

# The Design of Elastic Network Control Software based on Data and Control Separation

Chengsheng Pan<sup>1</sup>, Jinjin Shi<sup>1, a</sup>, Li Yang<sup>1</sup>, Zhixiang Kong<sup>2</sup>

<sup>1</sup>School of Dalian University, Dalian 116622, China;

<sup>2</sup> School of Nanjing University of Science and Technology, Nanjing 210094, China.

<sup>a</sup>1249716584@qq.com

**Abstract.** For the large-scale infrastructure of satellite networks, the implementation of control management deployment will face many difficulties such as huge cost, long cycle, immature technology and insecurity. Therefore, the design of flexible network control software based on SDN and NFV numerical control separation is designed. The purpose of SDN is to centrally manage the network, uniformly allocate resources and implement, and effectively grasp the remaining status of resource use. NFV is an operator to reduce resource construction costs and operation and maintenance output, mainly to improve resource utilization and achieve resource sharing. The prototype performs dynamic deployment and effective orchestration of the service chain under the SDN controller, and performs possible aggregation parallelization. The simulation implementation shows that the elastic network control software design can effectively reduce the time delay.

**Keywords:** satellite network; Software Defined Networking; Network Functions Virtualization; flexible network control software design; service chain parallel optimization; delay.

## 1. Introduction

For large complex facilities such as satellite networks, network performance evaluation of actual environments is often impossible. On the one hand, because construction costs are huge and the cycle is long, and even unpredictable dangers may occur. On the other hand, some of the existing technical research is not mature, or only theoretical, and experiments in the actual environment often cause huge the economic loss cannot achieve the desired effect, and the analysis and evaluation on the simulation platform also has the possibility of inaccuracy and large error. However, due to the continuous updating of research technology and the emergence of research ideas, in order to break through the development of traditional networks, only the software test platform that can effectively simulate the real network environment can correctly evaluate the authenticity and validity of the idea, so it is based on numerical control separation. The flexible network software design has important theoretical significance and practical value.

The elastic network software design based on numerical control separation is designed to verify the fast and efficient business arrangement of NFV [1] under SDN [2] control. SDN separates the control functions to form the control layer, and manages the data plane through the southbound API, which makes the network operation and maintenance simple, the network management capability transparent, and the network innovation capability enhanced. The software function implemented by NFV decoupling the underlying physical resources becomes a virtual resource pool, so that the user management platform or application does not need to understand the underlying resources to utilize and control network functions, and realize resource sharing. Infrastructures such as complex and large satellite networks have problems such as high construction cost, decentralized management system, and inefficient deployment of business services. SDN and NFV are introduced and effectively combined into the satellite network to optimize the satellite network architecture. Provides the possibility.

SDN uses the flow table to properly guide the VNF [3] to execute the link policy. NFV and SDN can implement flexible dynamic sequential service chain deployment. However, as the length of the service chain increases, the delay of the service chain increases linearly, which is unacceptable for delay-sensitive satellite networks. Therefore, in the design of elastic network software based on numerical control separation, the link optimization of the service chain is carried out, and the

aggregation of network functions is realized as much as possible, and the service chain delay is shortened.

## 2. Software System Design

As shown in Figure 1, the SDN controller corresponds to the GEO [4-5] satellite and is controlled by the satellite ground station and is responsible for data transmission. The MEO [4-5] satellite is used to receive the flow table delivered by the GEO satellite as a packet forwarding node. LEO satellites can implement network processing functions such as computing, storing, forwarding, and filtering data, so they can be functionally virtualized on LEO[4-5] satellites, such as firewalls, Performance Enhancement Proxy (PEP)[6], and Network Address Translation(NAT)[7], media transcoding, etc., to achieve resource sharing.

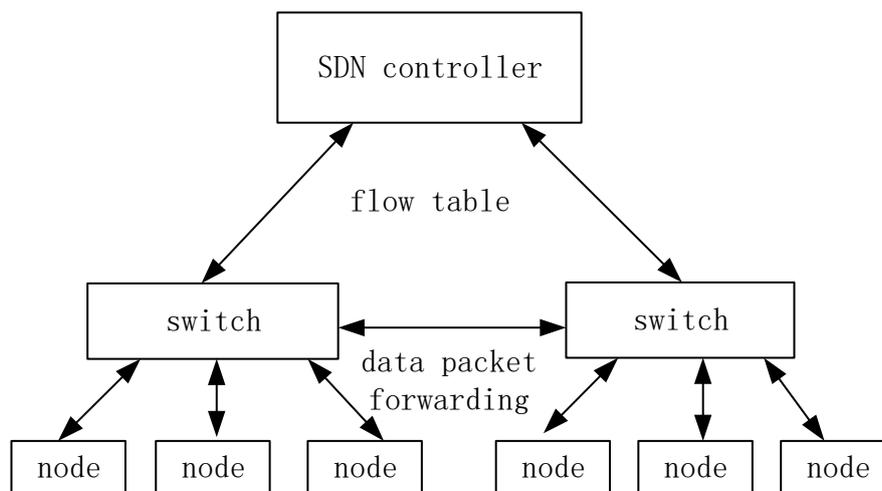


Fig.1 The structure of elastic network control software design

The spatial information network architecture based on SDN and NFV solves the traditional network problem and has the following advantages:

a. Resource sharing. The NFV technology concentrates the underlying physical hardware resources together to form a virtual resource pool through the virtual layer, and the virtual network function shares the resources in the pool, thereby realizing the rational use of resources.

b. Reduce the cost of network facility construction and maintenance. SFC can guide network automation deployment according to SDN controller traffic indication, without manual configuration, providing flexible and dynamic reprogrammable, configured network services, shortening service online time, and network upgrades do not require the purchase of additional hardware devices.

c. The deployment period is short. When the traditional network adds new services, the entire underlying network may be redesigned and deployed. The service function chain can change the flow of traffic according to the needs to select different virtual functions to realize the redeployment of the network.

d. According to the high topological characteristics of the spatial information network, the network function can be virtualized in time, the network invulnerability and flexibility can be enhanced, and parallel programming of possible virtual network functions can be performed to improve resource utilization efficiency.

### 3. Functional Realization of Software Design

#### 3.1 Separation of Data and Control Functions in Satellite Networks

SDN has a global view of the satellite network through flow table information. OpenFlow is used as an interactive interface between GEO and MEO satellites to control the entire network. The SDN controller can implement the following functions:

- a. Obtain network information of the entire network, including information about node resource usage, satellite nodes and topology operation status, communication link connection status, and the like;
- b. Collect user demand information, formulate a control strategy to generate a flow table, and effectively control VNFs through traffic guidance;
- c. Formulate the logical link strategy of SFC, realize the classification management of VNFs, and realize the possible aggregation parallel according to the controller forwarding strategy.

The entire software design process is: First, the user sends the service demand to the SDN controller, the controller collects the node and link information in the network, and formulates the policy and generates the flow table information according to the received information. [8] Then, the switch is configured to receive the flow table information sent by the controller and search for information that can be forwarded for transmission to the LEO satellite node, and simultaneously transmit the state information of the underlying satellite node to the controller. Finally, according to the flow table information, the service network function is completed from the top to the bottom, and the service function chain is arranged according to the service orchestration strategy, and the possible VNFs are aggregated in parallel to shorten the propagation delay.

#### 3.2 Software Design Service Chain Optimization

Parallelism has been well studied in computer programming and high-performance computing. Instruction Level Parallelism (ILP) [9] aims to reorganize sequential instructions in parallel and execute independent instructions in parallel. ILP has been widely adopted by modern processors [9]. Similar to ILP, we intend to use network functions in parallel to improve NFV. Performance, by identifying independent network functions and working in parallel, network function parallelism can significantly reduce network transmission delay, and through statistics of common network functions and their behavioral states, 53.8% of networks in the actual business service chain orchestration process are discovered. Functional pairs can be parallelized [10] and reduce propagation delay more or less.

For the traditional sequential service chain, the implementation of the user service needs to specify a policy to assign the correct location to the VNFs in a logical order. The SDN controller attempts to deploy VNFs to implement a possible parallel intuitive policy map. Therefore, in software design, you need to define a strategy that can combine multiple rules into one to describe the link intent.

Table 1. SFC connection intent

<b>Traditional service chain description</b>	Distribution (IPS, 1) Distribution (NAT, 2) Distribution (FW, 3) Distribution (LB, 4)
<b>Traditional service chain deployment strategy</b>	Forwarding rules (IPS, before, NAT) Forwarding rules (NAT, before, FW) Forwarding rules (FW, before, LB)
<b>Optimize deployment strategy</b>	Location (IPS, first) Control strategy (FW, before, LB) Control Strategy (NAT, before, LB)

The order rule indicates the VNFs connection order in the traditional SFC. An SFC passes IPS→NAT→FW→LB in turn, as shown in Table 1, where IPS stands for intrusion prevention system, NAT stands for network address translation, FW stands for firewall, and LB stands for load balancing. This order rule can be used to describe sequential VNFs. Combined intention. Priority indicates the position before and after the VNFs are specified. In the optimized deployment strategy, the network operator should be able to describe the intent to execute two VNFs in parallel, and the SDN controller further checks the dependencies of the VNFs in the flow table forwarding control policy to see if they can be parallelized. If yes, the flow table forwarding control policy is converted to a priority rule, and the VNFs with the order rule are assigned to a higher priority. If not, the two VNFs should still be linked together. Positional relationships refer to the placement of certain VNFs in specific locations in the SFC, which can be described by designing location rules. For example, we can specify that the IPS is the first location in the SFC, thus ensuring that the packet first passes through the IPS. Through the above rules, the network operator can define a service map by combining multiple rules into one policy (the third row in the table) to define the link intent. Compared with the traditional SFC sequence description, it is necessary to allocate all the positions of the NF in the chain. With the optimized orchestration strategy, only the location intention of some VNFs needs to be specified, which provides a greater possibility for parallel arrangement of VNFs as much as possible.

#### 4. Performance Evaluation

To control the variables of NF complexity, we generate sequential chains by implementing multiple instances of the repeater in the same way in OpenNetVM [11] and software design. We changed the chain length from 1 to 5, we used 64B to 1500B packets to evaluate throughput and focus on (minimum) 64B packet latency. The experimental results in Figure 2 show that the software design based on numerical control separation suffers from a small delay overhead. OpenNetVM is a CPU core dedicated to centralized switches to forward packets, while software design relies on distributed network operations that share CPU cores with network functions. As a result, software design may suffer from less latency overhead, but it can alleviate the performance bottleneck of centralized switches at high packet rates and achieve higher throughput.

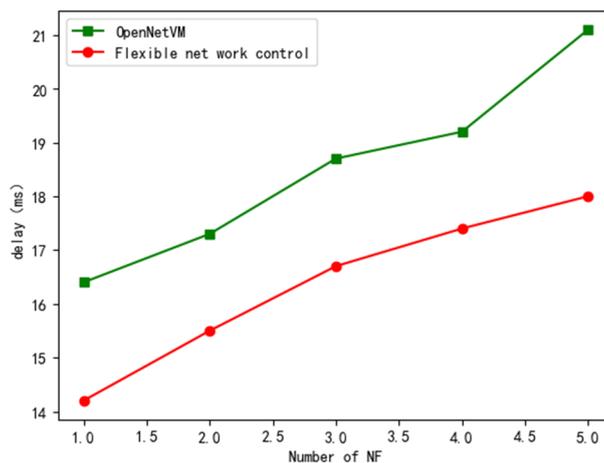


Fig.2 The relationship between the number of NF and delay in SFC

We evaluate the optimization of software design for different network functions. First, we measured the performance of six network functions implemented in software design (Agent Forwarder, Load Balancer LB, Firewall Firewall, Monitor Monitor, Virtual Private Network VPN, Intrusion Detection System IDS) [12]. To investigate the impact of complexity itself, we control the level of parallelism to two, which means that we compare the performance of sequential or parallel combinations of two instances of the same network function. Figure 3 shows the performance of

different network functions. It can be observed that the delay benefit of network function parallelism increases with the complexity of network functions.

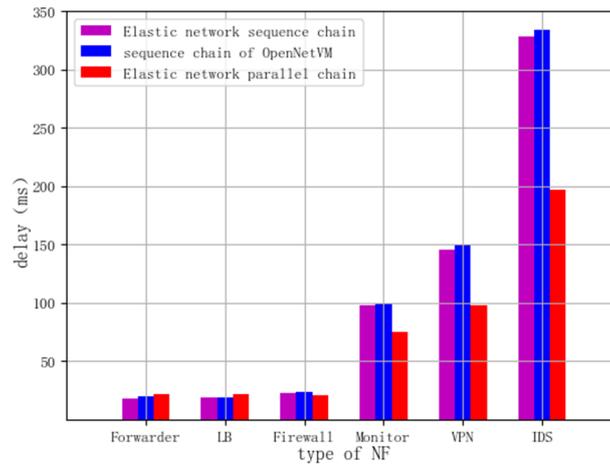


Fig.3 The relationship between NF type and delay

## 5. Summary

The goal of software design based on CNC separation is to implement the architecture of SDN and NFV collaborative deployment. In the design of the elastic network control software based on digital control separation, the data layer is virtualized by the NFV technology. The SDN control layer is responsible for guiding the deployment and scheduling of the data flow control network function, and the controller has a global view management for the entire network. Based on the collaborative deployment architecture, optimize the SFC layout and achieve the aggregation of VNFs as much as possible. Through simulation analysis, compared with the VNFs sequential connection, the effective aggregate parallel connection of VNFs can still meet the user's needs and effectively reduce the delay.

## References

- [1]. Rost M, Schmid S. It's a Match: Near-Optimal and Incremental Middlebox Deployment[M]. ACM, 2016.
- [2]. Han B, Gopalakrishnan V, Ji L, et al. Network function virtualization: Challenges and opportunities for innovations[J]. IEEE Communications Magazine, 2015, 53(2): 90-97.
- [3]. Li Y, Chen M. Software-Defined Network Function Virtualization: A Survey[J]. IEEE Access, 2015, 3:2542-2553.
- [4]. Lee S I, Shin M K. A self-recovery scheme for service function chaining[C]// International Conference on Information & Communication Technology Convergence. IEEE, 2015.
- [5]. Liu Y, Li Y, Canini M, et al. Scheduling multi-flow network updates in Software-Defined NFV systems[C]// Computer Communications Workshops. IEEE, 2016.
- [6]. Yan Wei, Chen Huowang. Overview of Model-Based Software Testing[J]. Computer Science, 2004, 31(2): 184-187.
- [7]. Li Yongwen. Distributed middleware deployment based on potential game [D].
- [8]. Shi R, Zhang J, Chu W, et al. MDP and Machine Learning-Based Cost-Optimization of Dynamic Resource Allocation for Network Function Virtualization[C]// IEEE International Conference on Services Computing. IEEE, 2015.

- [9]. Garraway H. Parallel Computer Architecture: A Hardware/Software Approach[J]. 1999, 10(5):63-68.
- [10]. Sun C, Bi J, Zheng Z, et al. NFP: Enabling Network Function Parallelism in NFV[C]// Conference of the Acm Special Interest Group on Data Communication. ACM, 2017, 62(3):23-31.
- [11]. Wood T, Rama K, Hwang J, et al. Toward a software-based network: integrating software defined networking and network function virtualization[J]. IEEE Network, 2015, 29(3):36-41.
- [12]. Duan Q, Ansari N, Toy M. Software-defined network virtualization: an architectural framework for integrating SDN and NFV for service provisioning in future networks[J]. IEEE Network, 2016, 30(5):10-16.