Reducing Network Power Consumption Using Dynamic Link Metric Method and Power Off Links

Shan Gao[1], Jia Zhou[1], Naoaki Yamanaka[1]
Contact: gao@yamanaka.ics.keio.ac.jp

Department of Information and Computer Science, Keio University
3-14-1 Hiyoshi, Kohoku, Yokohama, 223-8522, JAPAN

Abstract

As the high-speed network technologies developing, the traffic in the Internet increases rapidly. The network power consumption has come up to 10¥% of the worldwide energy consumption. In order to reduce this kind of consumption, we propose a novel method, which dynamically modifies the weights of links and makes traffic gather to certain links and then set some links of which the load no traffic Power-Off. We also did much work in the network simulations, and the result shows that the method has a good effect in reducing the network power consumption.

I. Introduction

In the past decade, the Internet has developed from the scope of the LAN into a particularly important communications media. It has become an integral part of the world. As the rapidly increasing number of the Internet users and the network transmission speed, Internet Service Providers have been expanding the capacity of routers and switches and increasing network bandwidth to satisfy the users' demands. Nevertheless, the increase of routers and switches bring a sharp rise in electricity consumption. As in the United States, 2¥% of nation's electricity consumption was attributed to the Internet in 2001. But by 2005, the Internet's power consumption has gone up to 10¥% of the entire power consumption [1]. As a result, how to use the network resources effectively, reduce power consumption in the Internet and establish "Green Internet" is a hot issue today.

The IEEE has formed an Energy-Efficient Ethernet study group that aims at developing a draft standard system to reduce the energy consumption of the Internet. The Energy-Efficient Ethernet group is concerned about not only the reduction of energy consumption in servers, desktop computers, laptop computers, but also the energy consumption in the local area network switches, routers and other network equipments. In particular, the network, which has the low utilization, should reduce more energy consumption. As the expectation of the Energy-Efficient Ethernet group, in the United States it can save 300,000,000 to 450,000,000 U.S. dollars of energy costs from only Ethernet ports, and for all users of the nation it's expected to save 25 billion U.S. dollars annually. If the Global Energy-Efficient Ethernet plans are implemented, the annual energy cost savings can be up to 75 billion U.S. dollars [2]. We can see that the reducing of power consumption can not only save the economic cost, what's more, it plays an important role against the global warming, in reducing carbon dioxide emissions and in other aspects of environmental protection.

Several traditional researches are focus on changing the speed of the NIC (Network Interface Card). In [3], a MAC layer protocol is proposed for changing link speed in the NIC. However, the effect of power consumption is limited, if only change the link rate of NIC. The more effective approach, link-shutdown method, is widely researched for reducing power consumption. In [4], the author proposed a network design method by sleeping ports of switch and shutting down links for minimizing the number of link utilization. This proposal calculates all patterns of network configurations. Therefore, a high-speed processor is required to execute this method. A more effective, more simple link shutdown method is proposed in this paper.

The rest of the paper is organized as follows. In section ¥ref{proposed}, we describe network status and explain the details of our method using dynamic link metric. Simulation results are provided to evaluate the performance of the proposed algorithm is Section ¥ref{performance}. Finally, Section ¥ref{conclusion} presents our conclusions.

II. Procedure for Paper

Network Architecture and State

In order to expand the bandwidth and capacities of link traffic load, Internet Service Providers are inclined to establish more links between nodes. So it could exist multiple links between two nodes in the network. The approach we proposed is aim at the network that has multiple links between nodes.

Usually in the busy time of the network, multiple links are very good for balancing traffic load and avoiding congestion. During the busy period the link utilization can reaches 80¥% commonly. However, in low-peak periods of network, such as 2:00 to 4:00 in the early morning, the traffic load is low and amount to only 20¥%Å `30¥% of links' capacity. But in that time, if the load balancing strategies is adopted, all the traffic load is fairly carried by links. All links are still power-on. And [6] have shown that there is little or no different in the power consumption between low-load and high-load states in links [5]. So the power consumption in low-load period is as the same as the network is busy. However, it's possible that all the traffic load can be carried by several links in the network. It's not necessary to set all link power-on.

Mechanism

We proposed an approach to reduce the power consumption in the low-load period of network. The approach is that we aggregate the traffic load to certain links and raise the utilization rate of these links. To other idle links, which no traffic pass through, we can set them power-off. In this way, we raise the utilization of links and power off the idle links so that we can save the power consumption of idle links. In addition, we set a threshold of the load of link in order to avoid congestion. While the load of link is below the threshold, it's considered that no traffic congestion happens.

In the Ethernet, the OSPF protocol is proposed. According to the OSPF protocol, traffic adopts the shortest path algorithm to choose paths. And the shortest path algorithm executes based on the weight of paths. In a word, the weight of link decides whether the link carry traffic or not. Therefore, we modify the weights of links, dynamically increase or decrease weights, to change the paths of traffic and achieve the aggregation of traffic.

Dataflow

The main process of the approach shows as follows.

Step1 Firstly, each node looks up any links that link to another node. Meanwhile, the traffic load of these links can't exceed the threshold. If the links that meet the conditions exist, go to next step.

Step2 Compare the load of links and choose one link which has larger load to decrease the weight.

Step3 Compare the load of links and choose one link which has larger load to decrease the weight.

Step4 After getting new information of weights of links, each node re-calculate the paths to reach other nodes. According to the new path vectors, the traffic re-chooses the transmission paths.

Step5 Each node checks the traffic load of links. If any link has no traffic load, determine the connectivity of the whole network without this link. If the network remains connected, set the link power-off. If not, end the process.

Step6 Check the load of links again. If the load is still below the threshold, go to step 2. Or when the number of all power-on links in the network becomes the minimum of number of links guaranteeing the network connectivity, the whole process stop.

Step7 If the load of link is beyond the threshold, the process should stop decreasing the weight and recover links to the previous status. That is, set the weight of link to the previous weight and if set certain link power-off in the previous step, reset the link power-on. After that, the whole process ends.

The function of decreasing the weight we choosed when the load is less than or equal to the threshold is

$$weight' = k * weight \quad (load < threshold)$$

(weight' is the modifyed weight of link. k is coefficient)

When the traffic load increases and the load of certain link exceed the threshold, we need to recover to the previous status. Hence, the weight of link is changed as follows:

$$weight' = weight / k \quad (load > threshold)$$

In a word, the weight of link is modified as follows:

$$weight' = \begin{cases} k * weight & (load \leq threshold) \\ weight / k & (load > threshold) \end{cases}$$

An Example

For a detailed description of our approach, take the following simple network with 3 nodes as an example. The weights of links are shown in figure 1.

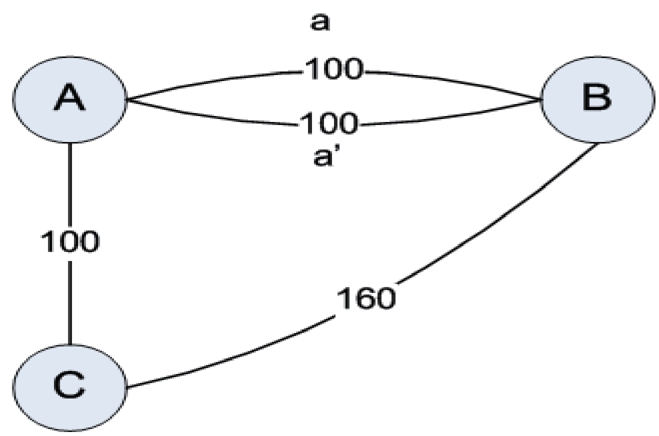

Fig. 1. 3-nodes network

We assume that the capacity of load of link is 1. Each node has the same traffic to other nodes and the traffic load is 0.1. That is, Node A has traffic to Node B and Node C, and the traffic load is 0.1; Node B also send traffic of which the load equal to 0.1 to Node A and Node C; So is the same as Node C. Based on the shortest path algorithm and the load balancing strategies, we can get the original load of each link and show in the table 1:

TABLE I
THE LINK LOAD TALBE

	Load
Link a	0.1
Link a’	0.1
Link AC	0.2
Link BC	0.2

Take Node A as an example. Node A is found to have two links leading to node B, and the sum of the load of two links does not exceed the threshold (the threshold is selected as 0.8 in this example). Therefore, Node A choose one link to modify the weight, that is, Link a is chosen and the weight of Link a is revised to 70 (here we have chosen k = 0.7). Next step is to notice the new link state to Node B and Node C. To Node B, it didn't need to modify the weights of Link a or Link a' because Node A had modified the weights of link. According to the new link states, each node re-calculates the paths to other nodes, and traffic re-selected paths to destinations.

A new links' load table was as table 2:

TABLE 2

THE LINK LOAD TALBE

	Load
Link a	0.2
Link a’	0
Link AC	0.2
Link BC	0.2

From table 2, It is found that no traffic pass through Link a'. Node A and Node C checked the network connection again without Link a'. The result showed the entire network was still connective. Hence we set Link a' power-off. At the same time the load of Link a didn't meet the threshold, so Node A modified the weight of Link a to 49. After exchanging the link states among nodes, each node got new path vectors. In particular, the traffic from Node C to Node B changed paths and passed though Link a and Link AC. At this time the load links table is as table 3.

TABLE 3

THE LINK LOAD TALBE

	Load
Link a	0.4
Link a’	0
Link AC	0.4
Link BC	0

As a result of Link BC no load, Node B and Node C can still reach the other nodes without Link BC. Therefore, set Link BC power-off. Right now the entire network only had the minimum number of links power-on, so end the algorithm.

Finally, the network connection diagram is as figure 2:

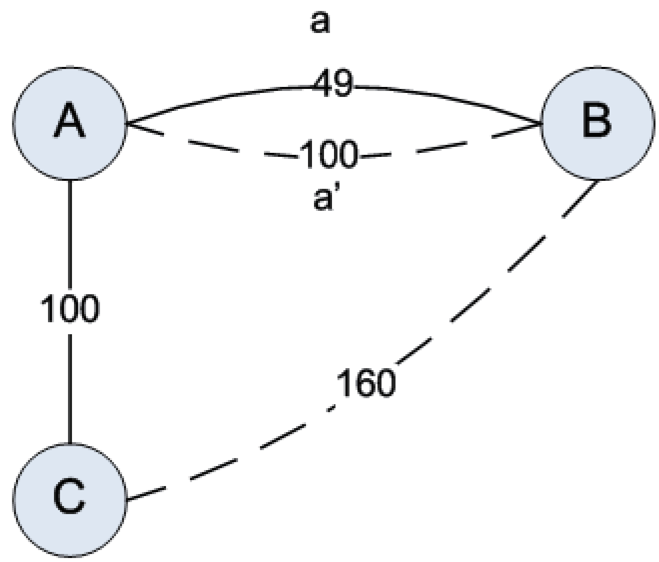

Fig. 2. 3-nodes network

III. PERFORMANCE EVALUATIONS

We evaluated the performance of the link power-off approach in computer simulation. The topology of network in simulation is shown as figure 3 :

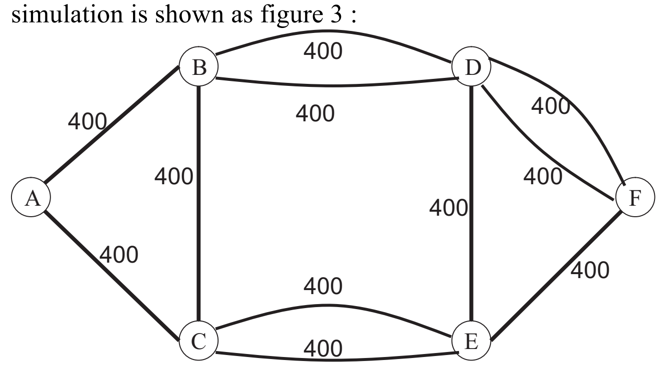

Fig. 3- The topology of network

In this network, there are 6 nodes and 11 links. Especially three pairs of links link to the same nodes and have the same weight originally. We defined the traffic capacity of each link as 1. In the simulation, traffic load is assumed in static state. Each node sends the same traffic $¥rho$ to other nodes. The traffic $¥rho$ varied from 0.01, gradually increased the load until the load of links was beyond the threshold. Every time the increase is 0.01. We did the simulation respectively when the threshold was selected as 0.7 and 0.8.

Figure 4 shows the relation of the percentage of power-on links and the traffic load $¥rho$ when the threshold of load of links is different. In this network, the minimum number of links which should be power-on is 5 to ensure that each node is connective. In other words, the minimum percentage of power-on links is about 45.5¥%. From figure 4, we can see that the number of power-on links remain the least even though the traffic load increase 7 times from 0.01 to 0.07 when threshold is selected as 0.8. And while the traffic load varies from 0.08 to 0.11, there is 80¥% links set power-on.

Similarly, when the threshold is 0.7, the power-on links are still the minimum link set until the traffic load is beyond 0.06. While the traffic load is medium, we also can set 20¥% links power-off.

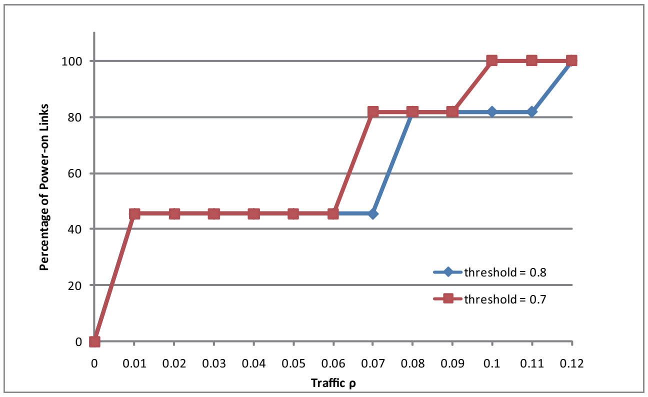

Fig. 4- The percentage of power-on links

Similarly, when the threshold is 0.7, the power-on links are still the minimum link set until the traffic load is beyond 0.06. While the traffic load is medium, we also can set 20¥% links power-off..

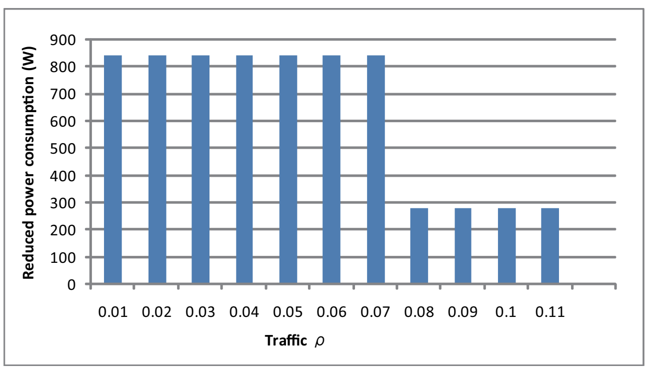

Fig. 5- The reduced power consumption

Figure 5 shows the probable reduced power consumption in the case of Cisco GSR 12008 as router. In [1], the power consumption of 1 port Oc-48/POS in a router line card is 70 watts. And one link setting power-off means 2 ports of router line can be shut down. In other words, one link set power-off can reduce 140 watts power. In the simulation, the maximum of the number of power-off links is 6. So the reduced power consumption can reach 840 watts in the period of low traffic and 280 watts in the period of medium traffic.

IV. CONCLUSION

How to raise the efficient of the Internet power consumption presents a challenge in the next generation network. In this paper, we proposed a dynamic Link metric method and power-off the idle links to reduce the network power consumption. We proved that our approach achieved to save power consumption. In the future, we plan to look for the better solution for the traffic jitter.

REFERENCES

[1] Joseph Chabarek, Joel Sommers, Paul Barford, Cristian Estan, David Tsiang, Steve Wright, "Power Awareness in Network Design and Routing", Infocom, April 2008.

[2] IEEE 802.3 Energy Efficient Ethernet Study Group, http://grouper.ieee.org/groups/802/3/eee¥_study/eee¥_5c.pdf, May 2007.

[3] C.Gunaratne, K. Christensen, and S.Suen, "Ethernet adaptive link rate(alr): Analysis of a buffer threshold policy," in IEEE Globecom, 2006.

[4] Aya TSURUSAKI, Yutaka ARAKAWA, Daisuke ISHI, Naoaki YAMANAKA, Hiroyuki

[5] Maruti Gupta, Suresh Singh, "Dynamic Ethernet Link Shutdown for Energy Conservation on Ethernet Links", in ICC '07. IEEE International Conference, June 2007

[6] C. Gunaratne, K. Christensen, and B. Nordman, Å gManaging energy consumption costs in desktop pcs and lan switches with proxying, split tcp connections, and scaling of link speed,Å h International Journal of Network Management, vol. 15, no. 5, pp. 297-310, Sept/Oct 2005.