미래 네트워크 제공을 위한 기계 학습 기반 스마트 서비스 추상화 계층 설계

Duc Tiep Vu, Gde Dharma N., 김경백, 최덕재
*전남대학교 전자컴퓨터공학과
e-mail: ductiep91@gmail.com, gdebig@gmail.com, kyungbaekkim@jnu.ac.kr, dchoi@jnu.ac.kr

Design of Machine Learning based Smart Service Abstraction Layer for Future Network Provisioning

Duc Tiep Vu, Gde Dharma N., Kyungbaek Kim, Deokjai Choi
Dept. of Electronics and Computer Engineering, Chonnam National University

요 약

Recently, SDN and NFV technology have been developed actively and provide enormous flexibility of network provisioning. The future network services would generally involve many different types of services such as hologram games, social network live streaming videos and cloud-computing services, which have dynamic service requirements. To provision networks for future services dynamically and efficiently, SDN/NFV orchestrators must clearly understand the service requirements. Currently, network provisioning relies heavily on QoS parameters such as bandwidth, delay, jitter and throughput, and those parameters are necessary to describe the network requirements of a service. However it is often difficult for users to understand and use them proficiently. Therefore, in order to maintain interoperability and homogeneity, it is required to have a service abstraction layer between users and orchestrators. The service abstraction layer analyzes ambiguous user's requirements for the desired services, and this layer generates corresponding refined services requirements. In this paper, we present our initial effort to design a Smart Service Abstraction Layer (SmSAL) for future network architecture, which takes advantage of machine learning method to analyze ambiguous and abstracted user-friendly input parameters and generate corresponding network parameters of the desired service for better network provisioning. As an initial proof-of-concept implementation for providing viability of the proposed idea, we implemented SmSAL with a decision tree model created by learning process with previous service requests in order to generate network parameters related to various audio and video services, and showed that the parameters are generated successfully.

1. Introduction

The introduction of Software Defined Network (SDN), and Network Function Virtualization (NFV) opens many opportunities for improving the network provisioning in future internet. By separating control plane and data plane, the SDN provides a global view of the whole network at the centralized controller, thus allows easy control and manipulation of data plane. On the other hand, the NFV decouples the software implementation of network functions from the underlying hardware to allow dynamic instantiation, migration and efficient placement of virtualized network functions. Both SDN and NFV technologies play important roles in the development of future networks.

The future services are expected to contain various type of services such as hologram games, virtual reality, livestreaming multimedia, and cloud services, which have unique service requirements. The current network provision of SDN and NFV is solely based network QoS and QoE parameters. QoS is the conventional approach to achieve balance network load between services and network elements while QoE measures the satisfaction level of user. However, QoS and QoE lack a common formalized description model to describe the dynamic service requirements. Therefore, a Service Abstraction Layer is required to improve future network provisioning and maintain the interoperability and homogeneity of the system [4].

Knowing the service requirements is essential to provide resources efficiently and enhance performance for specific applications. However, identifying the service requirements is a challenging task. In practice, a network service is often provided in a number of package services, which differs in terms of service requirements. User often only knows the basic feature of the service such as the resolution and the framerate of the video provided by the service [1]. Thus, we must specify the service category based on user's knowledge of the services to determine the corresponding network requirement of the service.

Machine learning method such as decision tree is commonly used to solve such problem. In [2], the ATLAS framework is proposed to classify application by using machine learning to learn traffic characteristic from previous traffic. Similarly, the service category can be predicted by learning decision rules inferred from previous data.

To address the problem, we present our initial effort to design a Smart Service Abstraction Layer (SmSAL) for future network architecture. The SmSAL seats between user and orchestrator, and employs a decision tree method to generate refined network parameters from ambiguous and abstracted user-friendly input parameters to more suitable service requirements. We implement an initial prototype of the SmSAL for audio and video services to demonstrate the viability of the proposed design.

2. Smart Service Abstraction Layer Architecture

The detailed design architecture of our smart service abstraction layer (SmSAL) is depicted in Fig. 1. The general scenario of our SmSAL is that the users or applications send a service request with their service requirements to SmSAL. Then, SmSAL utilizes the decision tree to analyze and classify these service requirements. The result of the decision tree is a service class which describe the proper network parameters such as bandwidth, jitter and latency. Finally, SmSAL sends these derived parameters to an orchestrator using a JSON message communicator.

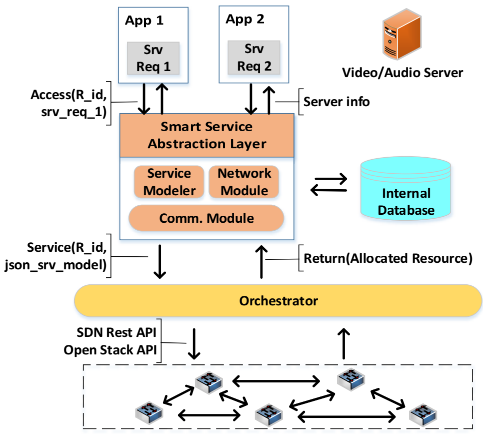

Figure 1. SmSAL Architecture

SmSAL consists of three modules, 1) Service Modeler; 2) Network Module; and 3) Communication Module. An additional module is an internal database which is used to store the status of the service request with service parameters and the information of underlying network deployed earlier with previous service requests with service parameters.

Service Modeler is an important module in SmSAL, which is responsible for analyzing the service request with service requirements received from users or applications and classifying the requested service into a service class. Based on the result of service classification, Service Modeler derives proper network parameters regarding to the classified service class in JSON message format.

Network Module is responsible for monitoring the status of underlying networks and communicating with the internal database in order to exploit the already deployed underlying network regarding to a classified service. Once Network Module receives a JSON message from service modeler, it access the message whether a deployed underlying network for a service described in the message is found in the internal database or not. If there is a matched underlying network, Network Module returns the information of the matched underlying network to users or applications. Otherwise, Network Module passes the JSON message through Communication Module to the orchestrator for network provisioning. Communication Module is responsible for any communication between SmSAL and an orchestrator.

3. Machine Learning-based Smart Service Abstraction Layer

As an initial design and proof-of-concept implementation, in Service Modeler, we use a decision tree algorithm as a machine learning technique to classify the service request with parameters which is submitted by a user or an application. We define the mandatory parameters, which have to be provided by the user. The decision tree algorithm then analyzes these mandatory parameters to define the rules and condition to make the tree-graph decision. The decision tree technique provides results as the service classification on each leaf. A new service request, which has mandatory parameters, will be analyzed using the decision tree model to decide which leaf the service request belongs. The flowchart of decision tree technique inside the service modeler is depicted in Fig. 2.

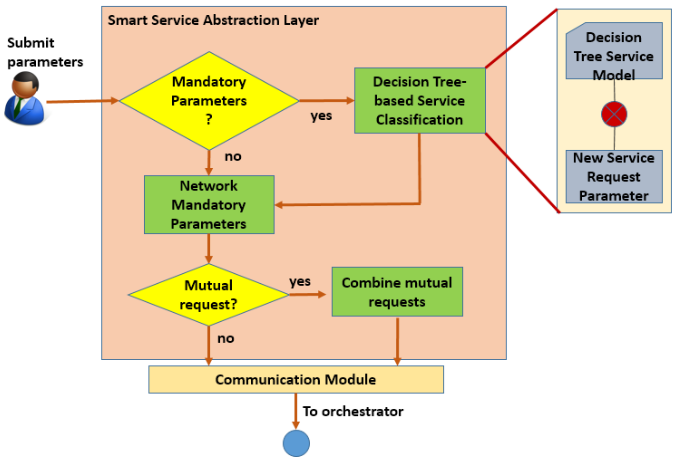

Figure 2. Service modeler flowchart based on decision tree technique

Our Service Modeler classifies a service request using one parameter for each step in the decision tree technique. This concept is similar with the concept of univariate decision tree which is implemented in J48 algorithm. Therefore, in our prototype, we use J48 algorithm from Weka Library to conduct the decision tree algorithm in our initial implementation [3]. To learn the decision tree, we use a sample dataset of multimedia service parameters as a training set for J48 algorithm to create a decision tree model.

Demonstration. As a proof-of-concept, we implemented a prototype of SmSAL as a client-server model by using Java and Weka Library. A SmSAL server is used to classify multimedia services such as audio and video services. We define the mandatory parameters of a service request for a multimedia service such as type, bitrate, resolution, frame-per-second (fps), and duration. The mandatory parameters are submitted by a user using our SmSAL client. Fig. 3 shows sample outputs of our implementation for a SmSAL client.

As depicted in Fig. 3, there are ten parameters which submitted by the SmSAL client. The ten parameters are service id, multimedia type, bitrate, resolution, frame per second, duration, bandwidth, delay, jitter and throughput. The last four parameters (bandwidth, delay, jitter and throughput) are optional for the users, and “0” value is used if the user did not know the value for the parameter. These ten parameters are handed to the SmSAL server by packing them into a JSON message. The format of a sample JSON message sent by the SmSAL client is depicted in Fig. 4.

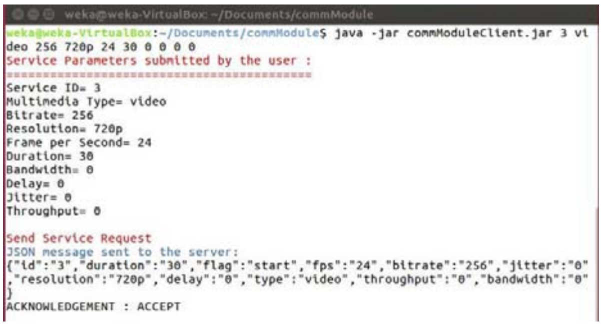

Figure 3. Sample output of a SmSAL client.


```
{
    "id":"3",
    "type":"video",
    "bitrate":"256",
    "resolution":"720p",
    "fps":"24",
    "duration":"0",
    "bandwidth":"0",
    "delay":"0",
    "jitter":"0",
    "throughput":"0"
}
```


Figure 4. A sample format of a JSON message sent by a SmSAL client.

The SmSAL server receives the JSON message sent by the SmSAL client. The SmSAL server then extracts the values from the JSON message and separates the values into the mandatory parameters and the optional parameters. If the SmSAL server detects “0” value for the optional parameters, then the SmSAL server initiates the decision tree function to classify the requested service based on the mandatory parameters. As an initial implementation we have four service classes; Audio_Silver, Audio_Gold, Video_Silver and Video_Gold. Each class has its own values for network parameters, and they are used to define the missing network parameters in the service request.

Fig. 5 shows a sample output of a SmSAL server. As depicted in Fig. 5, we show a service request is received by the SmSAL server, and it initiates the decision tree function in order to find the optional parameters with “0” value. For this demo, the decision tree function classifies the requested service as Video_Gold class. Then, based on the network parameters of Video_Gold class, the SmSAL server define the values of network parameters, and these parameters will be sent to the orchestrator. Fig. 6 shows a sample format of JSON message which contains the generated network parameters.

4. Conclusion

This paper presents our initial work on designing a Smart Service Abstraction Layer for future network provisioning with a machine learning technique. To this end, we have developed a thin smart service abstraction layer with a machine learning module to convert the dynamic user-friendly requirements of services into the corresponding complex network parameters. In future, we are going to improve the service abstraction layer for various other future services with diverse requirements.

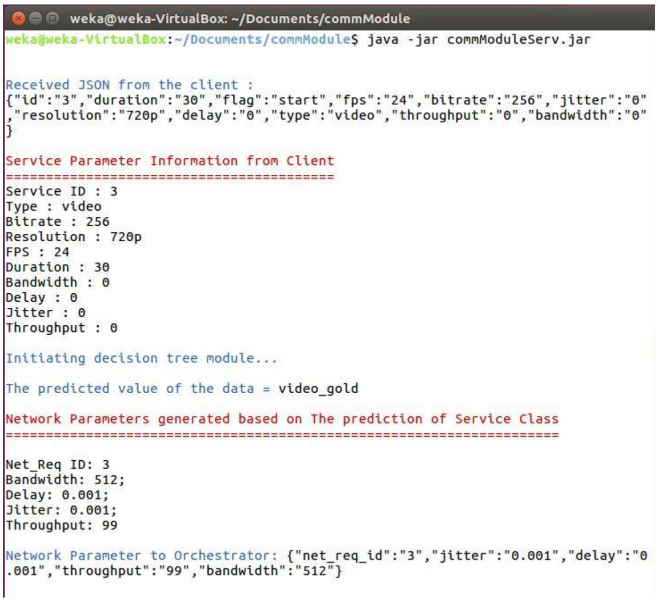

Figure 5. Sample output of a SmSAL server


```
{
    "net_req_id":"3",
    "bandwidth":"512",
    "delay":"0.001",
    "jitter":"0.001",
    "throughput":"99"
}
```


Figure 6. A sample formate of a JSON message generated by a SmSAL server.

Acknowledgement

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No. B0717-16-0033, Study on the multi-dimensional Future Network System Architecture for diversity of Services, Terminals and Networks).

참고문헌

[1] P. Juluri, V. Tamarapalli and D. Medhi, "Measurement of Quality of Experience of Video-on-Demand Services: A Survey," in *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 401-418, Firstquarter 2016.

[2] Qazi, Zafar Ayyub, et al. "Application-awareness in SDN." ACM SIGCOMM computer communication review 43.4 (2013): 487-488.

[3] Machine Learning Group at the University of Waikato (2016). Weka 3: Data Mining Software in Java [Online]. Available http://www.cs.waikato.ac.nz/ml/weka/

[4] Gde Dharma N., Quyet Nguyen-Van, Tiep Vu Duc, Ngoc Ngyen-Sinh, Alvin Prayuda J. D., Kyungbaek Kim, Deokjai Choi, “Design of Service Abstraction Model for Enhancing Network Provision in Future Network”, in Proc. APNOMS 2016, October 2016.