ISSN 2383-630X(Print) / ISSN 2383-6296(Online)
Journal of KIISE, Vol. 44, No. 7, pp. 719-725, 2017. 7
https://doi.org/10.5626/JOK.2017.44.7.719

Markov Approximation 프레임워크 기반 네트워크 서비스 체인 임베딩 기법 연구

(A Markov Approximation-Based Approach for Network Service Chain Embedding)

팜 츄 안 † (Pham Chuan)　뉴 엔 후 낫 민 † (Minh N. H. Nguyen)　홍 충 선 †† (Choong Seon Hong)

요 약 네트워크의 관리 비용을 줄이고 성능을 향상시키기 위해 ETSI(European Telecommunication Standards Institute)는 클라우드·데이터 센터에서 네트워크 기능(Network Function)을 소프트웨어 형태로 구현할 수 있는 네트워크 기능 가상화(Network Function Virtualization) 개념을 도입했다. 네트워크 기능 가상화 구조 내에서 네트워크 기능을 물리적 노드(예: 범용 서버)에 네트워크 기능을 호스팅하여 실제 리소스를 공유할 수 있다. 네트워크 기능 가상화를 지원하는 네트워크 서비스 제공 업체의 경우, 효율적인 자원 할당 방법을 통해 운영비용(OPEX) 및 자본 비용(CAPEX)를 줄일 수 있다. 이에 본 논문에서는 최적화 방법을 통해 Network Service Chain Embedding 문제를 분석하고 Markov Approximation 프레임워크 기반 최적의 솔루션을 제안한다. 제안사항에 대한 시뮬레이션 결과는 평균 CPU 사용률이 73%, 링크 사용률이 최대 53% 증가함을 보여준다.

키워드: 네트워크 기능, 네트워크 기능 가상화, 데이터센터, 서비스 체인

Abstract To reduce management costs and improve performance, the European Telecommunication Standards Institute (ETSI) introduced the concept of network function virtualization (NFV), which can implement network functions (NFs) on cloud/datacenters. Within the NFV architecture, NFs can share physical resources by hosting NFs on physical nodes (commodity servers). For network service providers who support NFV architectures, an efficient resource allocation method finds utility in being able to reduce operating expenses (OPEX) and capital expenses (CAPEX). Thus, in this paper, we analyzed the network service chain embedding problem via an optimization formulation and found a close-optimal solution based on the Markov approximation framework. Our simulation results show that our approach could increases on average CPU utilization by up to 73% and link utilization up to 53%.

Keywords: network functions, network function virtualization, datacenters, service chains

· This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIP) (B0190-16-2017, Resilient/ Fault-Tolerant Autonomic Networking Based on Physicality, Relationship and Service Semantic of IoT Devices)

· 이 논문은 제43회 동계학술발표회에서 'Markov Approximation Based Approach for Network Service Chain Embedding'의 제목으로 발표된 논문을 확장한 것임

† 학생회원 : 경희대학교 컴퓨터공학과
pchuanvn@gmail.com
nhatminhhcmut@gmail.com

†† 종신회원 : 경희대학교 컴퓨터공학과 교수(Kyung Hee Univ.)
cshong@khu.ac.kr
(Corresponding author임)

논문접수 : 2017년 1월 23일
(Received 23 January 2017)
논문수정 : 2017년 4월 21일
(Revised 21 April 2017)
심사완료 : 2017년 4월 24일
(Accepted 24 April 2017)

정보과학회논문지 제44권 제7호(2017. 7)

1. 서 론

Traditionally, service provision of network providers, such as telecommunication service provider (TSP), infrastructure provider (InP), etc., has been based on deploying physical proprietary devices and equipment for specific services [1], [2]. There are many aspects of the resource management in a network operator, such as efficient resource allocation, reducing power consumption, improving the resource utilization and performance. Working in the legacy networks, network functions (NFs), such as network address translation (NAT), load balancer (LB), firewall (FW), etc., are implemented on physical middleboxes [2]. Middle-boxes are based on special-purpose hardware and very expensive as well as too difficult to maintain and upgrade. To reduce the operational cost and improve the performance in implementation, the European Telecommunications Standards Institute (ETSI) recently introduces the concept of network function virtualization (NFV) that can run NFs on industry standard servers (commodity servers) [1]. This opens significant ways to manage the network system of providers efficiently in terms of reducing the operation and management cost. Using the NFV architecture, NFs can share the physical resources by implementing virtual network functions (VNFs) as software embedded on physical nodes.

However, there are some challenges in embedding VNFs on physical nodes. *First*, VNFs do not run independently in the network. For example, to require a web service, a network provider has to implement an FW to secure the packets, an LB to distribute the workload, a web server to execute HTTP requests. That means a set of VNFs is specified and the flows traverse these VNFs in a specific order. The set of VNFs for a specific service is called a service chain. *Second*, in implementation on cloud nodes, computing resources (such as CPU, memory, etc.) are explicit however sharing network traffic is still abstract and is often ignored. Finally, although VNFs now can embed on distributed cloud nodes by using algorithms of virtual machine placement approaches [6-8], they cannot be applied directly on the SC embedding due to the relationship of VNFs.

Considering the implementation for service chains, we focus on how to reduce the cost that is incurred in the system when implementing VNFs on physical nodes (named network service chain embedding). The diverse of service chains and VNFs increases the complexity of this problem. Currently, allocation approaches (Best-fit, First-fit) [4] cannot achieve a good result due to the relationship between VNFs in a service chain. Fig. 1 shows an example of service chain embedding with 3 service chains and 4 physical nodes. The relationship of NFs is represented as links in example of Fig. 1. To implement these service chains, all NFs can be hosted on nodes and virtual connections between NFs can be shared physical links that connect physical nodes. Reducing the number of active nodes in implementation is one of the targets that can reduce the operational cost of the network provider.

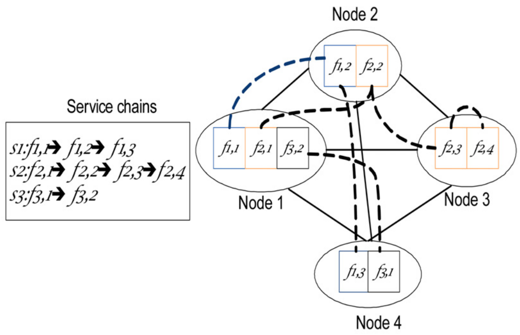


Fig. 1 An example of network service chain embedding

In this paper, we formulate the service chain embedding problem as a combinatorial optimization problem to minimize the operational cost. We then study a Markov approximation mechanism that can find a close optimal solution for this combinatorial NP-hard problem. By using log-sum-exp to transform the original problem, we can apply the Markov approximation framework [10]. We design a specific Markov chain and derive the transition rate between states. Based on the stationary distribution, we design a distributed embedding algorithm to implement service chains.

The rest of the paper is organized as follows. Section 2 discusses about the related work. The problem statement is presented in Section 3. We then represents the Markov approximation approach for solving the service chain embedding problem in Section 4.

The algorithm is proposed in Section 5. Section 6 presents about the simulation results. Finally, we present the conclusion in Section 7.

2. Related work

To support the NFV technology, ETSI and IETF have defined network architectures that enable to allocate resource for VNFs as well as to manage and orchestrate NFV to support services. NFs can be deployed and reassigned to share physical resource pool of the infrastructure, so as to satisfy scalability and performance requirements in implementation of the network. Fundamentally, the NFV architectures includes three main components: Network service chains, NFVI and the NFV Management and Orchestration (NFV-MANO), as shown in Fig. 2.

Resource allocation in NFV is a problem widely studied in literature [1], [3]. In these existing works, the resource allocation in NFV deals with the minimization of the operational cost, the network latency, and the number of rejected service chains. However, the service chain placement problem is NP-hard so that the current solution cannot guarantee the small gap compared to the optimal solution.

Furthermore, the current service chain placement solutions in [4], [5], [6] ignore the complicated relationship between VNFs in service chains. Because it increases the combination in the placement problem, which is more difficult to solve.

To find the solution for service chain embedding, we choose the Markov approximation framework, a famous framework to solve the combinatorial optimization problem. The robustness of this framework is convinced in several complicated applications such as maximum weighted independent set, optimal neighbor problem in peer-to-peer streaming system [9], virtual machine placement [7]. This approach not only guarantees the close-optimal solution but also enables a simple implementation in practice.

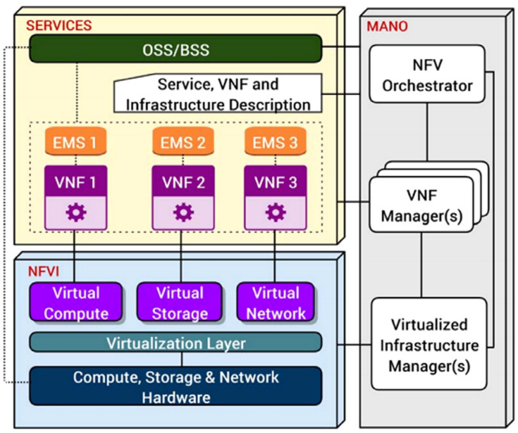


Fig. 2 ETSI-NFV architecture [1]

3. System model and problem formulation

3.1 System model

We consider a service provider scenario that users request a set $S=\{s_1,s_2,\ldots,s_I\}$ of service chains. Each service chain i includes a set N_i of VNFs. Using physical infrastructure, the network operator determines a scheme to embed VNFs on physical nodes (servers). In Fig. 1, we show an example that embeds 3 service chains on 4 physical nodes.

Furthermore, VNFs in a specific service chain require virtual connections (links) to forward packets (e.g., in a web service chain, an NAT is implemented in front an FW. This implementation requires a virtual connection to forward packets from NAT to FW). The virtual connections are also embedded and shared the network traffic on the physical links. We consider the computation resource of each VNF with the unit of CPU, and define f_{ij}^c as the amount of CPU unit requirement of VNF j in service chain i. When embedding all service chains on physical nodes, the network operator has to ensure following constraints:

$$\sum_{i\in S}\sum_{j\in N_i} x_{ijn} f_{ij}^c \leq n^c, \forall n\in N, \quad (1)$$

$$\sum_{s\in S} x_{ijs} = 1, \forall i\in S, \forall j\in N_i, \quad (2)$$

where N is the set of physical nodes, and n^c is the capacity of node n, and x_{ijn} is the indicator variable ($x_{jin}=1$ if node n is hosting VNF j of service chain i, otherwise $x_{ijn}=0$). Constraint (1) guarantees that all VNFs embedded on a node n cannot exceed the capacity of that node. (2) implies the system has to embed all VNFs and each VNF is only hosted at one node.

Moreover, each service chain i is a directed path, including a set V_i of virtual links. We define $d(l_v^i)$ as the data rate flowing in the link l_v of service chain i. Also, we denote $d(l_p)$ as the capacity data rate of the physical link l_p belong to the set L of physical links. Thus, we have the following constraints:

$$\sum_{i\in S}\sum_{v\in V_i} y_{vp} d(l_v^i) \leqq d(l_p),\ \forall p\in L, \quad (3)$$

$$\sum_{p\in L} y_{vp} = 1,\ \forall i\in S, v\in V_i. \quad (4)$$

where y_{vp} is the indicator variable to indicate virtual link l_v^i of service chain i embedded on the physical link l_p. (3) implies that the total amount of traffic passing through the shared link l_p should not exceed its processing capacity. (4) ensures that each virtual link l_v^i of each service chain i is embedded on only one physical link l_p.

The objective in embedding service chains is to reduce the cost to operate the physical nodes. This cost can be measured based on the amount of energy consumption of each node. Consider p^a and p^0 as the amount of power consumption of each server in the active status and in the idle status. The energy usage of each node n can be calculated by $p^0 + U_n p^a$, where U_n is the utility of each server which can be measured by the percentage amount of allocated CPU and its capacity $U_n = \frac{\sum_{i\in S}\sum_{j\in N_i} x_{ijn} f_{ij}^c}{n^c}$.

We define a_n as the active variable for node n, where $a_n = 1$ if $\sum_{i\in S}\sum_{j\in N_i} x_{ijn} \geqq 1$ otherwise $a_n = 0$. The objective function now can be formulated as follows: $\min \sum_{n\in N} a_n U_n$.

3.2 Problem formulation

Combining the constraints (1)-(4), we formulate the optimization problem of service chain placement as follow:

$$P_{SCP}\colon\ \min \sum_{n\in N} a_n U_n,$$

s.t. constraints (1) - (4).

P_{SCP} aims to find a service chain placement to minimize the total cost incurred in the system, which is represented based on the number of active nodes. However, the problem above cannot be found in the polynomial time since it is NP-hard.

Among the many choices of optimization methods, the one we advocate below, Markov approximation-based approach has an advantage in giving an approximated solution for the high complexity of the combinatorial optimization problem.

4. Markov approximation approach for P_{SCP}

4.1 Log-sum approximation

Let $f = \{x, y\}$ be a configuration for P_{SCP}, and F be the set of feasible configuration defined by constraints (1)-(4). Configuration f indicates a specific VNF mapping scheme on a set of active node N. A change of any VNF in the allocation scheme will create a new configuration (or a new state in the Markov chain). For ease of representation, we let $C_f = C(x,y) = \sum_{n\in N} a_n U_n$ be the cost of P_{SCP} corresponding to the configuration f. Thus, we have $\min_{f\in F} C_f$, consequently, P_{SCP} can be rewritten as follows:

$$\min_{p>0} \sum_{f\in F} p_f C_f$$
$$s.t. \sum_{f\in F} p_f = 1,$$

where p_f is the probability of choosing configuration f. Following [10], we apply log-sum-exponential approximation to P_{SCP} as follows:

$$\min_{p>0} \sum_{f\in F} p_f C_f + (1/\sigma) \sum_{f\in F} p_f \log(p_f)$$
$$s.t. \sum_{f\in F} p_f = 1$$

where σ is a large positive constant.

This is the convex optimization problem that can obtain the optimal solution p^* [10] as follows

$$p_f^*(C_f) = \frac{\exp(-\sigma C_f)}{\sum_{f'\in F} \exp(\sigma C_{f'})},\ \forall f\in F. \quad (5)$$

Unfortunately, (5) is too difficult to find in practice due to the large solution space. Thus, based on the stationary distribution of Markov chain, we consider f as a state of time reversible of Markov chain. As the Markov chain converges to its stationary distribution, we approach p_f^* as an optimal solution.

4.2 Markov chain and transition rate

We next design the Markov chain and the transition rate between states for P_{SCP}. Consider two configurations $f, f'\in F$ that represent the states of the time-reversible ergodic Markov chain with the stationary distribution p_f^*. We define $q_{f,f'}$ and

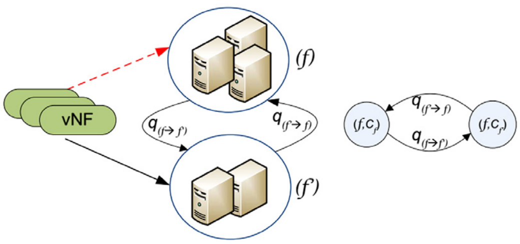


Fig. 3 Transition rate in Markov chain

$q_{f',f}$ as the non-negative transition rates from $f \rightarrow f'$ and $f' \rightarrow f$, respectively as shown in Fig. 3. Based on the balance equation for all states in Markov chain, we have

$$p_f^* q_{f,f'} = p_{f'}^* q_{f',f},$$
$$\exp(-\sigma C_f) q_{f,f'} = \exp(-\sigma C_{f'}) q_{f,f'}.$$

Consequently, we can derive

$$q_{f,f'} = \exp(-\delta)\frac{1}{1+\exp[-\sigma(C_f - C_{f'})]}, \quad (6)$$

$$q_{f',f} = \exp(-\delta)\frac{1}{1+\exp[-\sigma(C_{f'} - C_f)]}, \quad (7)$$

where δ is a constant.

The key idea in the Markov approximation method is to move randomly to a new configuration with a probability depending on the new cost of configuration f' that can approach to the optimal configuration.

5. Distributed service chain embedding algorithm

Consider the Markov chain and states, the system moves from configuration f to f' with only one change of an VNF. Thus, we can execute distributedly for each service chain, whenever a service chain change its configuration, others must be blocked. We schedule for all service chains following first-come-first-serve (FCFS and the next service chain has to wait the free message to execute its procedure).

The algorithm for service chain embedding based on the Markov approximation is named SEMA, which can obtain a close-optimal solution. The steps of SEMA are shown in Algorithm 1.

The SEMA algorithm solves P_{SCP} in a repeated manner with two phases. The first phase is to find a random feasible configuration to implement. Then, the second phase is to compare the current cost with the previous cost. A chosen configuration owns the smaller cost based on (6) and (7). These phases are repeated until underlying the Markov chain converges to the stationary distribution.

Algorithm 1: SEMA-Distributed service chain embedding.

For service chain i:

1. Randomly choose feasible nodes in the set N that have sufficient resources to host the service chain i.
2. Randomly replace one chosen node by selecting another node in the set of N. The new chosen subset must have sufficient resources to host the service chain i.
3. Migrate to the new configuration f' with the transition probability q according to (6) and (7).
4. Broadcast a "free message" to other sessions.
5. Return to Step 1 until the stopping criteria is met

Given a value of σ, we implicitly solve an approximation of P_{SCP} with an entropy term $\frac{1}{\sigma}\sum_{f \in F} p_f \log(p_f)$. The optimal gap is bounded by $\frac{1}{\sigma} log|F|$, where $|F|$ is the size of the configuration set F. Therefore, when increasing σ, the optimal gap will be reduced (5).

6. Simulation results

6.1 Settings

In this section, we make a simulation for our work with 20 service chains including 150 VNFs. We create 30 physical nodes and create a network connection between them. The link capacities of physical links are set from 5 to 10 and the virtual link bandwidth requirements are set from 1 to 2. For power consumption, we set $p^0 = 200\text{kWh}$ and $p^a = 400\text{kWh}$ for the physical nodes

6.2 Results

Convergence. We evaluate the convergence of SEMA during 1000 iterations. As shown in Fig. 4, based on the trace line, it shows the movement of all states following the Markov chain and the probability of transition rate. Fortunately, at iteration 32,

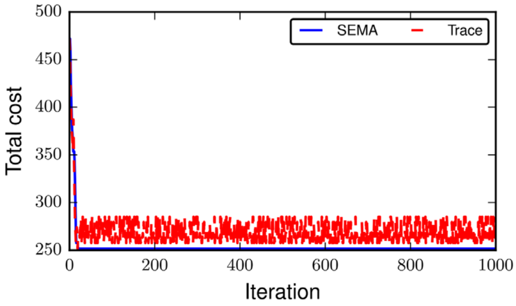


Fig. 4 Evaluation of the convergence of SEMA

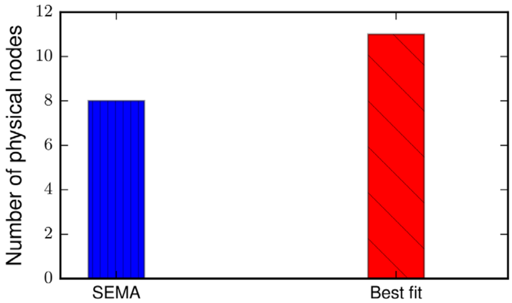
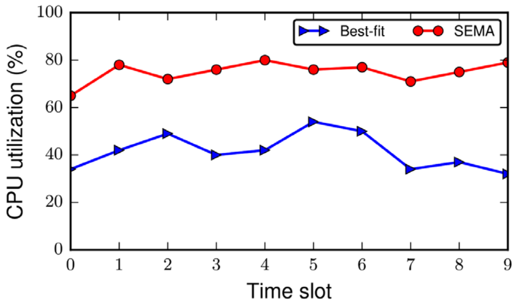
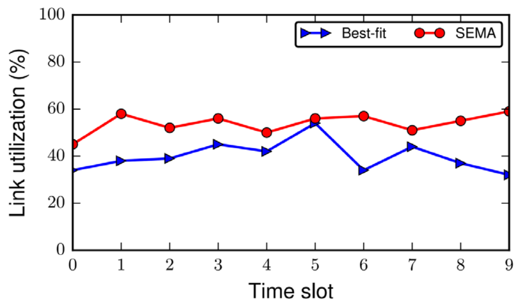


Fig. 5 Evaluating the number of active nodes



(a) Evaluation of the CPU utilization



(b) Evaluation of the link utilization

Fig. 6 Evaluation the resource utilization

the system selects an optimal solution and keeps it until ending. The total cost in our simulation is calculated by multiplying the total power of all active nodes with the electricity price (0.01 USD/kWh).

Resource utilization. Furthermore, we make a comparison to Best-fit approach as shown in Fig. 5. Without considering the network traffic between VNFs in service chains, Best-fit cannot reduce the number of active nodes. Best-fit always aims to increase the CPU utilization, however this way makes the physical links over the capacity.

Specifically, we make an evaluation that measures the resource utilization of CPU and physical links, as shown in Fig. 6. The utilizations of CPU and physical links in SEMA are higher and more stable than Best-fit during 10 time slots. By controlling both CPU and link bandwidth resources, SEMA can find suitable subsets which do not scale up and scale down too much. The link bandwidth allocation does not over as in Best-fit that keeps the system operate in stability. In average, the CPU utilization increases up to 73% and link utilization up to 53%.

7. Conclusion

In this paper, we have studied about service chain embedding problem. This problem is combinatorial and NP-hard problem that cannot be found a solution in polynomial time. Furthermore, due to the diversity and the relationship of NFs, current allocation approaches (First-fit, Best-fit) cannot achieve the efficient solution. We have proposed a distributed service chain embedding based on the Markov approximation method. Our solution can obtain the close optimal solution and outperforms compared to Best-fit, where the CPU utilization increases up to 73% and link utilization up to 53% in average.

References

[1] ETSI ISG NFV (Operator Group), "Network functions virtualisation Network operator perspectives on industry progress," White Paper, Oct. 2013.

[2] P. Quinn and T. Nadeau, "Service Function Chaining Problem Statement," Active Internet-Draft, IETF Secretariat, Apr. 2014.

[3] H. Jamjoom, D. Williams, and U. Sharma, "Don't Call Them Middleboxes, Call Them Middlepipes," *Proc. of the Third Workshop on Hot Topics in Software Defined Networking*, ser. HotSDN '14. New York, NY, USA: ACM, pp. 19-24, 2014.

[4] R. Mijumbi et al., "Network function virtualization: State-of-the-art and research challenges," *IEEE Commun. Surveys Tuts.*, Vol. 18, No. 1, pp. 236-262, 1st Quart. 2016.
[5] S. Clayman, E. Maini, A. Galis, A. Manzalini, and N. Mazzocca, "The dynamic placement of virtual network functions," *Proc. IEEE NOMS, Kraków*, Poland, pp. 1-9, May 2014.
[6] M. Xia, M. Shirazipour, Y. Zhang, H. Green, and A. Takacs, "Network function placement for NFV chaining in packet/optical datacenters," *J. Lightw. Technol.*, Vol. 33, No. 8, pp. 1565-1570, Apr. 15, 2015.
[7] Chuan Pham, Nguyen H. Tran, Cuong T. Do, Eui-Nam Huh, and Choong Seon Hong, "Joint Consolidation and Service-Aware Load Balancing for Datacenters," *IEEE Communications Letters*, Vol. 20, No. 2, pp. 292-295, Feb. 2016.
[8] Chuan Pham, Nguyen H. Tran, Minh N.H. Nguyen, Shaolei Reny, Walid Saad, Choong Seon Hong, "Hosting Virtual Machines on a Cloud Datacenter: A Matching Theoretic Approach," *IEEE/IFIP Network Operations and Management Symposium (NOMS 2016)*, Istanbul, Turkey, Apr. 25-29, 2016.
[9] Reingold, Edward M., Jurg Nievergelt, and Narsingh Deo, "Combinatorial algorithms," (1977).
[10] Chen, Minghua, et al., "Markov approximation for combinatorial network optimization," *IEEE Transactions on Information Theory 59.10 (2013)*, pp. 6301-6327.

Pham Chuan received the BS degree from HCM City University of Transport and MS degree from University of Science HCM City in 2004 and 2008, respectively. After graduation, he has been a lecturer in the Department of Computer Science, HCM City University of Transport. In 2013, he was awarded scholarship for his graduate study at Kyung Hee University, where he is currently working toward Ph.D. degree in Computer Science and Engineering

Minh N. H. Nguyen received the BE degree in Computer Science and Engineering from Ho Chi Minh City University of Technology in 2013. After graduation, he joined The University of Danang where he worked as a lecturer in Information Technology. In 2015, he was awarded scholarship for his graduate study at Kyung Hee University, where he is currently working toward Ph.D. degree in computer engineering

홍 충 선
정보과학회논문지
제 44 권 제 1 호 참조