Journal of Network and Computer Applications 93 (2017) 123–136

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Enabling Adaptive Routing Service Customization via the integration of SDN and NFV

Chao Bu[a], Xingwei Wang[a,*], Hui Cheng[b], Min Huang[c], Keqin Li[d], Sajal K. Das[e]

[a] *College of Software, Northeastern University, Shenyang 110819, China*
[b] *School of Computing and Mathematical Sciences, Liverpool John Moores University, Liverpool L3 5UX, UK*
[c] *College of Information Science and Engineering, Northeastern University, Shenyang 110819, China*
[d] *Department of Computer Science, State University of New York, New Paltz, New York 12561, USA*
[e] *Department of Computer Science, Missouri University of Science and Technology, Rolla 65409, USA*

ARTICLE INFO

Keywords:
Routing service
Software Defined Networking
Network Function Virtualization
Customization
Pareto efficiency
Benefit equilibrium

ABSTRACT

The Internet needs to provide the diversified functions and services beyond simple packet forwarding for different network applications. It calls for supporting different communication demands with diversified and customized routing services. However, the current routing service configuration is not based on the global network information to manage network resources and functions, and cannot dynamically attain the adaptively and optimality. The Software Defined Networking (SDN) and Network Function Virtualization (NFV) have inspired a good way to solve these problems. In this paper, based on SDN and NFV, an Adaptive Routing Service Customization (ARSC) mechanism is proposed. In ARSC, the suitable routing services are adaptively customized for different applications with the user utility and the ISP profit considered jointly. In addition, in order to deal with the simultaneously arrived application requests, an efficient matching algorithm is devised to match different applications with appropriate candidate routing services. The matching is optimized with Pareto efficiency introduced, and the benefit equilibrium of the users and the ISPs can be achieved. Simulation results show that ARSC is feasible and effective.

1. Introduction

Currently, quite a lot of new types of network applications are emerging fast. Meanwhile, their communication demands are also becoming more and more diversified and specialized, which brings great challenges to provide appropriate services for them. The current Internet should provide diverse and special services for different applications beyond just simple packet forwarding. There should be different choices from the available network functions (e.g., traffic shaping, buffer management, load balancer, error control) on the packet transmission paths. The objective of the routing service composition is configuring appropriate functions on the communication paths to compose specialized routing services for different applications (Shanbhag and Wolf, 2011). For example, e-mail just needs path calculation and error control functions, while telemedicine often needs more complex functions such as packet scheduling, traffic shaping and failure recovery to guarantee Quality of Service (QoS) (Aamir and Zaidi, 2012). Especially for the novel and complex applications, such as video teleconference, live TV and online gaming, the routing service customization is needed urgently (Lima and Carvalho, 2011). However, the traditional routing service configuration methods are still mainly in manual or command-default mode (Jiang et al., 2016), which makes them difficult to adaptively adjust the QoS according to the changing network status and the application demands in an effective way. Therefore, the Internet is in need of a kind of Adaptive Routing Service Customization mechanism with functional as well as non-functional requirements considered (Klein et al., 2014).

Although quantities of researches (see Section 2) have been done on integrating the adaptation idea into the routing service configuration. In fact, the future Internet requires dynamically customizing routing services to accommodate new user demands by reusing existing functions and assembling new services (Xia et al., 2015). Thus, the network resources and functions should be managed flexibly, rapidly and conveniently with the global network status considered, and the customization mechanism should be extensible for adding new functions and be sustainable in evolving existing functions. In addition, even on the same transmission path, the different customized routing services can provide different QoS (Wang et al., 2013), which brings different service experiences and profits for the user and the ISP respectively (e.g., high price for high quality service). Thus, the benefits

[*] Corresponding author.
E-mail address: wangxw@mail.neu.edu.cn (X. Wang).

http://dx.doi.org/10.1016/j.jnca.2017.05.010
Received 24 November 2016; Received in revised form 21 March 2017; Accepted 25 May 2017
Available online 01 June 2017
1084-8045/ © 2017 Elsevier Ltd. All rights reserved.

should also be considered when composing different routing services. As the user is interested in optimizing his service experiences (Tsiaras and Stiller, 2014) and the ISP is interested in maximizing its own profit (Ma et al., 2011), the routing service customization mechanism should take the benefits of both the user and the ISP jointly into account when achieving the technical requirements.

As the easy-to-manage, easy-to-develop, and easy-to-evolve novel networking paradigms, the Software Defined Networking (SDN) (Kreutz et al., 2015; Nunes et al., 2014) and the Network Function Virtualization (NFV) (Mijumbi et al., 2016) have inspired a good way to deal with the above challenges. SDN decouples the control plane and the data plane with the network intelligence being highly concentrated in the control plane. Based on such networking paradigm, the control plane can get the global network view. Even for the large-scale network, the control plane can be logically centralized and physically distributed and get the global view by the hierarchical and sub-domain control of multiple controllers (Fu et al., 2014). NFV decouples the functions from the physical equipment by leveraging virtualization technology to offer a way to design, deploy and manage services. The integration of SDN and NFV provides the sound basis for customizing routing services by dynamically allocating resources and composing functions under the global view. Although SDN and NFV cannot directly achieve the benefit win-win of the user and the ISP, they can provide good support for this objective. For example, via the logically centralized control plane of SDN and the flexible function composition of NFV, the ISP can uniformly and properly allocate resources and assemble functions to customize the diversified services with different qualities and costs for different users, and thus help optimize its profits. Meanwhile, this enables the user to have more choices from multiple candidate services to optimize his service experiences at the reasonable price under the current network status.

The network operator rents out its owned network resources to ISPs, which can provide their customized routing services (i.e., routing as a service) with their rented resources to support communications among the users. The relationships among multiple ISPs are pluralistic. They can be competitors when providing routing services. Their provided services may be different even for the same type of application request because of their different service customization and marketing characteristics. For example, some ISPs provide high-quality high-price services while others provide cost-effective services due to their market positioning, and even some ISPs may be better at customizing services for certain types of applications due to their special technical features. The competition encourages them to provide distinctive services, which brings more service choices to the users. However, they also can be cooperators when individual ISP cannot accomplish some service provision tasks independently. For example, when many application requests from different users arrived simultaneously, it is almost impossible for any individual ISP to provide services to all of them with their personalized demands satisfied completely due to its limited capacity. To deal with such scenario, the combination of cooperation and competition of the ISPs is necessary. As the result, there can be multiple ISPs to provide their distinctive candidate services to the applications according to their available resources, thus the bidirectional selection relationship among the candidate services and the applications can be established. In this paper, a matching algorithm among multiple applications and multiple candidate services is devised with Pareto efficiency introduced to achieve the benefit equilibrium of the users and the ISPs.

In this paper, an Adaptive Routing Service Customization (ARSC) mechanism based on the SDN and NFV is proposed. It provides the appropriate Customized Routing Services (CRSs) to applications in the dynamic network environment and achieves the benefit win-win of the user and the ISP. The major contributions of this paper are summarized as follows:

- A system framework for routing service customization is devised, by which the different ISPs can adaptively assemble the appropriate functions to provide the CRSs to different applications.
- The user utility is defined to serve the user as the criteria to choose his favorite routing service at the same time the ISP profit is defined to serve the ISP as the objective to customize his offered routing service. Both of them are used jointly to support the benefit win-win of the user and the ISP.
- A pricing strategy is proposed for routing service to support the differentiated resource allocation to the application, in order to reasonably optimize the ISP profit with the user selectivity considered.
- To cope with the situation where multiple application requests arrive simultaneously, an efficient matching algorithm among multiple applications and multiple services is proposed. The Pareto efficiency is used as the optimal condition for each matching round to improve the matching efficiency with benefit equilibrium considered.

The rest of the paper is organized as follows. In Section 2, we review the related work and compare our work with them. In Section 3, we present the system framework of the proposed ARSC. In Section 4, we describe the details of the components in the ARSC. In Section 5, we present the proposed matching algorithm with Pareto efficiency introduced. In Section 6, we present simulations and performance evaluations. Finally, Section 7 concludes the paper.

2. Related work

Many researches have been done on dynamically allocating network resources to optimize the resource supply for routing services. In Tansupasiri et al. (2006), a dynamic QoS routing model was proposed. To ensure the provision of the best services, it assigns different levels of privileges to different users, and uses interrupt mechanism to dynamically reconfigure QoS setting according to the user demands. In Gu and Zhang (2011), based on service awareness, a self-adaptive QoS control mechanism was proposed, through which resources can be adaptively allocated to network flows and the end-to-end performance of the network can be optimized. In Cohen et al. (2005), a fuzzy method was presented for path selection under the additive service constraints to deal with the imprecise routing QoS information. In Mellouk et al. (2011), based on the trial/error paradigm combined with a continuous adaptive function, a bio-inspired QoS routing algorithm was proposed to find K best paths in terms of the cumulative link cost, dynamic residual bandwidth and end-to-end delay. In McAuley et al. (2008), under the diverse objectives and multiple constraints, an approximate shortest path tree approach was proposed to calculate the best path with routing service quality improved. In contrast, the proposed ARSC can not only support the dynamic resource allocation, but also combine the adaptive function composition together to provide the CRSs.

There are also researches on establishing the adaptive function composition frameworks, which assemble the appropriate functions to provide the demanded services. In Balasubramaniam et al. (2009), a bio-inspired network service management framework and a dynamic routing solution for the future Internet were proposed to address the challenges in network service discovery and composition, and deal with highly dynamic and frequent service changes. In Katsikogiannis et al. (2013), a QoS-aware policy-based management framework for adaptive routing decisions was proposed. In Li and Nahrstedt (1999), a middleware control framework was proposed, using the dynamic control of the internal parameters and re-organization of functions to enhance QoS adaptation decisions effectively. In Benaboud et al. (2011), a routing service discovery and selection framework was proposed based on intelligent agents. It uses domain ontologies to retrieve candidate routing services, sorts them by their QoS, and then assembles the demanded routing services. In Peculea et al. (2010), an adaptive end-to-end routing service framework was proposed, combining admission control and bandwidth reservation mechanism to satisfy the resource requirements of data flows, and using dynamic bandwidth reconfiguration mechanism to adapt to network load changes. Among the above researches, only (Peculea et al., 2010) considered resource allocation and function assembling jointly, the others just considered function composition.

When providing services, (Tansupasiri et al., 2006; Mellouk et al., 2011) just considered the user's utility, while (Gu and Zhang, 2011; Balasubramaniam et al., 2009; Peculea et al., 2010) just considered the ISP's utility. In addition, all the above works made decisions to allocate resources and assemble functions for services mainly according to the local information. In contrast, the proposed ARSC is based on the SDN and the NFV. It makes decisions to compose services with the global network status considered, and its logically centralized framework facilitates routing management and service customization, which cannot be achieved by the above works. Furthermore, the ARSC takes both the user's utility and the ISP's utility together into account when providing the service to the application, which also cannot be achieved by the above works.

In fact, there are some researches on routing configuration with SDN and NFV considered. In Jarraya et al. (2014), with the simplified and automated SDN management, it becomes easier to configure routing for the applications. In Das et al. (2012), by analyzing the unified control of the SDN, it devised the routing technologies which can take advantage of a common service standpoint. In Bozakov and Papadimitriou (2012), Drutskoy et al. (2012), by virtualizing the underlying network resources, the routing configuration scheme could allocate network resources in the form of network services and facilitate the new service generation. Some works consider using NFV to compose networking services in SDN environment. In Ding et al. (2015), an open framework for service chain as a service (OpenSCaaS) was proposed. It enforced service-chaining policy by leveraging the benefits of SDN and NFV. In Elias et al. (2014), based on NFV paradigm, it proposed some novel orchestration mechanisms to optimally control and reduce the resource congestion of a physical infrastructure. In Cheng et al. (2015), an effective service chain instantiation framework was proposed to cooperatively combine network functions in the optimal way. Although the above researches configured services based on SDN and NFV, they did not consider routing problems and also did not consider the benefits of both the user and the ISP when providing services, which can be supported by the proposed ARSC.

3. The system framework of the proposed ARSC

At the outset, we list in Table 1 the abbreviations used throughout this paper to help reading.

The proposed ARSC is established based on the idea of the integration of SDN and NFV. Its system framework is shown in Fig. 1. In the proposed system framework, the network forwarding infrastructure (e.g., switches) is owned by the NO (e.g., Verizon, AT & T), and can only be directly managed by the NO's CC. The CC rents out the underlying network resources to ISPs, for example, Virtual Network Operators (Coulibaly et al., 2015), which have no network infrastructure and provide services by their rented resources. The CC also provides the VNVs to the different ICs based on their rented resources. With the user utility and the ISP profit considered, the control plane in ARSC dynamically invokes and assembles the basic function components for providing the CRSs to applications.

The control plane is composed of one CC and multiple ICs, and each IC has the same internal structure as the ISP1 controller's shown in Fig. 1. As the routing service composition center of each ISP, the IC composes candidate services for applications according to its VNV, so that the users can have more choices to satisfy their own preferences on ISPs. This can not only release the NO from the heavy and fine-grained works on customizing personalized routing services for the users, but also encourage the competition among ISPs and thus promote the full utilization of the underlying network resources. The CC is the decision-making authority for routing service provision. It enables the matching

Table 1
Abbreviations.

Abbreviation	Full name
NO	Network Operator
CC	Central Controller
VNV	Virtual Network View
IC	ISP Controller
SRM	Service Request Message
CSM	Candidate Service Message
NLM	NULL Message
SAM	Service Allowance Message
SDM	Service Denial Message
SPM	Service Provision Message
AAM	Application Acceptance Message
ADM	Application Denial Message
FTE	Forwarding Table Entry
UCC	Utility Computing Component
AMC	Application Matching Component
ACC	Admission Control Component
SCC	Service Composition Component
SPC	Service Pricing Component
SMC	Service Matching Component
MIC	Message Interaction Component
AIMIB	Application-ISP Matching Information Base
RIB	Resource Information Base

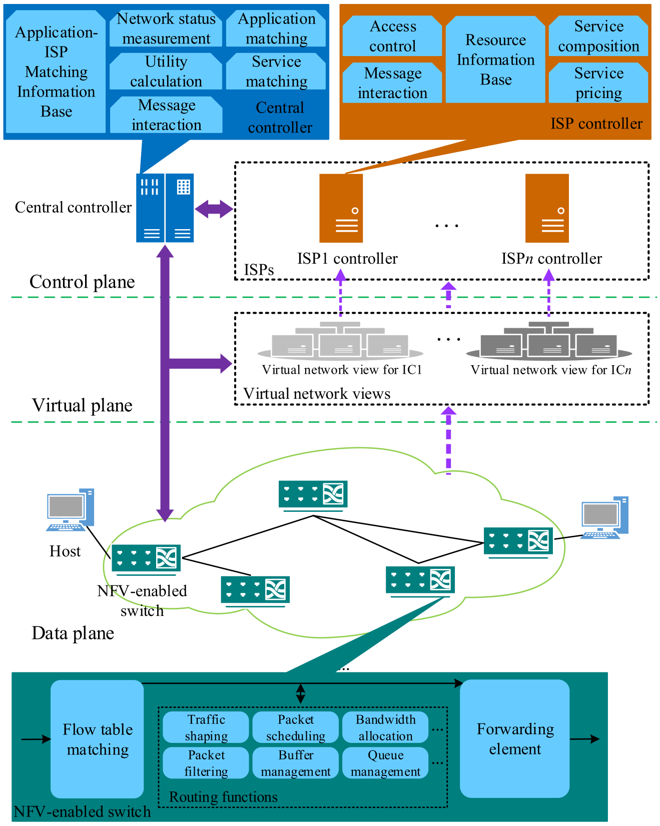


Fig. 1. The system framework of the proposed ARSC.

among the applications and the candidate services of different ISPs. According to the successfully matched services, the CC manipulates the network resources and functions to customize the routing services for the applications.

The virtual plane is established by the CC to virtualize the underlying network resources. For each VNV based on each ISP's rented network resources, the node independence of each ISP and the resource (e.g., bandwidth, computing capability, storage space, forwarding table, etc.) independence of each node are supported. In this way, each IC manipulates its rented network resources through the CC, so that the underlying network resources can be shared by multiple ISPs. Through the virtual plane, the global network resources can be unified, monitored and allocated by the CC.

The data plane consists of NFV-enabled switches, which are programmable and have the same internal structures as shown in Fig. 1. The proposed NFV-enabled switches (Mijumbi et al., 2016) not only contain general-purpose forwarding functions (shown in Fig. 1), but also can be programmed with special-purpose packet processing functions by the CC. These special-purpose functions are proposed by the ICs, and are only authorized to be used by the corresponding ICs to compose their customized services. Thus, each ISP can compose its CRSs with general-purpose and special-purpose functions (Bu et al., 2017). The switches are incapable of determining which resources and functions to be allocated and assembled by themselves, but only in charge of dealing with applications according to the forwarding tables distributed by the CC.

3.1. The routing service provision workflow

The CC and the ICs constitute the logical control center of the proposed ARSC. By message exchange and function component cooperation, they customize the routing services adaptively and generate forwarding table entries to guide the switches to forward packets. The messages are exchanged by the MICs of the CC and the ICs. The CC receives the SRM from the switch, and the CSM or NLM, SPM from the ISP. It sends SAM or SDM, SRM to the ISP, AAM and FTE, or ADM to the switch. The SRM carries information of QoS requirement, preference on ISP, and the affordable price of the application. The CSM carries information of the providable QoS and billing price of the candidate service composed by the ISP, and the resources and function components used by the service. The NLM carries NULL information, which indicates that the ISP cannot provide the desired service to the application. The SAM and SDM carry the information about the CC allowing and denying the ISP's candidate service respectively. The AAM and ADM carry the CC's notification of accepting and denying the application request respectively. The FTE is generated by the CC according to the matched candidate service.

The CC performs the necessary processing according to the carried information of the received messages. When SRM received, the CC sends it to each ISP and waits for the responses. If the ISP can provide the requested service, it sends CSM to CC, otherwise it sends NLM to CC. If multiple ISPs respond with CSMs, the CC executes the UCC to get the user utilities of the candidate services provided by different ISPs, and then executes the AMC to select the most suitable service to match the application request based on the obtained user utility; according to the matching result of AMC, the CC sends SAM or SDM to each involved ISP, then sends AAM and FTE to the switch which issued SRM. If only one ISP responds with CSM, the CC just sends SAM to the ISP, and then sends AAM and FTE to the switch which issued SRM. If all involved ISPs respond with NLMs, the CC just sends ADM to the switch which issued the SRM.

Algorithm 1.

```
Input: SRM /* the message from the switch */
Output: AAM or ADM, FTE,
/* the messages to the switch which issued SRM */
Begin
  if SRM is received do
    Send SRM to the ISPs;
    if multiple ISPs respond with CSMs then
      Run UCC to obtain the user utilities;
      Run AMC to obtain the matched service;
        for each involved ISP do
          if the candidate service is matched then
            Send SAM to the ISP;
          else Send SDM to the ISP;
          end if
        end for
      Update AIMIB;
        return AAM and FTE;
      else if only one ISP responds with CSM then
          Send SAM to the ISP;
          Update AIMIB;
          return AAM and FTE;
        else /* all ISPs respond with NLMs */
          return ADM;
        end if
    end if
end
```

Algorithm 2.

```
Input: SRM /* the message from the CC */
Output: CSM or NLM, SPM
/* the messages to the CC */
Begin
  if SRM is received then
    Run ACC according to RIB;
    if the available resources are sufficient then
      Run SCC to obtain the candidate service plan;
      Run SPC to obtain the service price and profit;
      Send CSM to the CC;
      if the CC responds with SAM then
        Prepare the candidate service;
        Send SPM to the CC;
    else /* the CC responds with SDM */
      Cancel the candidate service;
    end if
  else /* the available resources are insufficient */
      Send NLM to the CC;
    end if
  end if
end
```

The CC maintains the AIMIB, which is used to record the already-matched pairs among applications and ISPs. Thus, AIMIB can serve as a kind of reputation records, according to which each ISP's market share for each type of application can be obtained (that is, the ratio of

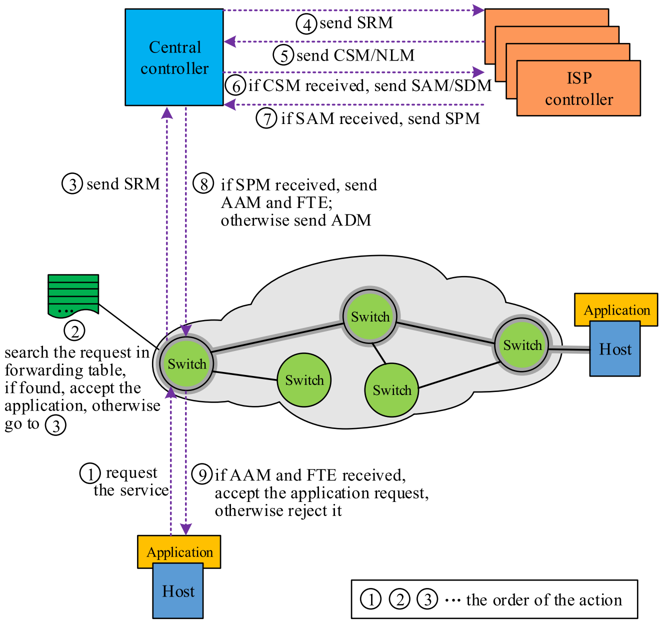

Central controller
ISP controller
④send SRM
⑤send CSM/NLM
⑥if CSM received, send SAM/SDM
⑦if SAM received, send SPM
③send SRM
⑧if SPM received, send AAM and FTE; otherwise send ADM
②
search the request in forwarding table, if found, accept the application, otherwise go to ③
Switch
Switch
Switch
Switch
Switch
Application
Host
①request the service
⑨if AAM and FTE received, accept the application request, otherwise reject it
Application
Host
① ② ③ ··· the order of the action

Fig. 2. The routing service provision workflow of ARSC.

the successfully matched number for one type of application by the ISP to the total successfully matched number of this type of application by all ISPs). The working process of the CC is shown in Algorithm 1.

The IC is in charge of composing candidate routing services in its domain. It updates its network RIB periodically according to its VNV. When SRM received, the IC executes the ACC to see whether the network resources are sufficient to accommodate the application request based on RIB. If yes, the IC generates the candidate routing service composition plan with the SCC, determines the service price and calculates the ISP's profit with the SPC, and then sends CSM to the CC; otherwise, it sends NLM to the CC. If the CC responds with SAM, the IC prepares its matched candidate service and sends SPM to the CC; if the CC responds with SDM, the IC cancels its candidate service. The working process of the IC is shown in Algorithm 2.

The switch deals with the application request according to the forwarding table distributed by the CC. The switch looks up the FTE to deal with the request in the forwarding table. If it finds the matched FTE, it accepts the application request directly and selects the necessary functions to compose routing services; otherwise, it sends SRM to the CC and waits for the response (i.e. AAM and FTE, or ADM). If the switch receives AAM and FTE, it accepts the application request and deals with it according to the received FTE; otherwise, that is, the switch receives ADM, it rejects the application request.

The routing service provision workflow of the proposed ARSC is shown in Fig. 2. In the proposed SDN and NFV based framework, the communication process can support the existing IP network protocols, such as TCP, UDP, TLS, and DTLS. The central controller and the ISP controller exchange messages, such as SRM, CSM, SAM and SPM, through their eastbound and westbound interfaces. The central controller communicates with the switch through its southbound interface. They exchange information through messages, such as SRM, AAM and ADM, by the *packet_in* and *packet_out* operations. Taking the OpenFlow as the example, the FTE based on it in this paper is shown in Fig. 3.

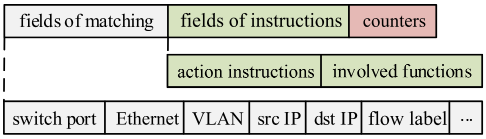


Fig. 3. The example of FTE.

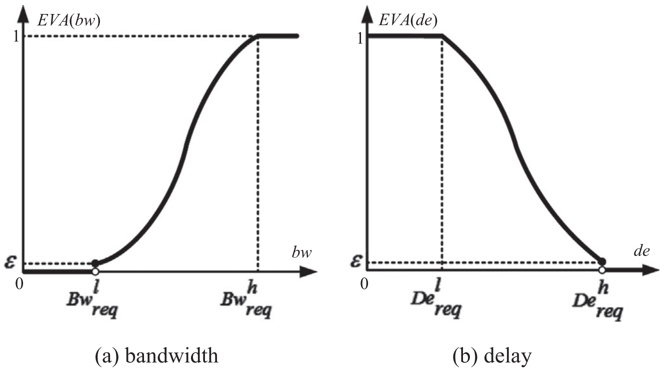
4. The components of the proposed ARSC

4.1. User utility

There are always multiple ISPs which can provide routing services demanded by the applications. It is challenging to choose which ISP to compose the most suitable service for the specific application request. In this paper, we develop a user utility scheme, which considers the following three metrics: the user satisfaction degree to the routing QoS, the user selectivity degree to the ISP, and the user acceptance degree to the routing service price.

Generally, the user can only clearly indicate his subjective requirements on the network provided QoS, that is, he mainly concerns his Quality of Experience (QoE) (Wang et al., 2015b). He can clearly express his qualitative requirements, for example, HDTV video, however, he usually does not have expertise to map such qualitative requirements into the accurately quantitative values of QoS parameters. For example, how much bandwidth is exactly required for HDTV to be transferred across the network is too hard for the common user to know. Considering that the user usually cannot express such quantitative requirements for QoS parameters accurately (Wang et al., 2014) and his judgments on things always follow the Gaussian distribution (Platenius et al., 2013), we use intervals to denote QoS parameters and use the Gaussian fuzzy membership functions (Hasan et al., 2014) to deal with their fuzziness to be expressed. In particular, we use $[Bw_{req}^l, Bw_{req}^h]$,$[De_{req}^l, De_{req}^h]$,$[Jit_{req}^l, Jit_{req}^h]$ and $[Err_{req}^l, Err_{req}^h]$ to specify the bandwidth, delay, delay jitter and error rate requirement of the user desired routing service. We denote the actual bandwidth, delay, delay jitter and error rate of the route provided by the ISP as *bw*, *de*, *jit* and *err*. Then, we define the user's evaluations on the actually received QoS as follows:

$$EVA(bw) = \begin{cases} 0, & bw < Bw_{req}^l \\ \varepsilon, & bw = Bw_{req}^l \\ e^{-(Bw_{req}^h - bw)^2/(bw - Bw_{req}^l)^2}, & Bw_{req}^l < bw < Bw_{req}^h \\ 1, & bw > Bw_{req}^h \end{cases} \tag{1}$$



(a) bandwidth (b) delay

Fig. 4. The user evaluation on the bandwidth and delay.

Table 2
The floating price.

Floating price	Delay	Delay jitter	Error rate
$FloPri_1$	de_1	jit_1	err_1
$FloPri_2$	de_2	jit_2	err_2
…	…	…	…
$FloPri_n$	de_n	jit_n	err_n

$$EVA(de) = \begin{cases} 1, & de \le De_{req}^{l} \\ 1 - e^{-(De_{req}^{h} - de)^{2}/(de - De_{req}^{l})^{2}}, & De_{req}^{l} < de < De_{req}^{h} \\ \varepsilon, & de = De_{req}^{h} \\ 0, & de > De_{req}^{h} \end{cases} \tag{2}$$

$$EVA(jit) = \begin{cases} 1, & jit \le Jit_{req}^{l} \\ 1 - e^{-(Jit_{req}^{h} - jit)^{2}/(jit - Jit_{req}^{l})^{2}}, & Jit_{req}^{l} < jit < Jit_{req}^{h} \\ \varepsilon, & jit = Jit_{req}^{h} \\ 0, & jit > Jit_{req}^{h} \end{cases} \tag{3}$$

$$EVA(err) = \begin{cases} 1, & err \le Err_{req}^{l} \\ 1 - e^{-(Err_{req}^{h} - err)^{2}/(err - Err_{req}^{l})^{2}}, & Err_{req}^{l} < err < Err_{req}^{h} \\ \varepsilon, & err = Err_{req}^{h} \\ 0, & err > Err_{req}^{h} \end{cases} \tag{4}$$

Here, $0 < \varepsilon < < 1$, $EVA(bw)$, $EVA(de)$, $EVA(jit)$ and $EVA(err)$ are the user's evaluations on the actually received bandwidth, delay, jitter and error respectively. Apparently, the more the actual bandwidth received, the less the actual delay, delay jitter and error rate experienced, the higher the user's evaluations on them. Fig. 4(a) and (b) show the user evaluation curves on bandwidth and delay respectively, and the shapes of the user evaluation curves on delay jitter and error rate are similar to that of Fig. 4(b). When the actually received bandwidth by the user approaches the lower bound of the requirement, and the delay, delay jitter and error rate approach the upper bounds of the requirements, the user's evaluation values on them tends to ε. Once the actual QoS cannot satisfy the user's bottom-line requirement, the corresponding evaluation value drops to 0.

Based on the above, we define the user satisfaction degree to the routing QoS as follows:

$$\begin{aligned} SaDeg_{QoS} = {} & \omega_{bw} \cdot EVA(bw) + \omega_{de} \cdot EVA(de) \\ & + \omega_{jit} \cdot EVA(jit) + \omega_{err} \cdot EVA(err) \end{aligned} \tag{5}$$

Here, ω_{bw}, ω_{de}, ω_{jit}, and ω_{err} indicate the relative importance of bandwidth, delay, delay jitter, and error rate to the user QoS requirement, $0 \le \omega_{bw}, \omega_{de}, \omega_{jit}, \omega_{err} \le 1$, $\omega_{bw} + \omega_{de} + \omega_{jit} + \omega_{err} = 1$.

The user selectivity degree to the specific ISP is mainly related to the user's loyalty to the ISP and the ISP's market share (Chiou, 2004), denoted as ULo and $ISPMS$ respectively,$0 \le ULo \le 1$, $0 \le ISPMS \le 1$. The higher the value of ULo, the more loyal the user to the ISP. We define the user selectivity degree to ISP as follows:

$$SeDeg_{ISP} = ISPMS + (1 - ISPMS) \cdot ULo \tag{6}$$

According to (6), as ULo or $ISPMS$ approaches 1, the user selectivity to the ISP approaches 1, and when ULo or $ISPMS$ is 1, the user selectivity reaches its maximum, that is, 1. When the user's loyalties to different ISPs are the same, the ISP with high market share will be selected by high probability. In the meantime, if the market shares of the ISPs are the same, the ISP to whom the user has the high loyalty will be selected by high probability.

The user acceptance degree to the routing service price is related to the offered service price, the user expected price and the highest user acceptable price, which are denoted as $RSPri$,$EPri$ and $HAPri$ respectively, $EPri < HAPri$. We define the user acceptance degree to the routing service price as follows:

$$ADeg_{Pri} = \begin{cases} 1, & RSPri \le EPri \\ 1 - \frac{RSPri - EPri}{HAPri - EPri}, & EPri < RSPri < HAPri \\ \varepsilon, & RSPri = HAPri \\ 0, & RSPri > HAPri \end{cases} \tag{7}$$

Here, $0 < \varepsilon < < 1$. If $RSPri$ is higher than $HAPri$,$ADeg_{Pri}$ is 0, that is, the user will not accept the offered routing service; if $RSPri$ is less than or equal to $EPri$, the user acceptance degree to the routing service reaches the maximum; otherwise, the lower the $RSPri$, the higher the user acceptance degree to the routing service.

The user utility is defined as follows:

$$Uti = \omega_{QoS} \cdot SaDeg_{QoS} + \omega_{ISP} \cdot SeDeg_{ISP} + \omega_{Pri} \cdot ADeg_{Pri} \tag{8}$$

Here, ω_{QoS}, ω_{ISP} and ω_{Pri} are the weights of the three metrics which reflect the relative significance of QoS, ISP and price to the user, $0 \le \omega_{QoS}, \omega_{ISP}, \omega_{Pri} \le 1$,$\omega_{QoS} + \omega_{ISP} + \omega_{Pri} = 1$. All candidate routing services can be sorted by their user utility values in descending order, and then the sorting result is sent to the Service Matching Component (SMC).

4.2. Status measurement and admission control

With virtual plane between control plane and data plane, it is easy for the CC to periodically measure network status. In this paper, we leverage the method of Link Layer Discovery Protocol (LLDP) according to Tarnaras et al. (2015) as the switch port discovery protocol. The CC exchanges messages (such as *packet_in* message and *packet_out* message) with switches. In this way, according to the information collected by the LLDP, the CC obtains the global network view. Based on each ISP's rented network resources, the CC provides the VNV for each IC to update the RIB, which contains node status (CPU utilization, buffer utilization, etc.) and link status (available bandwidth, delay, delay jitter, and error rate, etc.).

In the proposed ARSC, the paths that can support the bandwidth requirements of the applications are found at first, then ACC judges whether these paths can satisfy the applications' delay, delay jitter and error rate requirements according to its RIB, finally decides whether to accept or reject the application requests.

4.3. Service composition and pricing

The NFV-enabled switch has been equipped with various function components, such as traffic shaping, queue scheduling, error control, congestion avoidance, and queue management in addition to packet forwarding (shown in Fig. 1). We assume that these function components have already been well modularly designed with interfaces standardized, which can be combined together as the service chain (Wang et al., 2015a). According to the service composition strategies of the ISPs, even composed on the same transmission path, the differently composed routing services can provide different QoS to applications. Therefore, the diversified routing services can be provided along the same transmission path to different applications, and the same routing service can present different QoS along different transmission paths. The ISP gets the QoS requirements, then the SCC allocates the necessary resources and function components to compose the routing services for the applications based on the current network status.

The routing service price affects the user selectivity to the specific ISP. The high price may increase the ISP profit, but will reduce the number of the users who are willing to use it. ISP should provide the service price reasonably to optimize its profit. In this paper, the service price consists of two parts: the fixed price and the floating price. The former refers to the base price, which is unchangeable within a certain period, for example, the base price of unit bandwidth; the latter is changeable and paid for the service customization to satisfy the user-specific requirement. In general, the lower the committed delay, delay jitter, and error rate, the higher the floating price asked by the ISP. In the proposed ARSC, the price of the routing service is calculated by the SPC.

Assume that the bandwidth has the fixed price $FixPri$. The floating price $FloPri$ depends on the committed delay, delay jitter, and error rate, and can be set and then used in Table 2.

The routing service price of the ISP is defined as follows:

$$RSPri = FixPri + FloPri_m\ ,\ 1 \le m \le n \tag{9}$$

Let $RSCo$ be the cost of the routing service, the profit of the ISP is defined as follows:

$$SP = RSPri - RSCo \tag{10}$$

5. The applications and the services matching

When an application request arrives, there are always multiple ISPs which can provide the candidate services according to their VNVs of their rented network resources, and the user always prefers the one with high performance and reasonable price. Meanwhile, the ISP always tends to provide services to the applications which can maximize its profit when multiple application requests of the same type arrive. Indeed, it is a kind of matching behavior among multiple applications and multiple candidate services. With the Pareto efficiency introduced, we propose an efficient matching algorithm to optimize the matching results and reduce the runtime overhead. The CC is in charge of the matching process by AMC and SMC.

Assume that in a time slot there are multiple application requests arrived simultaneously and we denote them as the set $APP = \{App_1, \ldots, App_n\}$. The ISPs which can provide services are denoted as the set $ISP = \{ISP_1, \ldots, ISP_m\}$ and the routing services which the ISPs can provide to applications are denoted as the set $Ser = \{Ser_1, \ldots, Ser_q\}$ at this slot. Sort the services in Ser with their utility Uti for App_i in descending order as $Ser^{APPi} = \{Ser_1^{Appi}, \ldots, Ser_q^{Appi}\}$.

In order to get the optimal matched pairs among multiple applications and multiple services, we introduce the Pareto efficiency as the optimal condition for each matching round to reach the benefit equilibrium among applications and ISPs. Assume that AU_i^{Serk} is the utility when the application App_i selects service Ser_k and the highest one is denoted as AU_i^{max} when App_i is matched with its most favorite service. Assume that SP_k^{Appi} is the utility when Ser_k selects App_i and the highest one is denoted as SP_k^{max} when Ser_k is matched with the application which brings the ISP the highest profit. We define $SDeg_i^{Serk}$ and $SDeg_k^{Appi}$ as the satisfaction degree of the matched pair App_i and Ser_k respectively as follows:

Algorithm 3.

```
Input: $APP$, $ISP$, $Ser$
Output: AMMs and FTEs
Define $MPs$ as the matched pair set, $MPs[j] = \left(App_j^{MPs}, Ser_j^{MPs}\right)$
    denotes its jth element, $App_j^{MPs} \in APP$, $Ser_j^{MPs} \in Ser$.
Begin
  while $(APP \ne \emptyset)\&\&(Ser \ne \emptyset)$ do
    Initialize $MPs$ into an empty array;
    for each $App_i$ in $APP$ do
      Update $Ser^{APPi}$ according to current $Ser$;
      Select the first service from $Ser^{APPi}$;
      Send the Service Identifier (SID) of the selected
          service to the involved ISP;
    end for
    for each involved ISP do
       Add the received SIDs into the set of the candidate
           service (CS);
       if the ISP cannot support all services in CS then
          for each service in CS do
             Calculate its profit according to (10);
          end for
          Sort the services in CS in descending order
             according to their profits;
          Choose the in-the-front services that can be
                supported by the ISP in CS;
          Remove the remaining services from CS;
       end if
     end for
     March applications and services according to CS;
     Add the matched pairs into $MPs$;
     for $j \in \{1, \ldots, |MPs|\}$ do
       Calculate the $PE$ of $MPs[j]$ according to (13);
       for each unmatched application $App_z$ do
         if the second service in $Ser^{APPz}$ is $Ser_j^{MPs}$ then
           if ( $(SDeg_{Ser_j^{MPs}}^{Appz} \ge SDeg_{Ser_j^{MPs}}^{App_j^{MPs}})$ &
&             $(PE(App_z, Ser_j^{MPs}) > PE\left(App_j^{MPs}, Ser_j^{MPs}\right))$ ) then
                Replace $App_j^{MPs}$ by $App_z$ in $MPs[j]$;
              end if
           end if
         end for
       end for
     Send AAMs and FTEs to the involved switches
       according to the $MPs$;
     Remove the applications and services in the $MPs$
       from $APP$ and $Ser$;
  end while
end
```

$$SDeg_{Appi}^{Serk} = \frac{AU_i^{Serk}}{AU_i^{max}} \tag{11}$$

$$SDeg_{Serk}^{Appi} = \frac{SP_k^{Appi}}{SP_k^{max}} \tag{12}$$

The Pareto efficiency over $\left(SDeg_{Appi}^{Serk}, SDeg_{Serk}^{Appi}\right)$ is denoted as *PE* and defined as follows:

$$PE(App_i, Ser_k) = \omega_{Appi}\cdot SDeg_{Appi}^{Serk} + \omega_{Serk}\cdot SDeg_{Serk}^{Appi} \tag{13}$$

Here,ω_{Appi} and ω_{Serk} are the weights of $SDeg_{Appi}^{Serk}$ and $SDeg_{Serk}^{Appi}$ respectively,$0 \le \omega_{Appi}, \omega_{Seri} \le 1$,$\omega_{Appi} + \omega_{Seri} = 1$. Apparently, the higher value of *PE* means the better benefit equilibrium of the matched user and ISP pair (Chen, 2012).

The proposed matching algorithm is executed by AMC and SMC, and is described in Algorithm 3. The applications select services (lines 4–8) and the involved ISPs decide service provisions (lines 9–19), then the initial matched pairs are obtained. With the Pareto efficiency considered, the necessary rematch process is done to optimize the matching results, and the matched pairs under benefit equilibrium are obtained and they do not take part in the next matching round in order to improve matching efficiency (lines 22–31). After each round of matching (line 33), the CC allocates resources and assembles functions to provide routing services for the already-matched applications.

Theorem. *The matching algorithm (i.e. Algorithm 3) can achieve the optimal matching result. That is, the matching result always reaches the optimal equilibrium for the matched applications and services under the current network status.*

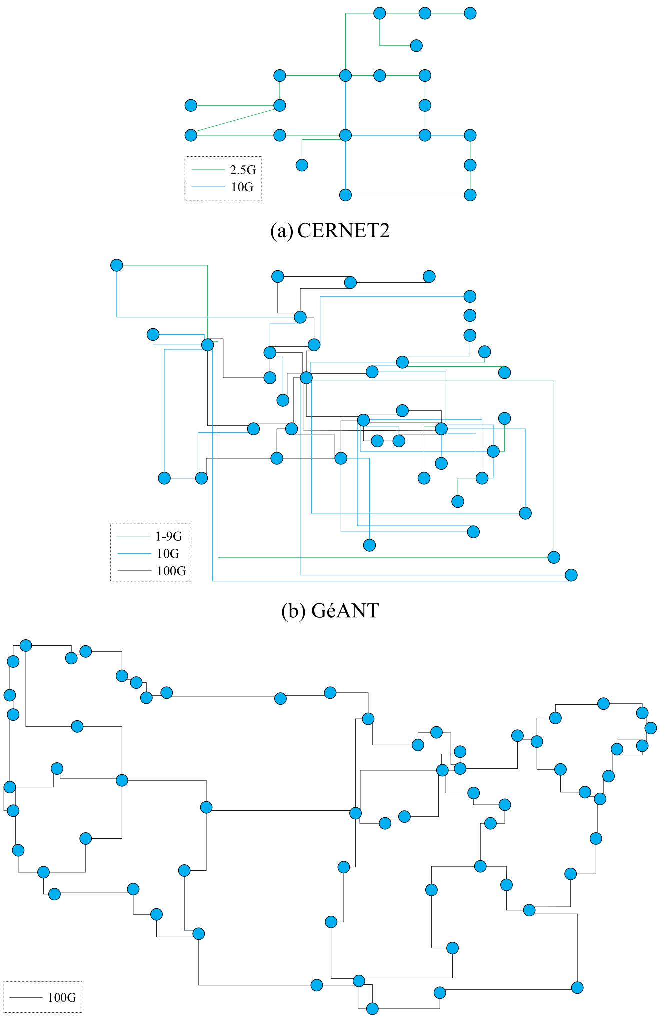
Proof. Assume that the initial *MPs* is defined as *IMPs* (line 21) and the rematch *MPs* is defined as *RMPs* (line 32). In *RMPs*, assume that the set of the matched applications is defined as *MAPP* $=\{MApp_1, \ldots, MApp_a, \ldots, MApp_b\}$, $MAPP \subseteq APP$, and the set of the matched services is defined as $MSer = \{MSer_1, \ldots, MSer_a, \ldots, MSer_b\}$, $MSer \subseteq Ser$. Thus, $MApp_a$ and $MSer_a$ are one of the matched pairs in *RMPs*. If $MApp_a$ is also one of the matched applications in *IMPs*, it is obviously that $MSer_a$ is the best service to $MApp_a$ (lines 5–7); otherwise, the best service to $MApp_a$ cannot be supported by the ISP (lines 15–17), and $MApp_a$ becomes the best service to $MApp_a$ (line 25) under the current network status. For $MApp_a$, there must be $\forall\ Ser_j \in Ser, SDeg_{MApp_a}^{MSer_b} \ge SDeg_{MApp_a}^{Ser_j}$ as well as $\forall\ Ser_j \in Ser$, $PE(MApp_a, MSer_b) \ge PE(MApp_a, Ser_j)$. $MApp_a$ is already one of the chosen services by ISP (lines 15–17), but its profit may not be optimal. Then, the rematch process is done to further optimize the profit of $MSer_a$ under the current network status (lines 22–31). For $MApp_a$, there must be $\forall\ App_i \in APP, SDeg_{MSer_b}^{MApp_a} \ge SDeg_{MSer_b}^{App_i}$ as well as $\forall\ App_i \in APP, PE(MApp_a, MSer_b) \ge PE(App_i, MSer)$.

Therefore, the proposed matching algorithm can get the optimal result, since in each matching round the matched applications always get the best services corresponding to them. Furthermore, those early matched applications get preferred services early without taking part in the next matching round, and these matched services can always bring the best profits to ISPs under the current network status.

6. Simulation and performance evaluation

To simulate the proposed ARSC, we choose Floodlight (Project Floodlight, 2015) as the controller in the control plane. It generates FTEs and distributes them to the switches on application communication paths. The FTE contains fields of matching and fields of instructions. The former is used to match requests for routing services. The latter contains action instructions (e.g. forward packets to ports, encapsulate packets and forward them to controller, send packets to CRSs, drop packets, etc.).

We choose OpenFlowClick (OpenFlowClick, 2011) to simulate the NFV-based switch, which is software-based and created based on Click modular router (Kohler et al., 2000) with OpenFlowClick element (Mundada and Rob Sherwood, 2012) embedded into it. In fact,



(a) CERNET2

(b) GéANT

(c) INTERNET2

Fig. 5. The network topologies.

Table 3
Characteristics of the three topologies.

	TNN	TNL	NLL	NLB	NLM
CERNET2	20	22	4	18	0
GéANT	41	65	8	30	27
INTERNET2	64	78	0	0	78

Table 4
Examples of applications.

	Non-interactive	Interactive
Elastic	E-mail(EM)	Web browsing(WB)
Non-elastic	Video on demand(VD)	Video teleconference(VT)

Table 5
Parameter settings.

	ω_{bw}	ω_{de}	ω_{jit}	ω_{err}	ω_{QoS}	ω_{Pri}	ω_{ISP}	ω_{Appi}	ω_{Serk}
EM	0.54	0.06	0.06	0.34	1/3	1/3	1/3	0.5	0.5
WB	0.26	0.35	0.04	0.35	1/3	1/3	1/3	0.5	0.5
VD	0.43	0.06	0.22	0.29	1/3	1/3	1/3	0.5	0.5
VT	0.12	0.37	0.37	0.14	1/3	1/3	1/3	0.5	0.5

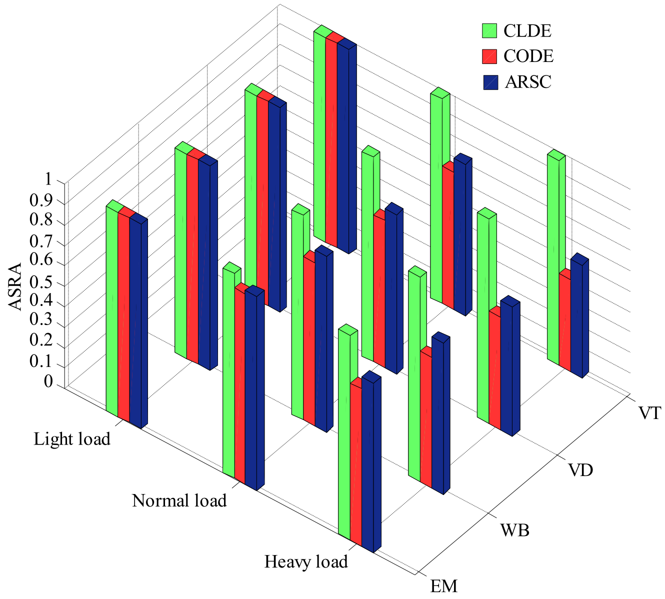


Fig. 6. ASRA over CERNET2.

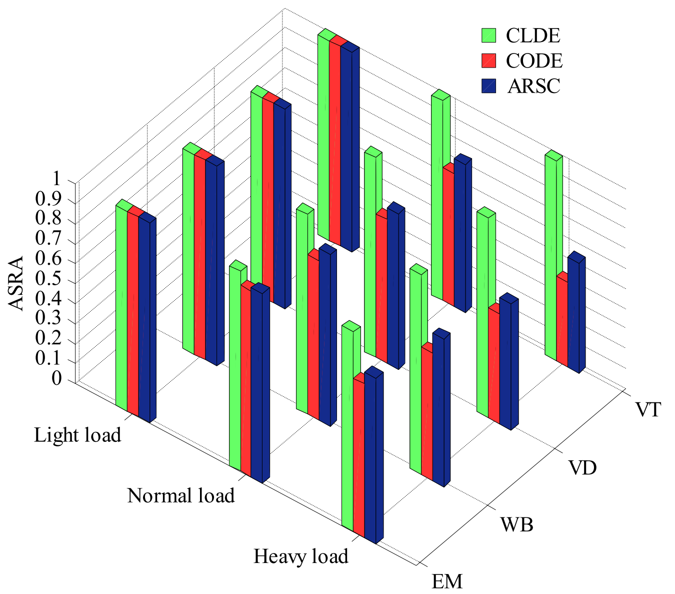


Fig. 8. ASRA over INTERNET2.

OpenFlowClick is extendable, programmable and assembled by a series of packet processing modules called elements which can be flexibly selected and added. In addition, the OpenFlowClick element allows a controller to install FTEs to guide packet processing through multiple network functions which serve as elements in Click modular router. It also allows multiple Click modular routers being controlled by one single controller.

The simulation environment is established on Linux platform with 3.3 GHz Intel core i5 and 16 GB DDR3 RAM. In order to evaluate the adaptability and stability of the proposed ARSC under different network topologies, we adopt three typical and real network topologies, which are CERNET2 (CERNET2 Topol., 2006), GéANT (The Internet Topol., 2013), and INTERNET2 (INTERNET2 Netw., 2014) as shown in Fig. 5. They have different characteristics, such as the Total Number of Nodes (TNN), the Total Number of Links (TNL), the Number of Links with bandwidth Less than 10 G (NLL), the Number of Links with bandwidth Between 10 G and 100 G (NLB), and the Number of Links with bandwidth More than 100 G (NLM), which are shown in Table 3. Packet forwarding, traffic shaping, error control, transcoding, content caching, packet scheduling, queue management functions are considered, which are component based and embedded into switches. These functions can be chosen and composed by the ICs to customize the special routing services for the arrived application requests. We select four typical applications based on their elasticity and interactivity according to (Stankiewicz et al., 2011), shown in Table 4. When do simulation, each type of application requests are randomly generated. The values of parameters used in simulations are set as shown in Table 5. In particular, we consider the weights of the parameters ω_{bw},ω_{de},ω_{jit} and ω_{err} based on the quality standard bounds (Network performance objectives, 2011; Definitions of terms, 2008) and set their values by the method of (Ahuja et al., 2014); we assume that the relative importance of QoS, ISP and price to the user is the same, and then we set $\omega_{QoS} = \omega_{Pri} = \omega_{ISP} = 1/3$; according to the purpose of benefit equilibrium between the user and ISP, we set $\omega_{Appi} = \omega_{Serk} = 0.5$. In order to simulate the real network traffic, we use network traffic traces according to (Gebert et al. (2012), and periodically collect the run-time status information by the method of Tarnaras et al. (2015).

For comparison purposes, we also simulate two other routing

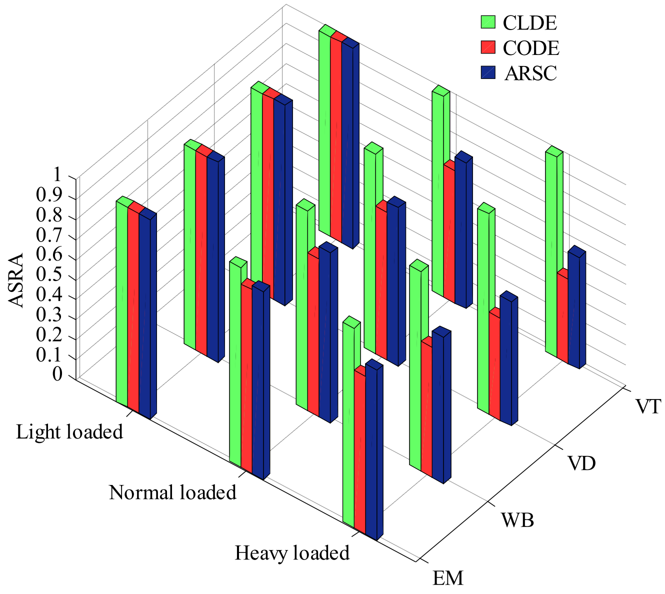


Fig. 7. ASRA over GéANT.

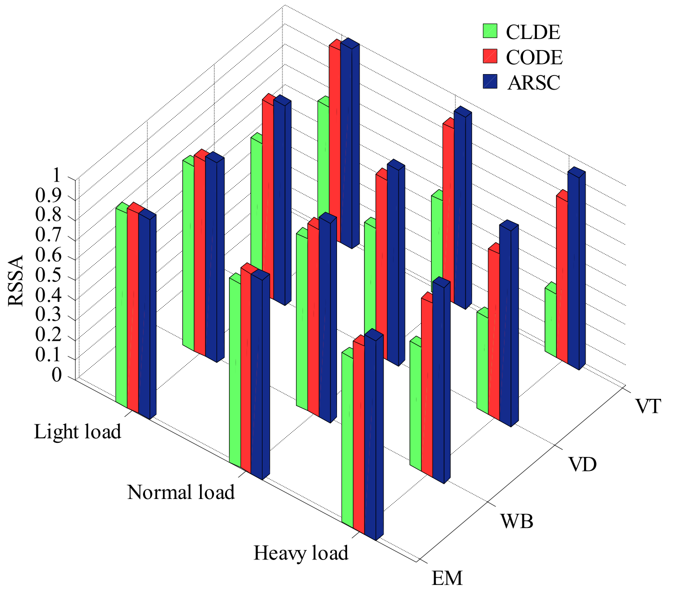


Fig. 9. RSSA over CERNET2.

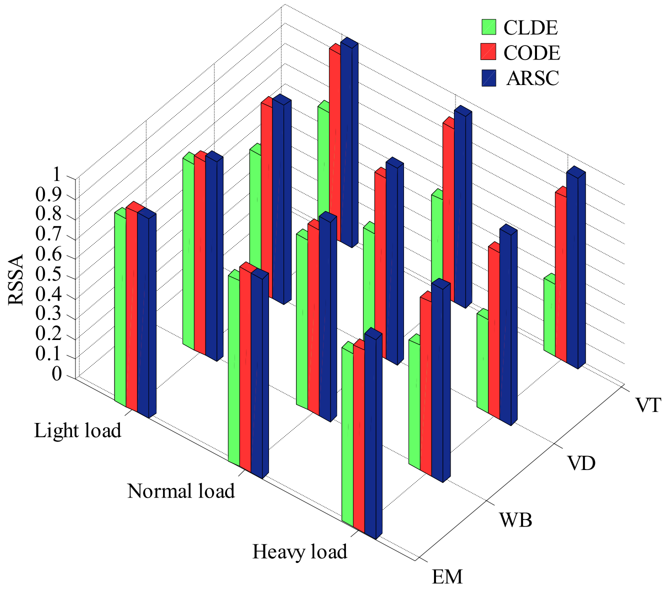


Fig. 10. RSSA over GéANT.

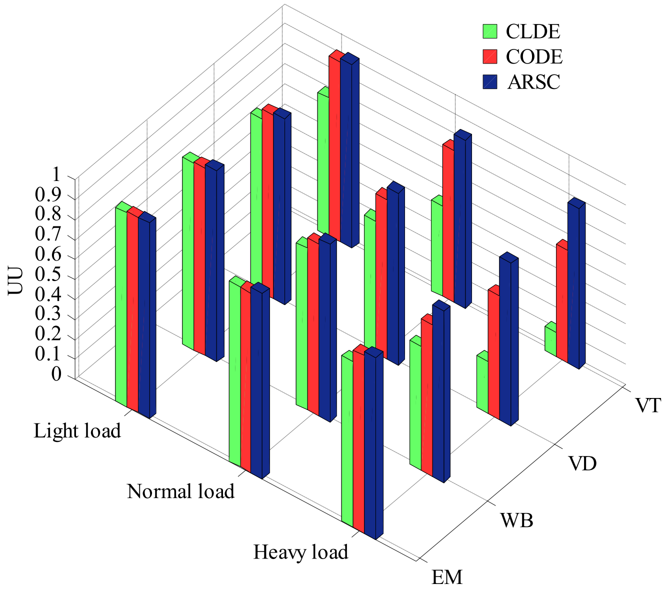


Fig. 12. UU over CERNET2.

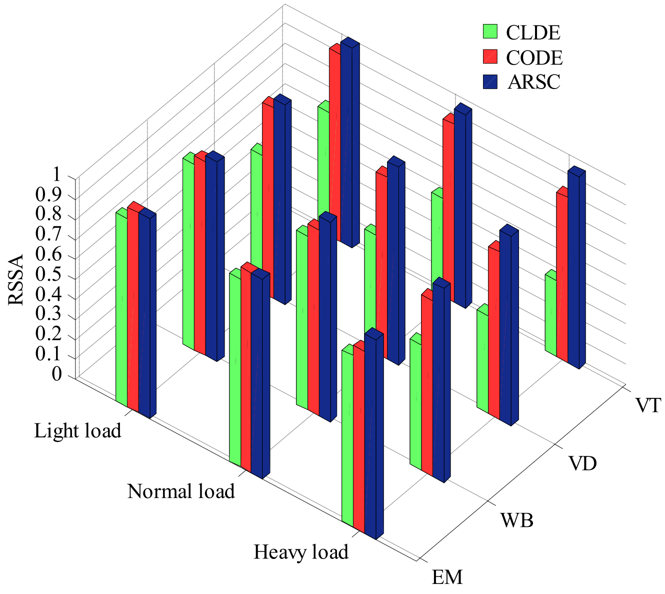


Fig. 11. RSSA over INTERNET2.

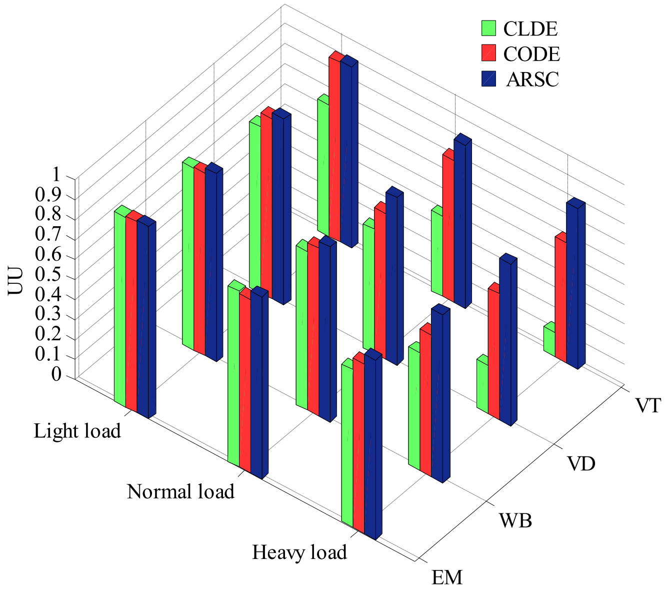


Fig. 13. UU over GéANT.

mechanisms: Connection-Oriented Data Exchange mechanism (CODE) and Connectionless Data Exchange mechanism (CLDE). CODE is simulated based on the framework of OpenSCaaS (Ding et al., 2015), and its supported QoS services are set according to the IntServ method. CLDE uses the typical best effort model and Dijkstra algorithm for routing. We conducted simulation experiments to compare ARSC, CODE, and CLDE under different network conditions, i.e., light load (the bandwidth utilizations of all links in the networks are below 30%), heavy load (the bandwidth utilizations of all links in the networks are above 70%), and normal load (otherwise). We use the following performance metrics: Access Success Ratio of the Application (ASRA), Routing Service Suitability to the Application (RSSA), User Utility (UU), ISP Profit (SP), Pareto Efficiency of the matched user and ISP pair (PE), and Runtime Overhead (RO).

6.1. Access success ratio of the application

ASRA is the ratio of the number of the successfully accessed application requests to the total number of the arrived application requests. Under the three different network conditions and with four types of applications, we compare the three mechanisms in terms of ASRA and the results are shown in Figs. 6–8.

The ASRA of CLDE is the highest, followed by ARSC and CODE. This is because CLDE only provides the best-effort service without admission control, even an already-accessed application may not be provided the effective service. Although its ASRA is always approaching 1, it cannot ensure that the accessed applications get their desired services. However, ARSC and CODE both can guarantee the provision of routing services for all the accessed applications, so that an application that has been accepted can always get the service, and their ASRAs decrease with the increase of the network load. The ASRA of ARSC is always higher than that of CODE, especially for non-elastic applications when the network load increases. Due to the video traffic consuming a big chunk of bandwidth resource nowadays, we take VD and VT as examples, the gap between ARSC and CODE are 4.5% and 5.6% under normal load respectively while 7.1% and 9.7% under heavy load respectively. The main reason is that ARSC can effectively provide more feasible accessed opportunities for applications by its adaptive

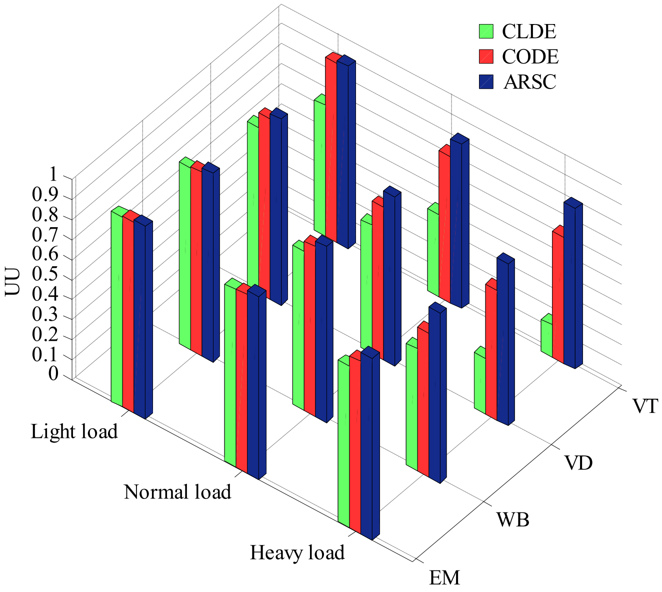


Fig. 14. UU over INTERNET2.

mechanism. It can choose the suitable route to avoid those unqualified links and nodes, which improves the ASRA. In addition, with the competition among multiple ISPs, an application request can get multiple responses, the central controller can dynamically select the appropriate one according to the current network situation to balance the traffic at the same time increase the ASRA of ARSC further for future coming requests.

6.2. Routing service suitability to the application

RSSA is the probability of the ISP-provided routing service satisfying the application requirements. The comparison results under different settings are shown in Figs. 9–11.

The RSSA of ARSC is the highest, followed by CODE and CLDE. Under CODE and CLDE, the RSSA decreases with the increase of the network load, especially for the non-elastic applications. For example, the gap between ARSC and CODE for VD and VT are 1.4% and 2.5% under light load respectively, 3.3% and 4.1% under normal load respectively, while 7.1% and 9.7% under heavy load respectively. The reasons are as follows. At first, ARSC accepts an application request only if the available network resources can satisfy the application requirements. Secondly, ARSC does not provide general-purpose service, instead it adaptively allocates resources and assembles function components to customize the specific routing service according to the network status and the application requirements. Thirdly, in ARSC the competition among different ISPs always facilitates the most suitable services to be selected by the applications. Although CODE can provide quality guaranteed services, these services are mainly general-purpose for applications. Furthermore, CODE and CLDE are incapable of adaptively adjusting the service configuration according to the changing network status.

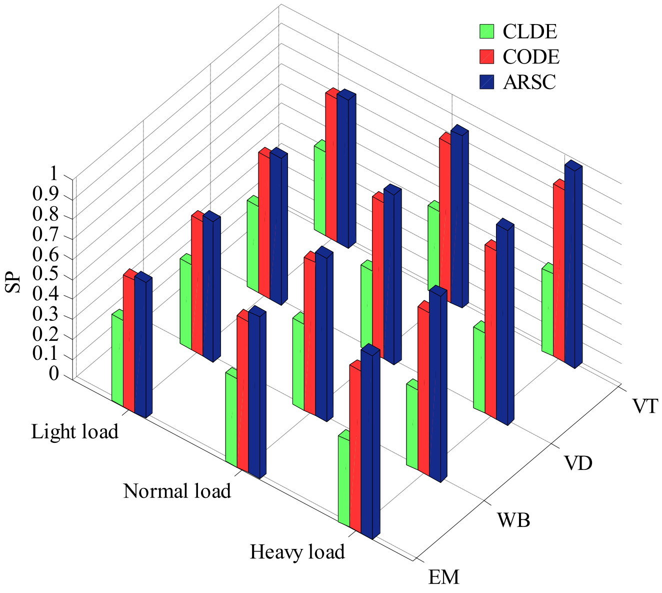


Fig. 15. SP over CERNET2.

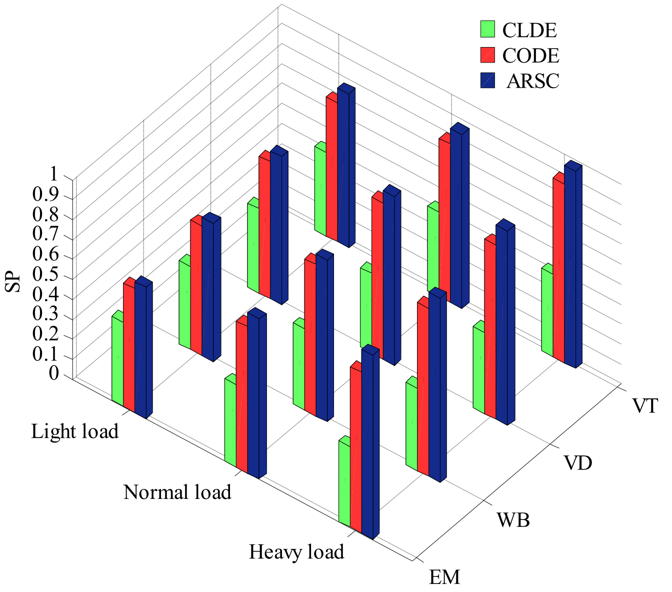


Fig. 16. SP over GéANT.

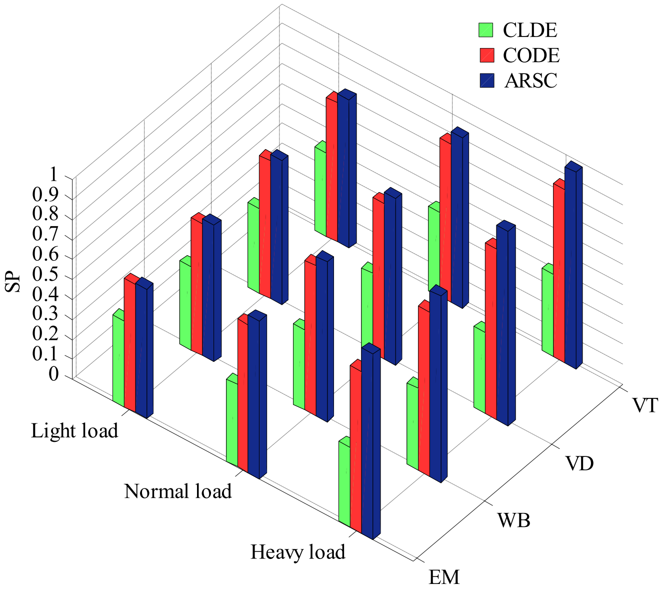


Fig. 17. SP over INTERNET2.

6.3. User utility

We compare the UUs of the four types of applications under ARSC, CODE and CLDE, and the results are shown in Figs. 12–14.

The UU under ARSC is much higher than those under CODE and CLDE, especially for the non-elastic applications when the network

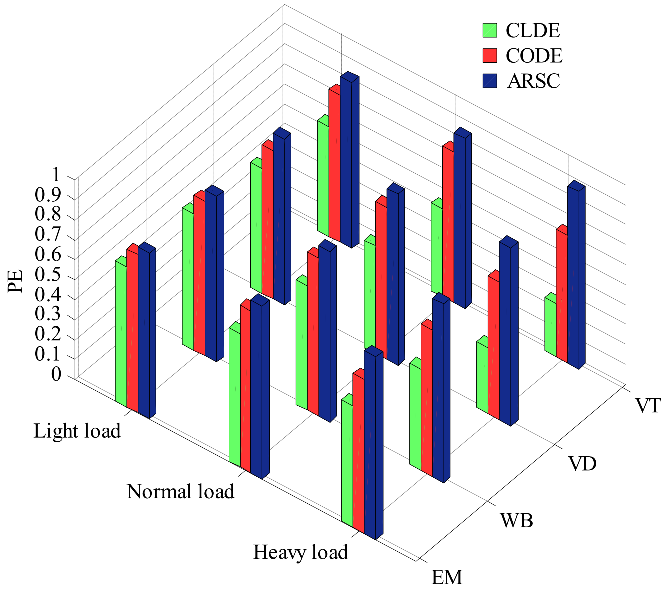


Fig. 18. PE over CERNET2.

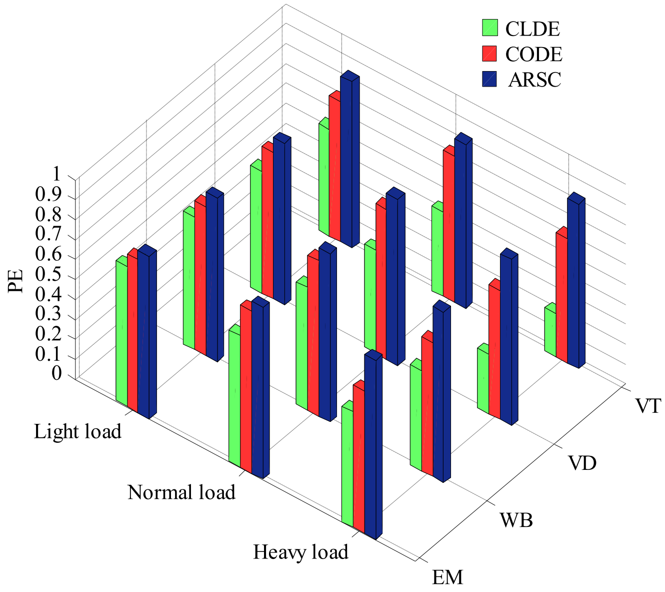


Fig. 19. PE over GéANT.

load increases. For example, the gap between ARSC and CODE for VD and VT are 1.7% and 2.6% under light load respectively, 7.3% and 8.1% under normal load respectively, while 14.1% and 18.2% under heavy load respectively. The reasons are as follows. Firstly, ARSC can provide special-purpose services by accurately analyzing the user primary demands (i.e., the corresponding QoS parameters) for different types of applications. Secondly, when providing the service, ARSC not only considers the user experience of service quality, but also takes the user experience of service price into account, which improves the user overall satisfaction on the customized service. However, CODE and CLDE do not focus on the user specialized and personalized quality requirements for different applications. And they either do not take the effect of the service price factor into account when providing services to different users, which also affects the UU.

6.4. *ISP profit*

We compare the SPs of the four types of applications under ARSC, CODE and CLDE, and the results are shown in Figs. 15–17.

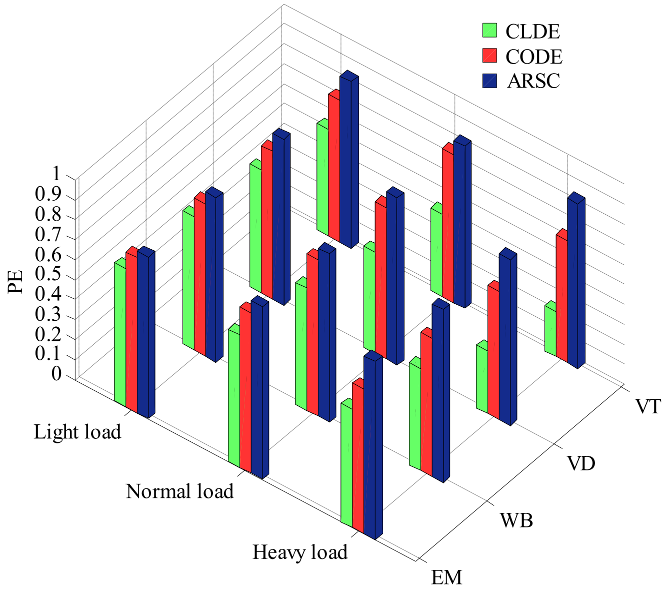


Fig. 20. PE over INTERNET2.

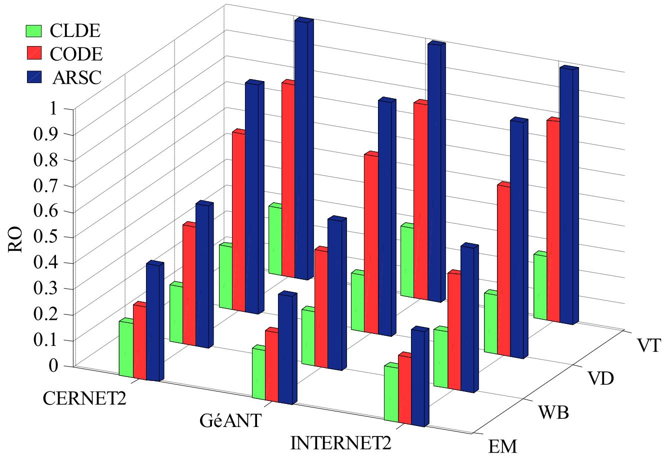


Fig. 21. RO over CERNET2, GéANT and INTERNET2.

We can see that the SP under ARSC is higher than those under CODE and CLDE, especially for the non-elastic applications when the network load increases. For example, the gap between ARSC and CODE for VD and VT are 2.8% and 3.5% under light load respectively, 5.1% and 7.3% under normal load respectively, while 10.1% and 11.2% under heavy load respectively. The main reasons are as follows. Firstly, in ARSC, each ISP can dynamically allocate its rented network resource for its provided services with the floating price considered according to its available resource. Secondly, each ISP can adaptively select its own functions to compose the customized services at the reasonable costs, which can effectively improve its profits. Thirdly, although there is competition among multiple ISPs, high quality service with high price still can attract the users who prefer good service experience. However, both CODE and CLDE only provide the standardized routing service, they do not consider to increase the ISP profit with the floating price according to the current network status. Furthermore, the non-customized services provided by them cannot make the user be willing to pay more even under the heavy network load.

6.5. *Pareto efficiency*

The PE of the matched user and ISP pair considers the benefit win-win of the user and the ISP. We compare the PEs of the four types of applications under ARSC, CODE and CLDE, and the results are shown

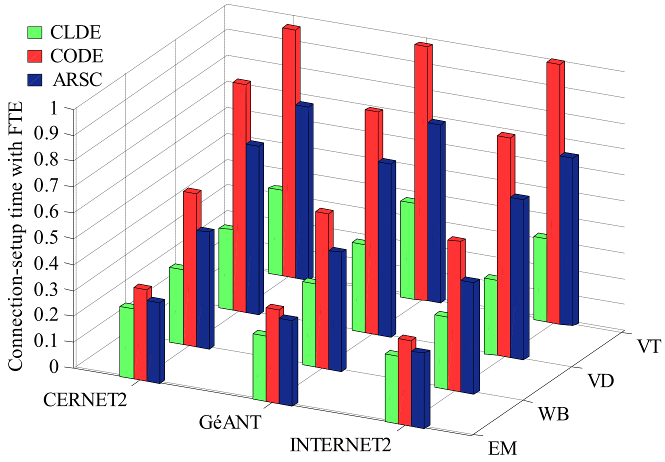


Fig. 22. Connection-setup time according to the matched FTE.

in Figs. 18–20.

The PE of ARSC is the highest and is more stable than that of CODE and CLDE. For example, the PE value of ARSC is between 85.6% and 91.5%, the PE value of CODE is between 61.7% and 81.3% and the PE value of CLDE is between 26.6% and 70.1%. In ARSC, the service selections are bidirectional choices between the users and the ISPs. Each ISP provides candidate services which optimize their profits, and the central controller provides the appropriate ones from the candidate services to the users, which optimize the user service experience. In addition, a matching algorithm is also devised in ARSC with the Pareto efficiency introduced to conduct matching among the applications and the services, which effectively improve the PE. However, CODE and CLDE do not consider the equilibrium among the users and the ISPs.

6.6. Runtime overhead

RO is defined as the time interval from the application request being received to the routing service being provided. We compare the ROs of the four types of applications under ARSC, CODE and CLDE. We use the relative values to show the comparison results, that is, we set the biggest value as 1 and others are their ratios to it. The biggest value is 0.225 s and the results are shown in Fig. 21. Being connectionless, the runtime overhead of CLDE is the smallest, it just need to calculate path. CODE is connection-oriented and needs to implement operations such as path calculation, admission control, connection setup and so on, thus its runtime overhead is bigger than that of CLDE. ARSC provides the customized routing services with the benefits of the user and the ISP considered jointly, it needs to do path calculation, admission control, resource reservation, utility calculation, service composition, service pricing, matching and so on, thus its runtime overhead is the biggest.

However, for the same type of application request which service has ever been successfully provided, ARSC can establish a quick connection for the application according to the matched FTE. That is, the corresponding FTE has already been distributed in the switch earlier, the second step in Fig. 2 (searching the forwarding table) is successful. The biggest value is 0.177 s and the results are shown in Fig. 22.

7. Conclusions

In this paper, an Adaptive Routing Service Customization mechanism is proposed based on the ideas of SDN and NFV. It customizes the routing services by adaptively allocating network resources and composing network function components. Considering the benefit win-win of the user and the ISP, we present a user utility scheme and a routing service pricing strategy. Then, an efficient matching algorithm for multiple applications and multiple services is proposed with Pareto efficiency introduced to achieve the optimal benefit equilibrium of the user and the ISP. Simulation results have shown that the proposed scheme can improve the routing service performance in the Internet.

In practical environments, the central controller needs to monitor network globally, and coordinate matching among multiple ISPs and a large amount of application requests. With only one single central controller, it inevitably brings serious scalability problem. Thus, the control and management by multiple controllers among network domains will be investigated in our future research.

Acknowledgments

This work is supported by the National Science Foundation for Distinguished Young Scholars of China under Grant no. 71325002, and the National Natural Science Foundation of China under Grant no. 61572123.

References

Aamir, M., Zaidi, S.M.A., 2012. QoS analysis of VoIP traffic for different codecs and frame counts per packet in multimedia environment using OPNET. In: Proceedings of the 15th International Multitopic Conference (INMIC), pp. 275–281.

Ahuja, K., Singh, B., Khanna, R., 2014. Network selection based on weight estimation of QoS parameters in heterogeneous wireless multimedia networks. Wirel. Pers. Commun. 77 (4), 3027–3040.

Balasubramaniam, S., Botvich, D., Carroll, R., Mineraud, J., 2009. Adaptive Dynamic Routing Supporting Service Management for Future Internet. In: Proceedings of the IEEE Global Telecommunications Conference (Globecom), pp. 1–7.

Benaboud, R., Maamri, R., Sahnoun, Z., 2011. Towards scalability of reputation and QoS based web services discovery using agents and ontologies. In: Proceedings of the 13th International Conference on Information Integration and Web-based Applications and Services, pp. 262–269.

Bozakov, Z., Papadimitriou, P., 2012. AutoSlice: Automated and Scalable Slicing for Software-Defined Networks. In: Proceedings of the ACM conference on CoNEXT student workshop, pp. 3–4.

Bu, C., Wang, X., Huang, M., Li, K., 2017. SDNFV-based dynamic network function deployment: model and mechanism. IEEE Commun. Lett.. http://dx.doi.org/10.1109/LCOMM.2017.2654443.

CERNET2 Topology.URL < http://www.edu.cn/cer2_1556/ > .

Chen, N., 2012. On computing Pareto stable assignments. In: Proceedings of the 29th International Symposium on Theoretical Aspects of Computer Science (STACS), pp. 384–395.

Cheng, G., Chen, H., Hu, H., Wang, Z., Lan, J., 2015. Enabling network function combination via service chain instantiation. Comput. Netw. 92 (2), 396–407.

Chiou, J.-S., 2004. The antecedents of consumers' loyalty toward Internet Service Providers. Inf. Manag. 41 (6), 685–695.

Cohen, A., Korach, E., Last, M., Ohayon, R., 2005. A fuzzy-based path ordering algorithm for QoS routing in non-deterministic communication networks. Fuzzy Sets Syst. 150 (3), 401–417.

Coulibaly, Y., Kaddoum, G., Ahmed, M.F.A., 2015. Mobile virtual network operator strategy for migration towards 4G. In: Proceedings of the International Conference on Information and Communication Technology Research (ICTRC).

Das, S., Parulkar, G., McKeown, N., 2012. Why OpenFlow/SDN can succeed where GMPLS failed. In: Proceedings of the 38th European Conference and Exhibition on Optical Communications (ECOC).

Definitions of terms, 2008. related to quality of service. ITU-T E. 800.

Ding, W., Qi, W., Wang, J., Chen, B., 2015. OpenSCaaS: an open service chain as a service platform toward the integration of SDN and NFV. IEEE Netw. 29 (3), 30–35.

Drutskoy, D., Keller, E., Rexford, J., 2012. Scalable network virtualization in software-defined networks. IEEE Internet Comput. 17 (2), 20–27.

Elias, J., Martignon, F., Paris, S., Wang, J., 2014. Optimization models for congestion mitigation in virtual networks. In: Proceedings of the 22nd International Conference on Network Protocols (ICNP), pp. 471–476.

Fu, Y., Bi, J., Chen, Z., Wu, J., Hao, B., 2014. Orion: a hybrid hierarchical control plane of Software-Defined Networking for large-scale networks. In: Proceedings of the 22nd International Conference on Network Protocols (ICNP), pp. 569–576.

Gebert, S., Pries, R., Schlosser, D., Heck, K., 2012. Internet access traffic measurement and analysis. In: Proceedings of the International Workshop on Traffic Monitoring and Analysis, pp. 29–42.

Gu, C., Zhang, S., 2011. Self-adaptive QoS control mechanism in cognitive networks based on intelligent service awareness. In: Proceedings of the International Conference on Web Information Systems and Mining, pp. 402–410.

Hasan, M.H., Jaafar, J., Hassan, M.F., 2014. Experimental study on the effective range of FCM's fuzzifier values for web services' QoS data. In: Proceedings of the International Conference on Computer and Information Sciences (ICCOINS).

INTERNET2 Network Infrastructure Topology, URL ⟨http://www.internet2.edu/media/medialibrary/2014/03/17/AboutInternet2.pdf⟩.

Jarraya, Y., Madi, T., Debbabi, M., 2014. A survey and a layered taxonomy of software-defined networking. IEEE Commun. Surv. Tut. 16 (1), 1955–1980.

Jiang, Y., Lan, J., Wang, Z., Deng, Y., 2016. Embedding and reconfiguration algorithms for service aggregation in network virtualization. Int. J. Commun. Syst. 29 (1), 33–46.

Katsikogiannis, G., Mitropoulos, S., Douligeris, C., 2013. Policy-based QoS management for SLA-driven adaptive routing. J. Commun. Netw. Kor. 15 (3), 301–311.

Klein, A., Ishikawa, F., Honiden, S., 2014. SanGA: A Self-adaptive Network-aware Approach to Service CompositionIEEE Trans. Serv. Comput 7, 452–464.

Kohler, E., Morris, R., Chen, B., Jannotti, J., Kaashoek, M.F., Kohler, E., 2000. The Click Modular Router. ACM Trans. Comput. Syst. 18 (3), 263–297.

Kreutz, D., Ramos, F.M.V., Verissimo, P.E., Rothenberg, C.E., Azodolmolky, S., Uhlig, S., 2015. Software-defined networking: a comprehensive survey. Proc. IEEE 103 (1), 14–76.

Li, B., Nahrstedt, K., 1999. A control-based middleware framework for quality of service adaptations. IEEE J. Sel. Area Comm. 17 (9), 1632–1650.

Lima, S.R., Carvalho, P., 2011. Enabling self-adaptive QoE/QoS control. In: Proceedings of the 36th Conference on Local Computer Networks (LCN), pp. 239–242.

Ma, R.T.B., Chiu, D.M., Lui, J.C.S., Misra, V., Rubenstein, D., 2011. On cooperative settlement between content, transit, and eyeball internet service providers. IEEE/ACM Trans. Netw. 19 (3), 802–815.

McAuley, A.J., Manousakis, K., Kant, L., 2008. Flexible QoS route selection with diverse objectives and constraints. In: Proceedings of the 16th International Workshop on Quality of Service (IWQoS), pp. 278–288.

Mellouk, A., Hoceini, S., Zeadally, S., 2011. A bio-Inspired Quality of Service (QoS) routing algorithm. IEEE Commun. Lett. 15 (9), 1016–1018.

Mijumbi, R., Serrat, J., Gorricho, J.L., Bouten, N., De Turck, F., Boutaba, R., 2016. Network function virtualization: state-of-the-art and research challenges. IEEE Commun. Surv. Tut. 18 (1), 236–262.

Network performance objectives, 2011. for IP-based services. ITU-T Y. 1541.

Nunes, B.A.A., Mendonca, M., Nguyen, X.N., Obraczka, K., Turletti, T., 2014. A survey of software-defined networking: past, present, and future of programmable networks. IEEE Commun. Surv. Tut. 16 (3), 1617–1634.

OpenFlowClick, URL ⟨http://archive.openflow.org/wk/index.php/OpenFlowClick⟩.

Peculea, A., Iancu, B., Dadarlat, V., Ignat, I., 2010. A novel QoS framework based on admission control and self-adaptive bandwidth reconfiguration. Int. J. Comput. Commun. 5 (5), 862–870.

Platenius, M.C., von Detten, M., Becker, S., Schäfer, W., Engels, G., 2013. A survey of fuzzy service matching approaches in the context of on-the-fly computing. In: Proceedings of the 16th International ACM Sigsoft Symposium on Component-Based Software Engineering, pp. 143–152.

Project Floodlight Open Source Software for Building Software-Defined Networks, URL ⟨http://www.projectfloodlight.org/⟩.

Shanbhag, S., Wolf, T., 2011. Automated composition of data-path functionality in the future internet. IEEE Netw. 25 (6), 8–14.

Stankiewicz, R., Cholda, P., Jajszczyk, A., 2011. QoX: what is it really? IEEE Commun. Mag. 49 (4), 148–158.

Tansupasiri, T., Kanchanasut, K., Barakat, C., Jacquet, P., 2006. Using active networks technology for dynamic QoS. Comput. Netw. 50 (11), 1692–1709.

Tarnaras, G., Haleplidis, E., Denazis, S., 2015. SDN and ForCES based optimal network topology discovery. In: Proceedings of the 1st IEEE Conference on Network Softwarization (NetSoft), pp. 1–6.

The Internet Topology Zoo. URL⟨http://www.topology-zoo.org/⟩.

Tsiaras, C., Stiller, B., 2014. A deterministic QoE formalization of user satisfaction demands (DQX). In: Proceedings of the Conference on Local Computer Networks (LCN), pp. 227–235.

Wang, X., Cheng, H., Huang, M., 2013. Multi-robot navigation based QoS routing in self-organizing networks. Eng. Appl. Artif. Intell. 26 (1), 262–272.

Wang, X., Cheng, H., Huang, M., 2014. QoS multicast routing protocol oriented to cognitive network using competitive coevolutionary algorithm. Expert Syst. Appl. 41 (10), 4513–4528.

Wang, P., Lan, J., Zhang, X., Hu, Y., Chen, S., 2015a. Dynamic function composition for network service chain: model and optimization. Comput. Netw. 92 (2), 408–418.

Wang, X., Qu, D., Huang, M., Li, K., Das, S.K., Zhang, J., Yu, R., 2015b. Multiple many-to-many multicast routing scheme in green multi-granularity transport networks. Comput. Netw. 93 (1), 225–242.

Xia, M., Shirazipour, M., Zhang, Y., Green, H., Takacs, A., 2015. Optical service chaining for network function virtualization. IEEE Commun. Mag. 53 (4), 152–158.

Yogesh, Mundada, Rob Sherwood, N.F., 2012. An OpenFlow switch element for Click. In: Proceedings of the Symposium on Click Modular Router.

Chao Bu received the B.S. degree in information security and the M.S. degree in software engineering from the Northeastern University, Shenyang, China, in 2010 and 2012 respectively, and is currently working toward the Ph.D. degree at the Northeastern University. His research interests include routing service composition in future Internet, etc.

Xingwei Wang received the B.S., M.S., and Ph.D. degrees in computer science from the Northeastern University, Shenyang, China, in 1989, 1992, and 1998 respectively. He is currently a Professor at the College of Software, Northeastern University. His research interests include cloud computing and future Internet, etc.

Hui Cheng received the B.S. and M.S. degrees in Computer Science from the Northeastern University, Shenyang, China, in 2001 and 2004 respectively, and the Ph.D. degree in Computer Science from the Hong Kong Polytechnic University, Hong Kong, China, in 2007. He is currently a Senior Lecturer at the School of Computing & Mathematical Sciences, Liverpool John Moores University, Liverpool, UK. His research interests include wireless mobile networks, optical networks, artificial intelligence, and dynamic optimization, etc.

Min Huang received the B.S. degree in automatic instrument, the M.S. degree in systems engineering, and Ph.D. degree in control theory from the Northeastern University, Shenyang, China, in 1990, 1993, and 1999 respectively. She is currently a Professor at the College of Information Science and Engineering, Northeastern University. Her research interests include modeling and optimization for logistics and supply chain system, etc.

Keqin Li received the B.S. degree in computer science from Tsinghua University, Beijing, China, in 1985, and Ph.D. degree in computer science from the University of Houston, Texas, USA, in 1990. He is currently a SUNY distinguished professor of computer science in State University of New York at New Paltz. His research interests include parallel and distributed computing and computer networking, etc.

Sajal K. Das received the B.S. degree in computer science from Calcutta University, Kolkata, India, in 1983, the M.S. degree in computer science from the Indian Institute of Science, Bangaluru, India, in 1984, and the Ph.D. degree in computer science from the University of Central Florida, Orlando, in 1988. He is currently with the Department of Computer Science, Missouri University of Science and Technology.