PAPER

Joint Bandwidth Assignment and Routing for Power Saving on Large File Transfer with Time Constraints

Kazuhiko KINOSHITA[†a)], ***Senior Member***, **Masahiko AIHARA**[††], ***Nonmember***, **Nariyoshi YAMAI**[†††], ***and*** **Takashi WATANABE**[††], ***Members***

SUMMARY The increase in network traffic in recent years has led to increased power consumption. Accordingly, many studies have tried to reduce the energy consumption of network devices. Various types of data have become available in large quantities via large high-speed computer networks. Time-constrained file transfer is receiving much attention as an advanced service. In this model, a request must be completed within a user-specified deadline or rejected if the requested deadline cannot be met. Some bandwidth assignment and routing methods to accept more requests have been proposed. However, these existing methods do not consider energy consumption. Herein, we propose a joint bandwidth assignment and routing method that reduces energy consumption for time-constrained large file transfer. The bandwidth assignment method reduces the power consumption of mediate node, typically router, by waiting for requests and transferring several requests at the same time. The routing method reduces the power consumption by selecting the path with the least predicted energy consumption. Finally, we evaluate the proposed method through simulation experiments.

key words: *power saving, scheduling, routing, file transfer, time constraint*

1. Introduction

The development of network technology in recent years has led to high-speed and huge computer networks. Unfortunately, the increase in the power consumption of communication devices has become a serious problem [1].

Many studies have been conducted as regards network power saving, focusing on making the link power consumption proportional to the link speed [2], by placing switches and routers in sleep mode in the absence of network traffic [3], and concentrating onto particular routers [4], [5].

On the other hand, the act of users transferring large files has increased with the diversification of services and the increase in the number of contents in a network. Real-time applications, such as multimedia or stock market information services, require immediate transfer, whereas bulk data transfer, such as backup applications, may require a large bandwidth, but not necessarily immediate transfer. In the latter case, the data transfer completion time is the key quality of service that users want. Any applications that need coordinated use of several resources can generally benefit from being deadline-aware [6], [7].

Many studies have been performed on file transfer, but most of them assumed a best-effort manner that does not guarantee the bandwidth and focused on shortening the average transfer completion time [8]–[10]. In these studies, predicting and/or guaranteeing transfer completion times was difficult because they strongly depended on the network conditions [11], [12].

To overcome this problem, other studies introduced a model whereby a transfer request must either be completed by a user-specified deadline or rejected if the deadline cannot be met [13], [14]. Note that in this model, shortening the transfer time below its deadline is not necessary, and accepting more requests is preferable, thereby reducing the number of rejected requests. The bandwidth assignment for each request must be considered to handle many requests that meet their deadline and reduce the call-blocking probability.

Some studies demonstrated scheduling with a hard deadline in this file transfer model. [15] proposed a scheduling method named DFP (Deadline Fitting Push) to reduce the call-blocking probability by considering the difference in deadlines. [16] studied an off-line algorithm and an online algorithm to optimally schedule packets for multihop wired networks. [17] developed a new protocol to meet the application deadlines for cloud datacenter networks. [18] proposed a flow scheduling method and a routing method for datacenter networks. Meanwhile, [20] studied a switch buffer control to meet the flow deadlines. [21] introduced a deadline-aware advance reservation model for media production networks.

Unfortunately, most of these studies did not consider the network energy consumption. Some studies considered energy consumption (e.g., [18]), but reduced only link consumption and ignored node consumption.

We propose herein a joint bandwidth assignment and routing method for power saving. The bandwidth assignment method, named Waiting Crossing Requests (WCR), reduces the network power consumption by transferring more requests via particular routers at the same time. For this purpose, in the proposed method, when a request comes, it waits for a later request that uses the same node, but does not use the same link as the original request [19].

The proposed routing method, named Least Additional Energy (LAE), makes a new graph to consider the node cost in addition to the link cost because the bandwidth assign-

---

Manuscript received March 27, 2019.
Manuscript revised July 28, 2019.
Manuscript publicized September 27, 2019.

[†] The author is with Tokushima University, Tokushima-shi, 770-8506 Japan.

[††] The authors are with Osaka University, Suita-shi, 565-0871 Japan.

[†††] The author is with Tokyo University of Agriculture and Technology, Fuchu-shi, 183-0054 Japan.

a) E-mail: kazuhiko@is.tokushima-u.ac.jp

DOI: 10.1587/transcom.2019EBP3072

ment methods schedule the bandwidth to reduce the node power consumption. The LAE method decides on the route using Dikstra's algorithm. The link cost of the LAE is the predicted power consumption increased by the arrival of the new request.

The combination of the proposed methods tries to reduce the total active time of nodes by shifting the transfer schedule and routes.

The remainder of this paper is organized as follows: Section 2 presents the background of the power-saving technologies and the time-constrained file transfer; in Sect. 3 we discuss the construction of a network model considering the power consumption and propose a bandwidth assignment method and a routing method; Sect. 4 evaluates their performance through simulation experiments; and Sect. 5 presents the conclusions.

2. Related Works

2.1 Power-Saving Technologies

The energy consumption of networks can be reduced by making redundant devices sleep or modulating the capacities of network chips [22], which leads to a dynamic trade-off between packet service performance and power consumption. This approach can be classified into three levels: link, node, and network levels.

(1) Link Level

The representative technology at the link level is Energy Efficient Ethernet (EEE) [2], which is standardized as IEEE802.3az, and has already been commercialized. EEE includes the following two methods: Low Power Idle (LPI) [23] and Rapid Phy Selection (RPS) [24]. LPI decides on whether to sleep or transmit according to the time fluctuation of the network traffic. The RPS selects the appropriate link rate from several kinds of prepared link rates.

The wakeup time of a line card is important in link sleep. It takes minutes to get a line card of a major vendor's high-end routers ready under the current design. To address this issue, [25] proposed a new line card design that keeps the host processor in a line card on standby, thereby only consuming a small fraction of power consumption, but saving considerable wakeup time. This design downloads a slim slot of popular prefixes with higher priority; hence, the line card will be ready for forwarding most of the traffic much earlier. Their experiment showed that the new line card can wake up from sleep mode in 127.27 [ms].

(2) Node Level

As the power-saving technology at the node level, Gupta and Singh proposed the method that set LAN switches in standby modes when packets arrive [3], [26]. They also discussed about packet delay and proposed a more effective algorithm [27], [28].

Dynamic Voltage Scaling (DVS) has also received much attention [29]. The DVS reduces the power consumption of line cards by scaling the capacity of processors according to the network traffic load.

(3) Network Level

Many energy-aware routing protocols are proposed at the network level [4], [5]. ECO-RP is one of the representative energy-aware routing protocols [4]. ECO-RP makes routers cooperate with each other based on Open Shortest Path Fast and transmits network traffic by part of routers in a period of time with less traffic.

2.2 File Transfer with Time Constraints

2.2.1 Problem Formulation

The best-effort manner in current networks aims to shorten the average completion time; hence, the request's deadline is not considered [11], [32]. Predicting and/or guaranteeing the completion time is difficult in such a situation. To overcome this problem, some studies have introduced a model where a file transfer request must either be completed by a user-specified deadline or be rejected if the deadline cannot be satisfied [13], [31].

This file transfer model is formally defined as follows: first, a user generates a transfer request with an allowable deadline; a feasible route with a large bandwidth to meet the new request's deadline is then searched; a bandwidth assignment is invoked if the request's route is searched; and the request is rejected if the route is not found or the bandwidth assignment is failed.

A transfer request with constraint $R_i (i = 1, 2, \ldots)$ is defined by parameters as follows:

$$R_i = (s_i, d_i, A_i, F_i, D_i), \tag{1}$$

where s_i is the source node; d_i is the destination node; A_i is the request arrival time; F_i is the file size; and D_i is the request deadline. Note that F_i and D_i will decrease as time passes. Therefore, we use $F_i(t)$ to describe the remaining size of the transfer and $D_i(t)$ to indicate the remaining time to deadline. We suppose that if the transfer request is finished by the deadline defined by the user, the user is sufficiently satisfied.

For each request R_i, $MinRate_i(t)$ is defined as the minimum average transfer rate that will meet the request's deadline [31]. This can be determined from the file size $F_i(t)$ and the deadline $D_i(t)$ as follows:

$$MinRate_i(t) = \frac{F_i(t)}{D_i(t)} \tag{2}$$

An accepted request can meet its deadline when the allocated bandwidth is $MinRate$ or more. The $MinRate$ will monotonically decrease with more allocated bandwidth.

2.2.2 Existing Methods

[31] proposed the ChangeRates method as an efficient

bandwidth allocation method that dynamically changes the assigned bandwidth. The method can decrease the call-blocking probability while simultaneously guaranteeing $MinRate$ and allocating the remaining bandwidth.

Instead of equally dividing the bandwidth for all ongoing requests, ChangeRates varies the bandwidth allocation according to the ratios of $MinRate$ of the requests using the link.

The bandwidth is allocated by selecting the minimum $Rate_j$ calculated in each link j. C_j is the link capacity, and $\sum MinRate_j$ is the sum of $MinRate$ of requests using the link j.

$$Rate_j = MinRate_i \frac{C_j}{\sum MinRate_j} \tag{3}$$

In the abovementioned existing method, however, a new request may be rejected without considering the bandwidth that will be released by future transfer completions. In other words, it always allocates at least $MinRate$ bandwidth. However, the transfer can be completed by the deadline even if the assigned bandwidth is temporarily less than $MinRate$.

[15] presented a bandwidth scheduling method that can meet a request's deadline without always allocating $MinRate$. This method considers the bandwidth that will be freed when an ongoing request is completed.

The length of the transfer time does not need to be considered, and all requests may complete just before their deadlines; hence, they consider scheduling a request's bandwidth starting with the request with the latest deadline as long as more bandwidth than $MinRate$ would be assigned later.

[15] also proposed a routing method for time-constrained file transfer named MLC (Minimum Link Congestion). It considers ongoing requests and their deadline.

The combination of the bandwidth assignment and the routing achieves a smaller call-blocking probability.

2.3 Problematic Issues

The existing methods for large file transfer with time constraints do not consider energy consumption. However, the file transfer model with time constraints has a potential to save the power consumption in the future network. In this paper, we propose a joint bandwidth assignment and routing method for power saving.

3. Proposed Method

3.1 Network Model

The following network characteristics are assumed herein: the network topology is represented by a graph $\boldsymbol{G}(\boldsymbol{V}, \boldsymbol{E})$ consisting of a node set $\boldsymbol{V}$ and a link set $\boldsymbol{E}$, and the bandwidth allocated to each transfer connection is guaranteed, as in software-defined networking [30]. A database for managing essential information, such as network topology, link capacity, and ongoing requests, is used for routing and bandwidth scheduling. The access networks are sufficiently fast; thus, they cannot become potential bottlenecks.

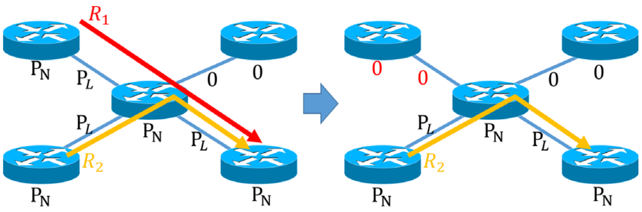

Fig. 1 Example of power consumption model.

In addition, the following power consumption model is assumed. The consumed power in a node includes two elements; core and link. A node has one or more link cards. A link card consumes constant power P_L [W] while providing any communications. The core of the node consumes constant power P_N [W] while providing any communications by at least one link card. In contrast, they consume no power while not supporting any communications. The time and the power consumption when a node or a link goes to sleep or restarts are sufficiently small. Consequently, the total energy consumption E_{all} [J] and the consumed power per second P_{all} [W] of the network are calculated as follows:

$$E_{all} = \sum_{e \in \boldsymbol{E}} P_L \cdot t(e) + \sum_{v \in \boldsymbol{V}} P_N \cdot t(v),$$
$$P_{all} = \frac{E_{all}}{T},$$

where $t(e)$ and $t(v)$ are the total active time of link e and node v, respectively, and T is the total observed time.

Figure 1 shows an example of the power consumption model. When two requests R_1 and R_2 exist in the network, the nodes and the links used by each request consume power P_N and P_L, respectively. The nodes and the links used by only R_1 consume no power after R_1 is completed.

3.2 Bandwidth Assignment Method

We first explain here the basic policy of our bandwidth assignment methods. These methods assign all available bandwidth to each request like the Earliest Deadline First (EDF) to maximize the sleeping time of nodes and links. However, the EDF method is not necessarily appropriate from the power saving viewpoint. Particularly when two requests have paths that are not node disjoint, but link disjoint, the power consumption can be reduced by assigning the bandwidth to these requests at the same time, since a mediate node is active for two requests and its energy is consumed efficiently. Hereinafter, this state is called *crossing*.

Figure 2 shows an example of the crossing requests. The network in Fig. 2 consists of 4 nodes and 4 links. Each link speed is 1 Gbps. Three requests (i.e., R_1, R_2, and R_3) increasing urgency in terms of deadline. R_1 and R_2 are not crossed because they do not use the same node. R_2 and R_3 are not crossed either because they use the same link C-D. In contrast, R_1 and R_3 cross each other. In this situation, assigning the bandwidth to R_1 and R_3 at the same

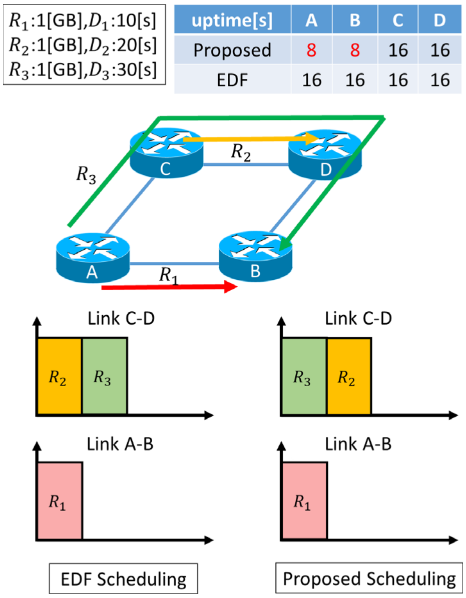

Fig. 2 Power saving scheduling of crossing requests.

time shortens the uptime of nodes A and B compared with assigning it at different times.

Hence, it is appropriate for saving power consumption to transfer crossing requests at the same time. However, such a pair of requests does not always exist. The probability that crossing requests exist is too small, especially in a network where requests do not frequently arrive. A method that makes an arrival request wait until the request crossing the arrival request arrives is further proposed to efficiently reduce the power consumption in such a situation.

We propose herein a power-saving bandwidth assignment method, which is a kind of dynamic bandwidth scheduling. Its specific procedure is presented below.

1. This method focuses on the request R_i with the earliest deadline among the requests, which have not yet been scheduled.
2. As for request R_j that is across R_i and has not been scheduled yet, all the available bandwidth to transfer $F_{min} = min(F_i, F_j)$ is assigned to requests R_i and R_j in the nearest interval to the current time. The bandwidth is allocated in an ascending order of the deadline if two or more requests correspond to request R_j.
3. If no request that is across R_i and has not been scheduled exists or the sufficient amount of bandwidth to meet R_i's deadline is not allocated, all the available bandwidth is allocated to transfer R_i's remaining file size after the waiting time W_i. W_i is calculated as follows:

$$W_i = \begin{cases} D_i - \frac{F_i}{C} \times \alpha & (D_i - \frac{F_i}{C} \times \alpha \geq T), \\ T & (D_i - \frac{F_i}{C} \times \alpha < T), \end{cases}$$

   where T is the current time and C_{R_i} denotes a capacity of the bottleneck link of the path for request R_i. In the proposed method, α is a predetemined parameter to balance the power saving and call blocking. Larger α leads to more efficient energy consumption. However, it may cause the increase of call blocking probability.
4. If the amount of bandwidth to complete R_i cannot be guaranteed after step 3, the bandwidth is assigned to R_i in the nearest time to W_i and before W_i.
5. If the bandwidth assignment does not complete after step 4, the request that arrived is rejected, and the assignment is restored to the previous assignment.
6. The new request is accepted if a sufficient amount of bandwidth is assigned to all requests to meet their deadlines; otherwise, return to step 1.

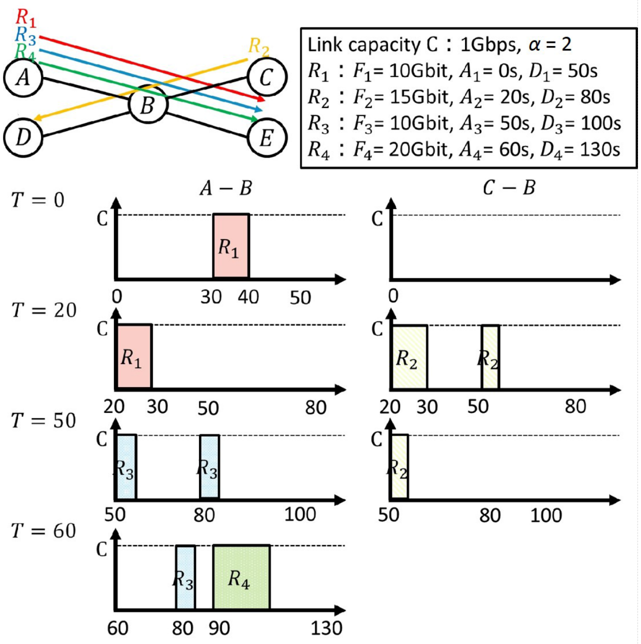

Fig. 3 Execution example of WCR method.

Figure 3 depicts an execution example of the proposed method with $\alpha = 2$, where requests from R_1 to R_4 with the different arrival time arrive.

First, at $T = 0$ [s], request R_1 from node A to node E arrives. At this time, file transfer of R_1 starts from the waiting time $W_1 = 50 - \frac{10}{1} \times 2 = 30$ [s] because no requests exist other than R_1. Note here that the proposed method waits for a new request to reduce the energy consumed by a mediate node B as much as possible by making multiple request crossing.

The proposed method then reschedules the bandwidth when R_2 arrives. In this scheduling, $F_{min} = 10$ [Gbit] is transferred at the current time because R_1 is across R_2, and 5 [Gbit] of R_2's remaining file size is transferred after $W_2 = 80 - \frac{15}{1} \times 2 = 50$ [s] because no request is across R_2, except for R_1. R_3 then arrives after R_1 is completed. In the same manner, $F_{min} = 5$ [Gbit] is transferred at the current time, and 5 [Gbit] of R_3's remaining file size is transferred after $W_3 = 100 - \frac{10}{1} \times 2 = 80$ [s]. Finally, R_4 arrives after R_2 is completed. The proposed method allocates all the available bandwidth to R_3 and R_4 after each waiting time because they are not crossing.

If the EDF method is applied, node B frequently restarts to transfer requests that newly arrives. However, the proposed method shortens the uptime of node B because the method lets a request wait until a crossing request arrives.

3.3 Routing Method

3.3.1 Core-Port Graph

The preparation of our routing method is first explained herein. As mentioned in Sect. 3.1, all nodes and links in the network model consume the constant power. The node power consumption is particularly very important because its consumption can be reduced by considering the crossing requests. Note here that the node power consumption depends not only on the number of ongoing requests, but also on its incoming/outgoing links. Accordingly, a directed graph named *core-port graph* that can express the cost metric reflecting the node itself and the incoming/outgoing links is made to consider this point in Dijkstra's algorithm.

We explain how to make the core-port graph with Fig. 4.

1. From an original node, one core node and p port nodes are created where p is the number of its neighbor nodes. At the beginning in Fig. 4, a circle indicates an original node. In Step 1, a core node indicated by double-lined circle and 2 or 3 port nodes indicated by single-lined circle are created from their original node. A square surrounding the core node and port nodes means their original node.
2. A port-port link connects a pair of port nodes. Each port node is responsible for presenting packet forwarding from an incoming link to outgoing links. Therefore, as shown in Step 2, each port node has one port-port link from one of neighbor original nodes and port-port links to the rest of them.
3. A core-port link connects a source core node created from an original node and a port node created from another original node. It shows the first link of a path. Therefore, as shown in Step 3, each core node has core-port links to all its neighbor original nodes.
4. A port-core link connects a port node to the core node, where they are created from the same original node. It shows the last link of a path. Therefore, as shown in Step 4, each core node has port-core links from port nodes in the same square.

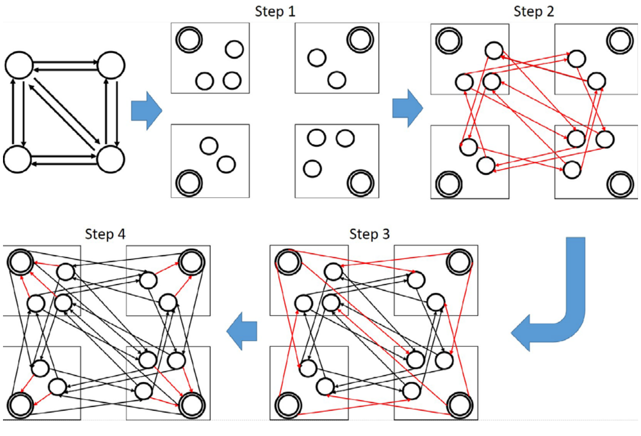

Fig. 4 Core-port graph.

In this core-port graph, only core node can be chosen as a source node or a destination node.

The cost of port-port link is the sum of link cost of original link between the original nodes whose port nodes are conneced by the port-port link and the core cost of the original node whose port node is destination-side of the port-port link. The cost of core-port link is the sum of link cost of original link between the original nodes whose port node and core node are conneced by the core-port link and the core cost of the original node whose core node is connected by the core-port link. The cost of port-core links is equal to the core cost of the original node node whose core node is connected by the port-core link.

The core-port graph is denoted by $G'(V', E'); v' \in V', e' \in E'$, where V' means a set of core nodes and port nodes and E' is a set of port-port links, core-port links and port-core links. By introducing the core-port graph, the number of nodes for routing algorithm gets larger by the number of ports. In a large scale network, however, the number of ports is much smaller than the number of nodes, so that the order of the time complexity is the same as Dijkstra's algorithm.

3.3.2 Least Additional Energy

A routing method, named LAE, is also proposed herein to enhance the power-saving effect of the proposed scheduling method. When a new request R_n arrives, the link cost $d_{link}(e)$ for link e and the node cost $d_{node}(v, e_1, e_2)$ for node v, where e_1 and e_2 means an incoming link and an outgoing link, respectively, are defined as follows:

$$d_{link}(e) = P_e \cdot \frac{F_n}{C_e}, \tag{4}$$

$$d_{node}(v, e_1, e_2) = \frac{1}{2} P_v \cdot \{ \frac{F_n}{min(C_{e_1}, C_{e_2})} + (\max_{e \in \boldsymbol{E}(v)} \sum_{R_i \in \boldsymbol{R}'(e)} \frac{F_i}{C_e} - \max_{e \in \boldsymbol{E}(v)} \sum_{R_i \in \boldsymbol{R}(e)} \frac{F_i}{C_e}), \} \tag{5}$$

where P_e is the power consumption of link e; P_v is the power consumption of node v; F_n is the file size of the new request R_n; C_e is the link capacity of link e; $\boldsymbol{E}(v)$ is the link set whose links connect with node v; $\boldsymbol{R}(e)$ is the request set whose requests, except the new request R_n, use link e; and $\boldsymbol{R}'(e)$ is the request set whose requests, including the new request R_n, use link e. These variables denote the predictive energy consumed by the new request R_n. The active time of node v is equal to the maximum active time of the links connecting with node v. Therefore, the node energy consumption can be predicted by calculating the difference of the maximum active time of the links before and after R_n

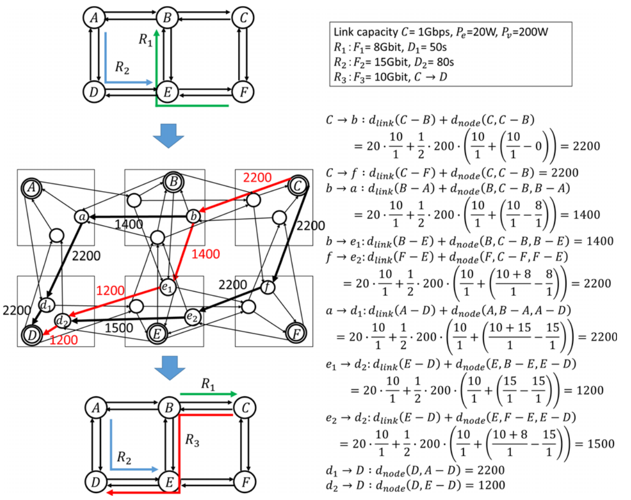

Fig. 5 Execution example of LAE routing method.

has arrived. This method also reduces the number of hops by averaging the difference and the maximum value of the difference $\frac{F_n}{min(C_{e_1}, C_{e_2})}$.

Figure 5 shows an execution example of the LAE routing method. For simplicity, all links have an identical capacity of 1 [Gbps]. All link power consumption and node power consumption are 20 [W] and 200 [W], respectively.

In this example, suppose that R_1 and R_2 are ongoing, and a new request R_3 has arrived. First, the LAE routing method makes a core-port graph to consider both link and node costs. The link cost is then calculated according to the formula 4 and 5. The link cost is equal to the summation of d_{link} and d_{node}. For simplicity, the links included by the paths with the least hops are only considered in this example. For example, focusing on core-port link C to b, no request exists on the links connected to node C. The cost of the link is equal to the maximum value of 2200 because node C does not reduce the energy consumption by using crossing requests. Moreover, port-port link e_1 to d_2 has the minimum value of 1200 because R_2 with a file size larger than R_3 uses a different link from R_3's on node E. In other words, the power consumption of node E is not affected by R_3 if R_3 uses node E. Finally, the LAE routing searches for the shortest path using Dijkstra's algorithm. The energy-efficient path across R_1 and R_2 is searched for in this example.

4. Performance Evaluation

4.1 Simulation Model

We used an inter-datacenter topology as a network model. This model had two $k = 4$ Fat-Tree topologies connected to each other by the network with 4 nodes and 32 links. Figure 6 shows the network configuration of this topology. The power consumption of the nodes and the links and the link capacity were almost the same as those in the intra-datacenter. The capacity of the inter-datacenter links was 10 [Gbps], and the power consumptions of the inter-datacenter nodes and links were 430[W] and 70[W], respectively [34]. The source and destination nodes for each request were randomly selected from the servers.

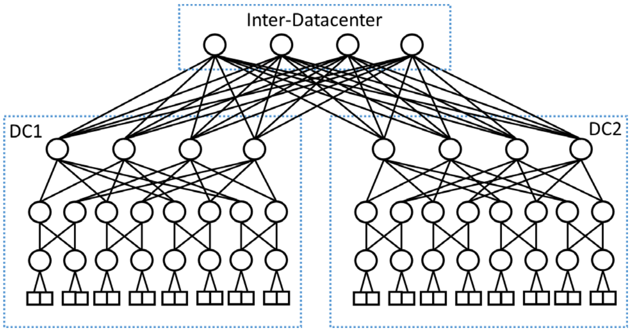

Fig. 6 Inter-datacenter topology.

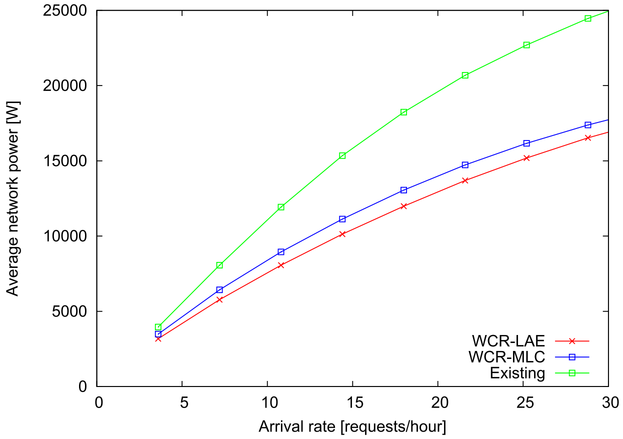

Fig. 7 Average network power consumption.

The transfer requests were generated via a Poisson arrival process with an average arrival rate of λ. All requests involved a file size of 100, 200 or 300 [GB] and a deadline of 10800, 21600 or 32400 [sec]. A combination of these parameter was selected with the same probability. The parameter α was set to 10 determined by preliminary experiment.

We confirmed that no requests was rejected among the 10^6 requests; therefore, we used the total network power consumption as a performance measure.

4.2 Simulation Results

Figure 7 shows the average power consumed in the whole network as a function of request arrival rate. According to the graph, the proposed bandwidth assignment method WCR reduces approximately 30% of the total network power consumption compared with the existing scheduling method DFP with the exsting routing method MLC. The proposed routing method LAE helps the proposed bandwidth assignment method and achieves additional power saving.

Note that the proposed method may increase the number of rejected requests in exchange for power saving when the requests arrive too frequent. Figure 8 shows the call blocking probability in higher call arrival rate. Assuming

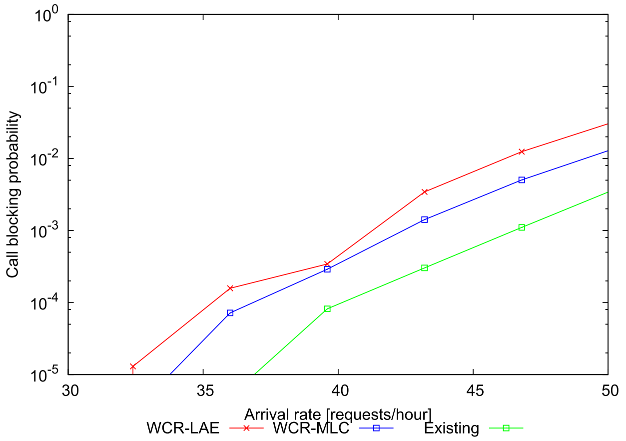

Fig. 8 Call blocking probability.

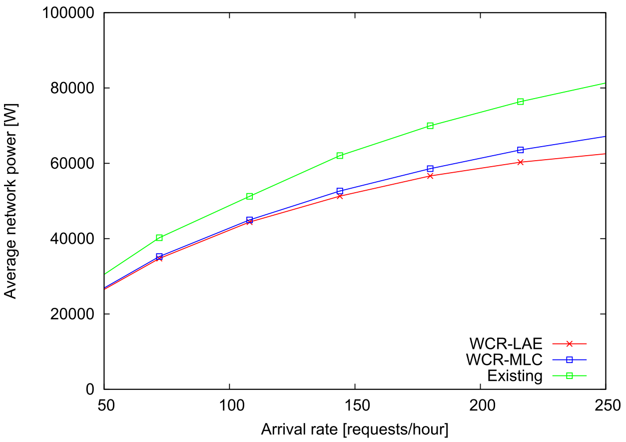

Fig. 9 Average network power consumption with random graph.

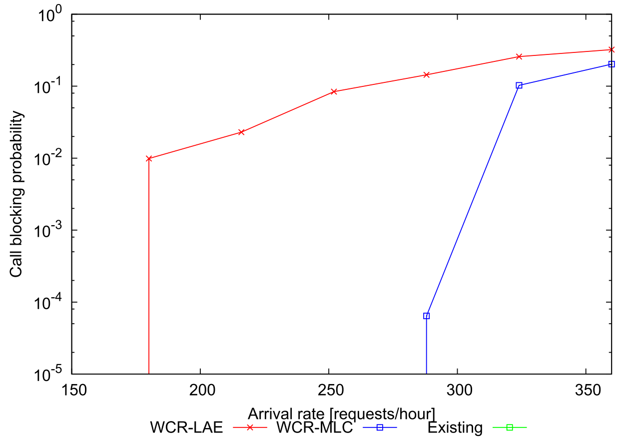

Fig. 10 Call blocking probability with random graph (no requests were rejected with the existing method).

the maximum acceptable call blocking probability is 1%, the proposed method is beneficial when the call arrival rate is less than 45.

Next, we evaluated the impact of network topology using random graph [35] with 100 nodes. Figures 9 and 10 show the average network power consumption and the call blocking probability, respectively. As shown in Fig. 10, more number of requests were rejected, since the probability of crossing increased for the connection through many hops. With smaller call arrival requests, however, the proposed method works well to achieve power saving.

5. Conclusion

This study described a model where a file transfer request must either be completed by a user-specified deadline or be rejected if the deadline cannot be satisfied. A joint bandwidth assignment and routing method was then proposed to reduce the network power consumption by applying the time-constrained file transfer model to data center networks. The simulation results showed the excellent performance of the proposed joint method.

Acknowledgments

This work is partly supported by the Kayamori Foundation of Informational Science Advancement.

References

[1] W.V. Heddeghem, S. Lambert, B. Lannoo, D. Colle, M. Pickavet, and P. Demeester, "Trends in worldwide ICT electricity consumption from 2007 to 2012," Comput. Commun., vol.50, pp.64–76, Sept. 2014.

[2] IEEE 802.3az-2010: Energy Efficient Ethernet, Amendment to IEEE Standard 802.3-2008 (CSMA/CD) Specifications, IEEE Standard, Sept. 2010.

[3] M. Gupta and S. Singh, "Greening of the Internet," ACM SIGCOMM 2003, Aug. 2003.

[4] D. Arai and K. Yoshihara, "ECO-friendly distributed routing protocol for reducing network energy consumption," International Conference on Network and Service Management (CNSM), pp.104–111, Oct. 2010.

[5] D. Li, Y. Shang, and C. Chen, "Software defined green data center network with exclusive routing," IEEE Conference on Computer Communications (INFOCOMM 2014), pp.1743–1751, April–May 2014.

[6] T. Ferrari, "Grid network services use cases from the e-science community," Open grid forum informational document, May 2007.

[7] P. Dharam, C.Q. Wu, and Y. Wang, "Advance bandwidth reservation with deadline constraint in high-performance networks," International Conference on Computing, Networking and Communications (ICNC 2014), pp.1041–1045, April 2014.

[8] H. Okada, T.N. Trung, K. Kinoshita, N. Yamai, and K. Murakami, "A cooperative routing method for multiple overlay networks," 6th Annual IEEE Consumer Communications and Networking Conference (CCNC 2009), pp.1–2, Jan. 2009.

[9] L. Toka, M. Dell'Amico, and P. Michiardi, "Data transfer scheduling for P2P storage," IEEE P2P, pp.132–141, Sept. 2011.

[10] Y. Chiu and D.Y. Eun, "Minimizing file download time in stochastic peer-to-peer networks," IEEE/ACM Trans. Netw., vol.16, no.2, pp.253–266, April 2008.

[11] S. Kamei, "Status and traffic issues of peer-to-peer technology," Computer Software, vol.22, no.3, pp.8–18, Sept. 2005.

[12] S. Gorinsky and N.S.V. Rao, "Dedicated channels as an optimal network support for effective transfer of massive data," 25th IEEE International Conference on Computer Communications, pp.1–5, April 2006.

[13] B. Chen and P. Primet, “Scheduling deadline-constrained bulk data transfers to minimize network congestion,” Proc. 7th IEEE International Symposium Cluster Computing and the Grid (CCGRID), pp.410–417, May 2007.

[14] D. Andrei, “Provisioning of deadline-driven requests with flexible transmission rates in WDM mesh networks,” IEEE/ACM Trans. Netw., vol.18, no.2, pp.353–366, Oct. 2010.

[15] M. Aihara, S. Kono, K. Kinoshita, N. Yamai, and T. Watanabe, “Joint bandwidth scheduling and routing method for large file transfer with time constraint and its implementation,” IEICE Trans. Commun., vol.E101-B, no.3, pp. 763–771, March 2018.

[16] Z. Mao, C.E. Koksal, and N.B. Shroff, “Optimal online scheduling with arbitrary hard deadlines in multihop communication networks,” IEEE/ACM Trans. Netw., vol.24, no.1, pp.177–189, Feb. 2016.

[17] J. Hwang, J. Yoo, and N. Choi, “Deadline and incast aware TCP for cloud data center networks,” Computer Networks, vol.68, pp.20–34, Aug. 2014.

[18] L. Wang, F. Zhang, and K. Zheng, “Energy-efficient flow scheduling and routing with hard deadlines in data center networks,” 34th IEEE International Conference on Distributed Computing Systems, pp.248–257, June–July 2014.

[19] T. Badraa, M. Aihara, K. Kinoshita, N. Yamai, and T. Watanabe, “An energy-efficient bandwid th assignment for large file transfer with time constraints,” IEEE International Conference on Consumer Electronics Taiwan (2018 ICCE-TW), May 2018.

[20] J. Zhang, “Deadline-aware bandwidth sharing by allocating switch buffer in data center networks,” The 35th Annual IEEE International Conference on Computer Communications (INFOCOMM 2016), pp.1–9, April 2016.

[21] M. Barshan, H. Moens, J. Famaey, and F. De Turck, “Deadline-aware advance reservation scheduling algorithms for media production networks,” Comput. Commun., vol.77, pp.26–40, March 2016.

[22] R. Bolla, R. Bruschi, F. Davoli, and F. Cucchietti, “Energy efficiency in the future Internet: A survey of existing approaches and trends in energy-aware fixed network infrastructures,” IEEE Commun. Surveys Tuts., vol.13, no.2, Second Quarter, July 2010.

[23] R. Hays, Intel Corporation, “Active/idle toggling with low-power idle,” Presentation for IEEE 802.3az Task Force, Jan. 2008.

[24] F. Blanquicet and K. Christensen, “An initial performance evaluation of Rapid PHY selection (RPS) for energy efficient Ethernet,” 32nd IEEE Conference on Local Computer Networks, pp.223–225, Oct. 2007.

[25] T. Pan, T. Zhang, J. Shi, Y. Li, L. Jin, F. Li, J. Yang, B. Zhang, X. Yang, M. Zhang, H. Dai, and B. Liu, “Towards zero-time wakeup of line cards in power-aware routers,” IEEE/ACM Trans. Netw., vol.24, no.3, pp.1448–1461, June 2016.

[26] M. Gupta, S. Grover, and S. Singh, “A feasibility study for power management in LAN switches,” 12th IEEE International Conference on Network Protocols (ICNP), pp.361–371, Oct. 2004.

[27] M. Gupta and S. Singh, “Dynamic Ethernet link shutdown for energy conservation on Ethernet links,” IEEE International Conference on Communications (ICC 2007), pp.6156–6161, June 2007.

[28] M. Gupta and S. Singh, “Using low-power modes for energy conservation in Ethernet LANs,” 26th IEEE Conference on Computer Communications (INFOCOM 2007), pp.2451–2455, May 2007.

[29] M. Mandviwalla and N.F. Tzeng, “Energy-efficient scheme for multiprocessor-based router linecards,” International Symposium on Applications on Internet (SAINT 2006), pp.156–163, Jan. 2006.

[30] S. Ghorbani and M. Caesar, “Walk the line: Consistent network updates with bandwidth guarantees,” ACM workshop on Hot topics in software defined networks, pp.67–72, Aug. 2012.

[31] D. Andrei, M. Barayneh, S. Sarkar, C.U. Martel, and B. Mukherjee, “Deadline-driven bandwidth allocation with flexible transmission rates in WDM networks,” IEEE International Conference on Communications (ICC 2008), pp.5354–5358, May 2008.

[32] S. Gorinsky and N.S.V. Rao, “Dedicated channels as an optimal network support for effective transfer of massive data,” 25th IEEE International Conference on Computer Communications (INFOCOM 2006), pp.1–5, April 2006.

[33] M. Al-Fares, A. Loukissas, and A. Vahdat, “A scalable, commodity data center network architecture,” Proc. ACM SIGCOMM 2008 conference on Data communication, vol.38, no.4, pp.63–74, Oct. 2008.

[34] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright, “Power awareness in network design and routing,” Proc. 26th IEEE Conference on Computer Communications (INFOCOMM 2008), pp.1130–1138, April 2008.

[35] B.M. Waxman, “Routing of multipoint connections,” IEEE J. Sel. Areas Commun., vol.6, no.9, pp.1617–1622, Dec. 1988.

Kazuhiko Kinoshita was born in Osaka, Japan, on June 13, 1973. He received the B.E., M.E. and Ph.D. degrees in information systems engineering from Osaka University, Osaka, Japan, in 1996, 1997 and 2003, respectively. From April 1998 to March 2002, he was an Assistant Professor at the Department of Information Systems Engineering, Graduate School of Engineering, Osaka University. From April 2002 to March 2008, he was an Assistant Professor at the Department of Information Networking, Graduate School of Information Science and Technology, Osaka University. From April 2008 to January 2015, he was an Associate Professor at the same University. Since February 2015, he has been a Professor at Tokushima University. His research interests include mobile networks, network management, and agent communications. Dr. Kinoshita is a member of IEEE.

Masahiko Aihara received the bachelor’s degree in information systems engineering and master’s degree in information networking from Osaka University, Osaka, Japan in 2015 and 2017, respectively. Since April 2017, he has been worked for Canon Inc.

Nariyoshi Yamai received his B.E. and M.E. degrees in electronic engineering and his Ph.D. degree in information and computer science from Osaka University, Osaka, Japan, in 1984, 1986 and 1993, respectively. In April 1988, he joined the Department of Information Engineering, Nara National College of Technology, as a research associate. From April 1990 to March 1994, he was an assistant professor in the same department. In April 1994, he joined the Education Center for Information Processing, Osaka University, as a research associate. In April 1995, he joined the Computation Center, Osaka University, as an assistant professor. From November 1997 to March 2006, he joined the Computer Center, Okayama University, as an associate professor. From April 2006 to March 2014, he was a professor in the Information Technology Center (at present, the Center for Information Technology and Management), Okayama University. Since April 2014, he has been a professor in the Institute of Engineering, Tokyo University of Agriculture and Technology. His research interests include distributed system, network architecture and Internet. He is a member of IPSJ and IEEE.

Takashi Watanabe is a Professor of Graduate School of Information Science and Technology, Osaka University, Japan. He received his B.E., M.E. and Ph.D. degrees from Osaka University, Japan, in 1982, 1984 and 1987, respectively. He joined Faculty of Engineering, Tokushima University as an assistant professor in 1987 and moved to Faculty of Engineering, Shizuoka University in 1990. He was a visiting researcher at University of California, Irvine from 1995 through 1996. He has served on many program committees for networking conferences, IEEE, ACM, IPSJ (Information Processing Society of Japan), IEICE (The Institute of Electronics, Information and Communication Engineers, Japan). His research interests include wireless networking, mobile networking, ad hoc networks, sensor networks, ubiquitous networks, intelligent transport systems, specially MAC and routing. He is a member of IEEE, IEEE Communications Society, IEEE Computer Society as well as IPSJ and IEICE.