An Energy Efficient Approach for Service Chaining Placement in Satellite Ground Station Networks

Xiangqiang Gao[1], Rongke Liu[1] and Aryan Kaushik[2]
[1]School of Electronic and Information Engineering, Beihang University, Beijing, China
[2]Department of Electronic and Electrical Engineering, University College London (UCL), London, United Kingdom
Email: {xggao, rongke_liu}@buaa.edu.cn, a.kaushik@ucl.ac.uk

***Abstract*—In this paper, we investigate the service chaining placement problem for user requests in satellite ground station networks with minimum energy cost, which consists of server energy, switch energy, and link energy. We build the *Server-Switch-Link* energy model and formulate the energy optimization problem as an integer nonlinear programming problem. To address this problem, we implement a prediction-aided Greedy (PA-Greedy) algorithm depending on satellite mission planning in satellite control centers. We conduct the experiments to evaluate the proposed energy model and PA-Greedy algorithm in Fat-Tree networks, and compare the performance with the baseline Greedy algorithm and two energy models of *Server-Link* and *Server*. In a Fat-Tree network with 16 servers, the proposed *Server-Switch-Link* energy model with the PA-Greedy algorithm can reduce energy consumption by 17.28% when compared with the baseline Greedy algorithm, and outperform these *Server-Link* and *Server* energy models by 47.10% and 62.91%, respectively.**

***Index Terms*—Energy optimization, service chaining, virtual network function, satellite ground station, Greedy, Fat-Tree.**

I. Introduction

In data center networks, energy consumption accounts for a considerable proportion of operational expenditure and has been an important issue to be handled for saving operational expenditure [1], [2]. The energy consumption of a data center network derives from network hardware components, which can perform for computing service and routing traffic, such as servers, switches, and links. For existing solutions in green and energy-saving data centers [3], [4], these network components are implemented with multiple running states (e.g., *idle*, *on*, and *off*) in an energy efficient way and these states can be adaptively switched for network components depending on the network states and workloads. In the energy model in [5], [6], the energy consumption of network components in *on* states is linearly proportional to their workloads and there will be no energy consumption if a network component turns off. Energy efficient ethernet introduces a low power idle to reduce the link energy consumption in IEEE 802.3az [7].

Software defined network (SDN) combined with network function virtualization (NFV) has emerged as a new paradigm, which can decouple software and hardware, and enable service functions to run on commodity servers, for flexibly providing service provisioning and facilitating data centers saving energy [8]. However, most of existing work related to energy optimization in SDN/NFV-enabled data centers focuses on reducing the energy consumed by servers [9], or servers and links [10], rather than considering the overall energy cost jointly generated by servers, switches, and links, from the perspective of a network framework [11], [12].

In satellite application scenarios, for earth observation satellites, satellite missions (e.g., remote sensing, environment monitoring, target investigation, etc.) need to be planned beforehand in satellite control centers [13], [14] and the command information concerning satellite missions can be transmitted to target satellites by satellite networks. When the observation mission is completed, the produced data will be sent back to a satellite ground station by inter-satellite links for further processing. As shown in Fig. 1, supposing that satellite ground stations can obtain the content about satellite mission planning in satellite control centers by terrestrial networks in advance, then the downloaded lifetime of observation data produced by satellite missions can be easily predicted depending on the data volume and satellite network states, which can facilitate data centers improving the energy efficiency.

In this paper, we study the service chaining placement problem to minimize the total energy consumption of a satellite ground station network. Specifically, we assume that a satellite ground station consists of multiple radio remote units (RRUs) and a baseband processing unit (BBU) [15], where the BBU is considered as a general data center network, such as Fat-Tree. The procedure of receiving and processing baseband data produced by a satellite mission in a satellite ground station network is considered as a user request. Each user request is composed of multiple virtual network functions (VNFs) in a specific sequence and viewed as a service chaining (SC). We also build a *Server-Switch-Link* energy model by taking into account the energy consumption of servers, switches, and links, comprehensively, and formulate the energy optimization problem as an integer nonlinear programming problem. We assume that the running states for servers, switches, and links can be flexibly managed by a SDN controller, where the *idle/on/off* states of these network components can be arbitrarily switched depending on the current network states and workloads. As a result, we can automatically scale in or out these active network components to reduce energy consumption. Similarly to the procedure in [16], we introduce a prediction-aided Greedy (PA-Greedy) algorithm to address the energy optimization problem. The experiments in Fat-Tree networks are conducted to discuss the performance of the proposed energy model and PA-Greedy algorithm in satellite ground station networks.

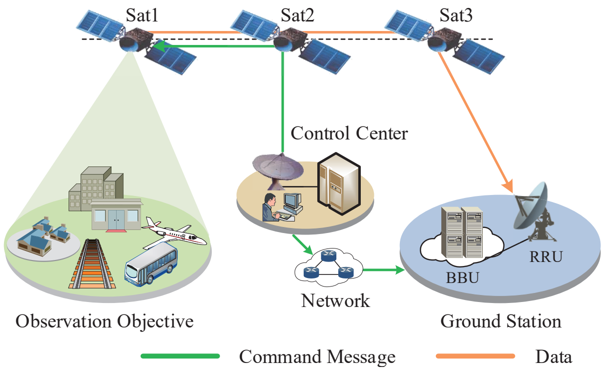


Fig. 1. Satellite application system.

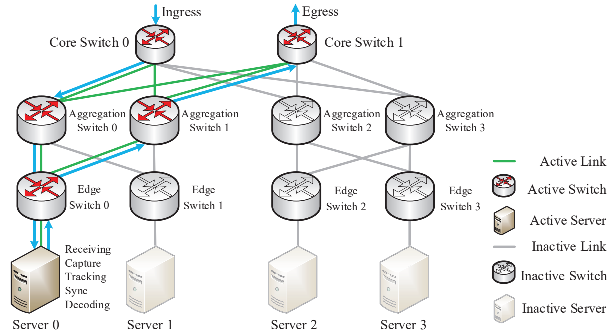


Fig. 2. Example of energy efficient service chaining placement.

II. System Model

A. Satellite Ground Station Network

1) Network Model: We represent a satellite ground station network as a directed graph $G(V, E)$, where V and E denote the set of network nodes and links, respectively. The set V consists of servers V_{svr}, core switches V_{cs}, and aggregation and edge switches V_{sw}. The number of servers in V_{svr} is N and the set of resources offered by each server is denoted as R, where central processing unit (CPU), memory, and graphics processing unit (GPU) are considered in this paper. We indicate the r-th resource capacity of the server v_n as C_n^r. For the link $l \in E$, we denote the bandwidth capacity as B_l and the transmission delay as t_l.

2) Energy Model: For the server energy model, we assume that each server can be in *idle*, *on*, and *off* states and these three states can be switched depending on the current network loads. The server v_n that does not provide any computing resources for user requests is considered into an *idle* state and the energy consumed by the server is indicated as the idle power p_n^{idle}. When the VNFs from user requests are deployed to an *idle* server, the server can be converted into an *on* state to provide computing service for user requests. We represent the energy consumption of the active server v_n as p_n^{on} and the energy consumption is linearly proportional to the resource utilization of the server [5]. When the idle time for the server v_n in an *idle* state is more than the pre-defined maximum idle time, which is indicated as $t_{n,idle}^{max}$, then the server will be converted into an *off* state. We consider that an *off* server will not consume energy. When the VNFs from user requests are assigned to an *off* server, the server will be activated to provide computing service for user requests. We indicate the startup energy consumption of the server v_n as the maximum power p_n^{max}. To avoid the switching state too often for a server [11], we assume that the server v_n can not be turned on until the pre-defined minimum off time, which is indicated as $t_{n,off}^{min}$, has been exceeded.

For the switch energy model, we assume that the switch energy consumption is mainly from switch base hardware, line-cards, and active ports [6], where the energy consumed by active ports is considered in the following link energy model. The energy consumption of switch base hardware and line cards is considered as fixed as they can not scale with the transmission rate and we also assume that all core switches are active all the time [2]. The energy consumption of the switch v_n is indicated as p_{sw}^n. According to reconfiguring network switches [10], we can power down these unused aggregation and edge switches and any unused switch ports to save energy consumption [4], [10].

For the link energy model, we assume that if both source and destination of a link are powered up then the link will be always in an *on* state. However, when any one of source and destination for a link is powered down, the link will be off and can not consume any energy. For an *on* link, there are two states of *sleep* and *wake*, the link can be switched into the *sleep* state during an inactive period which can consume about 10% of the link total energy consumption [7]. We represent the energy consumption for the *sleep* link l as p_l^{idle} and for the *wake* link l as p_l^{on}. We also indicate the maximum energy consumption of the link l as p_l^{max}. A hybrid link energy model [12] is used in this paper, where the energy consumption of a link is linearly proportional to the transmission rate.

B. User Requests

We represent the set of user requests as U including M user requests. The use request u_m is viewed as a directed acyclic graph $G(F_m, H_m)$ and has an acceptable maximum delay, which is denoted as t_m^{max}. The set of virtual network functions is indicated as $F_m = \{f_{m,1} = s_k, f_{m,2}, \cdots, f_{m,|F_m|} = d_m\}$, we use s_m and d_m to indicate the source and the destination, respectively. For the VNF $f_{m,k}$, we denote the r-th resource requirements as $c_{m,k}^r$ and the computing delay as $t_{m,k}$. As the ingress and the egress, we assume that there are no computing resource and delay requirements for the source s_m and the destination d_m. The set of edges for the user request u_m is represented as H_m. We denote the edge between f_{m,k_1} and f_{m,k_2} as $h_m^{k_1,k_2}$ and the bandwidth requirements as $b_m^{k_1,k_2}$.

C. Energy Efficient Model for Service Chaining Placement

In this paper, we assume that the source and the destination for a user request need to be deployed on two core switches as the ingress and the egress, respectively. All the VNFs except the source and the destination for each user request can be assigned to these active servers for performing service functions. In order to save the energy of a satellite ground station network, we can deploy more VNFs to an active server as far as possible and reduce the number of used servers. Furthermore, the adjacent VNFs for a user request should be placed on the same server to decrease the used bandwidth resources and save the link energy. Depending on reconfiguring the virtual network, we can temporarily switch off these unused network components including servers, switches, and ports for saving the energy while guaranteeing the resource requirements of user requests [4]. The unused links can be also converted into the *sleep* state. According to the changes of network loads, these network components in the *off* or *sleep* states can be adaptively activated to provide computing or routing service for user requests. An example of energy efficient service chaining placement is shown in Fig. 2. We just activate the used network components (e.g., server 0, edge switch 0, aggregation switch 0 and switch 1) to provide computing and routing service for a user request. However, the unused servers, aggregation and edge switches, and links are still in the *off* state for saving energy.

III. Problem Formulation and Proposed Approach

A. Problem Formulation

In order to formulate the energy optimization problem, we use a binary variable $x_n = \{0, 1\}$ to indicate whether the network node v_n is active. $x_n = 1$ if the network node v_n is active, otherwise $x_n = 0$. We use another binary variable $x_l = \{0, 1\}$ to indicate whether the link l is active. $x_l = 1$ if the link l is active, otherwise $x_l = 0$.

We also denote a binary decision variable $x^n_{m,k} = \{0, 1\}$ to indicate whether the VNF $f_{m,k}$ is deployed to the network node v_n. $x^n_{m,k} = 1$ if the VNF $f_{m,k}$ is deployed to the network node v_n, otherwise $x^n_{m,k} = 0$. We represent the set of the d shortest paths between two network nodes v_{n_1} and v_{n_2} as $p^{n_2}_{n_1}$. We denote another binary decision variable $x^{k_1,k_2}_{m,p} = \{0, 1\}$ to indicate whether the path p is used by the edge h^{k_1,k_2}_m. $x^{k_1,k_2}_{m,p} = 1$ if the path p is used by the edge h^{k_1,k_2}_m, otherwise $x^{k_1,k_2}_{m,p} = 0$. We also use a binary variable $x^p_l = \{0, 1\}$ to indicate whether the link l is used by the path p. $x^p_l = 1$ if the link l is used by the path p, otherwise $x^p_l = 0$.

According to the above energy model, we consider that the energy consumption of a satellite ground station network is from servers, switches, and links. The energy consumption of the server v_n can be expressed as [5]

$$p^{on}_n = p^{idle}_n + \frac{\sum_{u_m \in U} \sum_{f_{m,k} \in F_m} x^n_{m,k} \cdot c^{cpu}_{m,k}}{C^{cpu}_n} \cdot (p^{\max}_n - p^{idle}_n), \quad (1)$$

and the energy consumption of servers can be indicated as

$$P_{svr} = \sum_{v_n \in V_{svr}} x_n \cdot p^{on}_n. \quad (2)$$

We represent the energy consumption of switch base hardware and per line-card port as p_{chasis} and $p^{port}_{linecard}$, respectively. The number of line-card ports is indicated as $n^{port}_{linecard}$. The energy consumption of the switch v_n can be expressed as [6]

$$p^n_{sw} = p_{chasis} + p^{port}_{linecard} \cdot n^{port}_{linecard}, \quad (3)$$

and the energy consumption of switches can be indicated as

$$P_{sw} = \sum_{v_n \in V_{cs} \cup V_{sw}} x_n \cdot p^n_{sw}. \quad (4)$$

The energy consumption of the link l can be expressed as [12]

$$p^{on}_l = p^{idle}_l + \frac{b^{used}_l}{B_l} \cdot (p^{\max}_l - p^{idle}_l), \quad (5)$$

where the variable b^{used}_l indicates the used bandwidth resources of the link l. For $\forall h^{k_1,k_2}_m \in H_m$, and $\forall v_{n_1}, v_{n_2} \in V_{svr} \cup V_{cs}$, then b^{used}_l can be expressed as

$$b^{used}_l = \sum_{u_m \in U} \sum_{h^{k_1,k_2}_m} b^{k_1,k_2}_m \sum_{v_{n_1}, v_{n_2}} x^{n_1}_{m,k_1} \cdot x^{n_2}_{m,k_2} \sum_{p \in p^{n_2}_{n_1}} x^{k_1,k_2}_{m,p} \cdot x^p_l. \quad (6)$$

Thus, the energy consumption of links can be indicated as

$$P_{link} = \sum_{l \in E} x_l \cdot p^{on}_l. \quad (7)$$

The total energy consumption of a satellite ground station network is the energy consumption sum of servers, switches, and links. We can indicate the energy consumption as

$$P_{net} = P_{svr} + P_{sw} + P_{link}, \quad (8)$$

When the service chaining placement for user requests is performed in a satellite ground station network, we need to consider the following physical constraints.

It needs to be ensured that each VNF $f_{m,k}$ should be deployed to only one network node that is a server or a core switch. The placement constraint can be represented as

$$\sum_{v_n \in V_{svr} \cup V_{cs}} x^n_{m,k} = 1, \forall u_m \in U, \forall f_{m,k} \in F_m. \quad (9)$$

It needs to be ensured that when two adjacent VNFs f_{m,k_1} and f_{m,k_2} are assigned on two network nodes a path p between the two network nodes can be used to route the traffic. For $\forall v_{n_1}, v_{n_2} \in V_{svr} \cup V_{cs}, \forall h^{k_1,k_2}_m \in H_k$, the path constraint can be indicated as

$$x^{n_1}_{m,k_1} \cdot x^{n_2}_{m,k_2} = \sum_{p \in p^{n_2}_{n_1}} x^{k_1,k_2}_{m,p}. \quad (10)$$

We also ensure that the total required resources for a server can not exceed the resource capacity. For $\forall v_n \in V_{svr}, \forall r \in R$, the server resource constraint can be expressed as

$$\sum_{u_m \in U} \sum_{f_{m,k} \in F_m} x^n_{m,k} \cdot c^r_{m,k} \le x_n \cdot C^r_n. \quad (11)$$

We also ensure that the total required resources for a link can not exceed the resource capacity. For $\forall l \in E, \forall h^{k_1,k_2}_m \in H_m, \forall v_{n_1}, v_{n_2} \in V_{svr} \cup V_{cs}$, the link resource constraint can be expressed as

$$\sum_{u_m \in U} \sum_{h^{k_1,k_2}_m} b^{k_1,k_2}_m \sum_{v_{n_1}, v_{n_2}} x^{n_1}_{m,k_1} \cdot x^{n_2}_{m,k_2} \sum_{p \in p^{n_2}_{n_1}} x^{k_1,k_2}_{m,p} \cdot x^p_l \le x_l \cdot B_l. \quad (12)$$

Furthermore, we also ensure that the source-to-destination delay for each user request should be less than the acceptable maximum delay. For $\forall u_m \in U$, the delay constraint can be indicated as

$$t_m^{trans} + \sum_{f_{m,k} \in F_m} t_{m,k} \leq t_m^{\max}, \tag{13}$$

where t_m^{trans} indicates the total transmission delay for the user request u_m. For $\forall h_m^{k_1,k_2} \in H_m, \forall v_{n_1}, v_{n_2} \in V_{svr} \cup V_{cs}$, the transmission delay t_m^{trans} can be indicated as

$$t_m^{trans} = \sum_{h_m^{k_1,k_2}} \sum_{v_{n_1}, v_{n_2}} x_{m,k_1}^{n_1} \cdot x_{m,k_2}^{n_2} \sum_{p \in p_{n_1}^{n_2}} x_{m,p}^{k_1,k_2} \sum_{l \in p} t_l. \tag{14}$$

The *idle* time of each server is guaranteed to be less than the pre-defined maximum *idle* time. We represent the earliest and current idle time for the server v_n as $t_{n,idle}^{early}$ and $t_{n,idle}^{curr}$, respectively. We can indicate the idle time constraint as

$$t_{n,idle}^{curr} - t_{n,idle}^{early} \leq t_{n,idle}^{max}. \tag{15}$$

The *off* time of each server is guaranteed to be greater than the pre-defined minimum *off* time. We represent the earliest and current *off* time for the server v_n as $t_{n,off}^{early}$ and $t_{n,off}^{curr}$, respectively. We can indicate the off time constraint as

$$t_{n,off}^{curr} - t_{n,off}^{early} \geq t_{n,off}^{min}. \tag{16}$$

The energy optimization problem for service chaining placement can be expressed as

$$\begin{aligned} \min \quad & P_{net} \\ \text{subject to} \quad & (9) - (16). \end{aligned} \tag{17}$$

B. Proposed Approach

The service chaining placement problem in a data center network has been proven to be NP-hard in [11], [16], [17], the optimal solution for the problem can be only obtained in a small scale problem. Therefore, a heuristic Greedy algorithm is used to obtain the near optimal solution in this paper. Considering that the user requests to arrive and to end over a time frame can be predicted by satellite mission planning in satellite control centers [16], we propose the prediction-aided Greedy algorithm to tackle the energy optimization problem in a satellite ground station network.

The procedure of the proposed PA-Greedy algorithm for a dynamic service deployment environment is described in Algorithm 1. A batch mode is used to allocate the available network resources for user requests in each time slot. We suppose that we can predict the user requests U_t^{end} to end in the current time slot t and the user requests U_{t+1}^{new} to arrive in the next time slot $t+1$, respectively. According to the predicted information about user requests to arrive and end, we can pre free the network resources used by the user requests U_t^{end} to be available for deploying the new user requests. For each user request $u \in U_{t+1}^{new}$, we search all the current available servers V_t^{svr} to obtain the best placement strategy $v^*(u,t)$ with minimum energy consumption. According to performing the Greedy search process for each user request

Algorithm 1 Prediction-aided Greedy Algorithm.

Initialize: $t = 0$;
while $t < T$ **do**
  Predict the user requests U_t^{end} to over before the end of time slot t and pre free these used network resources;
  Obtain the user requests U_{t+1}^{new} to occur at the beginning of time slot $t+1$;
  for each $u \in U_{t+1}^{new}$ **do**
    for each $v \in V_t^{svr}$ **do**
      if these constraints in (9)-(16) are matched **then**
        Calculate the current energy consumption $P_{net,t}^{u,v}$ of the network with $v(u,t)$;
      end if
    end for
    Find the solution $v^*(u,t)$ with minimum energy consumption according to $v^*(u,t) = \arg\min_{v(u,t)} P_{net,t}^{u,v}$;
  end for
  $v^*(t) = \{v^*(u,t) | u \in U_{t+1}^{new}\}$;
  Free the network resources used by the requests U_t^{end};
  Assign the available network resources to the user requests U_{t+1}^{new} by the best strategy $v^*(t)$;
  Update the remaining network resources and states;
  t ← t+1;
end while

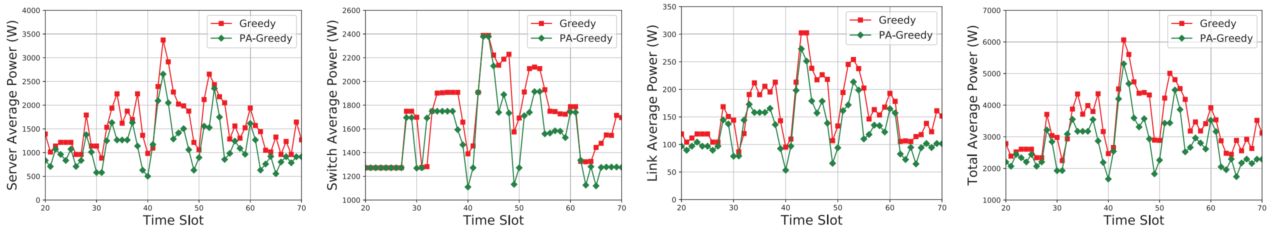
in U_{t+1}^{new}, we can obtain the best placement strategy profile $v^*(t) = \{v^*(u,t) | u \in U_{t+1}^{new}\}$. After the network resources used by the completed user requests U_t^{end} are freed, we will provide the available network resources for the user requests U_{t+1}^{new} via the best strategy profile $v^*(t)$.

IV. Simulation Results

We conduct the following experiments to simulate the energy consumed by servers, switches, and links in Fat-Tree networks. We perform the experiments for 50 times and obtain the results on average, where the running time of each experiment is 24 hours.

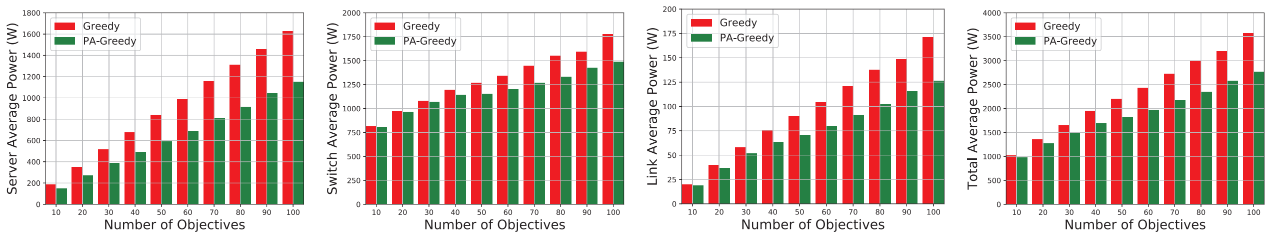
For Fat-Tree networks, all servers are homogeneous and each server includes 96 vCPUs, 112 GB Memory, and 12 GPUs. For each server, the idle and maximum active power is 49.9 W and 415 W, the maximum idle time is 3 time slots, the minimum off time is 1 time slot, and the startup time is 1 time slot. The power of switch base hardware is 143 W and the power of per line-card port is 4 W. The bandwidth capacity for a link between an edge switch and a server is 1 Gbps and for a link between switches is 10 Gbps. The idle and maximum active power of each link is 3 W and 30 W, respectively. The transmission delay for each link is 0.05 ms. We denote the number of the shortest paths between two network nodes as $d = 8$.

In addition, we assume that the observation objectives are randomly generated and observed by low earth orbit (LEO) observation satellites, where the downloaded data time for each user request can be considered as the visible window time between a LEO satellite and its observation objective. The



(a) Energy consumption of servers (b) Energy consumption of switches (c) Energy consumption of links (d) Total energy consumption

Fig. 3. Energy consumption for $K_{obj} = 90$ in a Fat-Tree network with 16 servers.



(a) Energy consumption of servers (b) Energy consumption of switches (c) Energy consumption of links (d) Total energy consumption

Fig. 4. Energy consumption for different objectives in a Fat-Tree network with 16 servers.

TABLE I
PARAMETER SETTINGS FOR USER REQUESTS

Name	vCPU	Memory	GPU	Throughput	Delay
Receiving	6	9 GB	0	100 Mbps	20 ms
Capture	7	11 GB	1	100 Mbps	1.5 s
Tracking	9	12 GB	1	100 Mbps	100 ms
Synchronization	14	12 GB	1	100 Mbps	10 ms
Decoding	3	5 GB	1	100 Mbps	25 ms

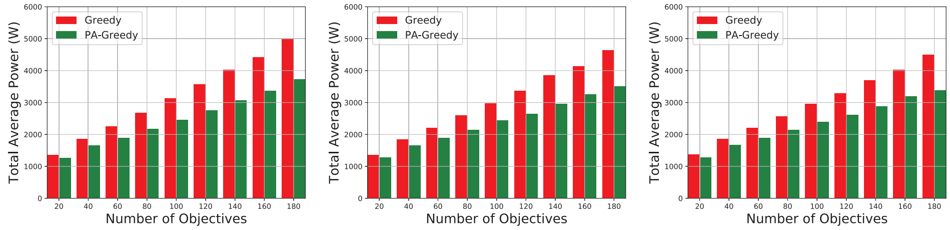
number of observation objectives is indicated as K_{obj}. Each user request consists of source, receiving, capture, tracking, synchronization, decoding, and destination, where these resource requirements are summarized in Table I. The acceptable maximum delay for each user request is 1.8 seconds.

In satellite ground station scenarios, we evaluate the proposed energy model with the PA-Greedy algorithm [16] and the baseline Greedy algorithm [17] in a Fat-Tree network with 16 servers. Fig. 3 shows the energy consumption results for $K_{obj} = 90$ in a Fat-Tree network with 16 servers. Fig. 3(a) describes the energy consumption of servers over time slots. We can find that the energy consumption obtained the PA-Greedy algorithm is less than that of the baseline Greedy algorithm. The energy consumption reduces by 28.36% for the PA-Greedy algorithm on average. The similar results can be also found in Fig. 3(b) and Fig. 3(c), which represent the energy consumption of switches and links, respectively. We can save about 10.58% of the energy consumed by switches and 22.27% of the energy consumed by links for the PA-Greedy algorithm. The total energy consumption of the Fat-Tree network is shown in Fig. 3(d). We can find that the PA-Greedy algorithm performs better than the baseline Greedy algorithm. On average, the total energy consumption decreases by 19.22% for the PA-Greedy algorithm.

To further discuss the proposed energy model, we conduct the experiments for different number of objectives in a Fat-Tree network with 16 servers, as shown in Fig. 4. For all cases, we can observe that the PA-Greedy algorithm outperforms the baseline Greedy algorithm in saving the energy consumed by servers, switches, links, and the network, respectively. On average, the energy consumed by servers, switches, links, and the network reduces by 28.54%, 9.10%, 21.61%, and 17.28% for the PA-Greedy algorithm, respectively.

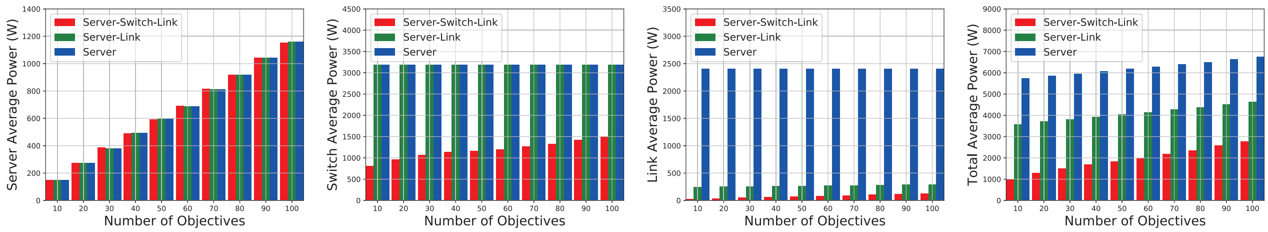
Furthermore, we conduct the experiments in Fat-Tree networks with 32, 48, and 64 servers to evaluate the effectiveness of the proposed energy model in different network scales. The energy consumption results for different number of objectives in the three Fat-Tree networks are shown in Fig. 5. We can also observe that the PA-Greedy algorithm performs better than the baseline Greedy algorithm. The total energy consumption obtained by the PA-Greedy algorithm in Fat-Tree networks with 32, 48, and 64 servers can reduce by 20.82%, 19.33%, and 18.98%, respectively.

We use the PA-Greedy algorithm to discuss the performance of three energy models, such as *Server-Switch-Link*, *Server-Link*, and *Server*, in a Fat-Tree network with 16 servers, as shown in Fig. 6. The energy consumption of servers is only considered in the *Server* energy model [9]. The *Server-Link* energy model considers the energy consumed by servers and links [12]. The *Server-Switch-Link* energy model is used in this paper. We can observe in Fig. 6(a) that the energy consumed by servers is relatively close for the three energy models as the energy consumption of servers is considered in the three energy models. However, due to the lack of considering the switch energy consumption in these *Server-Link* and *Server* energy models, The energy consumed by switches in these two energy models is not reduced, as shown in Fig. 6(b). Similarly,



(a) Fat-Tree with 32 servers (b) Fat-Tree with 48 servers (c) Fat-Tree with 64 servers

Fig. 5. Energy consumption in Fat-Tree networks with 32, 48, and 64 servers.



(a) Energy consumption of servers (b) Energy consumption of switches (c) Energy consumption of links (d) Total energy consumption

Fig. 6. Energy consumption for different energy models.

we can find in Fig. 6(c) that the link energy consumption in the *Server* energy model is also not decreased. The total energy consumption for these three energy models is described in Fig. 6(d). We can observe that the *Server-Switch-Link* energy model performs better than the other two energy models. Compared with these *Server-Link* and *Server* energy models, the energy consumption for the *Server-Switch-Link* energy model can reduce by 47.10% and 62.91%, respectively.

V. Conclusion

This paper studies the service chaining placement problem in satellite ground station networks with an aim to minimize the total energy consumption of a satellite ground station network. From the perspective of the network framework, we build the *Server-Switch-Link* energy model and jointly consider the energy optimization problem of servers, switches, and links, where we can switch off servers, switches, links, and active ports according to the virtual network reconfiguration. We use the PA-Greedy algorithm to address the energy optimization problem. We conduct the experiments in Fat-Tree networks to evaluate the proposed energy model and PA-Greedy algorithm for satellite ground station scenarios. The simulation results show the effectiveness of the proposed energy model and the PA-Greedy algorithm in satellite ground station networks. In a Fat-Tree network with 16 servers, the proposed PA-Greedy algorithm performs better 17.28% than the baseline Greedy algorithm, the energy consumed by the proposed *Server-Switch-Link* energy model is less 47.10% and 62.91% than that of these two *Server-Link* and *Server* energy models, respectively.

References

[1] M. Lin *et al.*, "Dynamic right-sizing for power-proportional data centers," *IEEE/ACM Trans. Netw.*, vol. 21, no. 5, pp. 1378–1391, 2013.

[2] D. Kliazovich *et al.*, "DENS: Data center energy-efficient network-aware scheduling," in *Proc. Int. Conf. Green Comput. Commun./Int. Conf. Cyber, Phys. Soc. Comput.*, Hangzhou, China, Dec. 2010, pp. 69–75.

[3] C. Canali *et al.*, "Joint minimization of the energy costs from computing, data transmission, and migrations in cloud data centers," *IEEE Trans. Green Commun. Netw.*, vol. 2, no. 2, pp. 580–595, 2018.

[4] Y. Zhang *et al.*, "HERO: Hierarchical energy optimization for data center networks," *IEEE Syst. J.*, vol. 9, no. 2, pp. 406–415, 2015.

[5] M. Dayarathna *et al.*, "Data center energy consumption modeling: A survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 732–794, 2017.

[6] O. Popoola *et al.*, "On energy consumption of switch-centric data center networks," *J. Supercomput.*, vol. 74, no. 1, pp. 334–369, 2018.

[7] K. Christensen *et al.*, "IEEE 802.3az: The road to energy efficient ethernet," *IEEE Commun. Mag.*, vol. 48, no. 11, pp. 50–56, 2010.

[8] Y. Tarutani *et al.*, "Virtual network reconfiguration for reducing energy consumption in optical data centers," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 6, no. 10, pp. 925–942, 2014.

[9] B. Yang *et al.*, "Stackelberg game approach for energy-aware resource allocation in data centers," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 12, pp. 3646–3658, 2016.

[10] R. Solozabal *et al.*, "Virtual network function placement optimization with deep reinforcement learning," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 2, pp. 292–303, 2020.

[11] B. Kar *et al.*, "Energy cost optimization in dynamic placement of virtualized network function chains," *IEEE Trans. Netw. Serv. Manag.*, vol. 15, no. 1, pp. 372–386, 2018.

[12] N. Huin *et al.*, "Energy-efficient service function chain provisioning," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 10, no. 3, pp. 114–124, 2018.

[13] M. Tipaldi *et al.*, "A survey on model-based mission planning and execution for autonomous spacecraft," *IEEE Syst. J.*, vol. 12, no. 4, pp. 3893–3905, 2018.

[14] D. Zhou *et al.*, "Mission aware contact plan design in resource-limited small satellite networks," *IEEE Trans. Commun.*, vol. 65, no. 6, pp. 2451–2466, 2017.

[15] F. Marzouk *et al.*, "On energy efficient resource allocation in shared rans: Survey and qualitative analysis," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1515–1538, 2020.

[16] X. Gao *et al.*, "Service chaining placement based on satellite mission planning in ground station networks," *IEEE Trans. Netw. Serv. Manag.*, 2020, doi:10.1109/TNSM.2020.3045432.

[17] J. Liu *et al.*, "Improve service chaining performance with optimized middlebox placement," *IEEE Trans. Serv. Comput.*, vol. 10, no. 4, pp. 560–573, 2015.