual nodes
N^P	The set of physical nodes
n^V	A virtual node
n^P	A physical node
(n^V, n^P)	Mapping candidate that means to assign n^V to n^P
C	A set of mapping candidates (n^V, n^P)
n^V_p	A virtual node that has a virtual link to n^V
n^V_s	A virtual node that has a virtual link from n^V
n^P_p	A physical node that n^V_p is assigned to
n^P_s	A physical node that n^V_s is assigned to
ε	A parameter that limits the physical path length between virtual nodes
ω	A parameter that limits the number of mapping trials

3 Modifications to the conventional method in order to take account of a load-dependent power consumption model

3.1 Load-dependent power consumption model

In this paper, similarly to [12, 13], we assume the power consumption of physical node follows load-dependent power consumption model (variable power consumption model), shown in Fig. 4 and given by equation (2).

$$P = \begin{cases} P_0 + uP_T & (u > 0) \\ 0 & (u = 0) \end{cases} \tag{2}$$

The detail of the model is as follows.

- The power consumption (P) of physical node consists of the base power consumption (P_0) and the variable power consumption (P_T).
- The variable power consumption is proportionate to the load (u) of the physical node. The load of a physical node is calculated as the ratio of the sum of the node demands of the assigned virtual nodes and the resource consumed by relaying the virtual links to the node resources the physical node has.
- If the physical node is not active, the power consumption is 0.

The power consumption of the physical link is considered negligibly small.

3.2 Modifications to the conventional method

When the physical nodes follow the variable power consumption model, the assignment assuming the constant power consumption model may not successfully reduce the total power consumption associated with mapping the given virtual network. We show the example to explain why the conventional method cannot find optimal assignment. Fig. 5 shows the example of mapping that assigns virtual node x on physical node A or B. The physical node A is active because some virtual nodes are assigned on A or A relays some virtual link. The conventional method assumes the physical nodes follow the constant power consumption model, so the virtual node x is assigned on A because assigning on A does not cause additional power consumption. However, when the physical nodes follow the variable power consumption model, both assigning on A and B cause the additional power consumption shown by the red arrows in Fig. 5. In this case, assigning on B can reduce the additional power consumption associated with assigning x. Thus, the active-node-first policy may not successfully reduce the total power consumption associated with mapping the given virtual network for variable power consumption, so we modify the conventional method to reduce the total power consumption with the variable power consumption model.

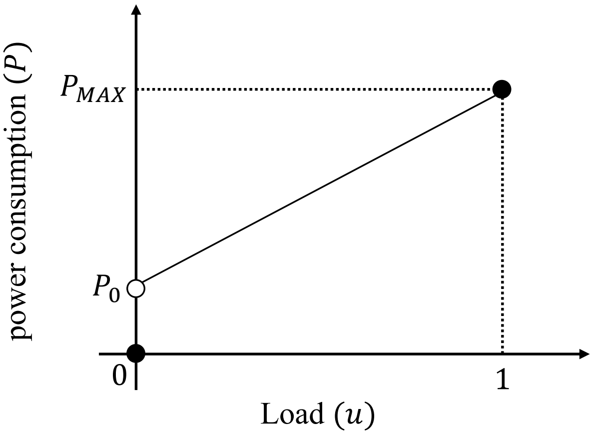


Fig. 4 The load-dependent power consumption model

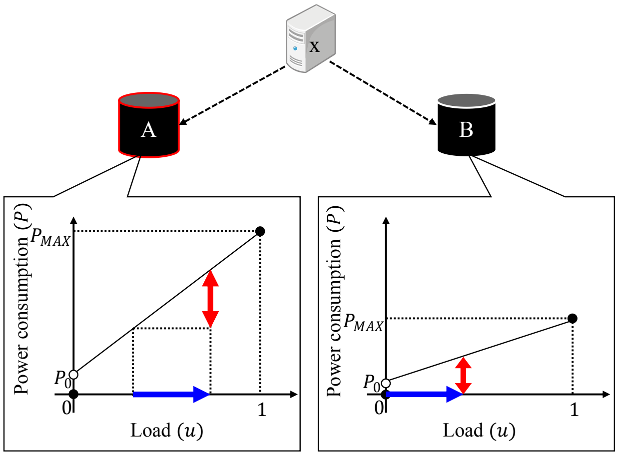


Fig. 5 An example of mapping

In order to minimize the additional power consumption, we modify the conventional method so that it maps a given virtual network with the minimum-additional-power-consumption-first policy instead of the active-node-first policy. To achieve this, the modified method calculates the actual additional power consumption based on the variable power consumption model, and use it for node and link mapping.

We propose two types of modifications, Modification A and Modification B. Modification A sort the mapping candidate by additional power consumption associated with node or link mapping at first and by node demand later in sort(C), same as the conventional method. Modification B tries to improve the mapping success rate, so we give higher priority to mapping virtual nodes in descending order of their node demands. The detail of Modification A is as follows.

A-1. In function sort(C) of Algorithm 2, we use the following two sort keys instead of the original ones.
  1. The additional power consumption caused by the corresponding node and link mapping when mapping candidate (n^V, n^P) is adopted (in ascending order)
  2. n^V's node demand (in descending order)

A-2. In obtaining a physical path assigned to a virtual link, we use the additional power consumption (ΔP) that is caused by mapping the virtual link to the physical link as the weight of the physical link. ΔP is calculated as follows.

$$\Delta P = \begin{cases} \Delta u_l P_T & (u_t > 0) \\ P_0 + \Delta u_l P_T & (u_t = 0) \end{cases} \qquad (3)$$

where Δu_l is the increase of load of the physical node pointed by the physical link and u_t is the current load of the physical node pointed by the physical link.

Modification A-1 tries to reduce the additional power consumption associated with both node and link mapping.

Modification A-2 tries to reduce the additional power consumption associated with link mapping. With the modification, the algorithm can choose the physical path that causes the minimum-additional-power-consumption.

Next, we introduce the details of the Modification B.

B-1. Two sort keys in function sort(C) of Algorithm 2 are as follows.
  1. n^V's node demand (in descending order)
  2. The additional power consumption caused by the corresponding node and link mapping when mapping candidate (n^V, n^P) is adopted (in ascending order)

B-2. Same as Modification A-2.

The difference between Modification A and B is the order of sort keys in function sort(C). In Modification B, we give higher priority to node demand than additional power consumption. This is because giving the first priority to additional power consumption seems to result in mapping a virtual node with smaller node demand preferentially, and consequently the number of virtual networks that are successfully mapped may be decreased.

Fig. 6 shows an example of mapping to explain how the modified methods work. Assume that the two-node virtual network on the left side of the figure will be assigned on the nine-node physical network on the right side of the figure. First, virtual node x is assigned on physical node F, because only F has enough residual node resource to accommodate virtual node x. The blue numbers beside the physical nodes show the additional power consumption assumed by the conventional method. The additional power consumption is 0W because F has been active before x is assigned. In the next step, virtual node y will be assigned on physical node A or G because of node resource constraint. The conventional method assigns y on G because only G is active among nodes A and G. The additional power consumption assumed by the conventional method is 0W despite the actual power consumption is 228W. On the other hand, the modified methods calculate the additional power consumption with load-dependent power consumption model. The red numbers besides the physical nodes show the additional power consumption assumed by the modified methods. The modified methods assign y to A with the minimum additional power consumption. The additional power consumption assumed by the modified methods with mapping this virtual network is 186W. The actual power consumption is same as that assumed by the modified methods. Therefore, the modified methods can find the optimal mapping. Like this, the conventional method cannot find the optimal mapping when physical nodes follow the load-dependent power consumption model.

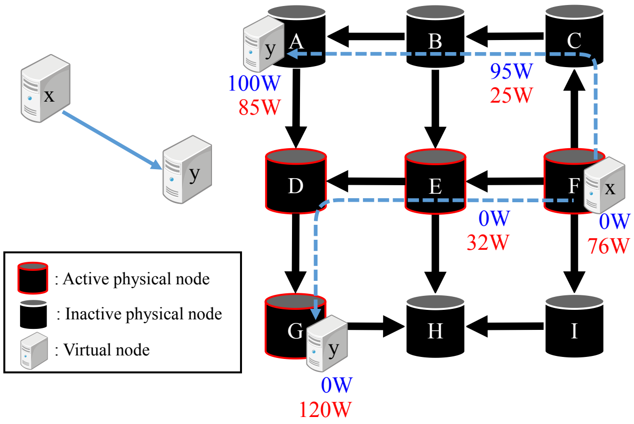


Fig. 6 An example of mapping

4 Performance Evaluation

4.1 Simulation model

We compare the performances of the modified methods and the conventional method by simulation. In the simulation, the actual total power consumption associated with mapping the given virtual networks is calculated assuming the variable power consumption model. The modified methods maps given virtual network assuming the variable power

Table 2 Parameters for physical networks

Parameter	Value
Number of nodes	100
The amount of node resources	An integer randomly chosen from [1:100]
The amount of link resources	An integer randomly chosen from [1:100]
The maximum power consumption [W]	An integer randomly chosen from [100:500]

Table 3 Parameters for virtual networks

Parameter	Value
Number of nodes	Fixed to an integer between 5 and 15
The amount of node resources	An integer randomly chosen from [1:50]
The amount of link resources	An integer randomly chosen from [1:50]

consumption model while the conventional method maps given virtual networks assuming the power consumption of active physical nodes is the same as the power consumption when the load of the node is 1.0 (i.e., the constant power consumption model where P_{MAX} in Fig. 3 is the same as that in the variable power consumption model). In the three methods, ω is set to quadruple of the number of virtual nodes (e.g., 60 when the number of virtual nodes is 15) and ε is set to 10. In the conventional method, the weights α and β are set to five. In the variable power consumption model, the fixed power consumption P_0 is set to 0, $0.33P_{MAX}$ or $0.5P_{MAX}$.

We use the total power consumption associated with mapping the given virtual networks and mapping success rate as metrics in the evaluation. The mapping success rate is defined as the ratio of the number of the virtual networks mapped successfully to the number of all the generated virtual networks.

Both physical networks and virtual networks are generated by Waxman model [14]. In Waxman model, both of the parameters α_W and β_W are set to 0.5. Tables 2 and 3 show the parameters for physical networks and virtual networks, respectively. Physical nodes are assumed to consume one node resource for relaying a virtual link.

At one trial in the simulation, we generate five virtual networks, try to map each of them in turn on the physical network and obtain the total power consumption, the mapping success rate and average physical path length per virtual link. For each value of the number of virtual nodes in a virtual node, we perform the trial for three different physical networks and five sets of different virtual networks (i.e., fifteen trials). When we calculate the averages of the performance metrics expect for the mapping success rate, we do not select the result for failed trial as a sample (i.e., the result for a trial where less than five virtual networks are mapped is not selected as a sample) for the fairness. If we select such failed trial as a sample, we cannot observe whether the result is caused by the effectiveness of a mapping method or by just mapping few virtual networks.

4.2 Results

Fig. 7 shows the results for $P_0 = 0.5P_{MAX}$. The horizontal axis is the number of virtual nodes in a virtual network.

In term of the total power consumption (Fig. 7(a)), the Modification A achieves 4-27% lower total power consumption than the conventional method, and the Modification B achieves 13-31% lower total power consumption. In order to investigate the total power consumption in detail, the power consumptions caused by node mapping and link mapping are depicted in Fig. 7(c) and Fig. 7(d), respectively. In the figures, the modified methods show higher power consumption than the conventional method in node mapping while the former does much lower power consumption than the latter in link mapping.

In terms of mapping success rate (Fig. 7(b)), the modified methods show better performance than the conventional method. The reason is assumed as follows. As shown in Fig. 7(e), the conventional method tends to map virtual links to physical routes with longer hop counts than the modified methods. Therefore, physical links run out of bandwidth and the physical network is divided into multiple sub-graphs earlier in the conventional method than in the modified methods. Because the divided sub-graphs have few resources than the total physical network, they tend to fail to accommodate the requested virtual network. Consequently, the conventional method shows lower mapping success rate than the modified methods.

Then we focus on the difference between two modified methods. In terms of mapping success rate, the results are almost the same, but, in term of power consumption, Modification B shows better performance than Modification A. Hence the order of sort keys in function sort(C) has influence on power consumption rather than mapping success rate.

We next focus on the influence of P_0 on the performance of the all methods. Fig. 8 shows the results for $P_0 = 0.33P_{MAX}$. Modification A achieves 4-26% lower total power consumption than conventional method, and Modification B achieves 18-31% lower power consumption. Fig. 9 indicates the results for $P_0 = 0$. In Fig. 9(a), Modification A shows 15-37% lower total power consumption than conventional method and Modification B shows 21-40% lower total power consumption.

We can see that the smaller P_0 is, the more power consumption can be reduced by the modified methods compared to the conventional method. This is because, the modified methods can always estimate the additional power consumption caused by node and link mapping for any P_0 while the conventional method generates larger error between the estimated additional power consumption and the actual one, as P_0 is larger.

From the above evaluation results, we can say that the minimum-additional-power-consumption-first policy is more effective than the active-node-first policy under the variable power consumption model.

5 Conclusion

In this paper, we modified the conventional energy efficient virtual network mapping method so that it can reduce the total power consumption under a load-dependent power consumption model. Simulation results clarified that 1) the modified methods achieve 4-40% lower total power consumption than the conventional method, 2) the smaller P_0 is, the more total power consumption can be reduced by the modified methods compared to the conventional method and 3) Modification B shows lower power consumption than Modification A.

One of our future work is to clarify the cause of the difference of the result between two modified methods.

Acknowledgement

This work was supported by JSPS KAKENHI Grant Number 15K00129.

References:

[1] B. Lantz, B. Heller and N. McKeown, “A network in a laptop: rapid prototyping for software-defined networks," *in Proc. of Workshop on Hot Topics in Networks*, 2010.

[2] K. Tutschku, T. Zinner, A. Nakao and P. Tran-Gia, “Network Virtualization: Implementation Steps towards the Future Internet,” *in Proc. of Workshop on Overlay and Network Virtualization*, 2009.

[3] N. M. M. K. Chowdhury, “Network Virtualization: State of the Art and Research Challenges,” *IEEE Communications Magazine*, vol. 47, no. 7, 2009, pp. 20-26.

[4] A. Fischer, J. F. Botero, M. T. Beck, H. D. Meer, and X. Hesselbach, “Virtual Network Embedding: A Survey,” *IEEE COMMUNICATIONS SURVEYS & TUTORIALS*, vol. 15, no. 4, 2013.

[5] M. Yu, Y. Yi, J. Rexford and M. Chiang, “Rethinking virtual network embedding: substrate support for path splitting and migration,” *ACM SIGCOMM Computer Communication Review*, vol. 38, Iss. 2, 2008.

[6] N. M. M. K. Chowdhury, M. R. Rahman and R. Boutaba, “Virtual Network Embedding with Coordinated Node and Link Mapping,” *in Proc. of INFOCOM*, 2009, pp. 783-791.

[7] X. Cheng, S. Su, Z. Zhang, H. Wang, F. Yang, Y. Luo, and J. Wang, “Virtual Network Embedding Through Topology-Aware Node Ranking,” *ACM SIGCOMM Computer Communication Review*, vol. 41, No. 2, Apr. 2011.

[8] Network Photonics Research Center, AIST, “AIST:Vertically Integrated Center for Technologies of Optical Routing toward Ideal Energy Savings (VICTORIES):Outline,” http://www.aist-victories.org/en/about/outline.html

[9] A. Fischer, M. T. Beck and H. D. Meer, “An approach to Energy-efficient Virtual Network Embeddings,” *in Proc. of IM*, 2013, pp. 1142-1147.

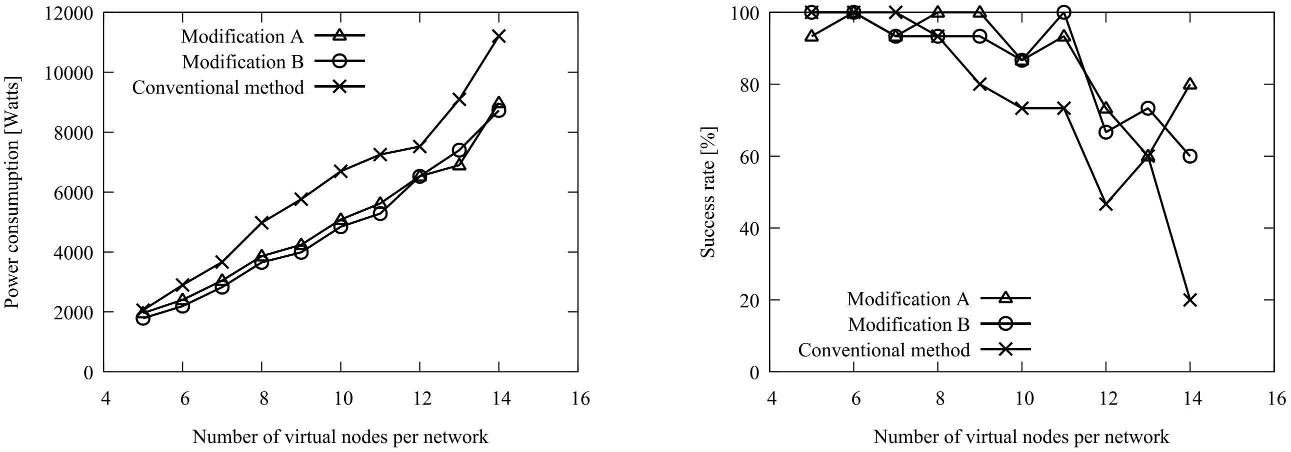
[10] S. Koyano, S. Ata, I. Oka and K. Inoue, “A High-Grained Traffic Prediction for Microseconds Power Control in Energy-Aware Routers,” *in Proc. of UCC*, 2012, pp. 347-352.

[11] J. Lischka and H. Karl, “A virtual network mapping algorithm based on subgraph isomorphism detection,” *in Proc. of VISA*, 2009, pp. 81-88.

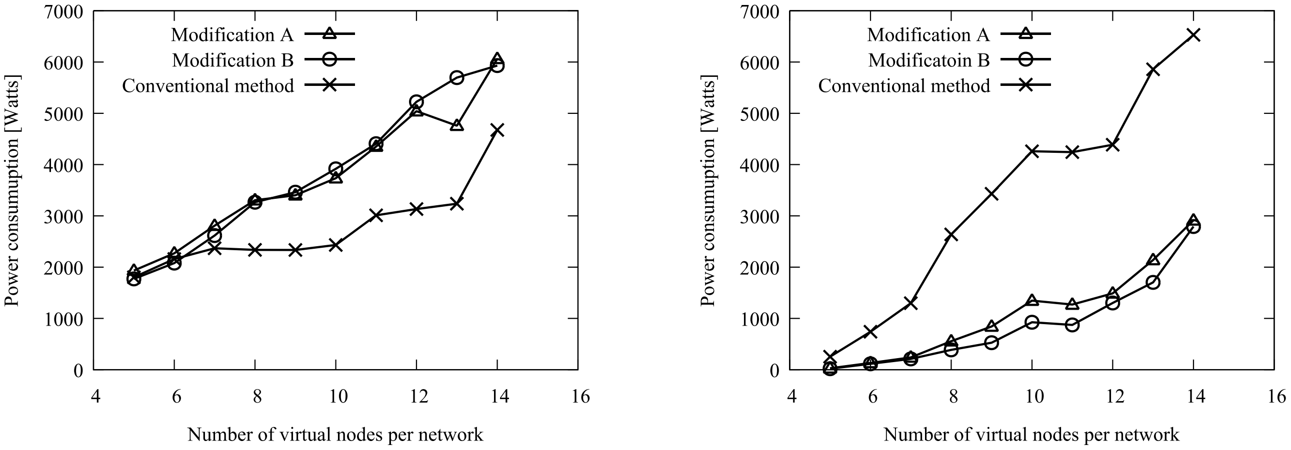
[12] M. Xia, M. Tornatore, Y. Zhang, P. Chowdhury, C. Martel and B. Mukherjee, “Greening the Optical Backbone Network: a Traffic Engineering Approach,” *in Proc. of ICC*, Jun. 2010.

[13] E. Yetginer and G. N. Rouskas, “Power Efficient Traffic Grooming in Optical WDM Networks,” *in Proc. of Globecom*, Dec. 2009.

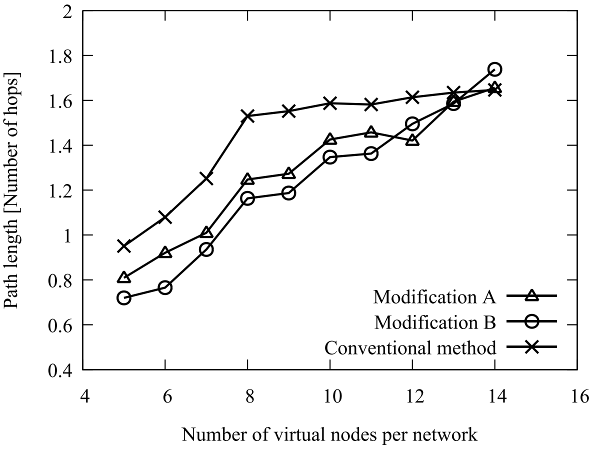
[14] H. Tangmunarunkit, R. Govindan, S. Jamin, S. Shenker, and W. Willinger, “Network topology generators: degree-based vs. structural,” *SIGCOMM Comput. Commun. Rev.*, vol. 32, 2002, pp. 147-159.



(a) The total power consumption (b) Mapping success rate

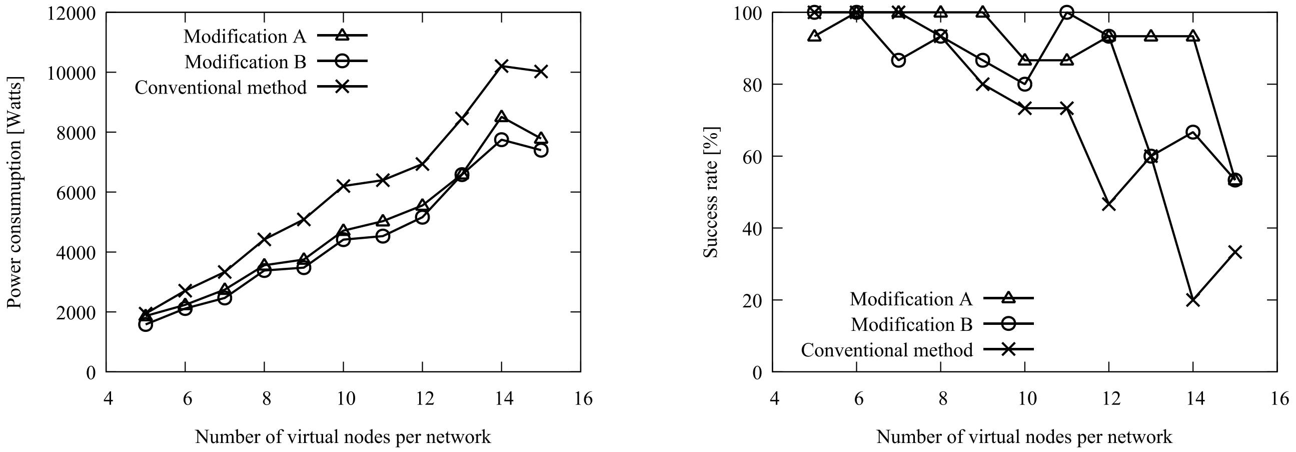


(c) Power consumption with node mapping (d) Power consumption with link mapping

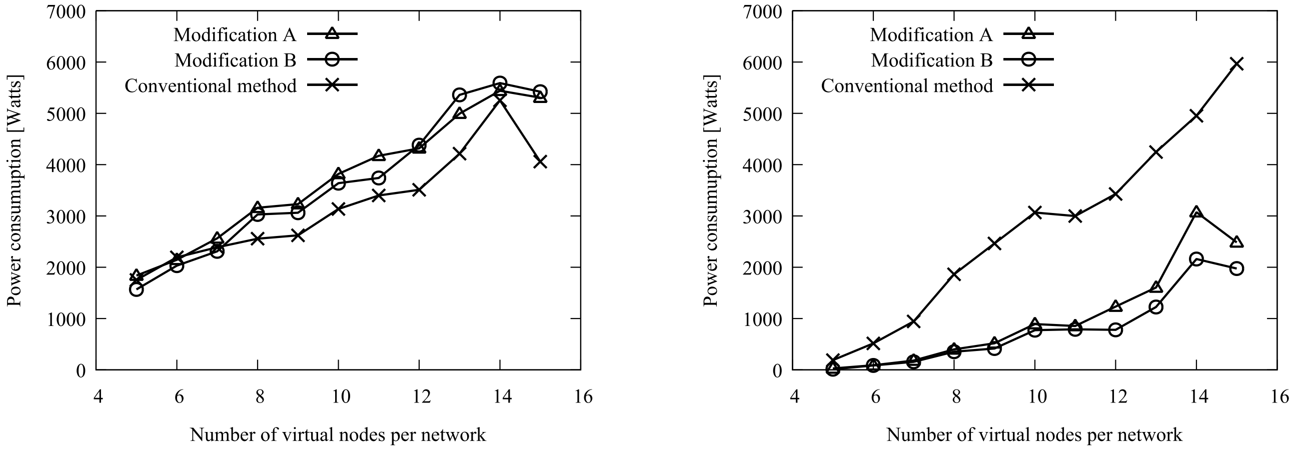


(e) Average path length

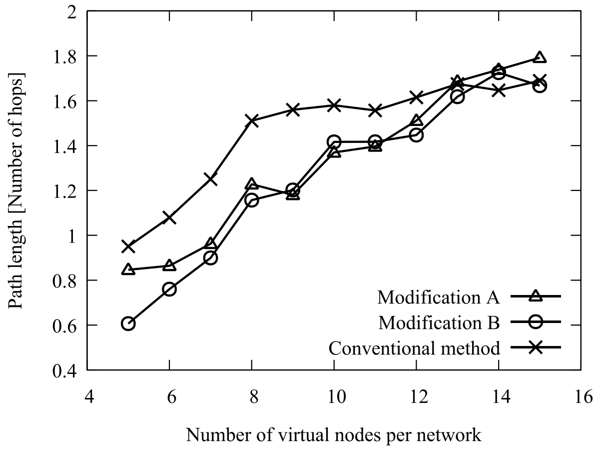
Fig. 7 $P_0 = 0.5P_{MAX}$



(a) The total power consumption
(b) Mapping success rate

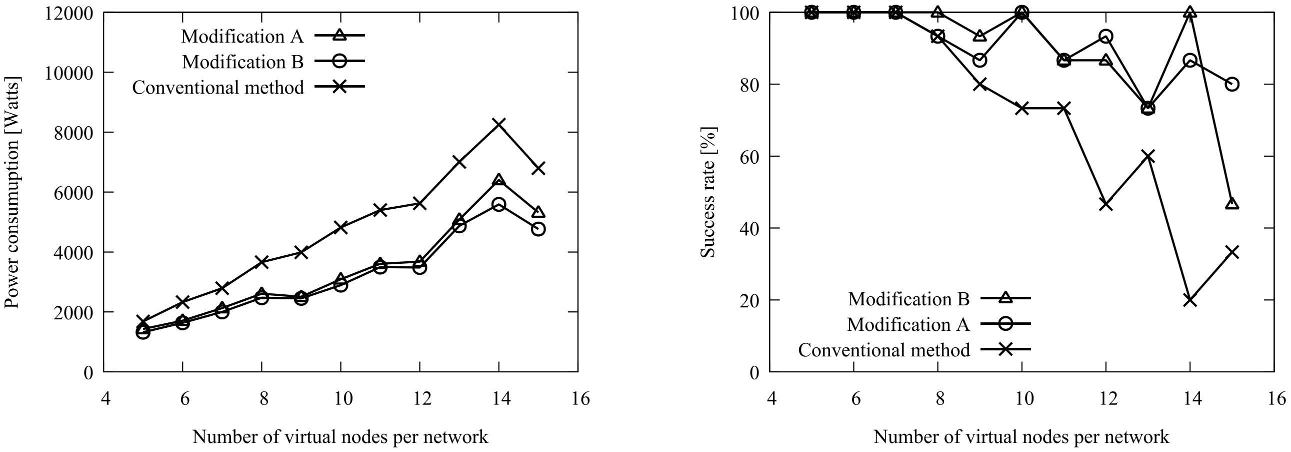


(c) Power consumption with node mapping
(d) Power consumption with link mapping



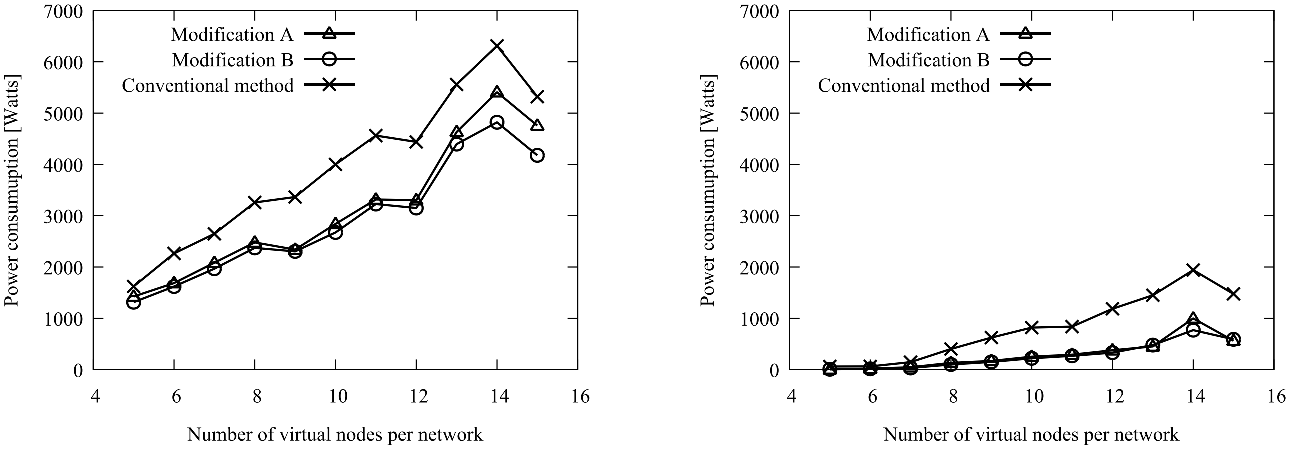
(e) Average path length

Fig. 8 $P_0 = 0.33P_{MAX}$



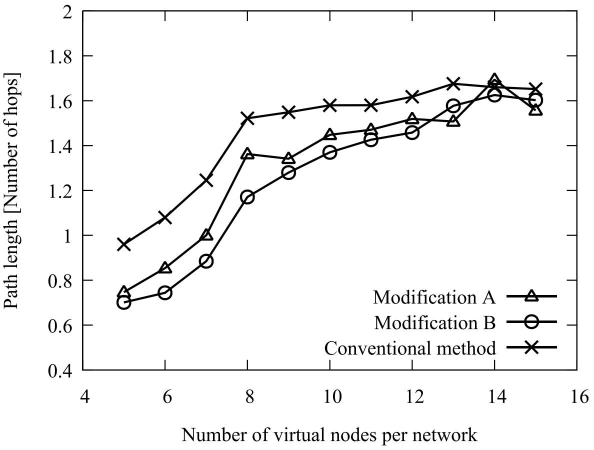
(a) The total power consumption

(b) Mapping success rate



(c) Power consumption with node mapping

(d) Power consumption with link mapping



(e) Average path length

Fig. 9 $P_0 = 0$