ATLANTIS PRESS

Advances in Engineering Research, volume 149

2nd International Conference on Mechanical, Electronic, Control and Automation Engineering (MECAE 2018)

A Dynamic Shortest Path Deployment of Virtual Network Function

Xiaolei Wang [a], Shiqing Sun [b] and Hongbo Tang [c]

National Digital Switching System Engineering & Technological R &D Center, Zhengzhou 450002, China;

[a]18638673901@163.com, [b]sunshiqingq@163.com, [c]1480236389@qq.com

Keywords: 5G, network function virtualization, vEPC, service function chaining, Q-learning.

Abstract. In the 5G mobile communication network virtualization scenario, how to deploy service function chaining of the core network efficiently is the key problem to realize the efficient deployment of virtual Evolved Packet Core network services. In order to solve the problem that the existing deployment methods are difficult to meet the requirement of the mobile communication with low latency, this paper proposed a method for service function chaining deployment based on Q-learning. This method solved the problem by applying establish a Markov decision process model to the latency optimization in the context of VNF deployment, and then design a Q-learning algorithm to found the deployment solutions with minimum delay cost of network services. Simulation results show that the proposed method achieves better performances in terms of average processing time, request acceptance rate, gain and execution time.

1. Introduction

As the era of 5G is coming, mobile service is growing significantly diversified and the service which used to be provided only by fixed internet has emerged on the mobile termination. With the popularity of cloud service, new type of service requires a ‘soft’, ‘green’ and ‘high speed’ infrastructure platform. To meet these demands of future network, mobile operators entail numerous high volume dedicated hardware devices to improve network capability and performance, which could lead to continuous increase in capital expenditure and operating expenditure. In order to optimizing the architecture of mobile network, evolved packet core (EPC) involves network function virtualization (NFV). NFV can provide a centralized management and orchestration plane, and thus improve the flexibility and scalability of network [1]. In [2], Akyildiz I, et. al. proposed that introducing NFV in EPC can save 20-35% of the capital expenditure.

NFV is proposed to decouple the function of network entities and hardware by moving the software-oriented network function on commercial data center. Through instantiating a series of virtual network function (VNF) as a service function chain (SFC), operators finally obtain a virtual EPC (vEPC) on data center, which can implement centralized deployment, orchestration and management of VNF and physical resource. Compared to the fixed and static deployment of dedicated hardware in current mobile network, NFV platform provision a software-oriented deployment, which render the operators to customize SFC according to the personalized demands of tenants.

To support the diverse online interaction application, 5G standard requires a millisecond E2E latency. However, the latency of current 4G LTE system can only reach 10-100ms. To solve this problem, this paper is focus on how to utilize the flexibility of NFV platform to meet the low latency requirement in network service deployment.

Most existing works relegate the problem of low latency deployment to a shortest path problem. The main solution can be classified to two types: (1) approaches resort to integrate programming for global optimal solution; (2) approaches resort to heuristic algorithm for local optimal solution. In terms of global method, [3] and [4] proposed a model via linear and non-linear programming respectively. These method set the embedding relationship between VNFs and physical servers as the decision variable and introduce binary placement constraints to simplify the problem in order to improve the computing efficiency and obtain the optimal deployment policy with lowest latency. However, this kind of deployment can only describe the embedding relationship by integrate variable, which means that the granularity of deployment still remains on ‘network element’ level. These

approaches of rigid embedding relationship can not scale resource allocation according to fluctuation of network traffic. In the cases of large scale network, they may fall in negative performance. To solve this problem, [5] proposes a deployment based on node-splitting method. This method selects unit data packet (UDP) in the network as a decision variable to further refine the granularity of resource allocation and improve the existing integer programming method request acceptance rate. Considering the existing closed method comprehensively, this kind of method has higher solution accuracy and can guarantee the optimality of the deployment results, but its computational complexity is high. When the scale of the problem increases, it will lead to combination explosion. As a result, some researches on heuristic algorithms have emerged in the industry and the algorithm solves the problem of high computational complexity of closed-form methods. However, correspondingly, such methods can not guarantee the optimality of the results. [7] designs a heuristic algorithm based on simulated annealing, which can be used in a short time to find an approximate optional solution, but this method only considers one type of VNF. In [8], a greedy minimum load (GLL) algorithm and tabu search (TS) –based algorithm are designed. The former deploys the VNF first on the underlying node with the largest available cache resources and the latter keeps searching by TS to find the optimal solution which meets the conditions. A deployment method based on Viterbi algorithm is proposed in [9]. By modeling the problem as a multi-stage directed graph with joint overhead and then using the Viterbi algorithm for the deployment of SFC, the proposed algorithm tends to assign the VNFs requested by each SFC on the same underlying node, ignoring the impact of centralized deployment on the network processing latency and only optimizing the network's transmission latency. In addition, the methods mentioned above pay more attention to the resource utilization of the underlying infrastructure, not considering the features of the mobile communication service. They divide the SFC deployment into two phases, that is, the node deployment of VNF is completed first, and then the data flow is processed by a sequence constraint of VNF and makes link selection. However, this two-stages deployment strategy can not obtain the optimal solution of both nodes and links at the same time, but finally obtain the second-best solution.

Aiming at the shortcomings of the above SFC deployment methods, this paper designs a deployment method based on Q-learning with the background of the mobile core network vEPC, and models the SFC deployment problem as Markov decision progress to achieve a joint deployment of nodes and links in the first phase. In order to minimize the latency of network traffic, a dynamic adaptive SFC deployment method is propose in this paper. This method can better meet the demand of low latency network service in mobile communication network. Experiments show that this method has better performance in service request acceptance rate, revenue and algorithm execution time.

2. Problem Description and Network Module

5G network architecture will utilize SDN/NFV to realize EPC network element function based on centralized deployment, orchestration, and management of data center network. EPC network element function will be transplanted from costly exclusive hardware to unified high-performance commercial server to realize network element function decoupling from the proprietary hardware. VEPC implements the main NE functions of EPC's current architecture in the form of software, including MME, SGW, PGW and Home Subscriber Server (HSS), As shown in Fig.1, vEPC separates the control and data plane of the traditional core network and divides the EPC service chain into two types of services chains—control plain and data plain. Mobile operators can dynamically instantiate SFCs according to user needs, making deployment and orchestration more attractive and promoting the sharing of the underlying resources.

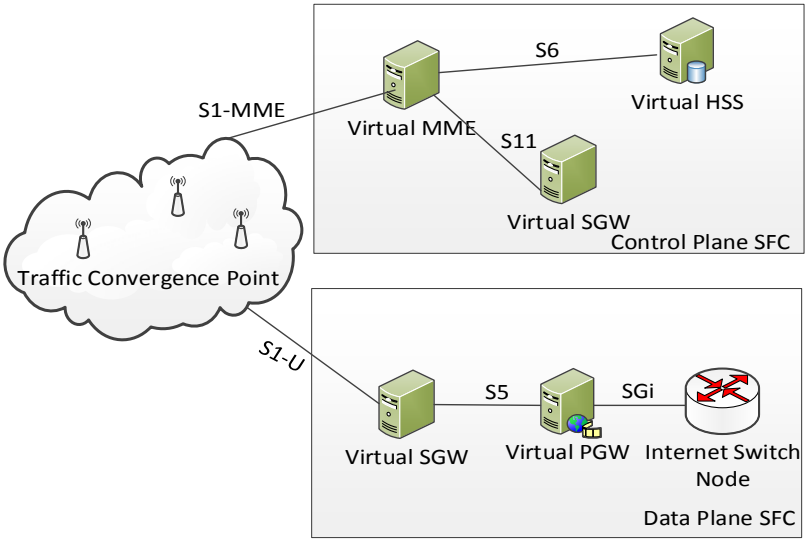

Fig.1 vEPC service chain model

The mathematical model of VNF service chain deployment is shown in Fig.2. The physical network is an operator data center, and the physical resource required for service provided by the commercial server is represented as aweighted undirected graph G_S. The virtual network which is represented as a weighted directed graph G_V is a service function chain consisting of a set of VNF sequences. The NFV management and orchestration module maps each VNF to the underlying network node according to the resource type and quantity requested by the service chain $f: G_V \rightarrow G_S$ [9].

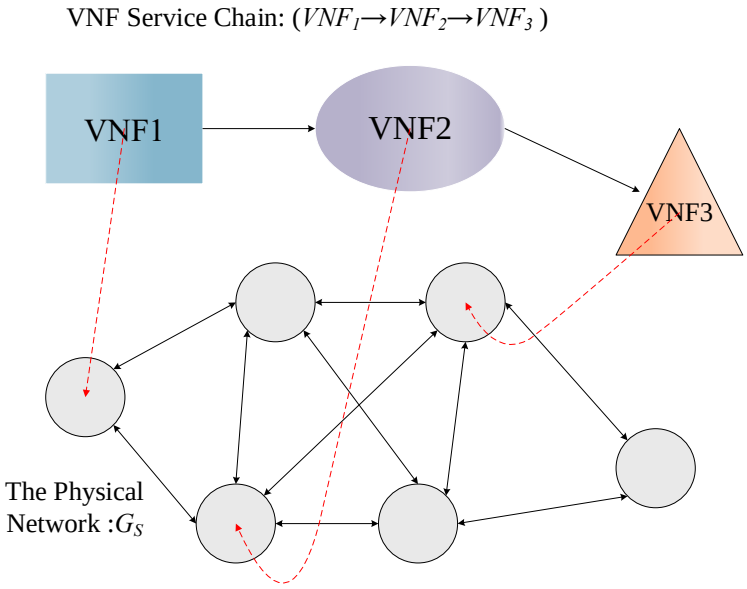

Fig.2 SFC deployment model

2.1 The Underlying Network

The underlying network which is consisted of physical nodes and links is represented as a weighted undirected graph $G^S = (N^S, E^S)$, where N^S is the set of physical nodes and the total number of physical nodes in the underlying network is n, k represents the type of physical resources (eg, compute, storage, bandwidth, etc.) provided in the underlying network. The underlying network resource matrix $C_{n\times k}$ presents the capacity of each type of resource for all physical nodes. E^S represents the link bandwidth of the physical nodes. The matrix $B_{n\times n}$ represents the link bandwidth of the physical nodes and the elements B_{ij} represents the minimum link bandwidth of the

communication loop between the physical nodes i and j The matrix $D^T_{n\times n}$ represents the transmission delay of the communication between the physical nodes.

2.2 Service Chain

We define an ordered set of VNF sequences as service chains. m represents the maximum number of VNFs that can be accommodated in a resource pool. l represents the total number of VNF types requested by the service chain. The numbers $1,2,\ldots,l$ represents different types of VNFs $\text{vMME},\text{vHSS},\ldots,\text{vSGW}$ respectively. The VNF composition vector $L_{1\times m}$ indicates the rank of each VNF in the tenant request chain. The resource matrix of the VNF service chain request is expressed as $R_{m\times k}$, where the row represents the resource vector required for deploying one VNF, and the column represents the VNF sequence requested by the service chain. $A_{m\times m}$ represents the service chain adjacency matrix, where the element A_{ij} represents the virtual link bandwidth between VNF_i and VNF_j. $D^P_{1\times l}$ represents the processing delay cost of each type of VNF. In [6], the experiment shows that the control plane and data plane of VNF are extracted and separated based on SDN, and the control plane is instantiated separately in the server. Therefore, the processing delay of the VNF is not affected by the data plane traffic and it is approximately a constant. Similarly, a weighted logical graph $G^V=(N^V,\ E^V)$ is used to represent the logical view of all the VNF nodes and their relationship in the service chain. N^V represents the logical node set of VNFs, and E^V represents the logical link set.

2.3 Service Chain Deployment Relationship

Define a deployment relationship matrix $X_{m\times n}$ that represents that the mapping between VNFs and service nodes, where the element $X_{ij}\in\{0,1\}$ is a binary variable that $X_{ij}=1$represents the i type of VNF deployed on the service node j. In this paper, the shortest path of strategy of service chain deployment is obtained by solving $X_{m\times n}$.

Table 1 Main parameters symbol defination

Parameter	Defination
m	The maximum number of VNFs requested by the tenant
n	The total number of physical nodes in the underlying network
k	The type of physical resource provided in the underlying network
l	The requested VNF type in the service chain
$C_{n\times k}$	The capacity of each type of resource for all physical nodes
$R_{m\times k}$	The number of different types of resourcesrequested by the VNF service chain
$B_{n\times n}$	Physical node adjacency relationship
$A_{m\times m}$	The delay of the physical link $e\in E$ between the physical network forwarding node s_1 and s_2
$D^T_{n\times n}$	Transmision delay of communication between physical nodes
$D^P_{1\times l}$	Processing delay of each type of VNF
$L_{1\times m}$	VNF composition of the service
$X_{m\times n}$	Service chain deployment relationship

2.4 Optimization Goals

The network model proposed in this paper aim to optimize the service delay of service chain deployment. Under the condition of resource constrains, we can transform the service chain deployment problem into an integer programming problem:

The target of optimization:

$$\min(Delay^{transmit} + Delay^{process}) \tag{1}$$

Constraint condition:

$$X_{m\times n} \cdot I_{n\times 1} = I_{m\times 1} \tag{2}$$

$$(X_{m\times n})^T \bullet R_{m\times k} \le C_{n\times k} \tag{3}$$

$$\sum_{(i,j)\in E^V} X_{i,s_1} \bullet X_{j,s_2} \bullet A_{i,j} \le B_{s_1,s_2}, \forall 1 \le s_1, s_2 \le n, s_1 \ne s_2 \tag{4}$$

$$X_{i,s_1} \in \{0,1\}, \forall 1 \le s_1 \le n, \forall 1 \le i \le m \tag{5}$$

Equation (1) is the objective function, whose purpose is to minimize the network service latency. Network service latency consists of processing delay and transmission delay. The processing delay $Delay^{process} = \sum_{i=1}^{m} D^P(\mathrm{L(i)})$ can be expressed as the sum of the processing delay of the VNFs in the whole network. The network transmission delay $Delay_{transmit} = \sum_{(i,j)\in E^V} X_{i,s_1} \cdot X_{j,s_2} \cdot D_{i,j}^T$, $\forall 1 \le s_1, s_2 \le n, s_1 \ne s_2$ means the sum of each physical link transmission delay. (2) is the deployment relationship constraint, where I is the a vector with elements values of all 1. The model proposed in this paper does not support VNF segmentation mapping, that is, it can not segment the VNF and map it to different underlying services nodes at the same time. (3) is the node resource capacity constraint, which means that all kinds of resources requested in the service chain can not exceed the corresponding resources provided by the underlying network node. (4) is a link resource constraint, indicating that the virtual link bandwidth between the VNFs does not exceed the minimum link bandwidth between service nodes deployed in the underlying network. (5) represents that the elements in the deployment relationship matrix $X_{m\times n}$ are binarized. In the condition of $X_{i,s_1} = 1$, VNF_i is deployed on the service node s_1, otherwise, the deployment of VNF_i is not on the node s_1.

3. SFC Deployment Method Based on Q-Learning

3.1 Markov Decision-Making Process Description

According to the paper [13], the deployment model of VNF service chain nodes can be described by discrete-time smooth Markov decision process, and it is expressed as quaternion $\{S, A, r, J\}$ [14], in which S represents the state space of the service chain mapping and the event in the space is the deployment status of the service chain at a certain moment. As it is shown in equation (6), $X(t)$ is the deployment status matrix at time t, and A represents the behavior space of the service chain deployment ,whose basic event is the change of service chain deployment status. For example, when any VNF in the service chain is instantiated or moved on the physical node and r indicates the revenue function, if the state s_t the moment t satisfies the constraints shown in equations (2)-(5), the deployment revenue in this state can be denoted as $r = -(Delay^{transmit} + Delay^{process})$, otherwise, the deployment benefitsapproaches to a negative real number whose absolute value is sufficiently large. J is the total discount return on service chain deployment and satisfies the equation $J = \sum_{n=0}^{T} r(t)$.

$$X(\mathrm{t}) \in S, X(\mathrm{t}) = \begin{pmatrix} X_{11} & \cdots & X_{1n} \\ & \ddots & \\ \vdots & X_{i,j} & \vdots \\ & & \ddots \\ X_{m1} & \cdots & X_{mn} \end{pmatrix}, X_{i,j}, i \in [1,m], j \in [1,n] \tag{6}$$

3.2 Q-Learning Algorithm Description

In order to solve the Markov decision process model proposed in the previous section, this section presents a solution based on the Q-learning algorithm. The Q-learning system module is shown in

Fig.3. At the t moment, the system performs the behavior a_t to the s_t state and then reach the s_{t+1} state, meanwhile it uploads the revenue function $r(s_t, a_t)$ of the t moment and updates the behavior value function $Q(s_t, a_t)$. Agent saves the updated behavior value function into the behavior estimation table $Q(s, a)$. Then the system repeats the above operation again for the state s_{t+1} until $Q(s_t, a_t)$ reaches the optimal behavior value $Q^*(s_t, a_t)$. After obtaining the optimal behavior estimation table $Q^*(s, a)$, agent will choose the optimal strategy π_Q^* based on J and the discount benefits of all the combinations of behaviors in the table.

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha_t[r(s_t, a_t) + \max Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \quad (7)$$

$$Q^*(s_t, a_t) = E[r(s_t, a_t) + \max Q^*(s_{t+1}, a_{t+1})] \quad (8)$$

The update rule of $Q(s_t, a_t)$ is shown in equation (7), where (s_t, a_t) represents the moment-behavior pair of the deployment process at the t moment, $r(s_t, a_t)$ indicates the instant benefits of the t moment, and α_t is the learning factor satisfying $\alpha_t \in [0,1]$. When the optimal behavior value function satisfies the Bellman's optimal value equation shown in equation (8), agent stops the iteration and $Q^*(s_t, a_t)$ represents the optimal behavior value at moment t.

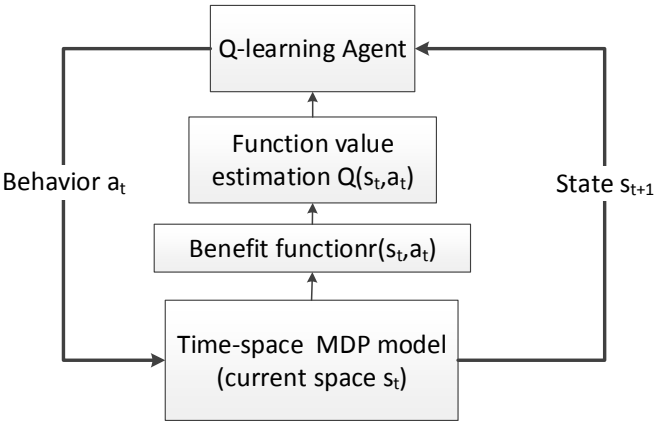

Fig.3 the reinforcement learning model

In the deployment process, it is assumed that the logical network and physical network view is shown in Fig.4. The logical layer in the figure indicates the VNF and the links needed to be mapped at the t moment. In the physical network, the nodes are distinguished by different numbers. The parameters $d_{ij}(t)$ indicates the instantaneous delay of the link (i, j), $i, j \in [1, n]$ and the processing delay of the nodes is indicated marked below the nodes. The bold arrows in the figure shows the behavior chosen by the agent at the moment t. As shown in the figure, VNF1 is mapped to node 2, VNF2 is mapped to node 1, and the virtual link between VNF1 and VNF2 is mapped to the physical link between node 1 and node 2. Because Q-learning uses the idea of dynamic planning, the system automatically updates the network view at the beginning of each iteration to obtain the immediate latency. Compared with the traditional static deployment method, the proposed algorithm can interact with the network view continuously during the iteration process. When problems such as node downtime and network congestion occur in the system, the method based on Q-learning can dynamically adjust the deployment strategy according to the current network environment.

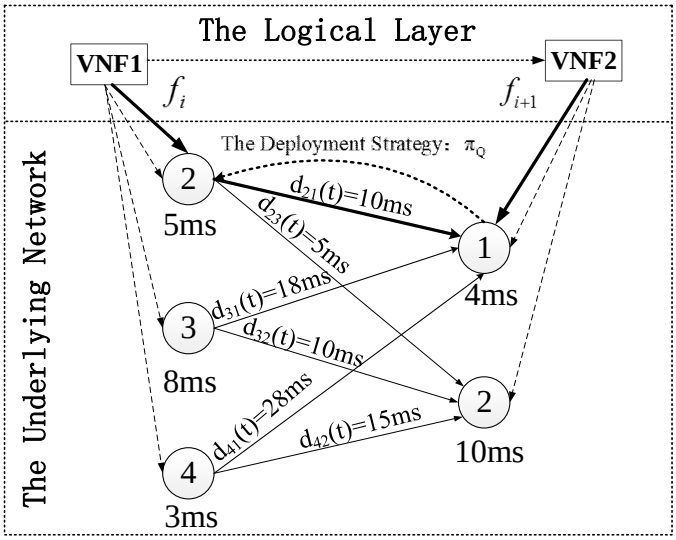

Fig.4 t moment network view

Table 2 VNF automated deployment algorithm based on Q-learning

Input :	The requested information of service chain l; the underlying network view G_S; The parameters of VNF
Output :	The delay-benefits optimal deployment strategy π_Q^* of service chain l
(1)	Initialize the Markov decision process state S, behavior space A and transition probability matrix P according to the input
(2)	Combine VNFs and get the order constraint φ_l of the service chain l according to the requested information of the service chain l
(3)	Initialize the selection strategy π_Q according to the order constraint φ_l
(4)	Initialize the learning factor α_0. Initialize the initial state s_0 randomly and let $t=0$ at the same time
(5)	Initialize the behavior value funtion estimation table $Q(s_t, a_t)$ randomly
(6)	Establish 0-1programming model
(7)	Initialize the instant benefitsfunction $r(s_t, a_t)$ according to 0-1 programming
(8)	while $Q(s_t, a_t) \neq Q^*(s_t, a_t)$
(9)	Decide the behavior a_t of the current state s_t at the moment t according to the strategy π_Q, and observe the state s_{t+1} of the next moment
(10)	Update $Q(s_t, a_t)$ of the current state-behavior pair according to equation (8)
(11)	Update the learning factor, and let $t=t+1$
(12)	end
(13)	Calcuate the discount benefits of the service deployment P and J
(14)	Output the optimal deployment sloution π_Q^*

4. Simulation and Performance Analysis

This section uses the total network delay, the requested acceptance and the execution time of the algorithm time as the performance evaluation indicators to fully evaluate the performance and complexity of the proposed algorithm, with the greedy GLL algorithm[8], TS-based algorithm[8], and Viterbi algorithm[9] for comparison.

4.1 Simulation Settings

In this experiment, the underlying network view and SFC requests are generated randomly by the discrete event simulator of the GT-ITM tool. The underlying network nodes and VNF parameters are

set according to Table 3 and Table 4. The arrival of an SFC request is subject to a Poisson distribution with arrival strength 0-1000. Each SFC request consists of one or more VNFs, with the number of VNFs subject to an evenly distributed range of [2, 5]. The algorithms are implemented by Matlab, on the PC with Intel i7 4790 and 4 GB memory.

Table 3 The parameter configuration of service network node

CPU	Storage capacity	Throughput
32	100GB	10Gbps

Table 4 VNFparameter configuration

VNF	CPU	I/O	Storage requirements
SGW	4	80	30
PGW	4	80	30
MME	6	40	50
HSS	2	100	200

4.2 Performance Analysis

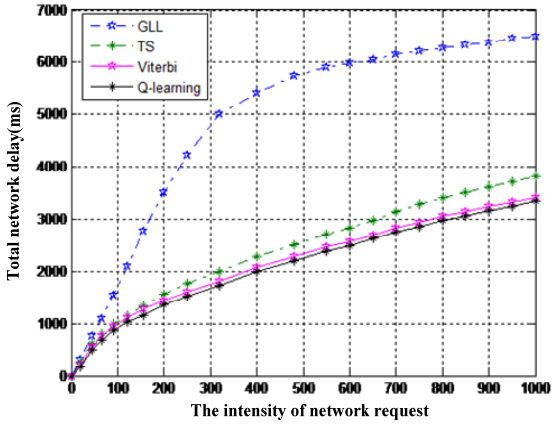

Fig.5 The average processing time of requests comparison

Fig.5 describes the total network delay with SFC request intensity changes. It can be seen from the figure that as the arrival strength of service requests increases, the total network delay increases. Under the same conditions, the proposed algorithm has the least increase in the total delay of the network. The node selection strategy of GLL is to select the available nodes with the most remaining resources. This method reduces the queuing time of service requests and reduces the processing delay, however, it does not consider the co-ordinated optimization of processing delay and transmission delay and can only obtain the minimum local optimal solution to processing delay. Similarly, the TS-based deployment method the solution space of the node selection problem by the local tabu search, avoiding the algorithm getting into the local optimum. The above two algorithms do not fully consider the state of the link which the data stream passes through, resulting in the long transmission delay of the underlying link. The Viterbi algorithm uses the dynamic programming idea to solve the sequential optimization problem and it adopts a static optimization algorithm when considering the link transmission delay optimization, without considering the randomness and uncertainty of the network delay parameter. At this point, the proposed Q-learning algorithm effectively co-ordinates the transmission delay and processing delay optimization, and considers the randomness of the experimental parameters. The experimental results show that this method has better performance under both large-scale and small scale requests.

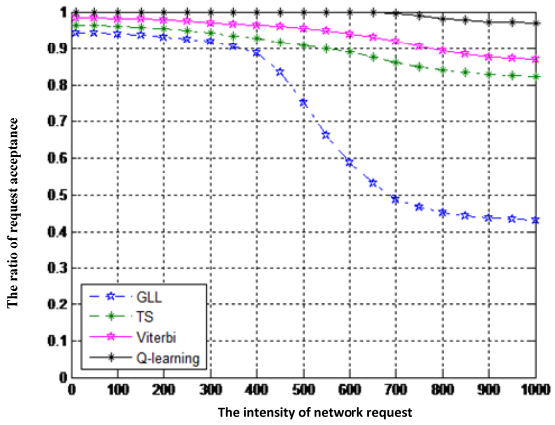

Fig.6 Request acceptance rate comparison

Fig.6 shows the change of request acceptance rate with the arrival strength of service requests. Due to the limited physical resources in the underlying network, as the strength of service request arrival increases, the system will not be able to provide sufficient resources. Therefore, the demand acceptance will decrease as the request intensity increases. The GLL and TS method have a limited choice of nodes with more remaining resources during deployment, and they will replace the node when the node usage is high. Each node in the network has a surplus of resources as the request intensity increases. Since this part of the resources can not carry a completed VNF, a large amount of resource fragments are formed, which results that the system is not able to fully utilize the physical resources. The Viterbi method does not consider the randomness of parameters such as delay, if a node is invalid and the network is congested, it can not dynamically adjust the deployment policy according to the instantaneous network traffic, leading to the failure of the requested deployment. We can see from the simulation results that the proposed method can fully utilize the physical resources with the consideration of network parameters' randomness to deal with the deployment of network requests in emergency situations.

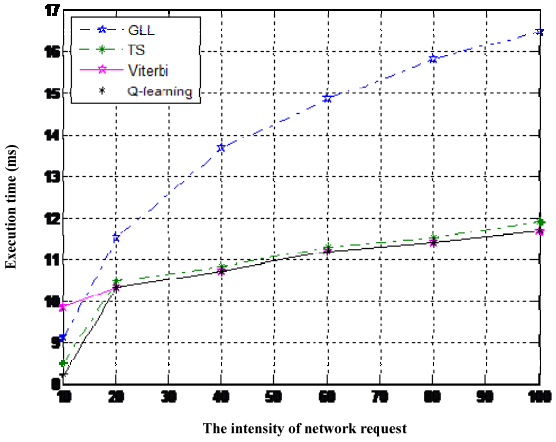

Fig.7 Algorithmexecution time comparison

The execution time of the algorithm depends on the size of the underlying network and SFC requests. As shown in Fig.7, the underlying network has 1000 nodes and the size of the SFC request varies from 10-100. With the longest processing time, the GLL method uses node-link two-stages mapping strategy: in the first stage, greedily select the optimal underlying node for VNF deployment so as to optimize the network processing delay; in the second stage, map the virtual link on the physical link with the least delay between nodes. This two-stages method can not deploy nodes and links at the same time. In the traversal process of two stages, each feasible solution needs to be judged whether it meets the conditions in the programming problem and the computational complexity is high. TS method based on greedy algorithm increases the search direction, but it can not change the flaw of two-stages algorithm. The Viterbi method adopts the idea of dynamic programming to search

for the shortest path of deployment recursively and its complexity of the algorithm is $O(n^2m)$, but there is no Agent to adjust the optimal-searching gradient in the process of recursive optimization and it belongs to non-directional search. In summary, the method based on Q-learning algorithm can obtain the immediate benefits according to the iterative equation (7) and greedily search for the deployment strategy in the most profitable direction, avoiding traversing the state space blindly, improving the efficiency of optimization and shortening the algorithm convergence time.

5. Summary

The introduction of SDN and NFV into the mobile core network enhances the scalability of the mobile communication network and enables the flexible deployment and on-demand generation of services. This paper mainly studies the problem of the shortest path of SFC deployment in virtualized environment. In view of the problem that the traditional SFC deployment method does not consider the dynamic optimization and delay co-ordination optimization, this paper proposes a vEPC service function chain deployment method based on Q-learning algorithm. Based on the traditional algorithm, the proposed method optimizes the processing delay and transmission delay of the deployment problem, and considers the randomness of the delay parameter, so it can dynamically adjust the deployment strategy to adapt to the impact of parameter changes. Experiments show that the deployment algorithm proposed in this paper has the computational complexity of polynomial time and significantly improves the total network delay, service request acceptance rate and algorithm execution time.

References

[1]. Mijumbi R, Serrat J, Gorricho J L, et al. Network function virtualization: state-of-the-art and research challenges. IEEE Communications Surveys & Tutorials, 2016, 18(1): 236-262.

[2]. Akyildiz I F, Nie S, Lin S C, et al. 5G roadmap: 10 key enabling technologies. Computer Networks, 2016, 106: 17-48.

[3]. Mohammadkhan A, Ghapani S, Liu G, et al. Virtual function placement and traffic steering in flexible and dynamic software defined networks.Proc of IEEE International Workshop on Local and Metropolitan Area Networks.Bolin 2015: 1-6.

[4]. Mehraghdam S, Keller M, Karl H. Specifying and placing chains of virtual network functions.Proc of the 3rd International Conference on Cloud Networking.chengdu.2014: 7-13.

[5]. Honbo TANG,Quan YUAN,Yu ZHAO,Xiaolei WANG.A Model for Virtualized Network Function Deployment Based on Node-splitting in vEPC,Journal of Electronics & Information Technology,2017,39(3):546-553.

[6]. Addis B, Belabed D, Bouet M, et al. Virtual network functions placement and routing optimization. Proc of the 4th International Conference on Cloud Networking.guangzhou.2015: 171-177.

[7]. Li X, Qian C. The virtual network function placement problem. Proc of IEEE Conference on Computer Communications.London.2015: 69-70.

[8]. Mijumbi R, Serrat J, Gorricho J L, et al. Design and evaluation of algorithms for mapping and scheduling of virtual network functions.NewYork.Proc of the 1st IEEE Conference on Network Softwarization.2015: 1-9.

[9]. Bari F, Chowdhury S R, Ahmed R, et al. Orchestrating virtualized network functions. IEEE Trans on Network and Service Management, 2016, 13(4): 725-739.

[10]. Herrera J G, Botero J F. Resource allocation in NFV: A comprehensive survey. IEEE Trans on Network and Service Management, 2016, 13(3): 518-532.

[11]. Baumgartner A, Reddy V S, Bauschert T. Mobile core network virtualization: A model for combined virtual core network function placement and topology optimization.Proc of the 1st IEEE Conference on Network Softwarization.Tyko.2015: 1-9.

[12]. FISCHER A, BOTERO J F, TILL BECK M, et al. Virtual network embedding: A survey. IEEE Communications Surveys & Tutorials, 2013, 15(4): 1888-1906. doi: 10.1109/SURV.2013.013013.00155.

[13]. Quan YUAN,Hongbo TANG,Yu ZHAO,Xiaolei WANG.Deployment Methmod for vEPC Virtualized Network Function via Q-learning,Journal on Communications,2017,38(8):172-182.

[14]. SONG H, LIU C, LAWARRÉE J, et al. Optimal electricity supply bidding by Markov decision process. IEEE Transactions on Power Systems, 2000, 15(2): 618-624. doi: 10.1109/59.867150.