Telecommun Syst
DOI 10.1007/s11235-015-9975-y

An optimization-based robust routing algorithm to energy-efficient networks for cloud computing

Dingde Jiang · Zhengzheng Xu · Jindi Liu · Wenhui Zhao

Abstract This paper studies the routing problem in energy-efficient networks for cloud computing. We propose a robust routing algorithm to reach the higher network energy efficiency, which is based on optimization problem. To attain the highly energy-efficient routing in energy-efficient networks for cloud computing, the link of low utilization is turned into the sleeping state to save the network energy. At the same time, the low link traffic is aggregated to the link with high utilization to enhance the link utilization and to sleep the links as many as possible. We present an optimized link sleeping method to maximize the number of the sleeping links. By targeting the network robustness, a weight adaptive strategy is brought forth to reduce the link congestion and enhance the robustness of the network. Simulation results indicate that our algorithm is effective and feasible to achieve energy-efficient networks for cloud computing.

Keywords Cloud computing · Energy-efficient networks · Modeling · Network measurement · Routing

D. Jiang (✉) · Z. Xu · J. Liu · W. Zhao
College of Information Science and Engineering,
Northeastern University, Shenyang 110819, China
e-mail: jiangdd@mail.neu.edu.cn

Z. Xu
School of Economics and Management,
AnQing Normal University, AnQing 246011, China

1 Introduction

According to the statistics of International Energy Agency (IEA), the global ICT electricity consumption accounted for 8 % of global energy consumption, and the trend is growing [1]. Particularly, network traffic is increasing exponentially and holds much complex features than before [2,3]. This leads to telecommunication companies and Internet service providers to pay more attention to the wired network energy efficiency problem. The rising energy prices and the increasing number of users have intensified the urgency of the research on energy efficiency issues, especially for cloud computing networks. Cloud computing networks are based on the current network infrastructures. However, further studies show that the energy management method of routers and links equipments is unreasonable in current network. The energy consumption of these devices accounts about a quarter of the entire energy consumption of Internet. Therefore, the topic of energy efficiency of cloud computing networks has become a hot issue in the current network community. How to maximize the energy efficiency of cloud computing networks is the main challenge faced by researchers and network operators.

In recent years, many researchers have studied the network energy efficiency from different aspects. Bolla et al. proposed an optimization strategy of equipment power consumption by sensing traffic load in the heterogeneous network [4]. Cianfrani et al. presented an energy efficiency management method for IP network [5], which used the OSPF routing algorithm to tag the link status and then to close the link with low traffic load in order to improve the energy efficiency. The two methods considered transferring the link traffic of low utilization links to the high utilization links, to reduce the energy consumption in the entire network when meet the soundness of network load, but not analysis and

Published online: 14 March 2015

research energy efficiency specifically. Restrepo et al. proposed a power management algorithm in traffic engineering, dynamically closing the low utilization links to reduce energy demand [6]. Chiaraviglio et al. proposed a heuristic approach to calculate the minimum number of links which can be slept for a known traffic matrix [7]. They used the traffic information to minimize network energy consumption [8]. Cianfrani et al. proposed an energy saving routing scheme based on OSPF routing protocol [9]. However, cloud computing networks hold more heavy load and more burst traffic. This leads to the fact that link utilizations in them are unbalance. Some links in cloud computing networks carry more heavy loads while other links are exploited sufficiently. Additionally, the energy efficiency problem of networks [10,11], the social nature of networks [12], and the dynamic properties of network traffic [13] add the difficulty in performing the effective cloud computing. The mobility of terminal users [14] and anomalous behaviors of network devices [15] can also lead to the failure of cloud computing tasks. In such a case, these methods in current studies are significantly difficult to provide the effective and robust routing to improve the energy efficiency of cloud computing networks.

In this paper, a robust energy-efficient routing algorithm, which is called optimization-based robust routing algorithm (ORRA), is proposed to solve the stability and energy efficiency problem in the current cloud computing network. In order to meet the QoS requests, typical ISP networks usually employ the redundant design to deal with unexpected situations of network congestion and routing failure. Although such a design can solve QoS issues, a problem of low energy efficiency is resulted in. Different from previous methods, we consider how to design a routing algorithm to meet the routing and QoS requirements as well as to achieve a highly energy-efficient and robust cloud computing network. We convert the link of low utilization into the sleeping state to save the energy of the cloud computing network. In such a case, the highly energy-efficient routing for cloud computing networks can be attained. Simultaneously, the low link traffic is aggregated to the link with high utilization to enhance the link utilization and to sleep the links as many as possible. For solving this problem, an optimized link sleeping method is proposed to maximize the number of the sleeping links in the cloud computing network. And we bring forth a weight adaptive strategy to reduce the link congestion and to ensure the network robustness for cloud computing. Simulation results indicate that the proposed algorithm is effective and feasible for the energy-efficient cloud computing network.

The remainder of this paper is organized as follows. We perform the problem statement in Sect. 2. Section 3 discusses and derives our robust routing method for the energy-efficient cloud computing network. Section 4 presents the simulation results and analysis. We then conclude our work in Sect. 5.

2 Problem statement

Here we investigate the energy-efficiency and robust routing problem in the cloud computing backbone network that connect different data centers and users. Figure 1 plots the cloud computing network model, where users include individual users and enterprise users, they are connected with cloud computing backbone networks via access links to visit data centers located at different geographical locations. Furthermore, the migrations of kinds of virtual resources (such as virtual machines, virtual CPUs, virtual memories, virtual service and so on) among different data centers have been achieved via the cloud computing backbone network in Fig. 1. Users' service requests for cloud computing and the virtual resource migrations hold the significant dynamic over the time. Thereby, in the cloud computing backbone networks indicated in Fig. 1, the utilization of each link is unbalanced. In such a case, some links in the cloud computing network are probably heavy loads and even become congestion, but other links are possibly used in the lower efficiency. Consequently, network devices can consume an large amount of energy to keep up the normal operation of the cloud computing network. At a result, the entire network runs in the low energy-efficient way. For the cloud computing network model, it is significantly important to find a routing method to implement the highly energy-efficient and robust communication.

From Fig. 1, we can extract the cloud computing backbone network topology and link utilization snapshot shown in Fig. 2, where different line widths denote the observed link utilization in a certain snapshot. From the network topology for cloud computing shown in Fig. 2, we can see that links *R1–R6*, *R2–R3*, and *R7–R9* hold the higher utility while other links are not nearly exploited sufficiently. Especially for links *R1–R2*, *R2–R5*, *R4–R5*, *R5–R6*, *R4–R7*, *R6–R7*, and *R8–r10*, they have the much lower utilization in a certain observation. Current studies show that the case as indicated in Fig. 2 exists practically. Moreover, for those links with lower utilization, they still consume the large amount of energy. To ensure the reliable communications, current cloud computing networks all adopt the redundant design. Accordingly, this lower energy-efficient situation becomes much more serious. Hence, how to appropriately raise the efficiency of each link and reduce their energy consumption is very important.

To this end, we can express the network shown in Fig. 2 as a graph model $G(V, E, W)$, where V is the set of the nodes in the network, E is the set of the links in the network, and W represents the weight of links. Let N and L be the cardinalities of V ($N = ||V||$) and E ($L = ||E||$), representing the number of nodes and links in the network, respectively. In the present paper, we only focus on the sleeping redundant links but not consider to close the nodes. And thus we define the energy efficiency of the entire network as the suc-

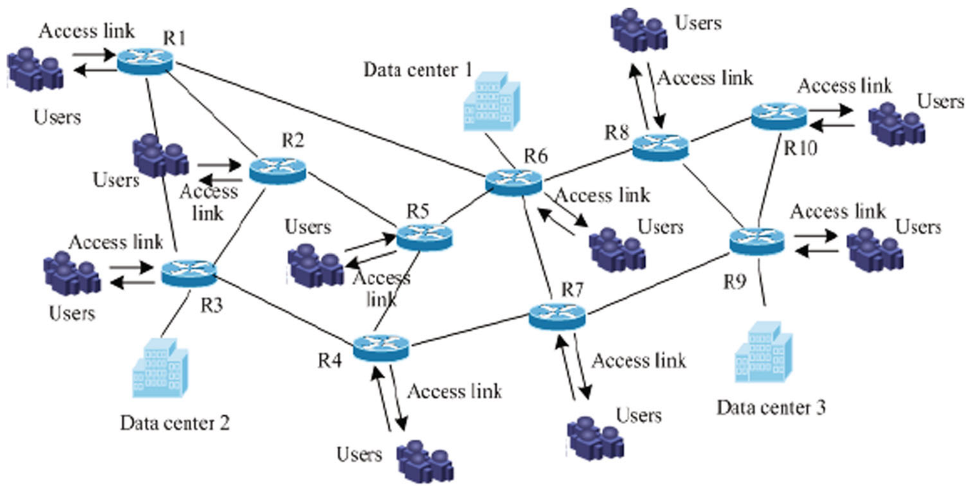

Fig. 1 Cloud computing network model connecting different data centers and access users

cessfully transmitting information bits of per joule energy consumption, namely

$$\bar{\eta} = \frac{\alpha_{bit}}{E_J} = \frac{\sum_{N_s \in V}\sum_{N_d \in V}\int_T h_{sd}(t)dt}{\sum_{L_{ij}\in E} x_{ij}\int_T P_{ij}(t)dt + \sum_{N_i \in V}\cdot p_0}$$
$$= \frac{\sum_{N_s \in V}\sum_{N_d \in V}\overline{h_{sd}}}{\sum_{L_{ij}\in E} x_{ij}\overline{P_{ij}} + \sum_{N_i \in V}\cdot p_0} \qquad (1)$$

where α_{bit} denotes the total throughput of networks, E_J stands for the total energy consumption of networks, $P_{ij}(t)$ represents the instant power of link L_{ij}, $h_{sd}(t)$ is the instant flow from router $N_s \in V$ to $N_d \in V$, $\overline{P_{ij}}$ indicates the average power of the link L_{ij} over the period T, $\overline{h_{sd}}$ signifies the average flow from access router N_s to N_d over the period T, and p_0 (which is usually set as a constant value) expresses the power of the nodes in the network.

Equation (1) formulates that the network energy efficiency is mainly associated with the forwarded information amount and the network energy consumption. When transmitting the same amount of information in a unit time, the smaller the network energy consumption is, the greater the network energy efficiency is, and vice versa. Here, we take into consideration to reduce the network energy consumption of sending unit information to raise the network energy efficiency. In the following, we will introduce an optimized link sleeping algorithm to sleep the network links as many as possible under the precondition of ensuring normal communication, to improve the energy efficiency of the whole network.

3 Robust routing for cloud computing networks

In this section, we are to derive our optimal robust routing algorithm (ORRA) for cloud computing networks. Figure 3 denotes the block diagram of ORRA algorithm proposed in this paper. ORRA algorithm proposed includes system initialization module, optimization link sleeping algorithm (OLSA) module, weighted adaptive shortest path first (WASPF) module, link weights feedback module, robust routing path output module, and user request interface. We can see from Fig. 2, when the user sends a request, the parameters obtained from the system initialization module would be transmitted to OLSA module to sleep the redundant links; and then the new network topology information was passed to WASPF module, using ORRA algorithm to route the traffic flow, outputting robust routing path and link weight parameters. The link weight parameters would be used as feedback parameters for the user requests of the next time. The system initialization parameters are obtained by analyzing the historical data, which was set appropriately based on the actual network conditions. In the following, we are to discuss ORRA algorithm in detail.

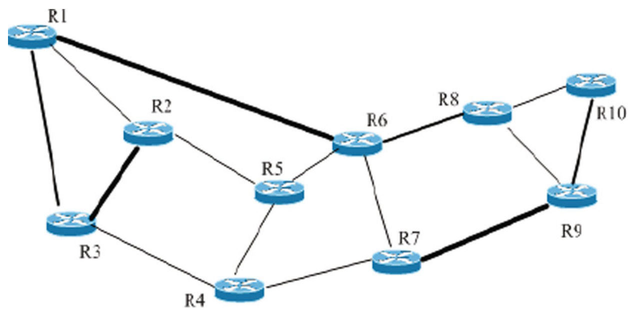

Fig. 2 Cloud computing backbone network topology and link utilization snapshot

3.1 Optimal link sleeping

Here we propose our OLSA to let the links sleep appropriately. To reduce the number of active links, we exploit the *move* to express a link revising process based on exportation degree mechanism [9]. By finding a node with the maximum

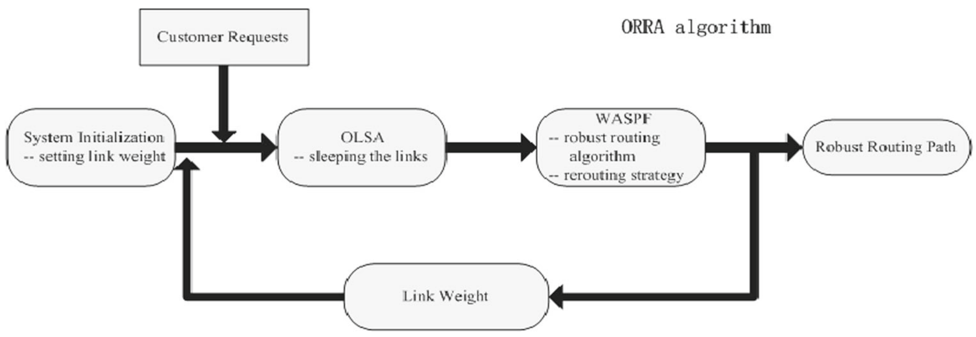

Fig. 3 Block diagram of ORRA algorithm

degree as the exporter the router (ER) in the network and searching the node with maximum degree within ER's neighboring nodes as importer the router (IR), a *move* process is that of the link direction change from IR to ER. Calculate the shortest path tree (SPT) of ER and IR, respectively, and then compare $SPT(IR)$ and $SPT(ER)$ to generate the modified path tree (MPT). By comparing the MPT and SPT of IR, we can sleep the redundant links. As mentioned in [9], the number of the sleeping redundant links can be defined as the move weight, namely

$$\mathrm{k} = \{k_i, \quad 1 \le i \le L\}. \tag{2}$$

Here, we employ the move compatibility idea to find the minimum sleeping link set. The move compatibility is defined as [9]: Given a couple of *moves* $m_1(i_1, e_1)$ and $m_2(i_2, e_2)$, if the *move* m_2 can still be carried out after executing the *move* m_1, the *move* m_2 is said to be compatible with the *move* m_1. The compatibility relationship is symmetric, i.e., if m_1 is compatible with m_2, so m_2 and m_1 are compatible. Then we will focus on how to maximize the number of the sleeping links in the network. The problem can be transformed into a typical Knapsack Problem, which can obtain the optimal solution by heuristic greedy algorithm. The objective function and the constraints are expressed as follows:

$$\begin{cases} \max & \sum_{i=1}^{L} k_i m_i, \\ s.t. & m_i + m_j \le 1, \quad \forall (i,j) \in C \\ & m_i \in \{0,1\}, \quad \forall i \in M, \end{cases} \tag{3}$$

where C is the compatibility relationship of m_i and m_j, M represents the set of all the possible moves, and k_i expresses the link move weight. The link move weight k_i can be obtained by Eq. (2), and then we can maximize the set of the sleeping links. Equations (2,3) show our optimized link sleeping algorithm (for short OLS).

3.2 Weighted adaptive shortest path first

in this subsection, we discuss our WASPF. To achieve the robust routing, now we propose a strategy with link weight adaptive robust routing, which can change the weights of the network links dynamically. According this strategy, we can change the network routing topology, realize load balancing, and improve network robustness. This strategy measures the robustness of communication networks using the network criticality which can map into the robustness of the network directly. The smaller the network criticality is, the more robust the network is, and vice versa. As discussed in [16], the network criticality τ can be defined as:

$$\tau = \sum_i \sum_j \tau_{ij} = 2n \sum_i l_{ii}^{+} = 2nTr(L^{+}), \tag{4}$$

where n is the number of network nodes, τ_{ij} denotes the criticality of link l_{ij}, L^{+} is the Moore-Penrose matrix inverse transform of the Laplacian L, $L^{+} = [l_{ij}{}^{+}]$, $l_{ij}{}^{+}$ is the diagonal elements in the inverse transform matrix, and $Tr(\cdot)$ represents the cumulative operation.

By normalizing the network criticality τ in Eq. (4), we can obtain the below equation:

$$\tau' = \frac{2}{n-1} Tr(L^{+}), \tag{5}$$

where τ' denotes the normalized network criticality. The link bandwidth capacity is related with criticality. When the link bandwidth capacity is changed, the link criticality is also altered [16]. For the link with heavy traffic load, the available link bandwidth capacity is relatively low, so the risk of choosing this link to route is larger. That is, the performance of this link becomes the bottleneck of the overall network. Thus, we can map the link criticality to the value of the link weight, so that the network link can be uniformly used to reduce the fluctuation of links utilization ratio. When the link bandwidth capacity in the network reduces Δc_{ij}, then the network criticality will change by $\Delta\tau$. Update the link weight value w_{ij} by the following formula:

$$\begin{cases} \tau & = n(n-1)\tau', \\ \Delta\tau & = \tau - \tau_0, \\ w_{ij} & = a \cdot \Delta\tau + w_{0ij}, \end{cases} \tag{6}$$

where τ_0 is the initial criticality in a time interval T, τ is the network criticality for the end of T, a is a constant which was set according to the actual situation of the simulation network, n is the number of network nodes, w_{0ij} is the initial link weight value of time interval T, and w_{ij} is the link weight value for the end of the time interval T.

We propose an adaptive robust routing strategy as follow:

- Find the link weight matrix W of the network according to Eq. (6);
- Build the best routing route based on W by the Dijkstra shortest path algorithm.

3.3 ORRA algorithm

Now, we discuss our robust routing with high energy efficiency for cloud computing networks, namely ORRA. As described in [9], in order to put forward our ORRA algorithm, we consider two cases. In the first case, we ignore the traffic matrix and QoS requirements, named the Simplified ORRA. In the second case, we consider the traffic matrix and QoS requirements, named the Advanced ORRA. The Simplified ORRA can be expressed as the classic MCP [17]. The advanced ORRA can be solved by the heuristic greedy algorithm based on 0–1 Knapsack problem [18]. Additionally,

For the Simplified ORRA, because it ignores the traffic matrix and the QoS requirements, it only require to maximize the network energy efficiency under the premise of realizing normal network communication, and ensure that the network has a good robustness. How to maximize the sleeping set of redundant links and to find the relationship between the network criticality and the energy efficiency of whole network is the key problem of the Simplified ORRA.

Given a network model $G(V, E, W)$, the link utilization ratio is defined as:

$$\overline{u_{ij}} = \frac{\overline{f_{ij}}}{c_{ij}} \tag{7}$$

where u_{ij} denotes the link utilization ratio of L_{ij}, c_{ij} presents the capacity of the link L_{ij}, and $\overline{f_{ij}}$ indicates the average flow of the link L_{ij} over the period T.

Let U be the variation matrix of network link, after a number of user requests within T (u_{0ij} and u_{ij} are the initial value and the final value of the time period T, respectively). The following equation can be obtained:

$$U_{ij} = \overline{u_{ij}} - \overline{u_{0ij}}, \quad L_{ij} \in E \tag{8}$$

Depending on the optimization problem described by Eq. (3), we can obtain the maximum set of the sleeping links. However, after sleeping the redundancy links, the network performance will be affected more or less. So we need to find a suitable routing algorithm to ensure the optimal robustness of the network performance. Equation (6) reflects the relationship of the network robustness and link weight. We use the network criticality to measure the stability of the network, which can calculate the link weight of the network at a stable performance state. And then we exploit the link weight as the input of Dijkstra algorithm to get a robust routing path. In addition, we also employ the idea of rerouting to reroute the request to other link when the link capacity can not meet the request of the user. Thus the refused probability of the network can be reduced. Through the above analysis, the Simplified ORRA problem can be expressed as the below equation:

$$\begin{cases} \max \quad \bar{\eta} \\ \quad s.t. \\ \qquad \forall N_s, N_d \in V \ \&\& \ \overline{h_{sd}}>0, Connection(N_s, N_d)=1 \\ \qquad \forall L_{ij} \in E, \quad c(\overline{f_{ij}}) = w_{ij} \\ \qquad \forall L_{ij} \in E, \quad \overline{f_{ij}} = \sum_{N_s \in V} \sum_{N_d \in V} \overline{h_{ij}^{sd}} \\ \qquad 0 \le U \le 1, \end{cases} \tag{9}$$

where $\overline{h_{sd}}$ represents the average flow from access router N_s to N_d over the period T, $\overline{h_{ij}^{sd}}$ denotes the average flow going through link L_{ij}, and $c(\overline{f_{ij}})$ is the link cost to deliver the link traffic load $\overline{f_{ij}}$.

In order to get the optimal robust network routing path and maximize network energy efficiency, we need to determine a series of parameters. OLSA based on Eq. (3) is relatively simple in setting parameters compared with other link sleeping mechanism. It is automatic convergence, and not sensitive to the initialization parameters. It is important to set a suitable time interval T, while meeting the constraints of link capacity limit and the number of user requests. The setting of T is simple in the actual simulation process, because the impact of T can be offset in the process of calculating the energy efficiency in the cloud computing network.

The setting of link capacity will also affect the performance of the algorithm. In this paper, we define a linear relationship between the link capacity and the initial link weight value, namely $c_{ij} = k \cdot w_{0ij} + a$ (k and a are constant). Therefore, we can set the link capacity based on link weight, just in terms of the reasonable k and a. To set the initial value of link weight, we use the link historical traffic matrix information to determine the state of each link, and then calculate the average flow value over a time period. Then we exploit this average value to set link weight. The pseudo code of ORRA algorithm is shown in Algorithm 1.

The Advanced ORRA considers the maximum load constraint of traffic matrix on the basis of the Simplified ORRA, so as to meet the QoS requirements of the user. The traffic matrix represents the traffic request between the source and destination nodes, which is expressed as M. For all the nodes, any two nodes will constitute a pair of source and destination nodes. And have a robust route path between any two nodes calculated by the above adaptive robust routing strategy. Here are an instance of network link capacity constraint with n nodes and l links.

A network with n nodes, then there are n^2 robust adaptive paths between the source and destination nodes. Each path is constituted by different link, and for each link, the link capacity is the sum of all user requests through the robust adaptive paths including this link. And then we can easily attain the below equation:

$$c_i = \sum_{(s,d)} m_{sd}, \quad \forall s \in V, \forall d \in V \& \quad i \in P_{WASPF}(s,d). \tag{10}$$

where m_{sd} is the element of traffic matrix M.

```
Algorithm 1: ORRA for a time slot
Input: $R(routingmatrix), T(trafficmatrix)$
Output: $P(robustroutingpath)$
Set: $W(linkweight), UR(users'requirementsmatrix)$,
  $Cmax(linkcapacity)$
% call the OLS algorithm to sleeping links
$[new\_W, sl] = OLS(R, W)$;
% $new\_W$ shows the new link weights
% sl shows the number of sleeping links
$m = Constant$; % set the number of users' requirements
for $i = 1, 2, ..., m$ do
  % call the Dijkstra algorithm
  $Path = Dijkstra(W, Source, Destination)$;
  % the weighted adaptive shortest path first
  if $Path \ = 0$ do
    $k = length(Path)$;
    % call the Criticality algorithm
**mark1:** $Cr1 = Criticality(Cmax)$;
    $temp = Cmax$;
    % modify the link weights
    for $j = 1$ to $k - 1$ do
      if $Cmax(Path(j), Path(j + 1)) - UR(band)$
        $>= low_limit$ do
        $Cmax(Path(j), Path(j + 1)- = UR(band)$
      Else
        % rerouting process
        goto **mark1**;
      end
    End
    % call the Criticality algorithm
    $Cr2 = Criticality(Cmax)$;
    $= (Cr2 - Cr1)$; % difference of criticality
    for $p = 1$ to $k - 1$ do
      $W(Path(p), Path(p + 1)) = W(Path(p),$
        $Path(p + 1)) + 100*$;
      % update the link weights
    end
  end
End %ORRA
```

After moving the link based on the optimized link sleeping algorithm proposed above, the capacity of network links will change. At the moment, we are able to obtain the new link capacity in terms of the new routing matrix by Eq. (10). It is defined as $c_i{}'$, $i \in E'$, where E'presents the set of new network link calculated by the optimized link sleeping algorithm. The Advanced ORRA problem is to make $c_i{}'$ less than the default maximum link capacity, namely

$$c_i{}' \leq C_{\max}(i), \quad \forall i \in E' \tag{11}$$

where $C_{\max}(i)$ is the maximum capacity of the link i. The constraint condition represented by formula (11) is added to Eq. (9) to solve this problem. Consequently, we can achieve the proposed Advanced ORRA. That is, by adding the link-capacity constraint denoted in Eq. (11) to the optimized link sleeping algorithm shown in algorithm 1, then we can overcome this problem and obtain the robust routing paths which can meet the QoS requirements of users.

So far, we have proposed our robust routing algorithm ORRA to reach the higher network energy efficiency of the cloud computing network. By turning the network redundant links into sleep state, simplifying network routing topology, reducing the number of active links, ORRA can reduce the power consumption of transmitting certain information and consequently raise the network energy efficiency. In the following section, we are to use a series of simulation to validate its performance.

4 Simulation result and analysis

In order to validate the proposed ORRA, we conduct a series of simulation experiments. We use the network topology and traffic data from the GEANT backbone network in the simulation process, which holds 23 router nodes and 74 links. EAR-OSPF in [9] is reported as a energy-efficient routing algorithm. Here we will compare the EAR-OSPF and ORRA algorithms. To further validate our algorithm ORRA, we also analyze the OLS algorithm based on OSPF (for short OLS-OSPF). In our numerical experiments, we are to discuss the performance of the three algorithms.

To verify the relationship between the energy efficiency and the number of user requests, we have carried out a corresponding simulation experiment, obtaining the simulation result shown in Fig. 4. The horizontal axis in Fig. 4 represents the 30 experiments, the number of user requests increasing by 20 for each experiment. The information transmitted of per request is the same, and the network energy efficiency is equal to the ratio of the amount of information to the network energy consumption. From Fig. 4, we can see that

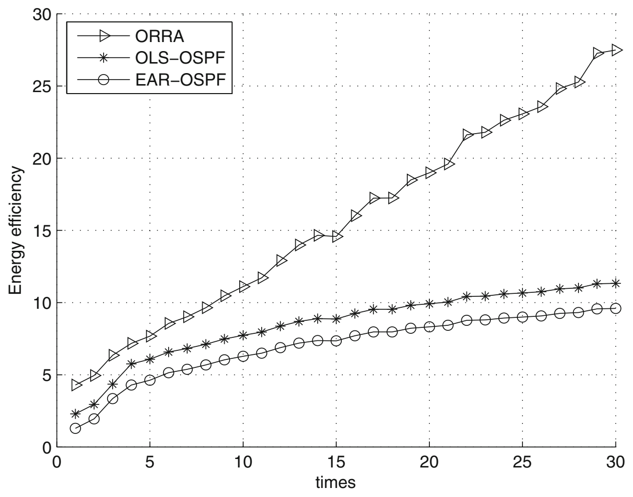

Fig. 4 Energy efficiency of ORRA

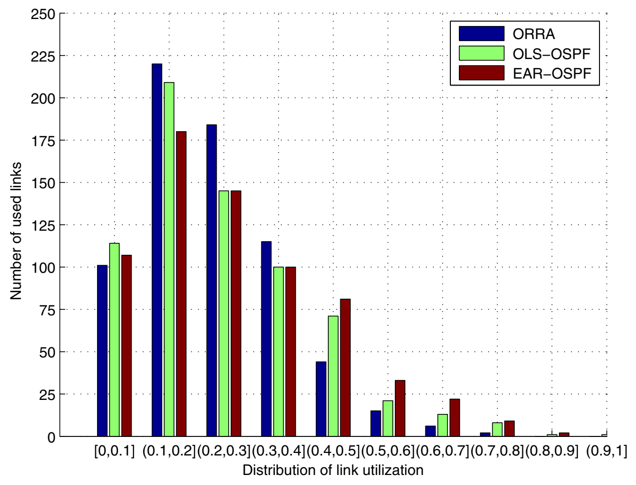

Fig. 5 Link utilization of three algorithms

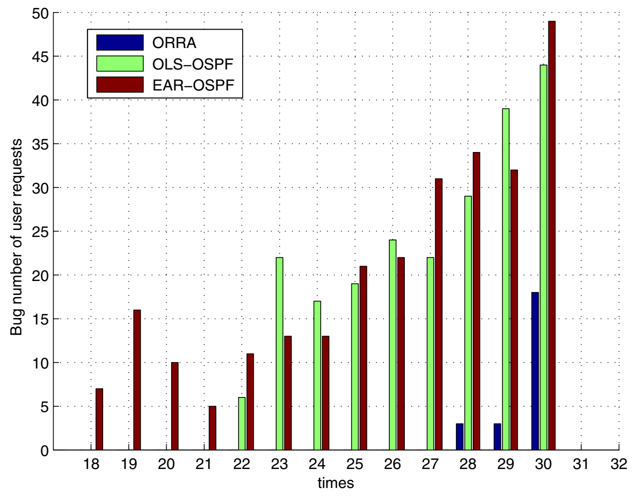

Fig. 6 Distribution of bug number of user requests

with the increase of user requests, the energy efficiency also increase gradually. This is significantly reasonable. According to ORRA, the more the user requests increase, the higher the link utilization is. Consequently, the throughput of networks is quickly enlarged, but due to the sleeping scheme of ORRA, the network energy consumption is slowly added. And thus the network energy efficiency becomes large with the increase of user requests. For OLS-OSPF and EAR-OSPF, we also achieve the similar conclusions. Howerver, in contrast to ORRA, OLS-OSPF and EAR-OSPF hold much lower energy efficiency although their ones increase with the growth of the number of customer requests. This indicates that ORRA can perform the highly energy-efficient packet delivery.

Now we analyze the performance of three algorithms. Fig. 5 is the histogram of link utilization of three algorithms, with the horizontal axis representing the distribution of the link utilization, namely (0, 0.1] denoting the probability of links whose utilization is below 10 %, (0.1,0.2] indicating the link utilization between 10 and 20 %, and others expressing similar meaning. The vertical axis in Fig. 5 represents the number of used links.From Fig. 5, ORRA holds the better performance in the range from 0.1 to 0.4. This indicates that the ORRA routing methods can keep most of link utilization between 20 and 40 % distribution. The link utilization is neither too high to result in network congestion and link failures, nor too low to lead to waste. This shows that ORRA exhibits the much better link utilization than OLS-OSPF and EAR-OSPF.

Figure 6 shows the distribution of the number of the failure user requests due to insufficient bandwidth capacity of the links. The horizontal axis in Fig. 6 represents the time intervals. The vertical axis in Fig. 6 indicates the bug number of user requests which can be routed successfully but the link bandwidth capacity of the routing path does not meet the user requirements. We can easily find from Fig. 6 that there is no bug number before the 28 time interval for ORRA, while OLS-OSPF and EAR-OSPF happen in the earlier time. The performance of EAR-OSPF algorithm is worst while the OLS-OSPF follows. After the 28 time interval, the bug number of ORRA is also far less than OLS-OSPF and EAR-OSPF. This further suggests that ORRA hold the better routing performance than other two algorithms.

Figure 7 plots the distribution of the utilization for each link in the network, where the horizontal axis represents the time intervals and the vertical axis indicates the numerical size of link utilization. The curve describes the usage of every link, and the transverse line is the average usage of three algorithms. Figure 7 tells us that the average link utilization in the descending order is ORRA, OLSA-OSPF, and EAR-OSPF. The average utilization of ORRA is 319.6486, which is greater than those of OLSA-OSPF and EAR-OSPF. Furthermore, we calculate the variance of three algorithms

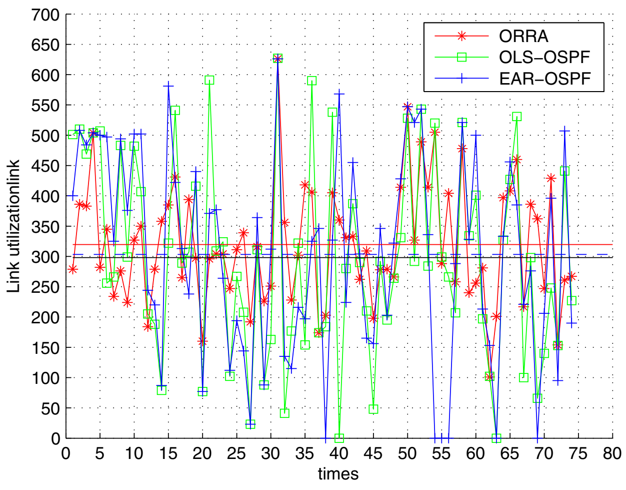

Fig. 7 Distribution of link utilization for three algorithms

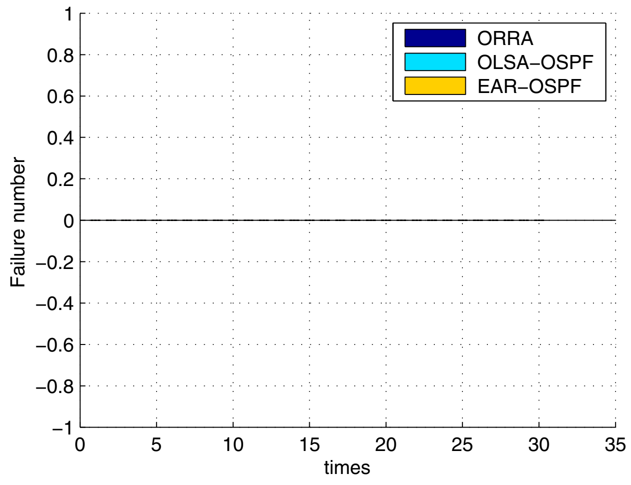

Fig. 9 Statistical distribution of failure routing

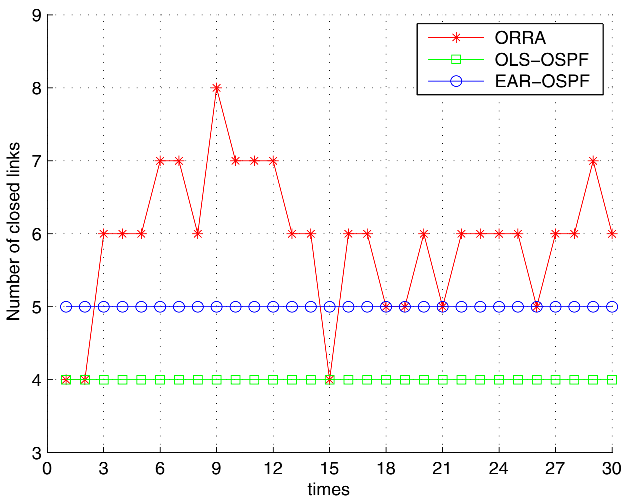

Fig. 8 Distribution of the number of closed links

to present the fluctuation of link utilization. The results show that the variances of ORRA, OLSA-OSPF, and EAR-OSPF are 0.971×10^4, 2.629×10^4 and 2.884×10^4, respectively. The variances of OLSA-OSPF and EAR-OSPF is almost twice more than that of ORRA. This explains that ORRA can effectively use the links. In contrast to OLSA-OSPF and EAR-OSPF, there are no too low or too high link utilizations for ORRA. This is very important for providing the robust and energy-efficient communications.

Figure 8 indicates the distribution of the number of closed links due to sleeping. ORRA can dynamically set the link weight in each time period, which will affect the sleeping and active state of the link and thus will have an impact on the routing path. EAR-OSPF always remains unchanged after initializing, which led to the network not to dynamically change the distribution of routing according to the state of user requests and not to achieve the load balancing. Thereby, the network robustness of EAR-OSPF is relatively weak. As can be seen from Fig. 8, ORRA can dynamically change the number of sleeping link so as to adapt to the varying network. EAR-OSPF and OSL-OSPF do not have such superiority. This explain that ORRA is effective, not only able to improve the network energy efficiency, but also able to meet the request of user QoS and improve the robustness of the network.

Figure 9 denotes the distribution of failure routing. From Fig. 9, we can find that the figure is blank. This shows that the routing performance of three algorithms is very good. This further indicates that these algorithms can obtain the fairly better robustness when perform the energy-efficient networking. More importantly, From the simulations above, we can see that ORRA does not only hold the robustness for cloud computing, but also it has the much higher energy efficiency of networks than other methods. This demonstrates that ORRA is feasible and promising to implement the energy-efficient network for cloud computing.

5 Conclusion

This paper propose a robust routing algorithm to solve the routing problem in energy-efficient networks for cloud computing. By describing the network energy efficiency problem into the optimal process of maximizing the number of the sleeping links in the network, we can find the maximum set of the sleeping links to save the network energy. We exploit the traffic rerouting technology to enhance the link utilization, with the result that the links as many as possible can stay in the sleeping state. Moreover, We present an optimized link sleeping method and a weight adaptive strategy to improve the energy efficiency and robustness of cloud computing networks. Consequently, our algorithm can avoid the traffic congestion and raise the network energy efficiency. Simulation results illuminate that our algorithm holds the better performance than previous methods.

Acknowledgments This work was supported in part by the National Natural Science Foundation of China (No. 61071124), the Program for New Century Excellent Talents in University (No. NCET-11-0075), the Fundamental Research Funds for the Central Universities (Nos. N120804004, N130504003), and the State Scholarship Fund (201208210013). The authors wish to thank the reviewers for their helpful comments.

References

1. Pickavet, M., Vereecken, W., & Demeyer, S. et al. (2008). Worldwide energy needs for ICT: The rise of power-aware networking, In *Proceedings of ANTS'08* (pp. 1–3).
2. Jiang, D., Xu, Z., Chen, Z., et al. (2011). Joint time-frequency sparse estimation of large-scale network traffic. *Computer Networks*, *55*(10), 3533–3547.
3. Jiang, D., Xu, Z., Nie, L., et al. (2012). An approximate approach to end-to-end traffic in communication networks. *Chinese Journal of Electronics*, *21*(4), 705–710.
4. Bolla, R., Bruschi, R., & Davoli, F., et al. (2009). Energy-aware performance optimization for next-generation green network equipment, In *Proceedings of PRESTO'09* (pp. 49–54).
5. Cianfrani, A., Eramo, V., & Listanti, M., et al. (2010). An energy saving routing algorithm for a green OSPF protocol, In *Proceedings of INFOCOM'10* (pp. 1–5).
6. Restrepo, J., Gruber, C., & Machoca, C. (2009). Energy profile aware routing, In *Proceedings of GreenComm'09* (pp. 1–5).
7. Chiaraviglio, L., Mellia, M., & Neri, F. (2009). Reducing power consumption in backbone networks, In *Proceedings of ICC'09* (pp. 1–5).
8. Chiaraviglio, L., Mellia, M., & Neri, F. (2009). Energy-aware backbone networks: A case study, In *Proceedings of ICC'09* (pp. 1–5).
9. Cianfrani, A., Eramo, V., Listanti, M., et al. (2009). An OSPF-integrated routing strategy for QoS-aware energy saving in IP backbone networks. *IEEE Transactions on Network and Service Management*, *9*(3), 254–267.
10. Lai, P., Yang, Q., & Wu, C., et al. (2011). Configuring network topology towards energy-efficient ip networks, In *Proceedings of ICCSN'11* (pp. 95–99).
11. Zhang, D., Yang, Z., Raychoudhury, V., Chen, Z., & Lloret, J. (2013). An energy-efficient routing protocol using movement trends in vehicular Ad hoc networks. *The Computer Journal*, *56*(8), 938–946.
12. Zhang, D., Zhang, D., Xiong, H., Hsu, C., & Vasilakos, A. V. (2014). BASA: Building mobile Ad-Hoc social networks on top of android. *IEEE Network*, *28*(1), 4–9.
13. Jiang, D., Zhao, Z., Xu, Z., Yao, C., & Xu, H. (2014). How to reconstruct end-to-end traffic based on time-frequency analysis and artificial neural network. *AEU-International Journal of Electronics and Communications*, *68*(10), 915–925.
14. Zhang, D., Chen, M., Guizani, M., Xiong, H., & Zhang, D. (2014). Mobility prediction in telecom cloud using mobile calls. *IEEE Wireless Communications*, *21*(1), 26–32.
15. Jiang, D., Xu, Z., Zhang, P., & Zhu, T. (2014). A transform domain-based anomaly detection approach to network-wide traffic. *Journal of Network and Computer Applications*, *40*(2), 292–306.
16. Tizghadam, A., & Leon-Garcia, A. (2010). Autonomic traffic engineering for network robustness. *IEEE Journal on Selected Areas in Communications*, *28*(1), 39–50.
17. Wood, D. R. (1997). An algorithm for finding a maximum clique in a graph. *Operation Research Letters*, *21*(5), 211–217.
18. Jaszkiewicz, A. (2002). On the performance of multiple-objective genetic local search on the 0/1 Knapsack problem—A comparative experiment. *IEEE Transactions on Evolutionary Computation*, *6*(4), 402–412.

Dingde Jiang received the Ph.D degree in communication and information systems from School of Communication and Information Engineering, University of Electronic Science and Technology of China, Chengdu, China, in 2009. He is an Associate Professor in College of Information Science and Engineering, Northeastern University, Shenyang, China. His research interests include energy-efficient networks, cognitive networks, network measurement, network security, Internet traffic engineering, and performance analysis. Dr. Jiang is a member of IEEE and IEICE.

Zhengzheng Xu received the Ph.D degree in management science and engineering in College of Information Science and Engineering, Northeastern University, Shenyang, China. She is working with School of Economics and Management, AnQing Normal University, AnQing 246011, China. Her research interests include supply chain and logistics management, decision analysis, modeling, and optimization.

Jindi Liu received M.S. and B.Sc. in Communication Engineering, Northeastern University. She is currently a graduate student for Ph.D degree for Electrical and Communication Engineering at Northeastern University. Her research interests include energy efficiency network, and anomaly detection.

Wenhui Zhao received B.Sc. in Communications Engineering, Northeast University at Qinhuangdao. She is currently a graduate student for a Master's degree for Electrical and Communication Engineering at Northeast University. Her research interests include network measurement, traffic engineering, and anomaly detection.