Computers, Materials & Continua CMC, vol.61, no.3, pp.1345-1364, 2019

QoS-Aware and Resource-Efficient Dynamic Slicing Mechanism for Internet of Things

Wenchen He[1, *], Shaoyong Guo[1], Yun Liang[2], Rui Ma[3], Xuesong Qiu[1] and Lei Shi[4]

Abstract: With the popularization of terminal devices and services in Internet of things (IoT), it will be a challenge to design a network resource allocation method meeting various QoS requirements and effectively using substrate resources. In this paper, a dynamic network slicing mechanism including virtual network (VN) mapping and VN reconfiguration is proposed to provide network slices for services. Firstly, a service priority model is defined to create queue for resource allocation. Then a slice including Virtual Network Function (VNF) placement and routing with optimal cost is generated by VN mapping. Next, considering temporal variations of service resource requirements, the size of network slice is adjusted dynamically to guarantee resource utilization in VN reconfiguration. Additionally, load balancing factors are designed to make traffic balanced. Simulation results show that dynamic slicing mechanism not only saves 22% and 31% cost than static slicing mechanism with extending shortest path (SS_ESP) and dynamic slicing mechanism with embedding single path (DS_ESP), but also maintains high service acceptance rate.

Keywords: Dynamic slicing, internet of things, load balancing, priority, QoS, resource allocation.

1 Introduction

In recent years, as technical progress of cloud computing and network virtualization moves far head, Internet of Things (IoT) has integrated into multiple domains (e.g., industry, power and transportation) and a large number of terminal devices and services is also slated to these scenarios. On the other hand, the QoS and resource requirements of services vary greatly in different scenarios [Chernyshev, Baig, Bello et al. (2017)]. Some of them even have extremely high priority and need to be done without considering QoS and resource requirements [Yousaf, Bredel, Schaller et al. (2017); Rossem, Tavernier, Sonkoly et al. (2015); Rossem, Peuster, Conceicao et al. (2017); David and Lin (2018)].

[1] State Key Laboratory of Networking and Switching Technology Beijing University of Posts and Telecommunications, Beijing, 100876, China.

[2] Global Energy Interconnection Research Institute Co., Ltd, Beijing, China.

[3] State Grid Information & Telecommunication Co., Ltd., Beijing, China.

[4] Carlow Institute of Technology, Ireland.

[*] Corresponding Author: Wenchen He. Email: wche@bupt.edu.cn.

However, traditionally tightly coupled network architecture of hardware and software makes it difficult to allocate network resources flexibly to different services. So improving resource utilization and guaranteeing QoS for different IoT service are still open issues.

The fifth generation (5G) technology has become the key to solving above problem, especially the network slicing. Network slices are defined as isolated, end-to-end logical networks running on a common underlying network. This standalone network can be flexibly created according to service requirements and has independent control and management. Network Function Virtualization (NFV) and Software Defined Networking (SDN) are used as enablers [Kuo, Shen, Kang et al. (2017); Baumgartner, Reddy and Bauschert (2015)]. Thus, different application can run on the independent and isolated virtual networks (VNs) by being allocated an appropriate resource [Gu, Chen, Jin et al. (2018)]. Furthermore, multiple virtual networks can share one physical network by network slicing, which greatly improves resource utilization.

Services are generated dynamically in actual IoT system and generally have temporal variations of resource requirements [Raza, Fiorani, Rostami et al. (2018)]. The fixed network obtained by static slicing will lead to a portion of idle resources when resource requirement of service decreases, which greatly reduces resource utilization. On the contrary, dynamic slicing can adjust the size of slice according to time-varying resource requirements of service, and then redistribute this part of resources to other services, which achieves reuse of idle resources. In addition, dynamic slicing can guarantee the protected services by reducing the slice size of non-protected services when resources are insufficient to support all services. Authors propose a network slicing framework for end-to-end QoS provisioning with differentiated resource types [Ye, Li, Qu et al. (2018); Benkacem, Taleb, Bagaa et al. (2018)]. Authors devise mechanisms for allocating a set of Virtual Network Function (VNF) for each slice to meet its performance requirements and minimize cost. However, above slicing mechanisms focus on static slicing but do not consider dynamic slicing. So, it is urgent to design a dynamic slicing mechanism meeting service requirements and improving resource utilization.

In summary, this paper proposes a network slicing optimization model to provide specific VNs consisting of VNFs and routing with the optimal cost. Then a dynamic slicing mechanism including VN mapping and VN reconfiguration (DS-MR) is designed to solve above problem. The main technical contributions are summarized below:

- A VN mapping mechanism is proposed to provide IoT services with complete slices containing VNF placement and routing in a cost-optimized way. The slice is obtained by extending the reachable end-to-end paths with optimal cost to the final cost-optimal path by improved depth-first search algorithm (DFS).
- A VN reconfiguration mechanism is proposed to match the time-varying requirements of resource and further improve resource utilization. The size of slices will be scale down/up by VN reconfiguration when resource requirement of current service increases/decreases. Meanwhile, the low-priority services may be scaled down the size of slices when resources are insufficient so that high-priority can be allocated enough slices preferentially.
- Load balancing factors are designed to maintain the traffic balancing of network by

selecting nodes and links with more resource. Besides, the delay is designed as a pruning factor to be added to the depth-first search algorithm.

The rest of this paper is organized as follows. Section 2 gives a brief review of related work. Section 3 presents system architecture and describes network slicing models. Optimization problems for resource allocation are also derived and presented in this section. Section 4 proposes dynamic slicing mechanism to solve above problems. Simulation results and analyses are demonstrated in Section 5, and some conclusions and future work are drawn in Section 6.

2 Related work

A significant amount of researches have been done recently about network virtualization and network slicing. Authors provide a comprehensive overview of network slicing solutions proposed by the research community and present a survey covering solutions for all network domains as well the management of network slices [Kaloxylos (2018)]. Authors incorporate capabilities of SDN into NFV architecture, and combine SDN and NFV technologies to address the realization of network slices [Ordonezet, Ameigeiras, Lopez et al. (2017)]. The focus of network slicing in core network is service orchestration and management. In term of QoS guarantee, authors devise an algorithm that derives the optimal number of virtual instances of 5G core network elements to sustain the QoS and meet the requirements of a specific mobile traffic [Bagaa, Taleb, Laghrissi et al. (2018)]. Authors focus on the QoS parameters (e.g., minimum guaranteed data rate, maximum end to end latency, port availability and packet loss) and present two QoS-aware placement strategies to support service differentiation between the users [Vizarreta, Condoluci, Machuca et al. (2017)]. Authors study the problem of VNF placement with replications, and especially the potential of VNFs replications to help load balance the network [Carpio, Jukan and Pries (2017)]. Authors use replications of VNFs to reduce migrations in DC networks, and then propose a Linear Programming (LP) model to balance the server allocation strategies and QoS [Carpio, Jukan and Pries (2017)]. These works mainly focus on the loading balancing in servers but neglect the importance of load balancing in links and dynamic slicing. So the load balancing in servers and links and dynamic slicing should be considered simultaneously.

In summary, this paper presents the VN mapping mechanism to provide slices with optimal cost and the VN reconfiguration mechanism to improve resource utilization by adjusting the size of slices. Additionally, a service priority model on the basic of QoS requirements and service attributes is defined to guarantee high-priority services.

3 System and network model

3.1 System model

Based on SDN/NFV, this paper proposes a network slicing architecture. The specific architecture is introduced by Fig. 1. The top layer is slice layer, which provides end-to-end channel slices for services. The middle layer is control layer and mainly includes SDN controller, NFV orchestrator, etc. It virtualizes and manages substrate resources, and provides resources to different slices. The bottom layer is infrastructure layer and provides core network resources (bandwidth, storage and computing capabilities, and

other physical resources). IoT devices communicate with the access point (AP) via Lora, Bluetooth, etc., and then the AP connects to the BS. IoT communication devices mainly include wearable devices, smart phone, and a plurality of sensors.

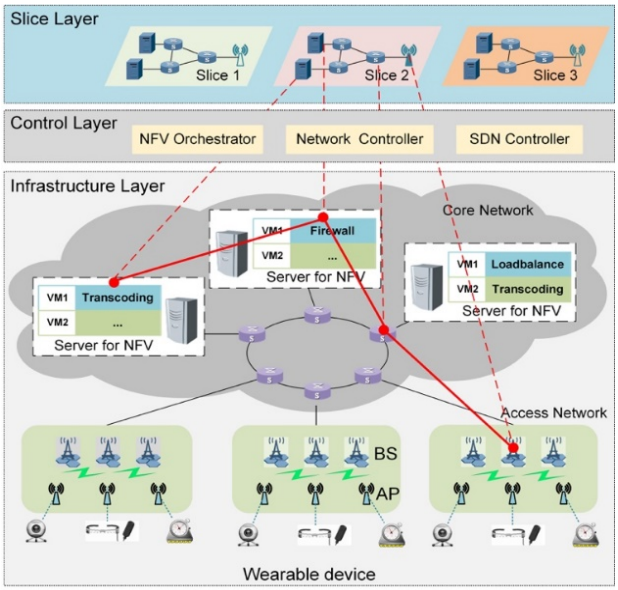

Figure 1: Network slicing model

3.2 Network model

The core network is represented by weighted undirected graph $G=(V,L)$, where V and L denote nodes and links. This paper classifies nodes into two types: 1) switch nodes forwarding traffic; 2) server nodes hosting virtual machines. The number of server nodes is represented by N. Virtual machine is used to install network functions. Capability of server for carrying VNFs is $Cap(n_i)$, which represents resources such as CPU computing capacity, storage, and other physical resources. Physical link l_{ij} connecting nodes n_i and n_j has bandwidth b_{ij} and transmission delay d_{ij}. Each service is completed in core network by a series of VNFs. $V=\{v_1,v_2,\cdots,v_k\}$ represents a set of VNFs and K represents the number of VNFs. $SC=\{sc_1,sc_2,\cdots,sc_M\}$ represents a set of service chains, and M represents the number of service chains. $N(sc_i)$ represents the number of VNFs in service chain sc_i. Server nodes consume $Cap(v_i)$ to host VNF v_i. VNF v_i has processing delay d_i. Similarly, virtual link l_{uv}^{v} between VNFs mapped to physical link l_{ij} needs to consume link bandwidth.

3.3 Service model

A tetrad $\left\{sc_i, Cap^i_{req}, B^i_{req}, D^i_{req}\right\}$ is used to represent a VN request s_i, where s_i represents the type of service chain required by s_i, and Cap^i_{req}, D^i_{req}, B^i_{req} represent the minimum server resource, end-to-end delay and bandwidth requirements respectively. With dynamic slicing, the resource requirement of the service is time-varying, and its variation is related to the type of service.

A service priority model is defined to determine the queue of resource allocation during mapping and reconfiguration of VNs. The service with high *priority* will be placed in front of the queue so that it can be completed preferentially. The specific model is defined as follows.

$$priority\left(s_i\right) = e\left(s_i\right) + QoS^*\left(s_i\right) \tag{1}$$

where $e\left(s_i\right)$ is a binary variable, and $e\left(s_i\right) = 1$ indicates that s_i belongs to protected service. It guarantees that the $priority\left(s_i\right)$ of protected service is higher than that of non-protected service. The service is non-protected in the case of $e\left(s_i\right) = 0$, and $priority\left(s_i\right)$ is related to the product of normalized QoS value. The normalized QoS value $QoS^*\left(s_i\right)$ is given by

$$QoS^*\left(s_i\right) = Q^* \cdot \left(Q_{cap} \cdot Cap^i_{req} + Q_{bw} \cdot B^i_{req} + Q_{delqy} \cdot D^i_{req}\right) \tag{2}$$

where Q^* is normalized factor and Q_{cap}, Q_{bw}, Q_{delqy} represent weights of server capability, link bandwidth and service delay. The weights are relevant to types of services. For example, Q_{delqy} consuming much resource.

3.4 Dynamic network slicing model

This paper defines two binary variables to describe service chain orchestration.

- $x_{i,j} : x_{i,j} = 1$ indicates that VNF v_j is mapped to physical node n_i;
- $y_{ij,uv} : y_{ij,uv} = 1$ indicates that virtual link l^v_{uv} is mapped to physical link l_{ij};

3.4.1 Network slicing cost

Installation of VNFs consumes CPU computing, storage in servers. The unit price of installed VNF is denoted by c_1. Additionally, effective measures are taken to avoid the already congested path or node under the condition of satisfying delay requirement, which can well maintain the traffic balance. Therefore, a load balancing factor Φ_i to indicate load status of nodes, and its value is inversely proportional to the remaining resource of nodes. The factor is given by

$$\Phi_i = \frac{\alpha_1}{Cap\left(n_i\right)_{remain} + \beta_1} + \gamma_1 = \frac{\alpha_1}{Cap\left(n_i\right) - \sum_{j \in K} x_{i,j} \cdot cap\left(v_j\right) + \beta_1} + \gamma_1, \forall j \tag{3}$$

where α_1, β_1, γ_1 is a set of adjustment factors. So related cost of installing VNFs is given by

$$cost(VNF) = c_1 \sum_{i \in N} \sum_{j \in K} x_{i,j} \Phi_i = c_1 \sum_{i \in N} \sum_{j \in K} x_{i,j} \left(\frac{\alpha_1}{Cap(n_i)_{remain} + \beta_1} + \gamma_1 \right), \forall i, j \tag{4}$$

We consider bandwidth cost when service chain orchestration in the core network. c_2 indicates unit price of and bandwidth. Similarly, a load balancing factor Θ_{ij} to represent the load status of links, and its value is also inversely proportional to the remaining resource of links. This factor is given by

$$\Theta_{ij} = \frac{\alpha_2}{b_{remain}^{ij} + \beta_2} + \gamma_2 = \frac{\alpha_2}{b_{ij} - \sum_{i,j \in N} \sum_{u,v \in K} B_{req}^{ij} y_{ij,uv} + \beta_2} + \gamma_2, \forall u, v \tag{5}$$

where α_2, β_2, γ_2 is a set of adjustment factors. So the cost of bandwidth consumption is given by

$$cost(bandwidth) = c_2 \sum_{i,j \in N} \sum_{u,v \in K} y_{i,j,uv} B_{req}^{ij} \Theta_{ij} = c_2 \sum_{i,j \in N} \sum_{u,v \in K} y_{i,j,uv} B_{req}^{ij} \frac{\alpha_2}{b_{remain}^{ij} + \beta_2} + \gamma_2 \tag{6}$$

It can be seen from (4) and (6) that links or nodes with larger remaining resource have relatively lower cost. Therefore, links or nodes of this type are more likely to be selected. Through Φ_i and Θ_{ij}, our routing scheme helps to balance the traffic load of network, and then improve network resource utilization.

Therefore, the total cost in the process of dynamic slicing including VNF instances and bandwidth is given by

$$cost(DS\_MR) = cost(VNF) + cost(bandwidth) \tag{7}$$

3.4.2 Node and link constraints

The server has finite resources including computing, storage, etc., and cannot continue to host VNFs when resources are occupied. So, the node must have sufficient remaining resources to install virtual machines for VNF instantiation.

$$C_1 : \sum_{j \in K} x_{i,j} cap(v_j) \le Cap(n_i), \forall j \tag{8}$$

Similarly, links in core network have finite bandwidth. Therefore, virtual links of service can only be mapped to physical links with sufficient bandwidth resource.

$$C_2 : \sum_{u,v \in K} y_{ij,uv} B_{req} \le b_{ij}, \forall i, j \tag{9}$$

The flow is assumed as indivisible so that the VNF and virtual link can only be mapped on one server node and one physical link.

$$C_3 : \begin{cases} \sum_{i \in N} x_{i,j} = 1, \forall j \\ \sum_{i,j \in N} y_{ij,uv} = 1, \forall u, v \end{cases} \tag{10}$$

The service prefers the same type of VNF that is installed, which will reduce orchestration cost. However, the number of services that a VNF can support is limited and depends on queue processing delay of services. Therefore, we set the service quantity constraint, and it is given by

$$C_4 : 1 \le num(v_i) \le num_0, \forall i \tag{11}$$

where $num(v_i)$ indicates the number of services already carried in v_i. In the process of routing, the sum of directional traffic of other nodes is 0 except source node and destination node.

$$C_5 : \sum_{i,j \in N} y_{ij,uv} - \sum_{i,j \in N} y_{ji,uv} = \begin{cases} 1, \ if \ z_{i,u} = 1 \\ -1, if \ z_{j,v} = 1 \\ 0, \ otherwise \end{cases} \tag{12}$$

3.4.3 QoS constraints

This paper considers services with requirements of minimum delay D_{req}. The end-to-end delay mainly includes processing delay of service node, the transmission delay on links in core network. The end-to-end delay is given by

$$D_{core} = D_{link} + D_{node} \tag{13}$$

In the process of orchestration, delay is one of pivotal reference factors. The routing scheme is valid only if its end-to-end delay meets requirement. Therefore, we should ensure that the routing scheme always meets delay requirements of service.

$$C_6 : D_{core} = D_{link} + D_{node} = \sum_{i \in K} \sum_{j \in N} x_{i,j} d_j + \sum_{u,v \in K} \sum_{i,j \in N} y_{ij,uv} d_{ij} \le D_{req} \tag{14}$$

3.4.4 Network slicing model

The slicing in core network involves VNF placement and traffic routing. Its objective is to minimize slicing cost, as well as meet above constraints. The optimization problem model is given by

$$\min \{cost(DS\_MR)\}$$
$$s.t. \begin{cases} C_1, C_2, C_3, C_4, C_5, C_6 \\ C_7 : x_{i,j}, y_{ij,uv} \in \{0,1\} \quad \forall i, j, u, v \end{cases} \tag{15}$$

4 Dynamic slicing mechanism description

Dynamic slicing mechanism based on time window model is introduced in this section. VN requests arriving within a time window will be processed in two mechanisms according to their types and priorities: VN mapping and VN reconfiguration. VN mapping provides the service with slice meeting its resource needs. VN reconfiguration includes dynamic adjustment of slice size, as well as processing solution when VNs compete for insufficient resources.

4.1 Time window model

The real-time character of services needs to be considered in the process of designing dynamic slicing algorithm, since VN requests are generated dynamically in actual service system. Therefore, a dynamic slicing mechanism based on time window model including VN online mapping and dynamic reconfiguration is proposed. During the time window, VN requests waiting for physical resource contain two types. *TypeI* represents newly arrived or unfinished VN in the previous time window. They need to be allocated a completed network slice by VN mapping; *TypeII* represents VN requests whose resource requirements change, and their size of slices needs to be adjusted by VN reconfiguration.

Dynamic slicing mechanism based on time window is shown in Fig. 2. There are three VN requests waiting to be mapped in current time window. VN1 and VN2 are a newly arrived request and an uncompleted request in previous time window respectively, and VN3 is a virtual request whose size of slice needs to be adjusted. In this time window, VN1 and VN3 are completed successively by being allocated enough physical resources as a result of their high priorities, but VN2 enters the next time window for mapping because no resources are available. Finally, VN3 still has not get resource and its mapping fails in the next time window since its waiting delay has exceeded delay requirement.

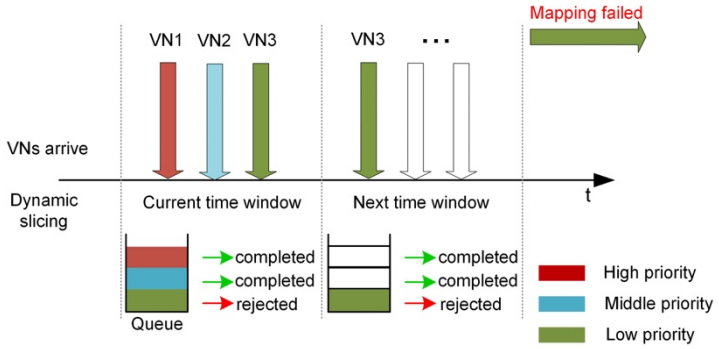

Figure 2: Dynamic slicing based on time window

4.2 VN mapping

VN mapping completes the service orchestration. VNFs in a service chain may be traversed by several distinct service flows in a certain sequence, so it becomes difficult to improve network resource utilization. Since network has created many VNF instances for previous services, so the newly arrived services have two options: use existing VNFs or install new VNFs. However, services may have to take long paths to reach existing VNFs and result in a high bandwidth consumption. On the other hand, installing new VNFs for services increases the network orchestration overhead and capital expenditure. So, a reasonable trade-off between aforementioned options can lead to optimal solution. Moreover, VNF placement and routing are two tightly coupled processes. The globally optimal is not equal to the optimal solution obtained by processing them separately. So this paper considers the two processes simultaneously and proposes an algorithm based on path extension.

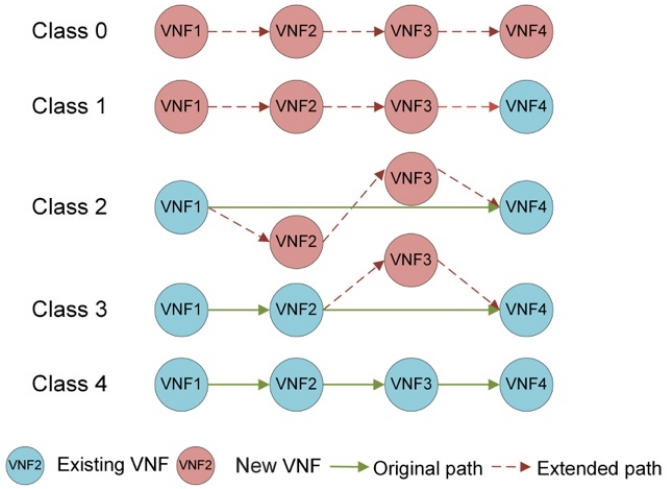

Figure 3: VNF placement and routing

The main idea of this paper is to obtain the reachable paths with the lowest cost and extend them to final orchestration path. As shown in Fig. 3, current service chain is assumed to require four network functions (VNF1, VNF2, VNF3, VNF4). Firstly, the end-to-end reachable paths that meet the delay requirements are obtained and divided into five classes according to the number of existing VNFs they have. So paths with insufficient VNFs need to instantiate new VNFs to meet service requirements. Next, the lowest cost path in each class is selected as the path to be extended. For extension, the servers available on current path is selected to instantiate the VNFs required for service. Otherwise, current path needs to be extended to the nearest available server which are not on this path to instantiate the VNFs. Finally, the path with the lowest cost after the extension will be selected as the final orchestration path.

4.3 VN reconfiguration

Fig. 4 is used to specify necessities for VN reconfiguration. Since services generally have temporal variations of resource requirements, the most ideal reconfiguration scheme is to detect changes in resource requirements in real time and redistribute resources to maximize resource utilization. However, this real-time detection and allocation bring huge calculations and costs, so granularity of resource variation is increased to simplify calculation as well as guarantee resource utilization. For example, some services may have peak resource requirements during the daytime and the others experience their peaks during the nighttime. The resource variation of VN1 and VN2 are as shown in Fig. 4.

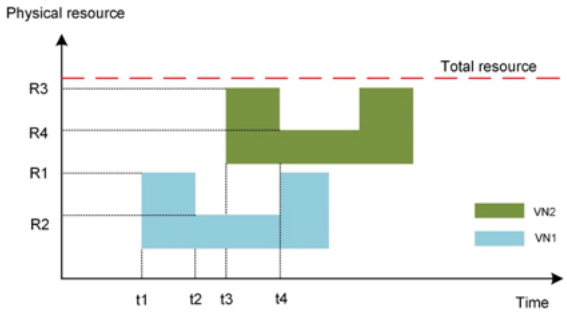

Figure 4: Dynamic adjustment of slice

According to the static slicing method, VN1 is allocated with the fixed physical resource of $R1$ (related to the peak resource requirement) when it arrives at $t1$. The resource requirement of VN1 will decrease when it reaches $t2$, but static slicing will not consider this part of idle resources $(R1-R2)$, which causes a great waste of resources.

$$\begin{aligned}\mu_{st} &= \frac{R_{req}}{R_{pro}} = \frac{R_2}{R_1} < 1 \\ \mu_{dy} &= \frac{R_{req}}{R_{pro}} = \frac{R_2}{R_2} = 1\end{aligned} \tag{16}$$

This paper makes a compromise between computational complexity and resource utilization. So the granularity of resource change is increased to simplify calculation as well as maintain resource utilization. As show in Fig.4, the resource requirement of VN1 decreases at $t3$. Dynamic slicing method monitors and matches changes of resource by releasing the idle resource. So, other VNs (e.g., VN2) can use the part of resource. The resource consumption of dynamic slicing method between $t2-t4$ is $R2$, while that of static slicing is $R1$. Through dynamic slicing, the physical resources can be better shared by different VNs. Resource utilization μ is further defined as the ratio of required resource R_{req} to provided resource R_{pro}. As shown in (16), resource utilization of dynamic slicing is obviously higher than that of static slicing.

$$\begin{aligned}\omega_{dynamic} &= \frac{Se_{Total} - Se_{Failed}}{Se_{Total}} = 100\% \\ \omega_{static} &= \frac{Se_{Total} - Se_{Failed}}{Se_{Total}} = 50\%\end{aligned} \tag{17}$$

Another scenario of VN reconfiguration is the VN competition mapping problem when overall network resources are insufficient. As shown in Fig. 4, VN2 with the higher priority than VN1 arrives at $t3$, and the control layer needs to allocate network resources for it. However, the total resources of network are insufficient to support two VNs at the same time. Therefore, the controller assigns corresponding network resources to VN2 firstly according to service priority. As shown in (17), the service acceptance rate of

dynamic slicing between $(t3-t4)$ is 100%, while that of static slicing is 50%. Therefore, VN reconfiguration in this scenario can well guarantee protected or delay-sensitive services and allocate network resources to meet their requirements of QoS and resource.

It might be happened that not all services are allocated enough resource when many VNs arrive or need to be scaled up. As a result, a part of services has different degrees of loss. Therefore, a service loss $Loss(s_i)$ is defined to indicate the degradation status of service. It is given by

$$Loss\left(s_i\right)=\frac{R_{req}-R_{get}}{R_{req}}=\frac{\int_t^{t+T_i}R_{req}\left(t\right)dt-\int_t^{t+T_i}R_{get}\left(t\right)dt}{\int_t^{t+T_i}R_{req}\left(t\right)dt} \tag{18}$$

where R_{req} and R_{get} represent resources service s_i needs and gets respectively. T_i represents the operation period of service s_i.

4.4 Dynamic slicing mechanism

Dynamic slicing mechanism in the VN mapping phase is to complete orchestration and routing of service chains with optimal cost.

For current time window, the specific steps are as follows.

1. The waiting time for VN requests in service system will be calculated. VN request s_i dissatisfies its delay requirement if $D_{wait}^{i}>D_{req}^{i}$, and then will be rejected and deleted. If not, it gets a chance to enter the queue.
2. Priorities of VN requests are calculated by service priority model so that the *Queue* is created. Then VN requests in *Queue* are classed into two types: *TypeI* and *TypeII*.
3. VN requests in *TypeI* will be completed by the VN mapping mechanism, and VN requests in *TypeII* will enter VN reconfiguration mechanism.

VN mapping mechanism allocates network slices to the former VNs based on their queue. The specific steps are as follows.

1. For current VN request s_i starting at n_s and ending at n_t, the set of reachable end-to-end paths $P_{rea}=\{p_1,p_2,...\}$ is obtained by the improved DFS algorithm. Paths in P_{rea} will be divided into $(N(sc_i)+1)$ classes on the basic of the number of existing VNFs they have. Then the path with the lowest cost in each class are selected to create an optional path set $P_{opt}=\{p_0,p_1,...,p_{N(sc_i)}\}$.
2. Next, these paths may have different number of existing VNFs to use. They need to be extended to have $N(sc_i)$ so that they can complete s_i. These $(N(sc_i)+1)$ paths in P_{opt} are extended on the basic of following specific method;

- If there is a server n_j available on this path, the required VNFs v_r will be instantiated on n_j;
- If there is no server available on this path, dynamic slicing mechanism will extend

this path to available servers n_k and instantiate required VNFs, while guaranteeing delay requirements.

3. Finally, the path with lowest cost after being expanded will be selected as final slice for current VN request. The pseudocode of VN mapping is shown in Algorithm 2.

Algorithm 1 Dynamic Slicing Mechanism including VN Mapping and Reconfiguration (DS_MR)

Input: G, $s_i = \left\{sc_i, Cap^i_{req}, B^i_{req}, D^i_{req}\right\}$

Output: slicing scheme; unfinished VN requests;

Initialize: parameters of network and VN requests;

for VN request s_i in current time window **do**

  if $D^i_{wait} > D^i_{req}$ **then**

    Reject and delete s_i from service system;

  end if

  Calculate $priority(s_i)$;

end for

Create the Queue of s_i based on $priority(s_i)$;

for s_i in *Queue* **do**

  if s_i in *TypeI* **then**

    Execute **VN mapping (** s_i **)**;

  else

    Execute **VN reconfiguration (** s_i **)**, adjust size of slices;

    if s_i has insufficient resource **then**

      Reject but not delete s_i;

    end if

    Update rejected VN requests;

    Scale down/up slice size of s_i;

  end if

end for

Put rejected VN requests to next time window;

Update status of physical network resource;

Return slicing schemes;

During the reconfiguration phase, the size of slice will be adjusted dynamically to match the resource requirement changes of VN.

1. If required server resource decreases from R_{t0}^{s} to R_{t1}^{s} and required bandwidth resource decreases from R_{t0}^{b} to R_{t1}^{b}, the part of resource that is not needed temporarily will be reinserted into resource pool by the control layer and then be reused by other services.
2. On the contrary, if required server resource increases from R_{t0}^{s} to R_{t1}^{s} and required bandwidth resource increases from R_{t0}^{b} to R_{t1}^{b}, the control layer will allocate more physical resources $R_{t1}^{s} - R_{t0}^{s}$ and $R_{t1}^{b} - R_{t0}^{b}$ from original path to s_i. The allocated resources cannot be from other paths since we assume that traffic is indivisible.
3. Network may still have insufficient resources to allocate when excessive VN requests arrive even if dynamic adjustment is made. However, the high-priority services must be provided with enough resource. Therefore, the services with higher $priority(s_i)$ will obtain resources they need firstly in the reconfiguration phase, while the services with lower $priority(s_i)$ may be sacrificed. The pseudocode of VN reconfiguration is shown in Algorithm 1.

Algorithm 2 VN Mapping Mechanism

Input: G, $s_i = \left\{sc_i, Cap_{req}^{i}, B_{req}^{i}, D_{req}^{i}\right\}$

Output: service chain orchestration scheme;

Initialize: parameters of network and VN requests;

Get $P_{rea} = \{p_1, p_2, ...\}$ by improved DFS;

Divide them into $(N(sc_i)+1)$ classes;

Select the path with the lowest cost in each class to create an optional path set $P_{opt} = \{p_0, p_1, ..., p_{N(sc_i)}\}$

for $p_i \in P_{opt}$ **do**

  for $n_j \in p_i$ **do**

    Instantiate required VNF v_r in n_j;

    Calculate $cost(VNF)$;

  end for

  Select scheme with lowest cost;

  if No server nodes available on p_i **then**

    for $n_j \notin p_i$ **do**

      Instantiate required VNF v_r in n_j;

      Extend p_i to n_j by Improved DFS;

Calculate $cost(VNF)+cost(bandwidth)$

end if

Select scheme with lowest cost;

end if

end for

if No orchestration scheme available for s_i **then**

Reject but not delete s_i;

end if

Return orchestration scheme for s_i;

In addition, the DFS algorithm has also been improved to reduce the complexity effectively. The delay is regarded as a judgement condition for pruning and further the binary value Ω is defined as the pruning factor. The definition of Ω is given by

$$\Omega=\left(\sum_{i\in K}\sum_{j\in N}z_{i,j}d_i+\sum_{u,v\in K}\sum_{i,j\in N}y_{ij,uv}d_{ij}\geq D_{req}\right) \tag{19}$$

$\Omega=1$ indicates that delay of path when reaching this node has exceeded delay requirement of service. So, whether or not to stop current search depends on the value of Ω. Specifically, when improved DFS is searching current node, it calculates delay of current routing scheme. If delay value exceeds the delay requirement, the current routing will be stopped, and return to the previous node to continue search until all paths are obtained. Finally, we calculate cost of all paths derived from improved DFS and select the path with the lowest cost. Some meaningless searches are deleted in advance by improved DFS, which greatly reduces the running time of the algorithm.

5 Evaluation analysis

Communication services in IoT are divided into following four categories according to differences in QoS and service attributes: control service; emergency service; video service; and voice service. The control service represents the internal service that guarantee the operation of network so that it has a high priority than other operation services. The emergency service is aimed at the emergency event. Although these services occur less frequently, the requirements of delay and reliability are high. The video and voice services belong to infotainment services, and they have different requirements of delay and resource.

Selecting two mechanisms including SS_ESP and DS_ESP as comparisons, this paper carries out a simulation experiment on three mechanisms.

1. Static slicing mechanism with extending shortest path (SS_ESP): It uses shortest path algorithm to obtain one cost-optimal path and extends it as the final orchestration path.
2. Dynamic slicing mechanism with embedding single path (DS_ESP): It also uses shortest path algorithm to obtain one cost-optimal path, and then supplements the

required VNFs but unavailable in shortest path.

5.1 Simulation setting

5.1.1 Service chain orchestration

A simulation network consisting of 100 nodes and 500 links is set up to represent core network by Java language. All substrate nodes are distributed randomly and the capability for server nodes is randomly set between 30-50. The link bandwidth is randomly set between 20-40. Each service chain required by VN requests contains 2-6 VNFs. The allowable embedding range of VN requests is 20, the connection rate between VNFs is 50%. The capability consumed by VNFs is uniformly distributed from 2 to 5, and the bandwidth of virtual link is uniformly distributed from 1 to 3. The width of a time window is set to 200 ms and 20 VN requests are generated in a time window. The lifetime of VN requests follows an exponential distribution with an average duration of 200 time units.

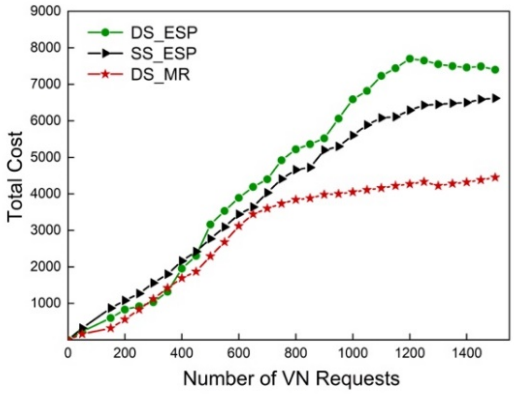

Figure 5: Total cost

Fig. 5 shows total cost of three slicing mechanisms. When the number of VNs is less than 400, cost of DS_ESP and DS_MR are slightly lower than SS_ESP, but they have not much different since network has sufficient resources to complete these mappings. Note that the cost of DS_ESP is slightly lower than DS_MR at 300-400, because it supplements the original path with VNFs, which has certain advantages when dealing with a small number of services. However, as the number of services increases, total cost of DS_ESP and SS_ESP are significantly higher than that of DS_MR. Because DS_MR uses the link extension method to achieve cost optimization when VN mapping, and adjusts the slice size in time when the service resource requirements change, which greatly improves its resource utilization.

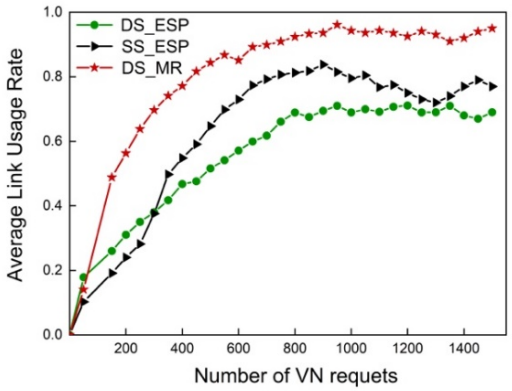

Figure 6: Average link usage rate

Fig. 6 shows average link usage rates of three mechanisms. It can be seen from Fig.6 that average link usage rates of the three mechanisms are basically equal when the number of services is less than 400, because the number of services at this time is small and three mechanisms can complete mappings well. As the number of services increases, the advantages of DS_MR become larger and its average usage rate is significantly higher than the other two. Taking VN requests=800 as an example, the average usage rate of DS_MR is 51% and 27% higher than those of DS_ESP and SS_ESP respectively. This is because DS_MR not only considers the dynamic adjustment of the slice to improve the network resource utilization, but also designs the link load balancing factor to ensure the traffic balance.

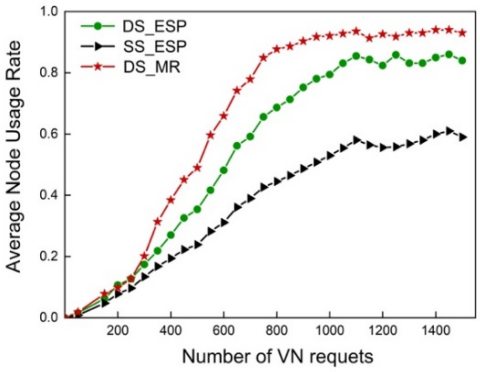

Figure 7: Average node usage rate

Average node usage rates of three mechanisms are shown in Fig. 7. As can be seen from above figure, average node usage rate of DS_MR is always higher than the other two mechanisms. Although this advantage is not obvious when the number of services is small, it is significantly higher than when dealing with a large number of services, since DS_MR guarantees load balancing of nodes during VN mapping and reconfiguration, and releases idle resources to improve utilization.

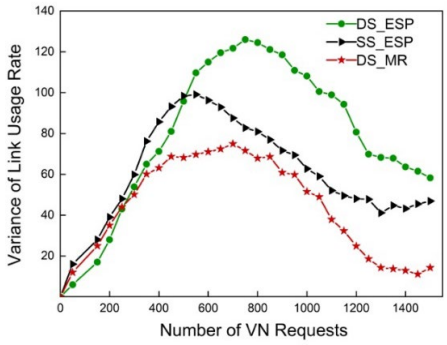

Figure 8: Variance of link usage rate

Variances of link usage rate can directly reflect the load balancing of all links in network. The comparison of three mechanisms is shown in Fig. 8. When the number of services is less than 400, variances of the three mechanisms are not much different, because network has enough resources to complete the mapping of services. Variances of DS_ESP and DS_MR are slightly lower than that of SS_ESP, because they consider the dynamic adjustment of the slice size, which improves resource utilization. As the number of services increases, the usage rate variances of DS_ESP and SS_ESP are significantly higher than DS_MR. Taking VN requests=1000 as an example, the variances of DS_ESP and SS_ESP are 61% and 73% higher than DS_MR respectively. DS_MR designs the link load balancing factor in the processes of VN mapping and VN reconfiguration, and ensures the load balancing by selecting the link with more remaining bandwidth as much as possible, so its variance is always smaller than DS_ESP and SS_ESP.

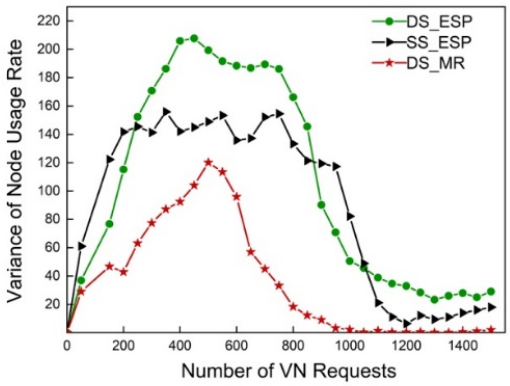

Figure 9: Variance of node usage rate

Fig. 9 shows the variances of node usage rate for three mechanisms. The usage rate variances of DS-ESP and DS_MR are slightly lower than SS_ESP when the number of services is less than 350. This is because the two consider dynamic adjustment of the slice size. However, as the number of services continues to increase, usage rate variances of DS_MR is significantly higher than DS_ESP and SS_ESP. Taking VN requests=800

as an example, usage rate variances of the two are 81% and 72% higher than that of DS_MR respectively. DS_MR designs the node load balancing factor in the processes of VN mapping and VN reconfiguration. It ensures the load balancing of the node by selecting the node with more remaining resources.

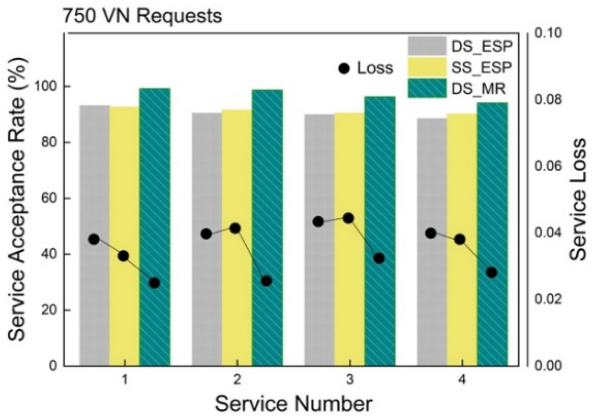

Figure 10: Acceptance rate and loss (750)

The service acceptance rate and loss of three mechanisms is shown in Fig. 10 when VN requests=750. As can be seen from the figure, DS_MR has higher acceptance rates than DS_ESP and SS_ESP when dealing with different priority services. Taking the service 1 with the highest priority as an example, the acceptance rate of DS_MR is 6.2% and 7.1% higher than those of DS_ESP and SS_ESP, respectively. In terms of service loss, DS_MR is always smaller than the other two mechanisms. Also taking Service 1 as an example, the loss of DS_MR is 29.8% and 17.6% lower than those of DS_ESP and SS_ESP respectively. This is because DS_MR considers the dynamic adjustment of the slice to complete more mapping, and designs the load balancing factors to reduce the probability of service mapping failure.

6 Conclusion

In this paper, a dynamic network slicing mechanism based on time window model including VN mapping and VN reconfiguration in core network is proposed to provide network slices for services. A service priority model on the basic of QoS requirements and service attributes is defined to determine the order of resource allocation as well as guarantee high-priority services. In the VN mapping phase, a complete slice containing service chain placement and routing with optimal cost is generated. Next, considering temporal variations of service resource requirements, the size of network slice is adjusted dynamically according to guarantee resource utilization in VN reconfiguration phase. Additionally, Load balancing factors are designed to achieve network traffic balance. Simulation experiments show that DS_MR has great advantages in terms of cost efficiency, and can also better guarantee QoS and load balancing of network.

Acknowledgement: This work is supported by National Natural Science Foundation of China (No. 61702048).

References

Bagaa, M.; Taleb, T.; Laghrissi, A.; Ksentini, A.; Flinck, H. J. (2018): Coalitional game for the creation of efficient virtual core network slices in 5G mobile systems. *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 3, pp. 469-484.

Baumgartner, A.; Reddy, V. S.; Bauschert, T. (2015): Mobile core network virtualization: a model for combined virtual core network function placement and topology optimization. *Proceedings of the 2015 1st IEEE Conference on Network Softwarization*, pp. 1-9.

Benkacem, I.; Taleb, T.; Bagaa, M.; Flinck, H. (2018): Optimal VNFS placement in CDN slicing over multi-cloud environment. *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 3, pp. 616-627.

Carpio, F.; Dhahri, S.; Jukan, A. (2017): VNF placement with replication for load balancing in NFV networks. *IEEE International Conference on Communications*, pp. 1-6.

Carpio, F.; Jukan, A.; Pries, R. (2018): Balancing the migration of virtual network functions with replications in data centers. *IEEE/IFIP Network Operations and Management Symposium*, pp. 1-8.

Chernyshev, M.; Baig, Z.; Bello, O.; Zeadally, S. (2017). Internet of things (IoT): research, simulators, and testbeds. *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1637-1647.

David, D. L. B.; Lin, F. J. (2018): Extending IoT/M2M system scalability by network slicing. *IEEE Noms IEEE/IFIP Network Operations & Management Symposium*, pp. 1-8.

Gu, L.; Chen, X.; Jin, H.; Lu, F. (2018): VNF deployment and flow scheduling in geo-distributed data centers. *IEEE International Conference on Communications*, pp. 1-6.

Kaloxylos, A. (2018): A survey and an analysis of network slicing in 5G networks. *IEEE Communications Standards Magazine*, vol. 2, no. 1, pp. 60-65.

Kuo, J. J.; Shen, S. H.; Kang, H. Y.; Tsai, M. J.; Yang, D. N. et al. (2017): Service chain embedding with maximum flow in software defined network and application to the next-generation cellular network architecture. *IEEE Conference on Computer Communications*, pp. 1-9.

Ordonez-Lucena, J.; Ameigeiras, P.; Lopez, D.; Ramos-Munoz, J. J.; Lorca, J. et al. (2017): Network slicing for 5G with SDN/NFV: concepts, architectures and challenges. *IEEE Communications Magazine*, vol. 55, no. 5, pp. 80-87.

Raza, M. R.; Fiorani, M.; Rostami, A.; Öhlen, P.; Wosinska, L. et al. (2018)**:** Dynamic slicing approach for multi-tenant 5G transport networks [invited]. *IEEE/OSA Journal of Optical Communications and Networking*, vol. 10, no. 1, pp. A77-A90.

Rossem, S. V.; Peuster, M.; Conceicao, L.; Kouchaksaraei, H. R.; Tavernier, W. et al. (2017): A network service development kit supporting the end-to-end lifecycle of NFV-based telecom services. *Network Function Virtualization & Software Defined Network*s, pp. 1-2.

Rossem, S. V.; Tavernier, W.; Sonkoly, B.; Colle, D.; Demeester, P. (2015): Deploying elastic routing capability in an SDN/NFV-enabled environment. *IEEE Conference on Network Function Virtualization and Software Defined Network*, pp. 22-24.

Vizarreta, P.; Condoluci, M.; Machuca, C. M.; Mahmoodi, T.; Kellerer, W. (2017): QoS-driven function placement reducing expenditures in NFV deployments. *IEEE International Conference on Communications*, pp. 1-7.

Ye, Q.; Li, J.; Qu, K.; Zhuang, W.; Li, X. (2018): End-to-end quality of service in 5G networks: examining the effectiveness of a network slicing framework. *IEEE Vehicular Technology Magazine*, vol. 13, no. 2, pp. 65-74.

Yousaf, F. Z.; Bredel, M.; Schaller, S.; Schneider, F. (2017): NFV and SDN-key technology enablers for 5G networks. *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2468-2478.