Indian Journal of Science and Technology, *Vol 9(24), DOI: 10.17485/ijst/2016/v9i24/96031, June 2016*

ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

A Strategy of Selecting a Maximal Set of Sleep Links for Energy Saving in Core IP Networks

Chimoon Han[1] and Sangchul Kim[2*]

[1]Department of Electronics Engineering, Hankuk University of Foreign Studies, Korea; cmhan@hufs.ac.kr
[2]Department of Computer and Electronic System Engineering, Hankuk University of Foreign Studies, Korea; kimsa@hufs.ac.kr

Abstract

Background/Objectives: Nowadays the energy consumption of ICT networks is about 10% of the total worldwide power consumption and is predicted to increase remarkably in the near future. **Methods/Statistical Analysis:** We study energy saving methods for IP networks in a way to select qualified links and place them in sleep mode under network-level QoS constraints. An energy saving strategy is proposed which is based on the concept of delegation process with a user-specified condition for limiting the increase in hop count of paths. The main feature of our strategy is delegation process through which a maximal set of sleep links can be found under the user-specified constraint. **Findings:** For maximizing energy saving, we present two heuristic algorithms for link selection under QoS constraints, and evaluate the characteristics of the algorithms by a computer simulation. Our two algorithms provide three kinds of methods, namely, max_set, min_traffic and min_traffic. This paper shows that in terms of link saving efficiency, the max_set is ranked first, the mini_traffic second, and the min_path last. The link saving efficiency of the max_set reaches about 65 % under certain conditions. Also, the max-set method is shown to find a larger set of sleep links than previous works, enabling more energy saving. On the certain conditions, we confirm the increased mean hops of the path within 1.2 hops in case of using the max_set for selecting sleep link. This value does not have a significant impact on the real end-to-end delay. **Application/Improvements:** We expect that our algorithms will provide the ISP with a mechanism for achieving the required level of energy saving by adjusting the delay of paths.

Keywords: Energy Saving, Delegation Procedure, Heuristic Algorithm, Hop Count, Link Reduction

1. Introduction

These days energy consumption saving is the most challenging one of the issues for the Internet, resulting in performing researches on the high level energy efficiency of the IP network[1–3]. Until now generally the design of IP networks has been carried out in consideration of the traffic during the peak time to provide the network-level QoS (Quality of Service).

It is known that the amount of traffic during the night time such as AM 2:00 to AM 6:00 is 30% less than the day time[4,17]. Hence, considering such a situation, the various energy saving methods on network-level have been actively investigated and the overview of them are reported[6–11]. Especially the basic concept of network-level energy saving is that during the period of light traffic, the routers (or interchangeably called *nodes*) on the network cooperatively find a way in which only a part of the network (i.e. routers and links) are utilized and the rest is placed in sleep state[12–15].

Among previous works for energy saving at ISP (Internet Service Provider) network level, the most interesting one is a method for energy saving by modifying the existing OSPF (Open Shortest Path First) protocol in a way that a set of links are selected and then placed in sleep state[14–17]. In the OSPF protocol each router uses Dijkstra's algorithm to provide SPT (Shortest Path Tree) paths from the router to all other routers. To save the energy of the IP network, research has been performed on a method in which specific nodes or links are put in sleep

**Author for correspondence*

state when the amount of total network traffic falls below a threshold[12,16]. If a node or link goes to sleep, rerouting is carried out in order to determine its new SPT using a routing protocol.

As mentioned earlier, the variation of peak time traffic and network-level QoS are important factors in the performance of ISP networks. Usually energy saving of network can be obtained by the reallocation of network resources such as links and nodes under those factors.

The traffic of Internet is expected to increase in the near future, and a router can be coupled with a lot of devices to accommodate the increased traffic. Required devices are NICs (Network Interface Cards), and their consumption energy is not negligible[23]. The energy handling of electronic devices on the IP layer is more flexible than that of optic devices on the WDM (Wave Division Multiplex) layer in IP network. Energy saving methods must put the minimal impact of the topological changes on the existing OSPF protocol[16]. For this reason, we focus on the link reduction of IP network in this paper.

We study methods for minimizing the number of active links under network-level QoS constraints. We can save the energy of the network through getting as many links as possible to sleep by deactivating the NICs at the endpoints of those links. However, some deactivating links tends to increase the hop counts of some SPT paths from ingress nodes to egress nodes. Also it may be unable to meet network-level QoS constraints.

To determine a link as one for sleeping (shortly called *sleep link*), we investigate a way to search an area near the link to find out an alternative link that can handle the traffic of the link[18]. The reason is that this process minimizes the overhead needed for the rerouting incurred by putting a link to sleep in the existing OSPF routing protocol. The overhead means an effort to modify the routing tables of nodes.

More precisely the start points of an alternative link and a sleep link must belong to the same router. Then such a router delegates the traffic of the sleep link to the router at the end point of the alternative link. To do so, the router at the starting points of those links modifies its SPT in order to redirect the flows of the paths through the sleep link toward the alternative link. We call the modified SPT of a node as the *MPT* (Modified Path Tree) of the node.

A similar approach to the method proposed herein was presented[17]. In the algorithm[17], the neighborhood of a link for sleeping is explored to find an alternative link. However, in the algorithm[17], the condition for checking whether two links can sleep at the same time is too strict so that the algorithm will find only a relatively small set of sleep links. Thus the energy saving efficiency of IP network can be relatively low.

In this paper, we study a method to find a maximal set of sleep links in consideration of the overhead for modifying the SPT's of routers. Our method also allows a user to specify the upper limit on the increment in SPT paths, so the user can adjust the number of sleep links at the expense of increment in path length. We propose two heuristic algorithms for selecting a maximal set of sleep links.

This paper is structured as follows. Chapter 2 briefly describes the general aspects and previous related works. In Chapter 3, we describe the problem modeling and basic concepts of this paper. In Chapter 4, two heuristic algorithms for maximal selection of sleep links are described. In Chapter 5, the characteristics of the proposed algorithm are analyzed through simulation. Chapter 6 concludes.

2. Background

2.1 General Aspects

An IP routing protocol, i.e. OSPF, is operated on an AS (Autonomous System) in IP networks and each router informs its own state to all neighbor routers using OSPF LSA (Link State Advertisements) messages. Each router calculates its SPT based on Dijkstra's algorithm, and determines the next hops for destinations, and then constructs its routing table. An IP network can be modeled as a weighted directed graph $G(N, E, W)$, where N is a set of nodes (i.e. routers), E is a set of directed links, and W a set of the weights associated with the links of E. Let R and L be the cardinalities of N ($R=\|N\|$) and E ($L=\|E\|$), respectively. For each directed link $l \in E$, s(l), e(l) and w(l) represent the source node, the end node and the weight of the link respectively.

Each router r of IP networks calculates its own SPT called $SPT(r)$ using OSPF LSA messages. In the calculation, the set E_A of active links is the set of links belonging to at least one SPT. The size L_D of E_A is defined as eq. (1).

$$L_D = \|E_A\| = \left\| \cup_{r \in N} SPT(r) \right\| \quad (1)$$

It is observed that $L_D \leq L$ for a certain upper bound L and a minimum number of undirected links must exist for transmitting the traffic between each pair of two routers in IP networks. So the minimum number L_{min} of undirected links is given as equation (2), where the R is the number of routers in IP networks.

$$L_{min} = 2(R-1) \tag{2}$$

As mentioned above, the objective of this paper is to find a method for reducing the number of the links used for transferring traffic in the IP networks. In previous works, various energy saving methods have been proposed to reduce the active links of equation (1). In applying an energy aware routing (EAR) for save energy in IP networks, a number of links, say L_e, will be put in sleep state. The efficiency of link saving, η_e, of the EAR can be defined as equation (3).

$$\eta_e = L_e/(L_D - L_{min}) \tag{3}$$

2.2 Related Studies

An IP network consists of routers together with links connecting them, and most of links are optical fibers. Routers are organized around a two-layered model, IP over WDM, and the logical topology of the IP network is mapped to an optical WDM network. Each undirected IP link corresponds to a dedicated WDM light path. Each router is roughly composed of a switch together with its control part in charge of routing functions, an NIC card in charge of link interface and optical functions such as E/O (Electronic to Optical Converter), O/E (Optical to Electronic Converter), OXC (Optical Cross Connect), etc.

Various techniques have been studied in order to save energy of ISP network. There have been publications on the survey and analysis of up-to-date technical solutions for network-level energy saving[6,9,10,11]. Those energy saving solutions can be classified as ones based on two layers such as WDM layer and IP layer.

The most of researches for energy saving of routers have been studied on IP layer due to the flexibility of IP layer reconfiguration[5,12–17,19]. For energy saving on IP layer, several methods have been proposed which minimize total energy consumption[19,20] and optimize the number of nodes and links[12] of IP network. Also, there is a solution to find all paths to minimize the energy consumption of network based on energy profiles of devices (i.e. router)[12,20,21].

The volume of energy consumption of devices is dependent on the load of operating devices in the IP network. So a solution is to search the routing paths to minimize the total energy consumption of the network, and various heuristic methods have been proposed[19,21]. But those methods have the disadvantage that the changing of paths is necessary according to the load variation of devices.

After putting some nodes and links to sleep, the rerouting of paths is performed in the newly reconstructed network [12]. In this case, the degradation of network-level QoS and the processing burden due to the rerouting of paths can increase. To overcome these problems, ESIR (Energy Saving IP Routing) algorithm is proposed[16]. This algorithm does not consider bidirectional links, which may be a limitation to the applications in the real network. Therefore, the concept of GPE (Green Partial Exportation) is proposed which is based on routing path sharing[17].

The two algorithms last mentioned have a disadvantage that the search space of finding sleep links is narrow. Only a small set of sleep links can be selected when the paths affected by the sleeping of those links are disjoint[18]. As a result, the algorithms have low efficiency of energy saving.

There can be a case that energy saving is more important than the overhead incurred by the sleeping of links. As mentioned earlier, the overhead includes the increase in some ingress-to-egress paths. This paper proposes heuristic algorithms for selecting a maximal set of sleep links to achieve energy saving in awareness of network-level QoS. One feature of our algorithms is to provide a mechanism for allowing users to control the size of the set of sleep links at expense of the increase in path length.

3. Problem Modeling and Basic Concepts

3.1 Problem Definition

This section describes the definition of the problem handled in this paper. For two edge nodes (i.e. ingress and egress routers) n_1 and n_2, notation *flow*< n_1, n_2> represents the incoming traffic of n_1 that is supposed to be delivered to n_2.

PROBLEM: Given a network $G(N, E, W)$ and a number K, find a maximal subset E' of links that can be put to sleep under the following conditions: where K is a user specified parameter that limits the maximum increment of hop count.

(C1) $E' \subseteq E$

(C2) In the network resulting from putting all links of E' to sleep, for each pair of two edge nodes in N, say n_1 and n_2, there is a path from n_1 to n_2 that can deliver traffic *flow*<n_1, n_2>. Also, there is a path from n_2 to n_1 that can deliver traffic *flow*<n_2, n_1>.

(C3) For the path from n_1 to n_2 mentioned in C2, its length does not exceed that of the shortest path for *flow*<n_2, n_1> in G by more than K. Similarly, the length of the path from n_2 to n_1 mentioned in C2 does not exceed that of the shortest path for *flow*<n_2, n_1> by more than K.

3.2 Basic Concepts

As mentioned earlier, the basic protocol for energy saving is that a number of link interface units (NIC's) at the end points of links are put to sleep under the network-level QoS, such as packet delay, packet loss, etc., and rerouting follows. Deactivating NICs leads to saving the energy consumed by those NICs. This enables the IP network to operate the OSPF protocol in normal conditions, and the energy saving protocol to replace the OSPF protocol when the total amount of network traffic is smaller than a certain threshold. In the OSPF protocol, since the state of all links can be monitored using LSA DB (Data Base) at all times[24], the total amount of traffic and the number of paths through each link can be obtained.

The selection of sleep links must be performed to achieve the minimization of energy consumption. The minimization can be achieved to find maximal set of sleep links without violating network-level QoS constraints. Moreover, putting those links into sleep mode requires the changes to the SPTs of the start routers of those links.

DEFINITION: When we check whether a link can sleep, we call it a *target link*. Another link that can satisfy condition ① is called a *delegation link*.

Condition ①: A target link and a delegation link for the target link have the same start router. Additionally, all traffic flows of the target link can be rerouted through the delegation link.

We modify the SPT of the start router of the target link in a way that the start router delegates its role of passing the traffic through the target link to a neighbor router, which is the end router of the delegation link. We call the modified SPT of a router, the *MPT* of the router.

DEFINITION: Given a target link and its delegation link, we call the start router of them a *delegating router*, abbreviated as *DgR*, and the end router of the delegation link a *delegated router,* abbreviated as *DdR*. The process of

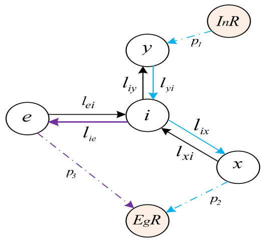

Figure 1. An example for selecting a DgR and a DdR.

delegating the traffic through the target link by modifying the SPT of a DgR is called a *delegation process*, abbreviated as *a delegation*.

An example of a delegation process is described using Figure 1. Nodes labeled *y, i, e,* and *x* are core routers. InR and EgR denote an ingress router and an egress router respectively. We will check whether l_{ix} can be qualified as a target link later in this section. The start node of l_{ix} is router *i*, and the end router *x*.

Thus router *i* has to find an alternative link to replace link l_{ix}, and then delegate the handling task of the traffic through l_{ix} to the end router of an alternative link. Then node *i* can serve as a DgR. Node *e* and *y* are adjacent to node *i*, so links l_{ie} and l_{iy} are considered as alternative links. As shown in Figure 1, router *e* and *y* are neighbors of router *i*, and since alternate link l_{ie} and l_{iy} exist, they form the set of candidate DdR's.

DEFINITION: Consider a target link and its corresponding delegation link in network *G*. The two links are called a *matching pair* in *G*.

DEFINITION: Given a set of the matching pairs in *G*, the set is *set-combinable* if (1) there is not a matching pair of two links that share the same target link, (2) there is not a matching pair whose delegation link is the target link of another pair, and (3) in the graph, resulting from applying all of the delegations represented by the set to *G*, the conditions ② and ③ are satisfied.

Condition ②: Consider a target link, say *l*, and each pair of two edge routers, say n_1 and n_2, in *G*. The length of the path for traffic *flow*<*n1, n2*> in *G'* does not exceed the length of the path for traffic *flow*<*n1, n2*> in *G* by more than *K*. *K* is a user specified parameter that limits the maximum hop count. *G'* is a graph obtained from applying the delegation procedure of *l* to *G*.

Condition ③: In IP networks, the total amount of the flow through each link or node can be limited by constraints (4) and (5) in consideration of network-level QoS. Constraint (4) is an upper bound on the total amount of the traffic through each unidirectional link, and constraint (5) is on the total amount of the traffic processed by each node. Node *j* is a neighbor of node *i*.

$$f_{ij} = \sum_{s\in M}\sum_{d\in M} f_{ij}^{sd} \le \alpha_{\max}\ C_{ij} \quad (4)$$

$$\sum_{j\in A(i)} f_{ij} + \sum_{j\in A(i)} f_{ji} \le \beta_{\max}\ C_i \quad (5)$$

where f_{ij}: the amount of the flow through link l_{ij}

C_{ij}: the maximum processing capacity of link l_{ij}

C_j: the maximum processing capacity of node *i*

α_{max}: the maximum utilization factor of links, $0 \le \alpha_{max} \le 1$

β_{max}: the maximum utilization factor of nodes, $0 \le \beta_{max} \le 1$

M : set of edge nodes,

A(i) : set of *i*'s neighbors

s: source node, *d*: destination node

N: number of nodes

As described in condition ②, we must be able to construct the MPT of the DgR in a way that for each MPT path, the increment in its length is less than or equal to *K* hops in comparison to the corresponding SPT path. For example, in Figure 1, router *e* can be a DdR if it satisfies constraint (6).

$$\text{Length of } (l_{ie}+p_3) \text{ by MPT of node } i \le K + \textit{Length of } (l_{ix}+p_2) \text{ by SPT of node } i \quad (6)$$

When we check whether the traffic through link l_{ix} can be transmitted via delegation link l_{ie} using the MPT paths that meet constraint (6), it is also checked whether the traffic through all the links and nodes of those paths does not exceed upper bounds on their capacities as follows. Each p_i denotes the path between a certain pair of nodes.

If conditions ② and ③ are satisfied, router *i* constructs its MPT by modifying its SPT using the routing tables of router *e* such that the traffic through link l_{ix} is transmitted via link l_{ie}. Finally link l_{ix} can gets placed in sleep mode.

DEFINITION: Consider the set, say *S*, of all matching pairs in *G*. A graph, say *CG=<V, H>*, is called the *bi-combinability graph* of *G* if *V*={*v*| *v*∈ *S*} and *H*={<*a, b*>| *a* ∈ *S*, *b* ∈ *S*, the set consisting of *a* and *b* is *set-combinable*}.

4. Algorithm for a Maximal Selection of Links for Sleeping

The problem of finding a maximal set of sleep links can be considered as a problem of finding the maximum clique that are set-combinable in a bi-combinability graph. The maximal clique problem is NP-hard. In the chapter, we describe two heuristic solutions to this problem.

4.1 A Simple Algorithm

Simple Algorithm: Algorithm for finding a maximal set of links that can sleep for energy saving.

Input: network *G*=<*N, E, W*>, *K*, α_{max}, β_{max}

Output: a set of links that can sleep

Procedure *SimpleMaxLinkSet* (*G, K*, α_{max}, β_{max})

1. Find the SPT's of all nodes of *G*;
2. Find all the *matching pairs* of G for link sleeping;
3. Order those pairs found in Step 2 using a certain criteria;
   Let the ordered sequence of matching pairs be *CE* = {p_1, p_2, ..., p_M};
4. *Res* =ϕ;
5. **for** *i* = 0 **to** *M* **do**
6.   **if** (there is not an element *q*∈ *Res* such that the target link of p_i is that of *q*) **then**
7.   **if** (a union of *Res* and {p_i} is *set-combinable*) **then**
8.   *Res* = *Res*∪ {p_i};
9.   **end if**
10.   **end if**
11. **end for**

Figure 2. A simple algorithm for selecting a maximal set of sleep links.

A simple algorithm of selecting a maximal set of sleep links is described in Figure 2. In Step 2, all matching pairs are found, and in Step 3 through 11, all those pairs are considered one at a time to select links for sleeping. Set *Res* stores sleep links, and it grows incrementally.

In Step 3, especially all of the matching pairs of Step 2 are ordered using a criterion. In the order each matching pair will be taken in consideration for link sleeping at the next step. In IP networks, if a link with the minimal number of paths or the minimal amount of traffic is selected, it is expected to have the minimal effect on the network-level QoS and the existing condition of the network. With the help of LSA DB of OSPF, the links can be arranged in an increasing order of the number of paths or amount of traffic through links, and the result of the arranging can be stored in the sorting DB.

The criteria can be one of the followings:

case 1: the minimum number of paths (called *min_path*)
case 2: the minimum amount of traffic (called *min_traffic*)
case 3: random selection (called *random*)

In Step 6, checking whether element q and p_i have the same target link is performed. If so, the delegation represented by p_i cannot be applied. As in the algorithm of Figure 2, the result of the algorithm is stored in *Res* variable, which represents a maximal set of links for sleeping.

4.2 A Clique-finding based Heuristic Algorithm

Figure 3 describes another algorithm for selecting a maxim set sleep of links. The main feature of this algorithm is to use a methodology for detecting a maximal clique of the bi-combinability graph. The bi-combinability graph is built at Step 2, and then a maximal set of matching pairs that are set-combinability is found at Step 3 through 19. For some node n and some graph Y, function *Degree* (n, Y) is to compute the degree of n in Y. Function *Neighbors* (n, Y) is to obtain the set of all nodes adjacent to n in Y.

Step 3 through 19 of procedure *MaxLinkSet* are a variation of a solution to the maximum clique problem[18]. Set *Res* stores matching pairs that are set-combinable. At Step 5 through 15, a maximal clique whose nodes are set-combinable is obtained in variable S for each node of CG. Variable *Res* stores the largest clique among the ones that have been found so far.

In procedure *CombSubgraph*, variable SS keeps track of a set of nodes that form a clique in a recursive manner. The number of nodes in variable SS grows incrementally one at a time at Step 12. Variable U contains a subset of *Neighbors* (v_i, CG) that have not been selected as members of SS, and the nodes of U will be considered later to check whether they can form a clique with the nodes of SS at Step 7.

Algorithm: Algorithm for finding a maximal set of links that can sleep for energy saving.
Input: network G=<N, E, W>, K, α_{max}, β_{max}
Output: a set links that can be put to sleep

Procedure *MaxLinkSet*(G, K, α_{max}, β_{max})
1. Find the SPT's of all nodes of G;
2. Build the *bi-combinability graph, CG,* from G;
Let the set of nodes of CG be $\{v_1, v_2, ..., v_M\}$;
3. $max = 0$; $Res = \varphi$;
4. **for** i=0 **to** M **do**
5. **if** (*Degree*(v_i, CG) $\geq max$) **then**
6. $U = \varphi$;
7. **for** each $v_j \in$ *Neighbors*(v_i, CG) **do**
8. **if** ($j>i$) **then**
9. **if** (*Degree*(v_j, CG) $\geq max$) **then**
10. $U = U \cup \{v_j\}$;
11. **end if**
12. **end if**
13. **end for**
14 $S = \{v_i\}$;
14. *CombSubgraph*(K, α_{max}, β_{max}, S, U, 1);
15. **end if**
16. **if** ($|S| > |Res|$) **then**
17. $Res = S$;
18. **end if**
19. **end for**

Procedure *CombSubgraph* (K, α_{max}, β_{max}, SS, U, $size$)
1. **if** ($U == \phi$) **then**
2. **if** ($size > max$) **then**
3. $max = size$; $S = SS$;
4. **end if**
5. **return**;
6. **end if**
7. Select a node $u \in U$ of maximum degree in CG such that
8. a union of SS and $\{u\}$ is *set-combinable*;
9. **if** (such a node u can be selected) **then**
10. $U = U - \{u\}$;
11. $NB = \{w |\ w \in$ *Neighbors*(u, CG) $\wedge$ *Degree*(w, CG) $\geq max\}$;
12. $SS = SS \cup \{u\}$;
13. *CombSubgraph* (G, K, α_{max}, β_{max}, SS, $U \wedge NB$, $size$+1)
14. **else**
15. *CombSubgraph* (G, K, α_{max}, β_{max}, SS, ϕ, $size$ + 1)
16. **end if**

Figure 3. A heuristic algorithm for selecting a maximal set of sleep links.

5. Characteristics Analysis and Evaluation

5.1 A Simulation Model

In order to analyze the characteristics our algorithms for energy saving, we constructed a network model for simulation as in Figure 4. In Figure 4, the network consists of 44 nodes and 124 unidirectional links. It is assumed that 22 ingress/egress nodes connect to computers or private networks, and the speed of links in the core network is higher or equal to that of the edge links. In our simulation, the speed of core links is set to be 1.5 times that of edge links.

For simplicity, we also assume that all the links consume energy equally and the input traffic volumes of all ingress routers are the same. However, the output traffic from each ingress router is randomly distributed to all of the egress routers in our experiments. The maximum utilization factors of each link and each node are certain values α_{max} and β_{max} respectively. The parameter η_e in equation (3) is used for the evaluation of characteristics of the algorithm. The algorithm depicted in Figure 2 and 3 was implemented in C++.

5.2 Analysis and Evaluation

5.2.1 Characteristics Analysis of our Algorithms

We evaluated the performance of the algorithms proposed in chapter 4. The performance of our algorithms is measured in terms of the efficiency of link saving, defined in equation (3), and mean increase in the length of paths. An experiment was carried out using the two algorithms of Figure 2 and Figure 3, and random link selection in the network of Figure 4. In the figures of this chapter, notation min_path, min_traffic and random denote the min_path method, the min_traffic method and random selection of Figure 2, respectively. Notation max_set denotes the algorithm of Figure 3.

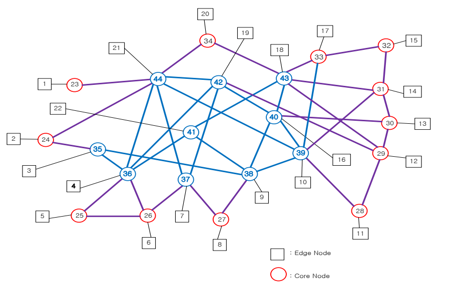

Figure 4. A network model for simulation.

The result of the experiment is summarized in Figure 5 and Figure 6. Here ρ_{ip}, the average input load enforced on each ingress node, is set to 0.3, and α_{max} is set to 0.7. The horizontal axis represents *K,* namely an upper bound on hop count increment. The vertical axis of Figure 5 represents the efficiency of link saving obtained from equation (3). That of Figure 6 represents the average increment of hop count per flow path.

As illustrated in Figure 5, the efficiency of energy saving differs between link selection methods. However, generally the efficiency reaches the maximum value when value of *K* is about 4, and if *K* is above 4, the efficiency is likely to remain constant.

As you can recognize in Figure 5, the efficiency of link saving of the max_set is about 68%, which is superior to other methods. This figure shows that the max_set is the most efficient among the four methods. Other three methods are given ranks in order of min_traffic, mini_path, and random selection. Especially at $K > 4$, it can be seen that the value of *K* does not have a significant impact on link saving efficiency.

Figure 6 illustrates the average increment in hop count of paths under the same condition as in Figure 5. The result shows that when $k \geq 4$ or 5, the average increment of hop count remains almost unchanged, enabling us to infer that there are nearly no paths with hop count increment above 4 or 5 hops. In the order of the average hop count increment per path, the max_set is ranked first, the min_path second, the random selection third, and the min_traffic last. In our experiment, the max_set is best if the efficiency of link saving is most emphasized, but the min_path is best if the hop count increment is the most important concern.

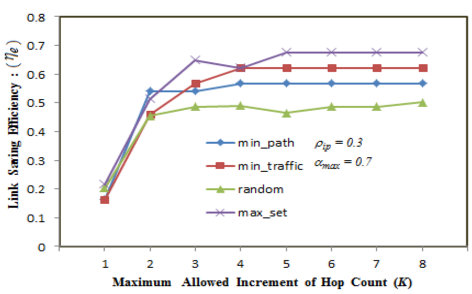

Figure 5. Efficiency of link saving according to the maximum allowed increment of hop count.

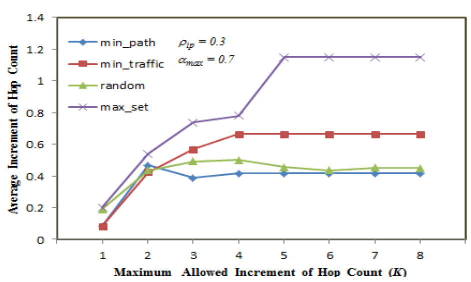

Figure 6. Average increment of hop count according to the maximum allowed increments of hop count.

Figures 5 and 6 show that there is a tradeoff between the efficiency of link saving and hop count increment. If network-level QoS constraints are loose enough so that all flows in IP network can easily satisfy them, we will focus on energy saving efficiency. In such a case, the max_set is shown to be the most effective for link saving.

Figure 7 illustrates the result of an experiment in which there is no limitation on K under various maximum utilization factors of links. Other conditions are the same as in Figure 5. It is shown that the max_set is most efficient in link saving than the other three methods. Also, the approximately 65% of link saving is achieved even when $\alpha_{max} \leq 0.6$.

Figure 8 illustrates the result of another experiment for observing the average increment of hop count, where the conditions are the same as in Figure 7. It is shown that the max_set has a larger increment of hop count than the other three methods. On the contrary, from the viewpoint of link saving, the max_set is more efficient than the other three methods. That means that the max_set gains an advantage in link saving efficiency, but suffers a loss in hop count increment.

Figure 9 shows the characteristics of hop count increment from an experiment carried out under the same condition as in Figure 5. The horizontal axis represents the increments of hop count, and the vertical axis represents the percentages of the paths with specific hop count increments. As stated earlier, the algorithm proposed in this paper aims at maximizing link saving and minimizing the hop count increment induced from path changes. However, hop count increment differs between the methods for target link selection, as shown in Figure 9. In the figure, TE (min-path) represents a method[12] which is frequently referred to in works on link saving.

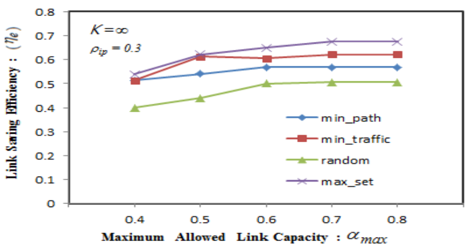

Figure 7. Efficiency of link saving according to the maximum utilization factors of links.

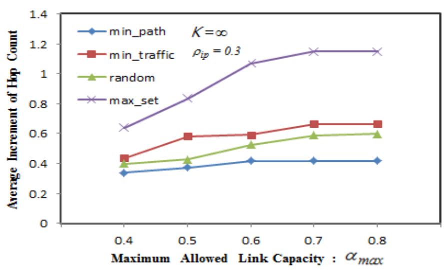

Figure 8. Average increment of hop count according to the maximum utilization factors of links.

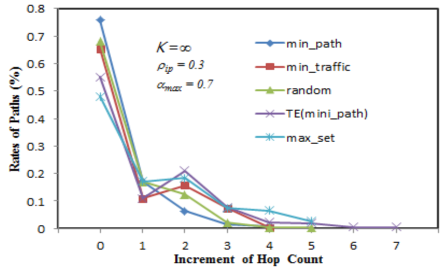

Figure 9. Rates of paths according to hop count increments.

As shown in Figure 9, the maximum increases of hop count are 4 hops in the min_path and the min_traffic, 5 hops in the max_set and random selection, and 7 hops in TE (min_path). However, more than 90% of the paths are within 3 hops at maximum in all the five methods.

Especially the min_path and the max_set induce no hop count increment in the about 75% and 50% of the paths respectively. On the contrary, the TE-energy (min_path) increases the hop count by up to 7 hops, and induces a larger mean and variation of delay due to hop count increment than the other methods.

From the experiment results mentioned above, we insist that our clique-finding based algorithm proposed herein can achieve high energy saving while sustaining network-level QoS (network delay and packet loss, etc.). In the method we can control the degree of link saving by adjusting the value of K, a maximum allowed increment of hop count, at expense of path length. By choosing the max_set and setting K to 3, the efficiency of link saving is expected to be 62%.

We compared our algorithms and previous works in terms of link saving at the various values of α_{max}. The result of the comparison is summarized in Table 1.

According to Table 1, the TE-energy (min_path) is the best in terms of link saving efficiency. This method seems like a better option than the others for real applications, but the rerouting by OSPF is required each time a link falls asleep. Therefore, we insist that the TE-energy (min_path) cannot be conveniently applied to real applications. As recognized in Figure 9, a certain path will undergo the increment of up to 7 hops. In the max_compatibility

Table1. Comparison of link saving with other methods

method \ α_{max}	0.4	0.5	0.6	0.7	K
Our algorithm (max_set)	0.51	0.65	0.65	0.65	$K = 3$
Our algorithm (min_path)	0.51	0.54	0.54	0.54	$K = 3$
Our algorithm (min_traffic)	0.49	0.56	0.57	0.57	$K = 3$
Our algorithm (max_set)	0.43	0.49	0.51	0.51	$K = 2$
Max_compatibility[16]	0.43	0.43	0.43	0.43	$K = 2$
TE-energy (min_path)[12]	0.59	0.68	0.81	0.89	$K = \infty$

method, the hop count of a path increases by at most 2 hops [16]. However, the efficiency of link saving is lower than the max_set under same condition. The max_set has several advantages over the previous methods. We can adjust a limit on the allowed increment of hop count. This feature results in the higher efficiency of link saving than the max compatibility method. We can also guarantee the network-level QoS by adopting a proper value of *K*. Also, the sleeping of links in our algorithms does not cause the entire rerouting by OSPF.

5.2.2 Network QoS Analysis

The network-level QoS described in this section includes the delay and packet loss of traffic flows between edge nodes. The delay and packet loss will vary according to the changes made to paths and loads on links during the execution of our algorithms.

Figure 10 illustrates a simplified model of a certain traffic flow in IP networks. In the figure, let us assume that node 1 and n are edge nodes, and all other nodes are core nodes. An arbitrary flow can be specified by a path between edge nodes, which is determined by the OSPF routing. Due to the modifications of paths during the execution of our algorithm, the length of some paths will increase compared to the result of existing OSPF routing, and the load on links of those paths will increase or decrease accordingly.

If the input of a node in the network is restricted to be below the maximum amount that the node can process and the internal memory of the node is designed to be large enough, there will be no packet loss. Therefore, we consider only the delay of a flow as a network-level QoS. For a flow between a source node and a destination node, its path can be obtained using Dijkstra's shortest path algorithm. The links on the path will carry some amount of traffic. In this case, the amount of traffic through link *k*, denoted by ρ_k, can be defined as follows.

$$\rho_k = \sum_{(s,d)} \rho_{sd} \ \forall (s,\ d)\ s.t.\ k \in P(s,\ d) \qquad (7)$$

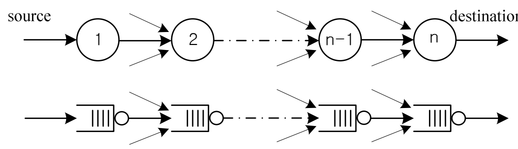

Figure 10. A simplified queuing model of a flow in IP networks.

$P_{(sd)}$ is the flow path between source node *s* and destination node *d* by Dijkstra's algorithm. ρ_{sd} denotes the amount of traffic through the flow path from *s* to *d*.

In the execution of our delegation algorithm, once a link *i*, say l_i, has fallen asleep, the loads on some links will be changed accordingly. Let $\Delta\rho_{ik}$ be the amount of the load change on link *k* due to the sleeping of l_i. $\Delta\rho_{ik}$ is the difference between the new amount of traffic on link *k* due to the sleeping of l_i and the initial amount of traffic determined by Dijkstra's algorithm. $\Delta\rho_i$ is defined as the set of all $\Delta\rho_{ik}$'s for the sleeping of l_i as follows:

$$\Delta\rho_i = \{\Delta\rho_{i1},\ \Delta\rho_{i2},\ \cdots\cdots\cdots\ \Delta\rho_{iL}\} \qquad (8)$$

By using equation (9), we can check whether each link l_k = <a, b> in the network can sleep

$$\rho_k + \sum_{i \in L} \Delta\rho_{ik} \le \alpha_{\max} C_{ab} \ \forall k \in E \qquad (9)$$

In equation (8) and (9), the traffic volume of sleep links gets zero. It can be recognized that the amounts of traffic through individual links are restricted by α_{max}.

For analyzing the characteristics of the delay of paths, let us assume that packets arrive randomly at source nodes, the length of packets is random, and the nodes (routers) on flows are independent of each other. Then the task of processing the traffic at each node can be modeled as M/M/1queuing model. The normalized average delay, denoted by D_{mean}, of path *p* between a source and a destination is defined as equation (10) [22], where *p* consists of link 1 through link n.

$$D_{mean}(p) = \sum_{i=1}^{n} \left(\frac{\rho_i}{1 - \rho_i}\right) \qquad (10)$$

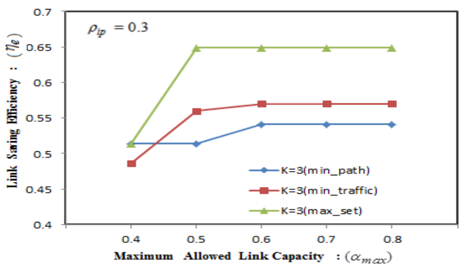

Figure 11. Efficiency of link saving according to the maximum utilization factor of links.

When the result of executing our algorithm is compared to the initial result of OSPF algorithm from path *p*'s point of view, it is seen that changes are made to each link ρ_i and *n*, the number of nodes of path *p*. Those changes have come from modifying the paths that are involved during the construction of MPT.

In our algorithms, we limit the maximum hop increment of paths to *K*, and the utilization factor of links to α_{max}. The maximum increment in the delay of a path, denoted by ID_{mean}, resulting from our algorithms, can be defined as equation (11).

$$ID_{mean}(p) = K \bullet (\frac{\alpha_{\max}}{1 - \alpha_{\max}}) \quad (11)$$

We performed an experiment to show the characteristics of delay for various values of α_{max}. The result of the experiment under $\rho_{ip} = 0.3$ and $K = 3$ is shown in Figure 11. It is observed that the efficiency of link saving remains almost the same for each method when $\alpha_{max} \geq 0.6$. According to equation (11), when we set $\alpha_{max} = 0.6$ and $K = 3$, the maximum average increment in the delay of paths is less than 6 times the average size length of packets. Owing to this property, we can say that the delay of a path does not cause a problem in the routers equipped with high speed processers. Also, as illustrated in Figure 9, the 50~70% of all the paths do not undergo any increment of hop count according to our heuristic methods, and have only delay variations due to the changing of the link utilization factor.

From the above results, we conclude that the requirement for path delay under network-level QoS constraints can be satisfied by adjusting α_{max} and *K*.

6. Conclusion

IP networks are usually designed based on peak time traffic without considering the variations of daily and timely network traffic. In this paper, we presented two heuristic algorithms for maximally selecting sleep links for energy saving under network-level QoS constraints in IP networks. Also we evaluated the characteristics of the algorithms.

We compared our three methods together with random selection. From the results of the simulation, we could see that in terms of the efficiency of link saving, the max_set is best among the four methods, and its efficiency is about 65% under $K=3$, $\alpha_{max} = 07$. Under the same network condition, those methods except for the max_set are ranked in order of the min_traffic, the mini_path, and random selection. Their efficiencies of link saving are 56%, 54%, and 48% respectively. Our max_set method was shown to find a larger set of sleep links than previous works, resulting in more energy saving.

Main features of our algorithms are to find a maximal set of sleep links through the delegation process and to allow a user-specified value for limiting the increase in hop count of paths. The delegation process enables minimizing the overhead incurred by the rerouting after putting links to sleep, similarly to a technique of the previous work. The sleeping of links causes the increase in hop count of SPT paths. There can be cases in which we emphasize energy saving more than the increase of path length. Our algorithm provides ISP with a mechanism for achieving the required level of energy saving by adjusting the delay of paths.

In the near future, we will perform a deeper research into various issues such as compatibility with OSPF protocol in large IP networks, practical application, etc.

7. Acknowledgement

This research was supported by Hankuk University of Foreign Studies Research Fund of 2015 and 2016, and based on the work of Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2011-0010070).

8. References

1. Roth KW, Goldstein F, Kleinman J. Energy consumption by office and telecommunication equipment in commercial buildings (Vol I: Energy consumption baseline). Arthur D. Little Inc.: Cambridge; 2002.
2. Bolla R, Bruschi R, Davoli F, Cucchietti F. Energy efficiency in the future internet: a survey of existing approaches and trends in energy-aware fixed network infrastructures.

IEEE Communications Surveys & Tutorial. 2011 Jul; 13(2):223–44.

3. Recupero DR. Toward a green internet. Science. 2013 Mar; 339(29):1533–34.
4. Gao S, Zhou J, Aya T, YamanakaN. Reducing network power consumption using dynamic link metric method and power off links. IEICE Technical Report (ICSS). 2009 Jun; 109(86):85–9.
5. Restrepo JCC, Gruber CG, Machuca CM. Energy profile aware routing. IEEE International Conference on Communications Workshops, Dresden; 2009. p. 1–5.
6. Silva T, Arsenio A. A survey on energy efficiency for the future internet. International Journal of Computer and Communication Engineering. 2013 Sep; 2(5):589–95.
7. Zhang Y, Chowdhury P, Tornatore M, Mukherjee B. Energy efficiency in telecom optical networks. IEEE Communications Surveys & Tutorials. 2010 Jul; 12(4):441–58.
8. Bolla R, Bruschi R, Davoli F, Cucchietti F. Energy efficiency in the future internet: a survey of existing approaches and trends in energy-aware fixed network infrastructures. IEEE Communications Surveys & Tutorials. 2011 Jul; 13(2):223–44.
9. Bianzino AP, Chaudet C, Rossi D, Rougier J-L. A survey of green networking research. IEEE Communications Surveys & Tutorials. 2010 Dec; 14(1):3–20.
10. Dharmaweera MN, Parthiban R, Ahmet Y¸ Sekercioˇglu. toward a power-efficient backbone network: the state of research. IEEE Communications Surveys & Tutorials. 2014 Jul; 17(1):198–227.
11. Idzikowski F, Chiaraviglio L, Cianfrani A, L´opezVizca´ıno J, Polverini M, Ye Y. A survey on energy-aware design and operation of core networks. IEEE Communications Surveys & Tutorials. 2015 Dec; PP(99):1–46.
12. Chiaraviglio L, Mellia M, Neri F. Reducing power consumption in backbone networks. IEEE International Conference on Communications, Dresten; 2009. p. 1–6.
13. Gupta M, Singh S. Dynamic ethernet link shutdown for energy conservation on ethernet links. IEEE International Conference on Communications, Glasgow; 2007. p. 24–8.
14. Cianfrani A, Eramo V, Listanti M, Marazza M, Vittorini E. An energy saving routing algorithm for a green OSPF protocol. IEEE International Conference on Computer Communications Workshops, San Diego; 2010. p. 1–5.
15. Cianfrani A, Eramo V, Listanti M, Polverini M. An OSPF enhancement for energy saving in IP networks. IEEE International Conference on Computer Communications Workshops, Shanghai; 2011. p. 325–30.
16. Cianfrani A, Eramo V, Listanti M, Polverini M, Vasilakos AV. An OSPF-integrated routing strategy for qos-aware energy saving in IP backbone networks. IEEE Transactions on Network and Service Management. 2012 Mar; 9(3):254–67.
17. Chiaraviglio L, Cianfrani A, Listanti M, Mignano L, Polverini M. Implementing energy-aware algorithms in backbone networks: a transient analysis. IEEE International Conference on Communications, London; 2015. p. 142–48.
18. Pattabiraman B, Patwary MMA, Gebremedhin AH, Liao W-K, Choudhary A. Fast algorithms for the maximum clique problem on massive sparse graphs. 10th Workshop on Algorithms and Models for the Web Graph, Cambridge; 2013. p. 156–69.
19. Bianzino AP, Chaudet C, Larroca F, Rossi D, Rougier J-L. Energy-aware routing: a reality check. IEEE Globecom Workshops; 2010. p. 1422–27.
20. Garroppo RG, Giordano S, Neucioni G, Pagano M. Energy aware routing based on energy characterization of devices: solutions and analysis. IEEE Interantional Conference on Computer Communications Workshops, Kyoto; 2011. p. 1–5.
21. Restrepo JCC, Gruber CG, Machuca CM. Energy profile aware routing. . IEEE International Conference on Computer Communications Workshops, Dresten; 2009. p. 1–5.
22. Kleinrock L. Queueing systems Vol 1: Theory A. Wiley-Interscience Publication: Newyork; 1975. p. 89–110.
23. Idzikowski F, Chiaraviglio L, Portoso F. Optimal design of green multi-layer core networks. Proceeding of the ACM e-Energy. Madrid; 2012. p. 1–9.
24. IETF RFC 2328 document [Internet]. Available from: http:// www.ietf.org/rfc/rfc2328.txt.