

A Component-based Middleware for Network Function Virtualization

Yung-Chiao Chen^{1,2}, Yi-Wei Ma², Cheng-Mou Chiang², Jiann-Liang Chen²

¹ Computer and Information Networking Center, National Taiwan University, Taiwan

² Department of Electrical Engineering, National Taiwan University of Science and Technology, Taiwan
chenyc@ntu.edu.tw, {yiweimaa, e619003}@gmail.com, Lchen@mail.ntust.edu.tw

Abstract

A Software Definition Network (SDN) supports programmable, flexible and convenient network management. Network Functions Virtualization (NFV) virtualizes multiple types of network services, reducing the complexity of network construction. The above two technologies work together to improve the efficiency of network construction and facilitate network maintenance. This study concerns the design of component-based middleware with high performance and reliability for NFV. The middleware responds to user needs in configuring network functions. When the load changes, it adjusts the virtual machine configuration to ensure that users can receive quality service. The proposed middleware platform includes the network, NFV and a Resource Manager to improve the operational effectiveness and reliability of the network. Network performance is analyzed identify the proposed method that minimizes the cost of construction a network resource service chain. The proposed middleware can eliminated the service bottleneck of NFV, and the results of the performance analysis demonstrate that it improves network performance by 30% and reliability by 1.5%.

Keywords: Software-defined Networking, Network function virtualization, Middleware, Service chain, Resource allocation

1 Introduction

The rapid development of network communication technology has resulted in a rapid change in network service type. Data storage, computing and services are becoming increasingly virtual. Important developments are being made in SDN and NFV technologies [1-2].

Network functionality is evolving toward diversity and complexity to meet operational needs of the cloud and virtualization. To improve service quality and customize network services, operators must use professional equipment and a software platform that supports several network uses. However, most current network construction methods are based on hardware,

and various devices commonly have compatibility problems, preventing them from communicating with each other effectively. Therefore, operators of diverse and customized network services commonly have to invest various time and resources to find effective solutions. As the hardware required to perform a network function becomes more extensive, the complexity and difficulty of network construction, deployment, integration and maintenance also increase. Therefore, the traditional network model cannot meet the needs of today's users [3].

SDN infrastructure is divided into a Control Plane and a Data Plane. An SDN depends on centralized management to provide programmable, flexible and convenient network management and relevant settings. NFV virtualizes many network services using a software-based build method. NFV performs network functions in a virtualized manner, executing them on standard server hardware, improving the efficiency of network configuration. NFV technology also solves the problem of compatibility between network-specific devices, reducing the difficulty of network building. NFV can help to meet the demands of users, support the rapid deployment and configuration of network functions, and improve the efficiency of network maintenance [4-6].

2 Background Knowledge

Traditional network architecture and service methods cannot meet current demand. SDNs and NFV represent important developmental directions. In this work, NFV is integrated with an SDN to manage the network environment [7].

2.1 Software-Defined Networking Technology

The main task of network managers is to establish rules for dealing with events that may occur in a network to optimize network performance. However, establishing network devices and rules is not easy. When a change to the state of a network demands the modification of rules, network managers must typically

set rules manually, one at a time, for all network devices, making system maintenance and tuning a challenging process. The concept of the SDN was developed to facilitate network management and construction [8-15].

Google has successfully implemented an SDN at its data center and thereby improved system efficiency. This case for SDNs has become a focus of public attention. SDN key technology is divided into a control layer and a data layer. SDNs provide the benefits of programmability, automation and network control, enabling operators to build networks that are highly scalable and flexible [16-23].

2.2 Network Function Virtualization Technology

To provide a wide range of network services and maintain high service quality, operators must deploy several professional network devices and software platforms, to satisfy various network usage requirements. However, networks are commonly unable to interoperate owing to issues of compatibility among hardware devices. Therefore, network service operators have spent substantial time and money in solving these problems when planning their network services. To solve these problems and to reduce the cost of building networks, network service providers have developed the concept of NFV. NFV virtualizes hardware resources as software resources [24-26]. NFV virtualizes hardware resources into software resources, as shown in Figure 1.

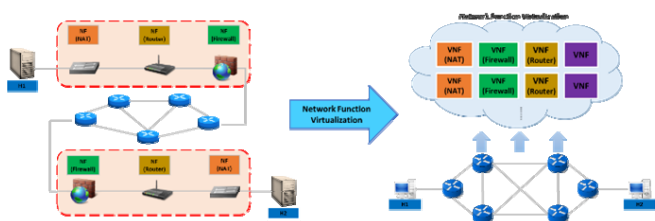


Figure 1. Traditional network and NFV environment

Network Functionality Virtualization is the virtualization of network functions in layers four through seven of the Open System Interconnection Reference Model, which include routers, firewalls, Quality of Experience and Load Balancer [27-28]. The standard protocol for NFV is driven by the European Telecommunications Standards Institute. Software-based network functions, implemented on a standard server, provide network services, enabling the dynamic deployment, deletion or migration of those functions without changing the physical network environment, favoring network construction efficiency [29-30]. To realize NFV, an NFV architecture model is utilized herein. The proposed system is composed of four layers infrastructure, virtual, application and service layers which cooperate with each other [31-40].

3 System Architecture

3.1 System Overview

The modification of settings, maintenance and management of a traditional network are not easy. If a user needs to install the NAT network function, then the operator must modify back-end settings to support an end-user connection. This process is prone to human error, which leads to network failure. NFV solves the problem of establishing a new network function in a traditional network. NFV provides network functions, reducing the risk of human error and difficulties with maintenance. The network managers need only to manage the virtual network layer, without having to manage physical hardware devices, reducing the possibility of human error.

3.2 Proposed NFV Architecture

This work proposes a component-based NFV system architecture, which can be divided into NFV Infrastructure (NFVI), NFV Communication (NFVC) and Middleware (NFVM) layers, as shown in Figure 2. NFVI is responsible for the settings that determine network environment, based on the decisions made in the NFVM layer. NFVM controls the network environment, manages it, and makes decisions regarding it. NFVI and NFVM layers communicate with each other by NFVC.

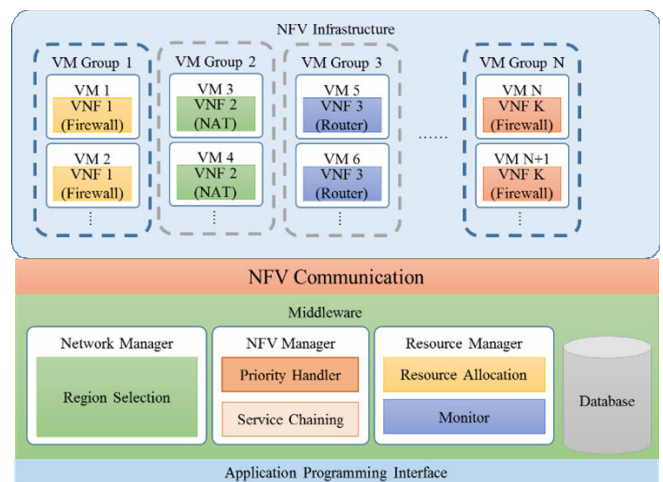


Figure 2. Proposed NFV architecture

NFVI consists of various physical devices, which incorporate operational, storage and network resources. Virtualization technology virtualizes hardware resources as virtual resources, and virtual machines are constructed to execute Virtual Network Functions (VNF) to provide services. NFVI consists of the network, NFV technology, and the Resource Manager, which is responsible managing network resources. NFVM communicates with NFVI by NFVC, to obtain various messages to support making decisions.

3.3 Notations for Proposed System

Table 1 presents the notations of problem formulation.

Table 1. Notations of problem formulation

Symbol	Definition
Res_{lower}	The lower bound resource utilization of the virtual machine.
Res_{upper}	The UPPER bound resource utilization of the virtual machine.
$Res_{Acceptable}$	The acceptable bound resource utilization of the virtual machine.
Res_T	The current used resource of the heavy-load/light-load VM.
NT_T	The current network throughput of the heavy-load/light-load VM.
NT_{MaxT}	The maximum network throughput of the heavy-load/light-load VM.
U_T	The current CPU utilization of the heavy-load/light-load VM.
Res_{exT}	The excessive resource of the heavy-load/light-load VM.
Res_{R_j}	The current used resource of the j^{th} receiving VM.
NT_{R_j}	The current network throughput of the j^{th} receiving VM.
NT_{Max_j}	The maximum network throughput of the j^{th} receiving VM.
U_{R_j}	The current CPU utilization of the j^{th} receiving VM.
Res_{REx_j}	The receivable resource of the j^{th} receiving VM.
$Res_{AvaialbeR}$	The receivable resource of all receiving VM.
W_{RA1}	The weight of network throughput for resource allocation.
W_{RA2}	The weight CPU utilization for receiving allocation.
Res_{U_i}	The current used resource of the i^{th} VM.
NT_{Max_i}	The current network throughput of the i^{th} VM.
U_{U_i}	The maximum network throughput of the i^{th} VM.
Res_i	The current CPU utilization of the i^{th} VM.
W_{SC1}	The available resource of the i^{th} VM.
W_{SC2}	The weight of network throughput for service chaining.
$T_{initial}$	The weight CPU utilization for service chaining.
T_{update}	The initial time that records in the database when monitoring.
$T_{duration}$	The update time that records in the database when monitoring.
$T_{tolerate}$	The toleration time that the virtual machine can be in heavy-load/light-load status.

3.4 Network Manager

A very large network environment is difficult to manage, so must be divided into many small areas to facilitate management. A network manager plans, divides and integrates virtualized areas; the network is divided into local and global regions to meet user needs. The network manager contains a Region Selection component.

Region selection component. The Region Selection component chiefly manages the division of the network area, as shown in Figure 3. Since the size of the network environment varies, managing a network is not easy; partitioning a network into many regions makes management more convenient.

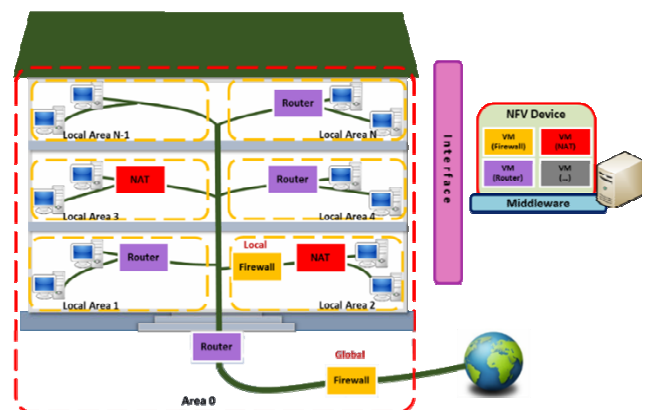


Figure 3. Region selection

A network can be divided into many areas, including local and global areas, such that different regions perform different types of network service. Managers can be assigned to different types of area, such as a company, a department, a floor or a studio. Region Selection is used to provide a range of network services in a particular area. When a user has a network service demand, will be configured as determined by the location of the user. Region Selection execution flowchart and time sequence is shown in Figure 4 and Figure 5.

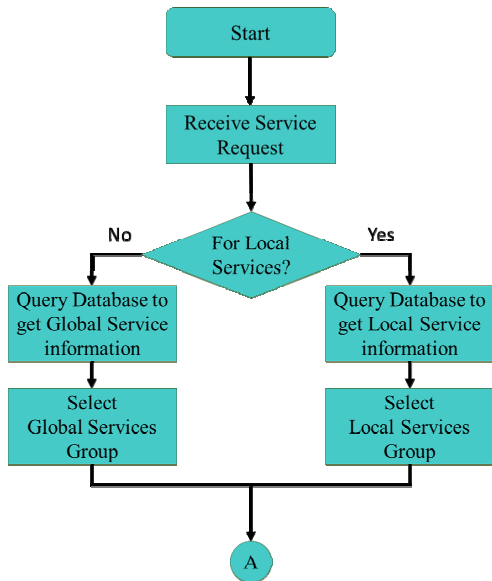


Figure 4. Region selection – Flowchart

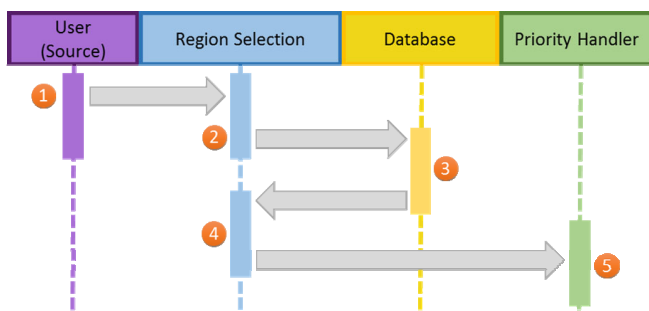


Figure 5. Region selection - Time sequence



3.5 NFV Manager

The main purpose of NFV Manager is to manage network functions, and to set the operating mode and the process method. Different resources are assigned to different users, based on user priority to ensure that users with higher priority receive services of a higher quality. NFV Manager includes the Priority Handler and Service Chaining components, which work together to provide users with requested network services.

Priority handler components. Priority Handler component is mainly used to provide users with the need to use the network function, the configuration

contains the corresponding to virtual network function of virtual machine level selection. Not all users, when using a network service, can be assigned to the same resource. To ensure that specific users enjoy a higher service quality than others, users are divided into two levels normal and preferred. Table 2 presents the two levels of VM type.

Table 2. VM classification

VM type	Computing resource	Users (Number)
Preferential VM 	High	Limited
Normal VM 	Low	Unlimited

In different virtual machines, more resources are deployed to preferential types, and fewer resources are deployed to normal types. The deployment of more resources to preferential types of VM to ensures that high-level users receive high services quality. Priority handler flowchart and time sequence as shown in Figure 6 and Figure 7.

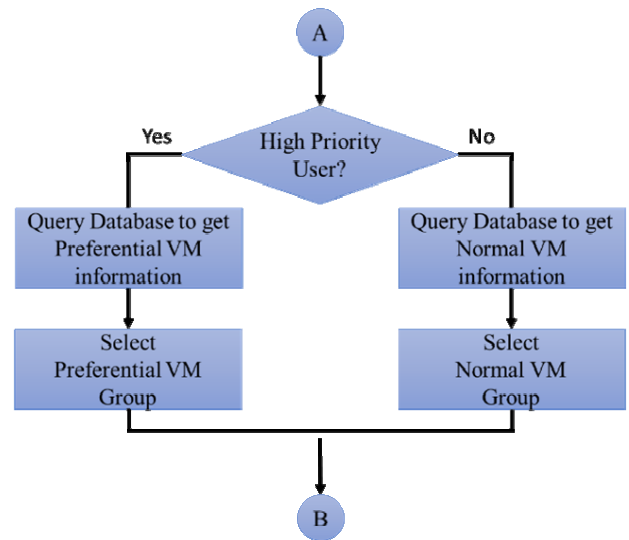


Figure 6. Priority handler – Flowchart

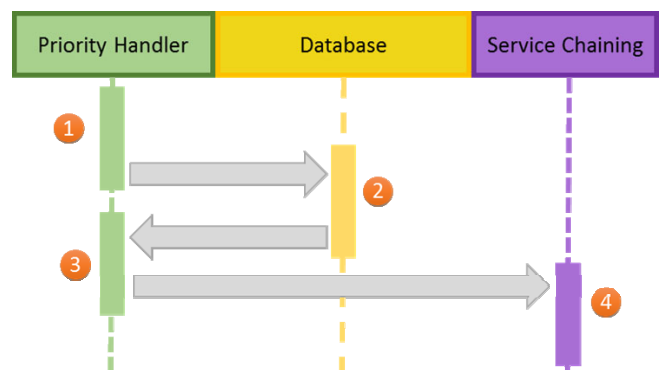


Figure 7. Priority handler - Time sequence

Service chaining component. Service Chaining deals with user's requirements for the configuration and implementation of a service chain to provide users with effective service selection methods that meet their needs. Service Chaining reduces the need for resource reconfiguration and improves service efficiency by analyzing virtual machines with network functions and selecting those with a low utilization rate.

This work proposes Service Chaining for establishing a service chain. The resource usage of virtual machines is computed, and a favorable virtual machine configuration is selected for the service chain to provide favorable service quality. Equation (1) specifies the current resource utilization of a virtual machine in executing a network function. NT_{U_i} and NT_{Max_i} represent the current network throughput of the i -th virtual machine and its maximum network throughput, respectively. U_{U_i} represents the current CPU utilization of the i -th virtual machine.

$$Res_{U_i} = W_{SC1} \times \frac{NT_{U_i}}{NT_{Max_i}} + W_{SC2} \times U_{U_i} \quad (1)$$

Equation (2) is mainly used to select the virtual machines that can provide the most resources. Res_i represents the amount of resources that the i -th virtual machine can currently provide.

$$\begin{aligned} Maximim : Res_i &= 1 - Res_{U_i} \\ &= 1 - (W_{SC1} \times \frac{NT_{U_i}}{NT_{Max_i}} + W_{SC2}) \end{aligned} \quad (2)$$

W_{SC1} and W_{SC2} represent the weights of the network throughput and the CPU utilization. The sum of W_{SC1} and W_{SC2} is unity. The network administrator can adjust the values of W_{SC1} and W_{SC2} according to the network environment.

$$W_{SC1} + W_{SC2} = 1 \quad (3)$$

When all network functions have been configured, the results of that configuration are sent to an NFVI to enable the design of the service chain and recorded in a database to enable the usage of the network to be monitored. Service chaining flowchart and time sequence as shown in Figure 8 and Figure 9.

3.6 Resource Manager

The main purpose of Resource Manager is to manage network devices, and to monitor the usage of NFVI. To reduce the waste of resources, the provision of virtual machine services is dynamically adjustable. Overloading occurs when too many users use a single virtual machine of VNF, reducing the efficiency with which services are provided. When a user does not

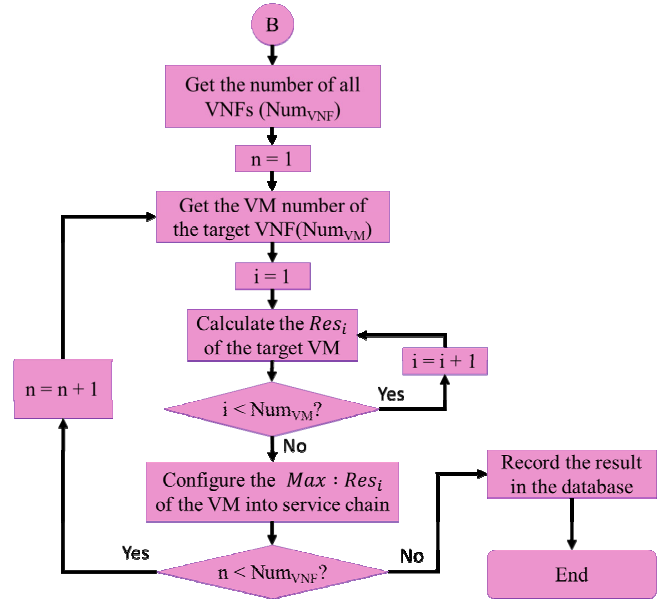


Figure 8. Service chaining – Flowchart

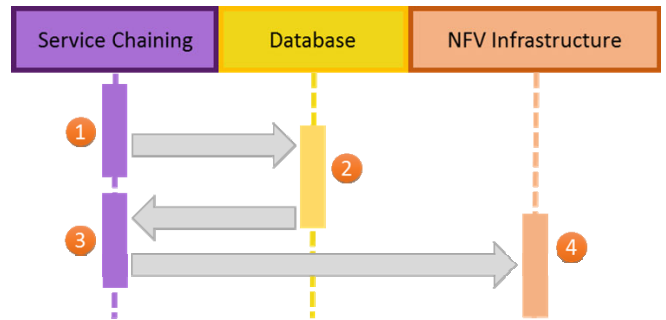


Figure 9. Service chaining - Time sequence

require a virtual machine, the load is light, resulting in wasted resources. Therefore, adjusting resource allocation and monitoring the overall environment are important tasks. Resource Manager contains Resource Allocation and Resource Monitoring components.

Monitor and database component. To improve the efficiency of virtualized network functions, the overall environment must be monitored and adjusted. To provide the correct service, the administrator must register the service user, record network information and the user's request in a database, as shown in Table 3. This table records user ID, priority level of the user, and the required network function.

Table 3. NFV user profile

User ID	Priority	VNF
1	Preferential	NAT, Firewall, Router
2	Normal	Router, Firewall
3	Normal	NAT, Router, Firewall
4	Preferential	NAT, Firewall
5	Preferential	Firewall
...

Since a user's region is not fixed, a network function may span multiple regions. Therefore, to records

information about the region in which the network function that is requested by the user is performed.

Table 4 presents the monitored records, which contain various forms of information.

Table 4. Monitoring data

VM ID	VM Level	Region	Network Function	Computing Utilization	Network Throughput	Initial Time	Update Time
1	Normal	Global	Firewall	23%	350Mbps	1459326998	-
2	Preferential	Local	Firewall	10%	100Mbps	1459326970	-
3	Preferential	Local	NAT	60%	800Mbps	1459326981	-
4	Normal	Global	Router	55%	700Mbps	1459326989	-
5	Normal	Local	Router	90%	1200Mbps	1459326981	1459327051
6	Preferential	Local	Router	40%	400Mbps	1459326985	-
...
m	Normal	Global	VNF	85%	1000Mbps	1459326970	1459326992

When a network service has been constructed, the configuration is recorded in a database. Each user requests a particular network function and has a particular priority. When a service chain is constructed, the user configuration is recorded in a database, which also records the network functions that are required by users and the virtual machine ID during the construction of the service chain, to support management by the administrator.

During the monitoring process, the Monitoring component periodically sends to NFVI requests for information about the virtualized networked environment. When this information is available, whether a virtual machine is under a light load, in a normal state, or overloaded is determined. If all virtual machines are in a normal state, then this result is recorded in the database and the initial record is updated with the current time, flowchart and time sequence as shown in Figure 10 and Figure 11.

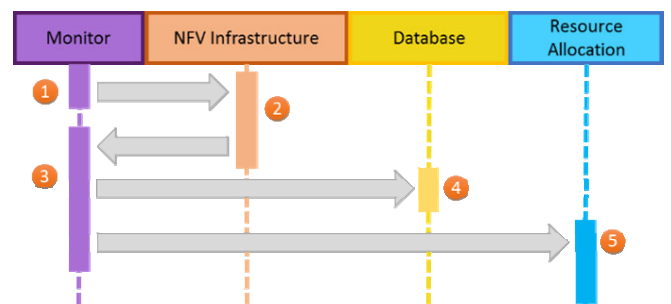


Figure 11. Monitor - Time sequence

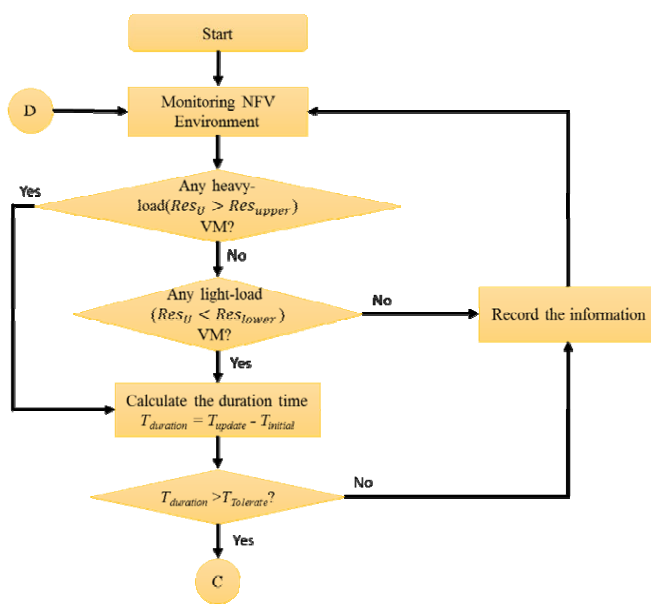


Figure 10. Monitor – Flowchart

Resource allocation component. Resource Allocation deals mainly with the reconfiguration of resources when virtual machines are overloaded or lightly loaded. The overloading of a virtual machine is prevented to avoid inefficiency and the wasting of resources through lack of use. In the first method, when overloading occurs, excess load is transferred to other virtual machines that perform the same network function, so these other virtual machines share the excess load, reducing the load on the original virtual machine; when a virtual machine is under a light load, all of that load is transferred to virtual machines that perform the same network function, releasing resources to reduce idleness and improve operating efficiency. In the second approach, when overloading occurs, a new virtual machine is established to receive the excess load.

Figure 12 shows resource usage and threshold setting for a virtual machine resource management. A load from 0% to the lower threshold (Res_{lower}) is defined as a light load, which is too low for resource utilization. Between the lower threshold and the upper threshold (Res_{upper}) the machine is in the normal state, which is acceptable, requiring no resource adjustment. Between the upper threshold and 100%, the machine is in the overloaded state, requiring the re-allocation of resources. To facilitate the resource re-allocation and to reduce its frequency, a load balance reference point ($Res_{acceptable}$) for resource adjustment is designed. When resource usage is below the load balancing

reference point, an excess work load can be accepted. When resource usage is above the load balancing reference point, the virtual machine is in a buffered state and can accept the rationing of new jobs, but cannot accept excess load. The buffer state is designed to ensure that the virtual machine does not frequently trigger Resource Reallocation requirements.

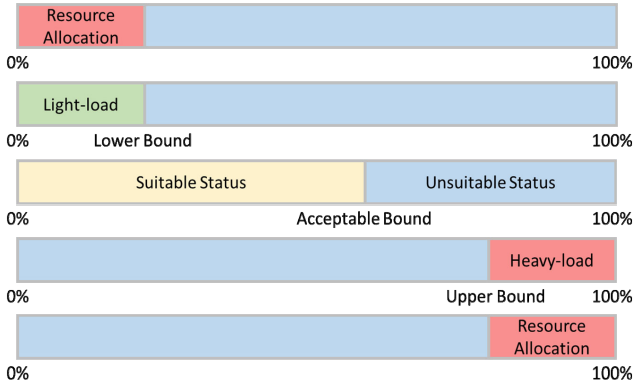


Figure 12. Expression of threshold

When a virtual machine is under a light load or overloaded, it will trigger the reconfiguration of resources. To ensure the smooth re- allocation of resources, whether the virtual machine is under a light load or is overloaded must be determined; whether other virtual machines that perform the same network function can accept the load, completing resource reconfiguration, must also be determined.

Equation (4) yields the current resource usage of a virtual machine. NT_T represents network throughput of a virtual machine. NT_{MaxT} represents the maximum network throughput of all virtual machine. U_T represents the CPU utilization by a virtual machine.

$$Res_T = W_{RA1} \times \frac{NT_T}{NR_{MaxT}} + W_{RA2} \times U_T \quad (4)$$

Equation (5) yields the current excess load. Res_{ext} represents the excess load of a virtual machine.

$$\begin{cases} Res_{ext} = Res_T - Res_{acceptable}, & \text{in Heavyload} \\ Res_{ext} = Res_T & , \text{in Lightload} \end{cases} \quad (5)$$

Equation (6) yields the current resource usage by other virtual machines that perform the same network function. Res_{R_j} represents resource usage of the j-th virtual machine. NT_{R_j} and NT_{Max_j} represent the network throughput and the maximum network throughput of the j-th virtual machine. U_{R_j} represents CPU utilization by the j-th virtual machine.

$$Res_{R_j} = W_{RA1} \times \frac{NT_{R_j}}{NR_{MaxT}} + W_{RA2} \times U_T \quad (6)$$

Equation (7) represents current acceptable load. Res_{REX_j} represents the acceptable load capacity of the j-th virtual machine. When the current utilization of virtual machines is below the load balancing reference point, the receivable load is calculated. Otherwise, the acceptable load is set to zero.

$$\begin{cases} Res_{REX_j} = Res_{Balance} - Res_{R_j}, & Res_{Acceptable} > Res_{R_j} \\ Res_{REX_j} = 0 & , < Res_{R_j} \end{cases} \quad (7)$$

Equation (8) yields the total acceptable load transfer.

$$Res_{Available_R} = \sum Res_{REX_j} \quad (8)$$

W_{RA1} and W_{RA2} represent the weights of network throughput and CPU utilization, The sum of these weights is one, as in Eq. (9).

$$W_{RA1} + W_{RA2} = 1 \quad (9)$$

The resource allocation flowchart and time sequence as shown in Figure 13 and Figure 14.

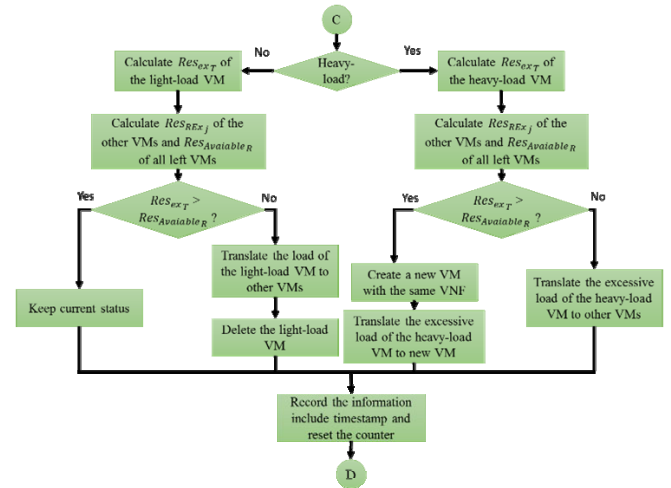


Figure 13. Resource allocation – Flowchart

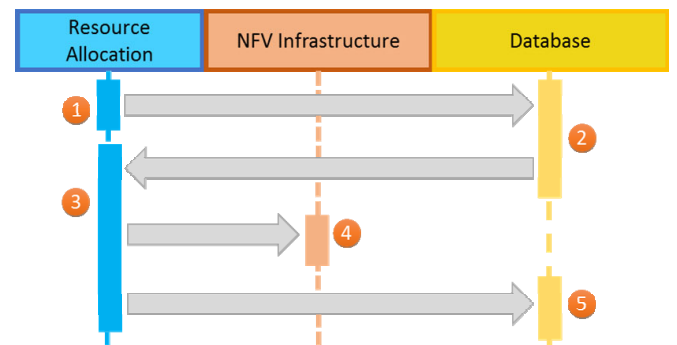


Figure 14. Resource allocation - Time sequence

4 Performance Analysis

In this work, an SDN and NFV are utilized to construct an experimental environment. SDN switches

are utilized to perform routing operations. An NFV server performs VNF to provide services. NFVM is used to manage NFVI, and to make decisions that ensure that it is the one wanted by the user of network operations. Table 5 presents the configuration of the parameters of Network Function Virtualization.

Table 5. NFV parameter configuration

Network function virtualization requirement information	
Number of NFV Types	5
Minimum Number of VMs per VNF	1
Maximum Number of VMs per VNF	6
Maximum Number of VMs in the Environment	30
Maximum Network Throughput of Network Interface	1024Mbps/s
Number of User Connect to the Environment	1 user per second

Figure 15 shows the analysis of service performance for various numbers of users. One user per second is added to the experimental environment; that user is assigned normal priority or high priority at random. High priority users have more resources than users with normal priority. The number of users is limited, so network transmission can be maintained in a short delay state.

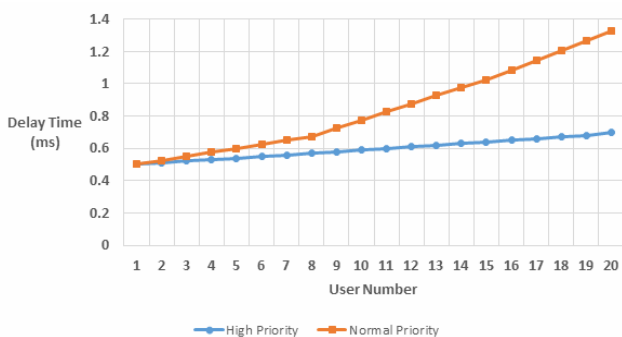


Figure 15. User priority analysis

This work proposes Resource Allocation to adjust a virtualized network environment to ensure smooth network operations. Resource Allocation adjusts the utilization rate to transfer load from heavy VM to light VN that perform identical network functions or establishes a new network function of VM to receive excess load when the network function usage reaches its upper limit. Figure 16 shows the results of a network function usage analysis, comparing the minimum virtual machine method, the maximum virtual machine method, and the method that is presented herein. In the minimum virtual machine method, as the number of user increases, the utilization increases, and at approximately 20 seconds, it reaches the load limit at a utilization rate of 100%, reducing network efficiency. In the maximum virtual machine method, many users are accommodated, and no

excessive load was generated in the experiment; a low utilization rate was maintained, so resources were wasted. In the proposed method, the latter problem is solved. With Resource Allocation, the utilization rate is maintained at a moderate level and the efficiency of network is improved, reducing the waste of network resources. By the proposed method, in 14 seconds, 30 seconds, 44 seconds, the network function usage exceeds the upper bound; the resource adjustment mechanism is triggered, and a new network function is added to the network to solve the problem of excessive loading.

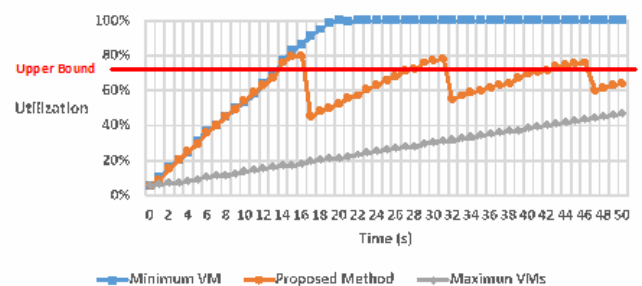


Figure 16. Virtual network function utilization analysis

Figure 17 shows the number of virtual machines that corresponds to the network function of VM that is tested in Figure 16. By the proposed method, when the network function usage exceeds the upper bound, Resource Allocation is triggered to add virtual machines.

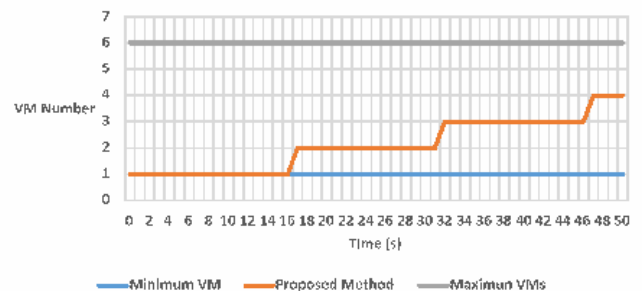


Figure 17. Number of VMs for VNF

To ensure the smooth operation of network services, the packet loss rate must not be too high. Figure 18 and Figure 19 show the packet loss rate and the results of a latency performance analysis, respectively. In the minimum virtual machine method, since the load reaches the upper limit at approximately 20 seconds, the usage rate reaches 100% at that time; the number of user increases, and the packet loss ratio and latency also increase. In the maximum virtual machine method, since resources are very abundant, the packet loss rate and latency are very low. In the proposed method, since the network function usage is maintained within a particular range to solve problem of excessive VMs load, a low packet loss rate and latency are maintained.

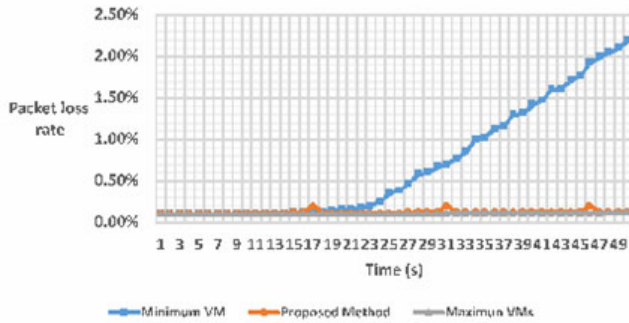


Figure 18. Packet loss rate analysis

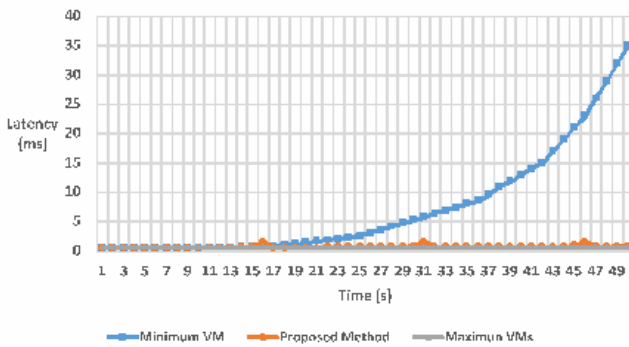


Figure 19. Latency analysis

The mechanism that is proposed in this work can effectively eliminate network inefficiency and the wasting of resources at the cost of increased computing time. Without service chain scheduling, a service chain can be established in more than one way, so construction time is short. However, in the mechanism that is proposed in this study, the establishment of a service chain involves complicated calculations and selection of network function. The proposed method selects an appropriate VM for network function and builds a suitable network function service chain to the user, increasing computing time. As the number of required network functions increases, the computation time increases. Figure 20 shown the computation time analysis.

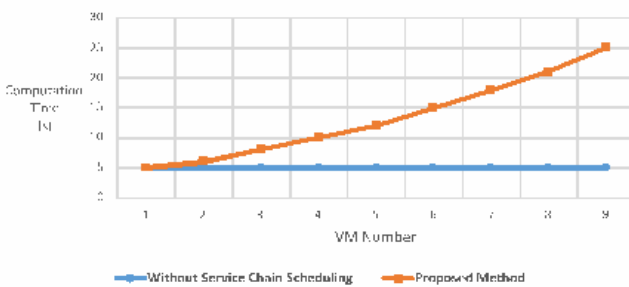


Figure 20. Computation time analysis

5 Conclusion

This work proposed a component-based Middleware, including a network, NFV, and a resource manager to

ensure high performance and reliability in a virtual network environment. Network Manager manages the planning of a network function, based on the area of user that is. NFV Manager manages the provision of NFV services and provides service chain planning based on the needs of users and the status of each individual VNF, improving service quality. Resource Manager manages resource allocation, the quantity of utilization and the adjustment of the load associated with each network service function. This study developed the NFV middleware system achieves enhanced network operational efficiency, reliability and scalability. The experimental results demonstrate that the proposed mechanism improve network performance by 30% and reliability by 1.5%.

References

- [1] F. Hu, Q. Hao, K. Bao, A Survey on Software-Defined Network and OpenFlow: From Concept to Implementation, *IEEE Communications Surveys & Tutorials*, Vol. 16, No. 4, pp. 2181-2206, Fourth Quarter, 2014.
- [2] J. J. Gil, J. F. Botero, Network Functions Virtualization: A Survey, *IEEE Latin America Transactions*, Vol. 14, No. 2, pp. 983-997, February, 2016.
- [3] Y.-D. Lin, P.-C. Lin, C.-H. Yeh, Y.-C. Wang, Y.-C. Lai, An Extended SDN Architecture for Network Function Virtualization with a Case Study on Intrusion Prevention, *IEEE Network*, Vol. 29, No. 3, pp. 48-53, May-June, 2015.
- [4] D. Gao, Z. Liu, Y. Liu, C. H. Foh, T. Zhi, H.-C. Chao, Defending against Packet-In Messages Flooding Attack under SDN Context, *Soft Computing*, Vol. 22, No. 20, pp. 6797-6809, October, 2018.
- [5] J. Matias, J. Garay, N. Toledo, J. Unzilla, E. Jacob, Toward an SDN-Enabled NFV Architecture, *IEEE Communications Magazine*, Vol. 54, No. 4, pp. 187-193, April, 2015.
- [6] R. Chaparadza, T. B. Menem, J. Strassner, B. Radier, S. Soulihi, J. Ding, Z. Yan, Industry Harmonization for Unified Standards on Autonomic Management & Control (AMC) of Networks and Services, SDN and NFV, *Proceedings of Globecom Workshops*, Austin, TX, 2014, pp. 155-160.
- [7] R. Muñoz, R. Vilalta, R. Casellas, R. Martinez, T. Szyrkowicz, A. Autenrieth, V. López, D. López, Integrated SDN/NFV Management and Orchestration Architecture for Dynamic Deployment of Virtual SDN Control Instances for Virtual Tenant Networks, *IEEE/OSA Journal of Optical Communications and Networking*, Vol. 7, No. 11, pp. B62-B70, November, 2015.
- [8] M. Jarschel, T. Zinner, T. Hoßfeld, P. Tran-Gia, W. Kellerer, Interfaces, Attributes, and Use Cases: A Compass for SDN, *IEEE Communications Magazine*, Vol. 52, No. 6, pp. 210-217, June, 2014.
- [9] J. Mambretti, J. Chen, F. Yeh, Software-Defined Network Exchanges (SDXs) and Infrastructure (SDI): Emerging Innovations in SDN and SDI Interdomain Multi-layer Services and Capabilities, *Proceedings of 2014 Science and*

- Technology Conference*, Moscow, Russia, 2014, pp. 1-6.
- [10] K. Govindarajan, K. C. Meng, H. Ong, W. M. Tat, S. Sivanand, L. S. Leong, Realizing the Quality of Service (QoS) in Software-Defined Networking (SDN) based Cloud Infrastructure, *Proceedings of 2nd International Conference on Information and Communication Technology*, Bandung, Indonesia, 2014, pp. 505-510.
- [11] R. Munoz, R. Vilalta, R. Casellas, R. Martínez, SDN Orchestration and Virtualization of Heterogeneous Multi-domain and Multi-layer Transport Networks: The STRAUSS Approach, *Proceedings of IEEE International Black Sea Conference on Communications and Networking*, Constanta, Romania, 2015, pp. 142-146.
- [12] S. Scott-Hayward, Design and Deployment of Secure, Robust, and Resilient SDN Controllers, *Proceedings of 1st IEEE Conference on Network Softwarization*, London, UK, 2015, pp. 1-5.
- [13] L. Li, W. Chou, W. Zhou, M. Luo, Design Patterns and Extensibility of REST API for Networking Applications, *IEEE Transactions on Network and Service Management*, Vol. 13, No. 1, pp. 154-167, March, 2016.
- [14] A. Blenk, A. Basta, M. Reisslein, W. Kellerer, Survey on Network Virtualization Hypervisors for Software Defined Networking, *IEEE Communications Surveys & Tutorials*, Vol. 18, No. 1, pp. 655-685, First Quarter, 2016.
- [15] Nishtha, M. Sood, Software Defined Network — Architectures, *Proceedings of the International Conference on Parallel, Distributed and Grid Computing*, Solan, India, 2014, pp. 451-456.
- [16] A Lara, A. Kolasani, B. Ramamurthy, Network Innovation Using OpenFlow: A Survey, *IEEE Communications Surveys & Tutorials*, Vol. 16, No. 1, pp. 493-512, First Quarter, 2014.
- [17] G. Tarnaras, E. Haleplidis, S. Denazis, SDN and ForCES Based Optimal Network Topology Discovery, *Proceedings of the 1st IEEE Conference on Network Softwarization*, London, UK, 2015, pp. 1-6.
- [18] S. Yamashita, A. Yamada, K. Nakatsugawa, T. Soumiya, M. Miyabe, T. Katagiri, Extension of OpenFlow Protocol to Support Optical Transport Network, and Its Implementation, *Proceedings of the 1st IEEE Conference on Standards for Communications and Networking*, Tokyo, Japan, 2015, pp. 263-268.
- [19] X. Cao, N. Yoshikane, T. Tsuritani, I. Morita, M. Suzuki, T. Miyazawa, M. Shiraiwa, N. Wada, Dynamic Openflow-Controlled Optical Packet Switching Network, *Journal of Lightwave Technology*, Vol. 33, No. 8, pp. 1500-1507, April, 2015.
- [20] J. H. Lam, S. G. Lee, H. J. Lee, Y. E. Oktian, Securing Distributed SDN with IBC, *Proceedings of the Seventh International Conference on Ubiquitous and Future Networks*, Sapporo, Japan, 2015, pp. 921-925.
- [21] X. Sun, T. S. E. Ng, G. Wang, Software-Defined Flow Table Pipeline, *Proceedings of the IEEE International Conference on Cloud Engineering*, Tempe, AZ, 2015, pp. 335-340.
- [22] E. D. Kim, Y. Choi, S. I. Lee, M. K. Shin, H. J. Kim, Flow Table Management Scheme Applying an LRU Caching Algorithm, *Proceedings of the International Conference on Information and Communication Technology Convergence*, Busan, South Korea, 2014, pp. 335-340.
- [23] J. L. Chen, Y. W. Ma, H. Y. Kuo, C. S. Yang, W. C. Hung, Software-Defined Network Virtualization Platform for Enterprise Network Resource Management, *IEEE Transactions on Emerging Topics in Computing*, Vol. 4, No. 2, pp. 179-186, April-June, 2016.
- [24] J. Garay, J. Matias, J. Unzilla, E. Jacob, Service Description in the NFV Revolution: Trends, Challenges and a Way Forward, *IEEE Communications Magazine*, Vol. 54, No. 3, pp. 68-74, March, 2016.
- [25] D. Cotroneo, L. D. Simone, A. K. Iannillo, A. Lanzaro, R. Natella, Dependability Evaluation and Benchmarking of Network Function Virtualization Infrastructures, *Proceedings of the 1st IEEE Conference on Network Softwarization*, London, UK, 2015, pp. 1-9.
- [26] B. Han, V. Gopalakrishnan, L. Ji, S. Lee, Network Function Virtualization: Challenges and Opportunities for Innovations, *IEEE Communications Magazine*, Vol. 53, No. 2, pp. 90-97, February, 2015.
- [27] D. Rajan, Common Platform Architecture for Network Function Virtualization Deployments, *Proceedings of the 4th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering*, Oxford, UK, 2016, pp. 73-78.
- [28] L. R. Battula, Network Security Function Virtualization (NSFV) towards Cloud Computing with NFV Over Openflow Infrastructure: Challenges and Novel Approaches, *Proceedings of the International Conference on Advances in Computing, Communications and Informatics*, New Delhi, India, 2014, pp. 1622-1628.
- [29] B. Nemeth, X. Simonart, N. Oliver, W. Lamotte, The Limits of Architectural Abstraction in Network Function Virtualization, *Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management*, Ottawa, ON, Canada, 2015, pp. 633-639.
- [30] B. Jaeger, Security Orchestrator: Introducing a Security Orchestrator in the Context of the ETSI NFV Reference Architecture, *IEEE Trustcom/BigDataSE/ISPA, Vol. 1*, Helsinki, Finland, 2015, pp. 1255-1260.
- [31] S. V. Rossem, W. Tavernier, B. Sonkoly, D. Colle, J. Czentye, M. Pickavet, P. Demeester, Deploying Elastic Routing Capability in an SDN/NFV-enabled Environment, *Proceedings of the IEEE Conference on Network Function Virtualization and Software Defined Network*, San Francisco, CA, 2015, pp. 22-24.
- [32] M. Miyazawa, M. Hayashi, R. Stadler, vNMF: Distributed Fault Detection Using Clustering Approach for Network Function Virtualization, *Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management*, Ottawa, ON, Canada, 2015, pp. 640-645.
- [33] H. Jeon, B. Lee, Network Service Chaining Challenges for VNF Outsourcing in Network Function Virtualization, *Proceedings of the International Conference on Information and Communication Technology Convergence*, Jeju, South Korea, 2015, pp. 819-821.

- [34] S. Q. Zhang, Q. Zhang, H. Bannazadeh, A. Leon-Garcia, Routing Algorithms for Network Function Virtualization Enabled Multicast Topology on SDN, *IEEE Transactions on Network and Service Management*, Vol. 12, No. 4, pp. 580-694, December, 2015.
- [35] T. Kim, S. Kim, K. Lee, S. Park, A QoS Assured Network Service Chaining Algorithm in Network Function Virtualization Architecture, *Proceedings of the 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, Shenzhen, China, 2015, pp. 1221-1224.
- [36] Y. L. Chen, Y. Qin, M. Lambe, W. Chu, Realizing Network Function Virtualization Management and Orchestration with Model Based Open Architecture, *Proceedings of the 11th International Conference on Network and Service Management*, Barcelona, Spain, 2015, pp. 410-418.
- [37] P. Iovanna, F. Ubaldi, T. Pepe, L. M. Contreras, V. Lopez, J. P. Fernandez-Palacios Gimenez, Main Challenges on WAN Due to NFV and SDN: Multi-layer and Multi-domain Network Virtualization and Routing, *Proceedings of International Conference on Optical Network Design and Modeling*, Pisa, Italy, 2015, pp. 74-79.
- [38] Y. Nam, S. Song, J. M. Chung, Clustered NFV Service Chaining Optimization in Mobile Edge Clouds, *IEEE Communications Letters*, Vol. 21, No. 2, pp. 350-353, February, 2017.
- [39] H. Masutani, Y. Nakajima, T. Kinoshita, T. Hibi, H. Takahashi, K. Obana, K. Shimano, M. Fukui, Requirements and Design of Flexible NFV Network Infrastructure Node Leveraging SDN/OpenFlow, *Proceedings of International Conference on Optical Network Design and Modeling*, Stockholm, Sweden, 2014, pp. 258-263.
- [40] D. V. Bernardo, B. B. Chua, Introduction and Analysis of SDN and NFV Security Architecture (SN-SECA), *Proceedings of IEEE International Conference on Advanced Information Networking and Applications*, Gwangju, South Korea, 2015, pp. 796-801.

Biographies



Yung-Chiao Chen received the M.S. degrees in Electrical Engineering from National Taiwan University, Taipei, Taiwan, in 1988. He is currently an assistant professor in Computer and Information Networking Center, National Taiwan University, Taipei, Taiwan. His current research focuses on Fault-tolerant WDM networks, Wireless Networks, Software Defined Networks and high speed computer networks.



Yi-Wei Ma is an assistant professor in National Taiwan University of Science and Technology. He received the Ph.D. degree in Department of Engineering Science at National Cheng Kung University, Tainan, Taiwan in 2011. He received the M.S. degree in Computer Science and Information Engineering from National Dong Hwa University, Hualien, Taiwan in 2008. His research interests include internet of things, cloud computing, multimedia p2p streaming, digital home network, embedded system, and ubiquitous computing.



Cheng-Mou Chiang received the M.S. degree in Electrical Engineering of National Taiwan University of Science and Technology, Taipei, Taiwan. His research interests include network function virtualization, and software-defined network.



Jiann-Liang Chen was born in Taiwan on December 15, 1963. He received the Ph.D. degree in Electrical Engineering from National Taiwan University, Taipei, Taiwan in 1989. Since August 2008, he has been with the Department of Electrical Engineering of National Taiwan University of Science and Technology, where he is a professor now. His current research interests are directed at cellular mobility management and personal communication systems.

