Energy efficient network design tool for green IP/Ethernet networks

Hidetoshi Takeshita, Naoaki Yamanaka *, Satoru Okamoto, Sho Shimizu, Shan Gao
Department of Information and Computer Science, Faculty of Science and Technology, Keio University, 3–14–1 Hiyoshi, Kohoku-ku, Yokohama, 223–8522, Japan

1. Introduction

The continued growth in the Internet service means that network throughput must be raised significantly. In particular, video data transfer and rich web services are generating heavy traffic loads. This will only further increase the energy consumed by IT equipment. Fig. 1 shows the energy consumption forecasts for IT equipment in Japan. According to this figure, power consumed by routers will increase dramatically [1]. In 2010, 13,000 million kwh/year (13 TWh/year) will be consumed. That is equivalent to two nuclear power plants.

To reduce network energy consumption, several techniques are being developed. The first one, by IEEE, is the Energy Efficient Ethernet (EEE). This technique changes link speed according to the data traffic demand. The second is to employ low power voltage LSIs. The third is to use DC power supply to reduce AC/DC conversion loss. The fourth is to use a network virtualization technique such as virtual servers and virtual routers.

We proposed the energy efficient network control concept named “MiDORi (Multi- (layer, path, and resources) Dynamically Optimized Routing).” “Midori” is the Japanese word for “green” [2]. To reduce the power consumption, we control the power off/on state of links and nodes. This operation is controlled by a Path Computation Engine/Element (PCE). A low energy consumption network design engine in the PCE determines the optimal network topology (path routes). IP routers/Ethernet switches are controlled by this PCE via GMPLS (Generalized Multi-Protocol Label Switching) based link power on/off control protocols [2].

This paper proposes a new energy efficient network design tool for IP/ Ethernet networks. The proposed tool...
Fig. 1. Energy consumption for IT equipment from Asami and Namiki, ECOC 2008, Tu.4.A.3, Brussels, Belgium, Sept. 23, 2008 [1].

Fig. 2. Basic network configuration using PCE for path control in the AS.

aggregates the traffic and powers down empty links by monitoring the amount of traffic being transferred in the network. In other words, by activating the fewest possible links at any one time, the network topology can be dynamically reconfigured in an energy saving manner.

The rest of the paper is organized as follows. In Section 2, the concept of the low power consumption network architecture is presented. Section 3 details the energy efficient network topology generation algorithm and its evaluation. Required link on/off protocol is proposed in Section 4. Section 5 presents the prototype Gigabit Layer-2 switch. Finally in Section 6, we summarize the paper.

2. Low power consumption network architecture

Today’s network structure is shown in Fig. 2. The network consists of ASs (Autonomous Systems) which includes tens to hundreds of routers with controllers. The network topology (path routes) is controlled by PCE. PCE calculates network topology to match traffic demands and network conditions. Our research target is a high-performance PCE that can efficiently calculate the network topology with the lowest power consumption and control the power off/on states of links and nodes.

The key energy saving approaches are as follows [3]:

2.1. Interface power off

Some parallel links and single links are deactivated if the traffic demand is low. In that case, both sides of the interface are automatically powered off. In addition, PCE considers these links as sleeping links; they physically exist but are inactive.

2.2. Node power off and partial power off

Transits routers that are not currently needed are powered off, low traffic routers will be partially powered off. Partially powering off a router is realized by partially deactivating the switch fabric and controllers.

Proposed design method is illustrated in Fig. 3 for the case of 6 nodes and \( n \) links. First, with regard to on/off states, total link combination number is \( 2^n \). The initial topology has all links powered off, so nodes cannot communicate with each other. Next, one link is powered on, this pattern has \( n \) combinations. Additional links can be powered on one by one. The network topology pattern that can carry all traffic demands is selected. With regard to routers, router power consumption is proportional to the traffic amount, and the traffic of a low load router can be moved to another router and the empty router can be powered off.
A simple diagram of parallel links is shown in Fig. 4. In the proposed network, parallel link number can be automatically changed to meet traffic demand. In this example, power savings equivalent to 10 Gb/s × 4 links is possible. However, the following QoS requirements may alter this process.

1. All traffic can be carried with proper margin.
2. Maximum hop number of paths can be guaranteed.
3. Disjoint Multi-route divergence is assured to achieve reliable communication.

QoS guarantees and minimal link topology are identified by calculating $2^n - 1$ topology patterns and checking each to confirm if they meet the QoS restrictions. A flowchart of the proposed method is shown in Fig. 5. The proposed algorithm creates all combinations of the network link topologies. This type of combination calculation problem is called the set cover problem. To get the optimum solution, all candidate patterns must be searched. Therefore, to get the optimum energy efficient topology, a high speed network topology generation method is essential.

3. Energy efficient network topology generation algorithm and evaluation

Combinatorial algorithms can be applied to the above problem, which is a variant of the set-covered problem. The calculation time of the problem rapidly increases with network scale [4]. We propose here a method that generates all combinations rapidly because no greedy algorithm can be assured of obtaining the optimum solution [5–7]. Beeler’s algorithm generates all combinations by picking $k$ outcomes from $n$ possibilities [8,9], and these combinations can be expressed in $n$-digit binary sequence in ascending order [10,11].

The proposed algorithm represents a network’s topology as an $n$-digit binary sequence where links are modeled as “1” for power on and “0” for power off. The $n$-digit binary sequence, in ascending order, is divided into several groups to minimize the computation time. The proposed algorithm generates the head $n$-digit binary sequence, divides it into groups, which we call seed data, and interactively generates the next $n$-digit binary sequence from the seed data [10,11]. This approach allows DAPDNA-2 to implement the proposed algorithm and automatically produce the link sets (network topology) by parallel pipeline calculation [12,10,13–15,11].

Fig. 6 shows the pipeline operation given 3 power-on links from a total of 6 links. The combination number, which is calculated as $6C_3$ (picking 3 outcomes from 6 possibilities), is 20 and is divided into 4 groups. The 1st, 6th, 11th and 16th data groups are seed data [10,11]. The DNA matrix in DAPDNA-2 outputs 2nd, 7th, 12th, and 17th data which are next seed data.

3.1. Beeler’s algorithm and any-order pattern algorithm

Beeler, Gosper, Schroppel proposed an algorithm that generates all combinations where $k$ outcomes are picked from $n$ possibilities [8,9]. These combinations can be expressed as $n$-digit binary sequences. For example, the binary sequence “010110” represents that link numbers (2, 3, 5) are power on when $n = 6$ as shown in Fig. 6. Combinations can be ordered in ascending order. The power on sequence (2, 3, 5) is smaller than sequence (2, 4, 5), because the binary sequence “010110” is smaller than the binary sequence “011010”. Beeler’s algorithm can generate all sequences from “000111” to “111000” in order. Details of the algorithm are as follows [11].

There are five steps to generate the next binary sequence $Y$ from the original binary sequence $X$.

1. Let $S_1$ be given as all bits are unset except for the least significant bit 1 in sequence-$X$.
2. $R_1 = X + S_1$.
3. Let $S_2$ be given as all bits are unset except for the least significant bit 1 in sequence-$R_1$.
4. $R_2 = (S_2/S_1) \gg i$ (i: shifts right i-bit)
5. $Y = R_1|R_2$, and $Y$ is next to $X$.

When $n = 6$, $k = 3$, $X = 001110$, for example, $Y$ is calculated as follows.

1. $S_1 = 000010$
2. $R_1 = X + S_1 = 010000$
3. $S_2 = 010000$
4. $R_2 = (S_2/S_1) \gg 1 - 1 = 001000 \gg 1 - 1 = 000100 - 1 = 000011$
5. $Y = R_1|R_2 = 010011$. 
Our prior paper proposed an algorithm that generates any-order patterns in sequences that are sorted in ascending order [12]. In general, a non-ordered sequence is generated by the following equation, assuming we pick \( k \) power on links from \( n \) links.

\[
C_k = \sum_{i=0}^{n-1} C_{k-1}.
\]

(1)

To get the \( m \)-th sequence, find the smallest \( x_1 \) that satisfies the following inequality.

\[
\sum_{i=k=1}^{x_1} C_{k-1} \geq m \quad (k - 1 \leq x_1 \leq n - 1)
\]

(2)

\( x_1C_{k-1} \) means the sequence whose most significant bit “1” is \( x_1 \)-th bit and there are \( k - 1 \) “1”s between 1st and \( (x_1 - 1) \)-th bit because there are \( k \) “1”s in total. Hence, the \( x_1 \)-th bit of the \( m \)-th sequence is “1”. The \( m \)-th pattern corresponds to the \( m - \sum_{i=k+1}^{x_1} C_{k-1} \)-th sequence in \( x_1C_{k-1} \). Thus \( m \) is replaced as follows.

\[
m \rightarrow m - \sum_{i=k=1}^{x_1} C_{k-1}.
\]

(3)

Next, find the smallest \( x_2 \) that satisfies the following inequality.

\[
\sum_{i=k=2}^{x_2} C_{k-1} \geq m \quad (x_2 \leq x_1 - 1)
\]

(4)

\( x_2C_{k-1} \) means the sequence whose most significant-bit “1” is the \( x_2 \)-th bit and the combination number is the number of combinations including \( k - 2 \) “1”s between the 1st and the \( (x_2 - 1) \)-th bit. Hence, the \( x_2 \)-th bit of the sequence is “1”. \( x_1, x_2, \ldots, x_6 \) can be obtained by repeating this process \( k \) times. Setting the corresponding bit to “1” yields the \( m \)-th sequence.

For example, the 6th pattern \( (m=6) \) in \( 6C_{3} \) can be obtained as follows.

\[
6C_{3} = 2C_{2} + 3C_{3} + 4C_{4} + 5C_{5} + 6C_{6} = 1 + 3 + 6 + 10.
\]

Apply Eq. (1) to \( 4C_{2} \) because \( 4C_{2} \) includes the 6th pattern. Hence, \( x_1 = 4 \), \( m \rightarrow 2 \).

\[
4C_{2} = 3C_{1} + 4C_{2} + 5C_{3} = 1 + 2 + 3.
\]

Apply Eq. (1) to \( 2C_{1} \) because \( 2C_{1} \) includes the 2nd pattern. Hence, \( x_2 = 2 \), \( m \rightarrow 1 \).

\[
2C_{1} = 1C_{0} + 2C_{1} = 1 + 1.
\]

The 1st pattern corresponds to \( 0C_{0} \). Hence, \( x_3 = 0 \). Set the corresponding bit to 1, which yields the 6th pattern “010101”.

3.2. Implementation on DAPDNA-2 and performance evaluations

Our proposed design tool expands the exhaustive search limit within reasonable calculation time by implementing the calculation algorithm as a parallel process. There are many reconfigurable processors [16]. We selected DAPDNA-2 and implemented the above algorithms to elucidate DAPDNA-2’s performance; to goal was to realize energy efficient network design [11]. DAPDNA-2 is a dynamically reconfigurable processor, and combines a Digital Application Processor (DAP) with Distributed Network Architecture (DNA). DAP is a dual-core RISC processor, and DNA is the dynamically reconfigurable core with 376 processing elements (PEs). The hardware configuration of DAPDNA-2 can be dynamically changed within one clock to provide the optimal circuitry for an application on demand.

The proposed algorithm applies 3 processes.

(1) Calculate \( m \)-th pattern (seed data).
(2) Execute Beeler’s algorithm.
(3) Using minimum link set topology, check the bandwidth limit.

Process (1) is executed by DAP, and processes (2) and (3) are executed by DNA. DAP generates all combinations and divides them into several groups to minimize the computation time, and calculates the head \( n \)-digit binary sequence (seed data) for each divided group, and stores seed data in main memory. DNA reads these results from main memory and executes Beeler’s Algorithm (generates next \( n \)-digit binary sequence from seed data continuously), calculates minimum link set topology, and checks the bandwidth limit by parallel pipeline calculation.

Let \( n \) be the total number of links and \( k (\leq n) \) be the number of power on links. In our implementation, \( n \leq 32 \) because one word is 32-bits long in PE (Processing Element). For example, when \( n = 7, k = 2 \), we generate all 21 combinations from “0000011” to “1100000”. Each network topology (link power off/on state) is represented by a 32-bit word. When the calculated topology is the minimum link set, its bandwidth limit is checked. PCE chooses the resulting topology and controls the link power off/on states as appropriate. Note that the proposed method can identify the most energy-efficient network topology with reasonable calculation time.

We compare the execution time of DAPDNA-2 (166 MHz) to that of a Pentium 4 (2.8 GHz). Let \( k \) be the number of power-on links, and \( n \) be the total number of links.

Fig. 7(a) plots the comparison of theoretical execution time and experimental execution time, when the value \( k \) (power on links number) = 8 links for small network sizes. Black plots represent Beeler’s method on Pentium 4, and white plots represent the proposed method on DAPDNA-2. Circle plots represent the theoretical execution time, and square plots represent measured execution time. When \( n = 30 \), DAPDNA-2 is 40 times faster than the Pentium 4 combination due to the parallel processing and pipeline operation.

Fig. 7(b) plots the theoretical execution time of the conventional method and the proposed method for an evaluation of scalability. Calculation conditions are the same as in Fig. 7(a), where power on links number \( k \) is 26% of total link number \( n \). Black plots represent Beeler’s method on Pentium 4, and white plots represent the proposed method on DAPDNA-2.
3.3. Evaluation of network power saving

We evaluated the link power off rate and network power saving rate for the NSFNET topology model which is composed of 14 nodes and 21 links, and each node has one 320 Gbps-router. The evaluation conditions are as follows.

- Link capacity: 1.0 for all links.
- Inter-node traffic: uniformly generated.
- Shortest path search: Dijkstra’s algorithm.

The maximum power consumption of the NSFNET topology model is 15.7 kw; the power consumption of 320 Gbps-router which has eight 40 Gbps-POS line interface cards is 2612 W, the packet switch fabric consumes 224.4 W, the board-to-board inter-connection takes 0.8 W, and 40 Gbps-POS line interface card takes 298.3 W [17].

Fig. 8 shows the link reduction rate and power saving rate versus the traffic load for randomly generated traffic patterns. The maximum power off link rate is 38% and the power saving is nearly 30%, which is calculated by dividing the maximum power saving, 4.8 kw, by the maximum power consumption 15.7 kw.

4. Link power on/off control protocol

The low energy consumption network design engine in PCE outputs the optimum network topology. IP routers/Ethernet switches are currently controlled by PCE using GMPLS. Since GMPLS does not provide for powering links on/off, a new control protocol, a GMPLS extension, has been proposed [2]. This protocol is an extension of OSPF (Open Shortest Path First), LMP (Link Management Protocol), and RSVP (Resource Reservation Protocol).

The OSPF extension offers a new logical (TE: traffic engineering) link state for the link power up/down state. Fig. 9(a) shows the basic OSPF extension images. In the current OSPF implementation, if a physical link fails, TE-link also fails and is removed from the link database. As a result, PCE is not able to recognize quiescent links. Quiescent links should be used when the traffic load becomes large and/or network failure occurs. The proposed OSPF extension keeps inactive TE-links in the link database. Therefore, quiescent TE links can be used when a request is received to recalculate the path network topology.

Link power off/on is controlled by RSVP and/or LMP protocol extension as shown in Fig. 9(b). To realize protocol communication between adjacent nodes, an out-of-fiber type communication channel is assumed. This is because if in-fiber type communication channel sleeps when the communication channel sleeps. Fig. 9(c) shows two RSVP extensions. One is an LSP (Label Switched Path) status flag extension. The added flag indicates whether the LSP is constructed on quiescent link(s) or not. Another is LSP reconfiguration escalation from lower layer LSP to upper layer LSPs. If the server layer TE-link enters the “power off” state, upper layer LSP segment can detect the status change and send NOTIFY message to the appropriate LSP ingress node. The ingress node changes the LSP status to “power off”. To retain the LSP for traffic delivery, a new LSP will be automatically provisioned as shown in Fig. 9(c).

5. Prototype system development

We developed a prototype Gigabit Layer-2 switch to demonstrate the energy efficient IP/Ethernet network, and
constructed a MiDORi network test-bed. Fig. 10 shows the network structure. The power controlled Gigabit Layer-2 switches are connected to MiDORi PCE. Traffic information is reported to the PCE, PCE calculates the most energy efficient network topology. The PCE hosts the proposed MiDORi GMPLS extension protocol.

Fig. 11 shows the prototype Gigabit Layer-2 switch which has eight 1Gbps-Ethernet (1GE)-links. The Gigabit layer-2 switch supports following features.

- The switch can be controlled remotely via the telnet interface.
- The switch can determine the traffic level of each LSP (VLAN) and 1GE-link.
- Link power on/off state can be controlled remotely.
- Power on/off of each switch fabric can be controlled remotely.
- Power consumption of the switch can be monitored via a command and a current meter.

We implemented a network topology with 6 switches and 31 links. The low traffic condition exhibits 80% less traffic than the high traffic condition. Using the proposed method, 10 links (32%) can be powered off in the low traffic condition. The results show that the proposed network, Gigabit Layer-2 switch, and algorithm realized effective power control. This is sufficient to confirm the algorithm output the optimum topology.

6. Conclusion

The key to developing MiDORi network technologies, is calculating the optimal network topology. In order
to obtain the optimal network topology (lowest power consumption), we have proposed a fast calculation method that runs on the reconfigurable processor DAPDNA-2 of IPFlex Inc. The proposed method enables parallel pipeline implementation because all combination patterns are generated by divided some groups. All combination patterns in divided groups are generated from the seeds data of each group. Experiments showed that the execution time of the proposed algorithm increases with \( n \) (Total number of links in the network) because DAPDNA-2 calculates in parallel using pipeline operation. When \( n = 30 \) and power on links \( k = 8 \), DAPDNA-2 is 40 times faster than the Pentium 4 alternative.

PCE uses the resulting optimum topology to power down unused physical links and routers/switches. We have also proposed control protocols based on extensions of GMPLS OSPF, RSVP-TE, and LMP. In addition, a prototype of the MiDORi Network was successfully demonstrated. Our calculation method can identify the ideal network topology in terms of power savings. The evaluation results show that network power savings of up to 30% can be realized for the NFSNET topology model.

Acknowledgment

This work is supported by the PREDICT program of the Ministry of Internal Affairs and Communications (MIC) of Japan. This work is also supported in part by a Grant-in-Aid for the Global Center of Excellence for Leading-Edge Platform on Access Spaces from the Ministry of Education, Culture, Sport, Science, and Technology in Japan.

References