Yi B, Wang XW, Huang M *et al.* A QoS based reliable routing mechanism for service customization. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 37(6): 1492–1508 Nov. 2022. DOI 10.1007/s11390-021-0686-4

A QoS Based Reliable Routing Mechanism for Service Customization

Bo Yi[1] (易 波), *Member, CCF, IEEE*, Xing-Wei Wang[1,*] (王兴伟), *Senior Member, CCF*, Min Huang[2] (黄 敏) and Qiang He[3] (何 强)

[1] *College of Computer Science and Engineering, Northeastern University, Shenyang 110169, China*
[2] *College of Information Science and Engineering, Northeastern University, Shenyang 110819, China*
[3] *College of Medicine and Biological Information Engineering, Northeastern University, Shenyang 110169, China*

E-mail: yibo@cse.neu.edu.cn; {wangxw, mhuang}@mail.neu.edu.cn; heqiang@bmie.neu.edu.cn

Received July 15, 2020; accepted June 2, 2021.

Abstract Due to the rapid development of the Internet technology such as 5G/6G and artificial intelligence, more and more new network applications appear. Customers using these applications may have different individual demands and such a trend causes great challenges to the traditional integrated service and routing model. In order to satisfy the individual demands of customers, the service customization should be considered, during which the cost of Internet Service Provider (ISP) naturally increases. Hence, how to reach a balance between the customer satisfaction and the ISP profit becomes vitally important. Targeting at addressing this critical problem, this work proposes a service customization oriented reliable routing mechanism, which includes two modules, that is, the service customization module and the routing module. In particular, the former (i.e., the service customization module) is responsible for classifying services by analyzing and processing the customer's demands. After that, the IPv6 protocol is used to implement the service customization, since it naturally supports differentiated services via the extended header fields. The latter is responsible for transforming the customized services into specific routing policies. Specifically, the Nash equilibrium based economic model is firstly introduced to make a perfect balance between the user satisfaction and the ISP profits, which could finally produce a win-win solution. After that, based on the customized service policies, an optimized grey wolf algorithm is designed to establish the routing path, during which the routing reliability is formulated and calculated. Finally, the experiments are carried out and the proposed mechanism is evaluated. The results indicate that the proposed service customization and routing mechanism improves the routing reliability, user satisfaction and ISP satisfaction by about 8.42%, 15.5% and 17.75% respectively compared with the classical open shortest path first algorithm and the function learning based algorithm.

Keywords quality of service (QoS), reliability, routing, service customization

1 Introduction

The rapid development of the Internet technology (e.g., 5G and 6G) promotes the appearance of many novel applications such as AR and VR (the 5th and the 6th generation mobile communication technology respectively). These new applications require high throughput and directly lead to the exponential growth of network traffic, which in turn causes great challenges for network routing and service provision [1]. In addition, for the customers using these applications, they may have their own individual requirements towards the same application and the same customer may also have different requirements towards different applications. Such phenomena are becoming very common and universal especially with the appearance of novel network paradigms such as Software-Defined Networking (SDN) [2] and Network Function Virtualization (NFV) [3]. Hence, there is an urgent need for the Internet service providers (ISPs) to offer different or cus-

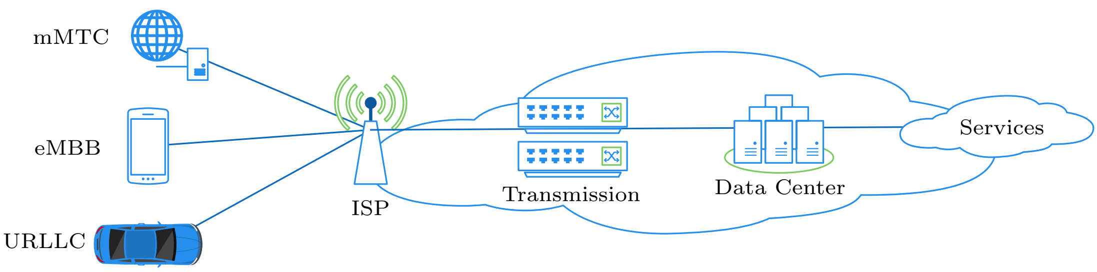
Regular Paper

This work was supported in part by the National Key Research and Development Program of China under Grant No. 2019YFB1802800, the National Natural Science Foundation of China under Grant Nos. 62032013, 62002055 and 61872073, the China Postdoctoral Science Foundation under Grant No. 2020M680972, the Post-Doctoral Research Fund of Northeastern University of China under Grant No. 20200103, the Fundamental Research Funds for the Central Universities of China under Grant No. N2016012, and the CERNET Innovation Project of China under Grant No. NGII20190107.

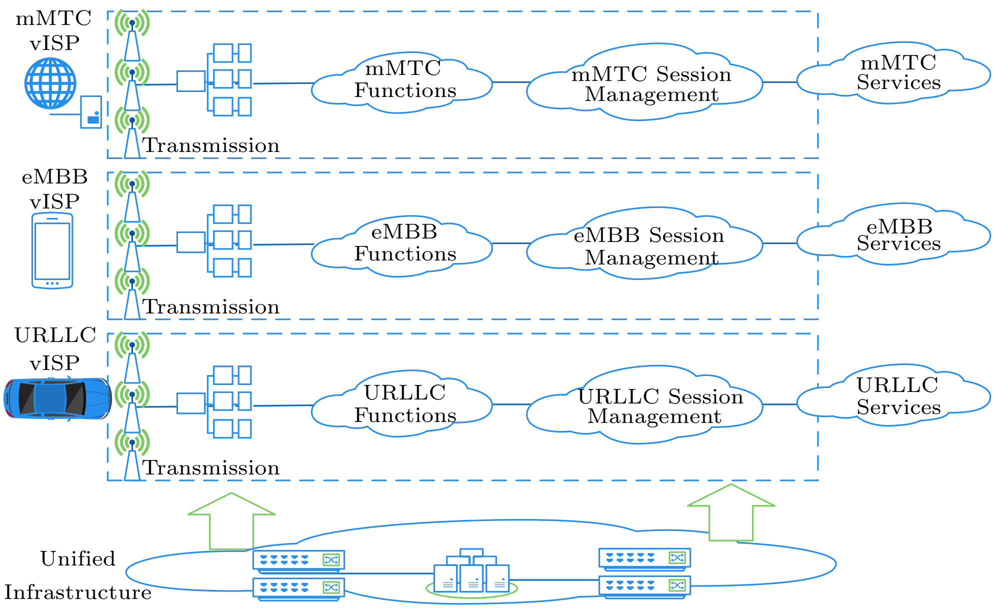
*Corresponding Author

tomized services to customers having different service-level demands. In particular, the offered service customization policy should not only satisfy the customers' requirements, but also guarantee the ISPs' profits to a certain extent, since the profit is the main driving force for ISPs to further improve service quality and user satisfaction [4]. For example, as shown in Fig.1, the three concrete application scenarios (the massive Machine Type Communication (mMTC), the enhanced Mobile BroadBand (eMBB) and the Ultra Reliable Low Latency Communication (URLLC)) of 5G are leveraged to illustrate our motivation. Specifically, in Fig.1(a), mMTC, eMBB and URLLC services are offered by the same ISP over the same infrastructure without service customization. In this condition, the traffic from these three application scenarios may affect one another and the resource preemption may also happen. This resource preemption may directly lead to unexpected results and low performance. On the contrary, as shown in Fig.1(b), by introducing multiple virtual ISPs (vISPs) [5], the service customization can be implemented, where each vISP is responsible for providing one specific kind of services. In addition, the vISPs are actually isolated by using the virtualization technology, despite the fact that they still share the same infrastructure. Hence, compared with the model in Fig.1(a), the one with service customization would achieve better performance.

However, due to the closely coupled characteristics between the data forwarding plane and the control plane in the traditional Internet, it is very hard to offer customized services for all users especially with



(a)



(b)

Fig.1. A concrete example to show the motivation: with the service customization, multiple vISPs can be generated to offer isolated and unique services. (a) One ISP offers all 5G services (without service customization). (b) Multiple vISPs offer different kinds of 5G services (with service customization).

low cost. Therefore, many researches rely on using SDN (software-defined networking) and NFV (network function visualization) to separate the data forwarding plane and the control plane, such that differentiated services can be offered in a flexible way. For example, Li *et al.* [6] constructed a service customization framework based on SDN. With the global network view and the centralized control ability, the network conditions and status can be easily monitored and collected. Based on it, this work [6] defines the mapping relationship between service configuration and routing in different granularities and explored the economic relationship between users and ISPs to achieve a better balance. Besides, by leveraging the concept of NFV, the service can be flexibly composed. For example, Gharbaoui *et al.* [7] re-programmed the hardware-based middle-boxes to virtual network functions (VNFs), such that these VNFs can be flexibly composed to offer a quick service customization [8].

As explained, the IPv6 protocol naturally supports service customization, since it allows to store customized information in the extended header fields. Despite this, such extended fields are used by most researches to implement a better QoS routing instead of designing proper service customization policies to guide the traffic forwarding and routing. For instance, Tomovic and Radusinovic [9] simply generated, maintained and deleted the flow label field to achieve a QoS routing solution. Targeting at addressing the security problems in network, a source address validation improvement (SAVI) platform [10] was proposed. However, this work [10] mainly focuses on validating the IP addresses, while the reliability is also very important in network. Nevertheless, the most typical actions responding to network failures are the fast retransmit and recovery, which would still occupy a lot of extra resources.

Aiming at addressing these challenges, this work proposes a service customization and reliable routing mechanism based on the idea of SDN. Specifically, the service customization is fulfilled in three steps. Firstly, the users' demands on services are analyzed for clustering. Secondly, the service classification is implemented from the perspective of ISPs. Thirdly, the Nash equilibrium is used to reach an economic balance between the user satisfaction and the ISP profit. Then, we need to transform the service customization into routing policies, where an optimized grey wolf algorithm is designed to calculate the corresponding path and the link failure model is also considered to finally achieve the reliable routing. The main contributions of this work include a new service customization framework, a service customization scheme and a reliable routing mechanism. In particular, the framework provides the working environment and interactive interface for the service customization scheme and reliable routing. The service customization scheme helps the proposed routing mechanism to make the reliable routing decision. The three contributions are summarized as follows.

• We construct a new service customization framework in this work, which is mainly composed of service customization and optimization routing. The former includes the functions of demand analysis, service classification and the corresponding economic model, while the latter offers a routing algorithm to support such service customization.

• We design a service customization scheme which tries to satisfy the customers' personal demands as well as the ISPs' expectations. In order to make them both satisfied, the concept of Nash Equilibrium is applied. Specifically, we try to reach a balance between customers and ISPs to finally achieve a win-win situation.

• We design a reliable routing mechanism on the basis of a grey wolf algorithm, in which the situations of both link failure and node failure are taken into consideration. In addition, we apply the mutation action into this mechanism to further optimize the routing process.

The rest of this work is organized as follows. Section 2 presents the related work and Section 3 designs a framework for service customization and routing. Section 4 explains the service customization scheme and builds a Nash Equilibrium between service providers and service consumers. Section 5 discusses the proposed reliable routing mechanism. Section 6 presents the experimental results and Section 7 makes a conclusion.

2 Related Work

According to the research of this work, we divide the related work into two kinds, that is, service customization and reliable routing.

2.1 Service Customization

Service customization is generally a process that builds a mapping relationship between users' requirements and the diverse service policies. The users' requirements can be further decomposed to the detailed demands of quality of service (QoS). Recently, many organizations or individuals are trying to apply the service customization methods into solving the routing configuration and service selection in many different scenarios.

In order to implement the idea of service customization, many researches were carried out. For example, Li *et al.*[11] proposed a customized method for service routing, in which two kinds of routing characteristics are defined. The first one was coarse-grained and abstracted, such that it was regarded as the fundamental part. The second one was fine-grained and service-oriented, such that it was more like polymorphism. Based on the two kinds of defined routing characteristics, different service paths could be dynamically adjusted to the users' needs, which finally achieved multi-dimensional routing to a certain extent. Instead of service customization, Kumara *et al.*[12] focused on implementing a service differentiated framework in the SDN environment. This method mainly addresses the service configuration problem for large-scale streaming applications and the service differentiation is guaranteed by mining and analyzing the differences between requirements. Bu *et al.*[13] proposed a big data driven service customization policy for SDN. This method can offer different routing configurations for traffic transmission by designing an accurate user demand model on the basis of a large amount of history data. Besides, from the perspective of the service provider, mining the interest relationship between users and providers could also provide the economic analysis basis for service configuration.

It is generally known that users all desire high service quality. However, most of them cannot specify their requirements about the QoS they want. In this regard, Masruroh *et al.*[14] proposed a fuzzy inference method for service selection without needing to consider the detailed context of users and it uses the service graph to represent the task workflow. Besides, this method takes the service QoS related parameters as input and the conditions that may violate the pre-defined QoS hypothesis as output. Meanwhile, Han *et al.*[15] also constructed a service graph to address the QoS sensitivity problem, where the methods of local search and global search were jointly used. Bhattacharya *et al.*[16] proposed a service aware and composition model which considers not only the differences of user requirements, but also the cost optimization objective when satisfying the QoS demands. Contrary to the customers, ISPs know exactly what they want, that is, the higher the profits, the better. For example, Xiao *et al.*[17,18] presented an in-depth study of the commercial mobile virtual network operators (MVNO), which introduces new measurement and optimization methods to address the fundamental concerns of MVNO.

In order to better support service customization, some researches[19,20] tried to use the technology of segment routing on the basis of the IPv6 protocol, which is short for SRv6 (segment routing over IPv6). In particular, SRv6 can naturally support service customization, by using the extra header fields of the IP packets. Taking this as the basis, Moravejosharieh *et al.*[19] proposed a risk-based QoS routing algorithm by using the SRv6 technology. This algorithm tries to calculate and build the maximum connections that can be supported by the available resource, thus to reduce the time spent on the request admission control. Meanwhile, Yan *et al.*[20] tried to solve the QoS problem among different IP domains by jointly using the SRv6 and flow labels. On the one hand, SRv6 enables to implement the service customization in an easier way to satisfy customers' requirements. On the other hand, the flow label is lightweight and it can be used to further improve the QoS performance, since the flow label based model can accelerate the data transmission and balance the resource utilization compared with the best-effort and differentiated models. Despite the use of the flow label, these methods cannot eventually offer the mapping between the service model and the final routing. Apart from this, these methods are hard to adapt the multi-constraint and multi-domain environment, since their calculation convergence rate and the solution diversity could not be guaranteed.

2.2 Reliable Routing

One critical factor for guaranteeing the service quality is reliability, particularly the reliable routing, since it can make sure that the customized service will not break during the service cycle. Taking this into consideration, a lot of researches have been proposed, including some classical methods such as redundancy and fast rerouting.

For example, Yang *et al.*[21] prepared the backup routing path for important traffic, under which the backup path is used immediately when faults occurred. This backup method is implemented in two cases, that is, with and without using labels. The label-based case intends to encapsulate the backup message in the packet header, while the other one would like to carry the backup information in the data transmitted. However, the routing strategy of [21] simply follows the shortest path first principle. Besides, as a result of the emergency of diverse novel network scenarios, these cases are no longer adaptive enough. For example, the novel SDN enables the IPv6 network with the capacities of high flexibility and programmability [22]. However, the centralized control and management characteristics of SDN make the centralized controller a performance bottleneck for IPv6 networks, which then may result in a big trouble for achieving reliability.

New technologies and scenarios bring many benefits as well as many challenges. For example, the centralized control of SDN enables to manage the network from a global view, which could also result in performance bottleneck. In this regard, to address these issues, Papan *et al.* [23] tried to optimize the process of fast re-routing by designing a new bit indexed explicit replication method. In specific, this work would like to avoid storing the state information of traffic in the intermediate devices, such that the corresponding routing information can be stored firstly in a bit sequence. After that, the traffic routing can be achieved in a distributed manner, which then avoids the performance bottleneck caused by the centralized control in a certain extent. Similarly, Nobakht *et al.* [24] also tried to solve such a performance bottleneck by building a distributed management and control model for reliable routing. In particular, this work calculated two totally different paths across multiple domains, which means that the two paths do not share the same intermediate nodes. Among them, one path is regarded as the main path, while the other is the backup path. Despite this, there are multiple paths between any two nodes; hence multiple paths can be calculated in the practical situation. Using part of these paths for data transmission and the others for backup, we can not only improve the traffic transmission efficiency, but also guarantee the service routing reliability to a certain extent.

It was generally aware that the reliability of traffic can also be reflected by how much the corresponding QoS requirements are satisfied [25]. For instance, some traffic may have great demand on security, such that the traffic would be regarded to be unreliable if its security demand could not be fully satisfied. Martínez-Peñas and Kschischang [26] tried to achieve a secure routing by applying the technology of network coding protection (NCP) into the protection process of shared links in IPv6 networks. By improving the security level offered, this work could guarantee the reliability of links in a great probability. Vignesh and Premalatha [27] proposed a new reliable routing framework, in which the problem is formulated as a multi-objective optimization model, that is, the reliable routing is evaluated by different QoS performance metrics. By jointly taking many QoS indicators (e.g., delay and throughput) into consideration, this proposed framework can calculate one optimal path, during which the path failure and interference are both considered. Moreover, to achieve a reliable routing, Milolidakis *et al.* [28] proposed a probability correlated failure (PCF) model for the case that one single physical link may be shared by multiple logical links. Specifically, PCF can prepare a better backup path based on the failure probabilities of each logical link when the physical link shared by them is failed. Furthermore, Wang *et al.* [29] adopted the technology of machine learning to automate the configuration of service routing and achieved better performance in scalability and adaptability. However, we should be aware that these methods are either proactive or reactive. The former enables to recover from failure quickly with the price of extra resources on backup paths, while the latter suffers from a slow calculation convergence rate with the need to reserve resources for backup paths.

3 System Framework for Service Customization and Reliable Routing

The service customization and reliable routing system framework is designed and shown in Fig.2, where it is composed of the network monitor, the service customization module, the reliability routing module, and the source address validation improvement (SAVI) based platform [10]. In particular, the SAVI platform is used to provide a secure environment for the experiment. Based on this platform, the network monitor module is in charge of monitoring the traffic and service states. Once the situation that the QoS demand is not satisfied has been detected, the service reconfiguration process will be triggered. Importantly, the service customization module consists of three steps which are the demand analysis, the service classification and the economic model. The three steps are used to parse and analyze the arriving requests, during which the users' demands on QoS and reliability are obtained and transformed into customized service configurations. After that, the routing module is used to calculate the routing path according to the customized service configurations, in which an optimized grey-wolf algorithm is designed. Finally, the evaluation module is used to verify and evaluate the performance of the proposed work.

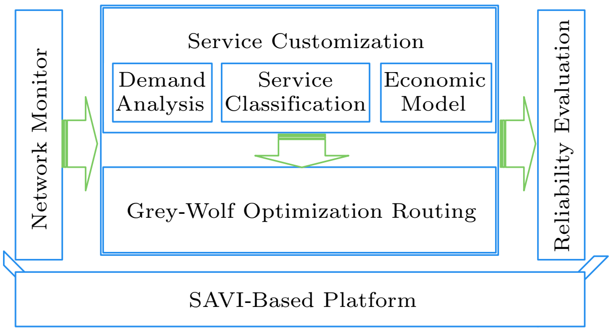


Fig.2. System framework.

Specifically, the network monitor runs periodically to monitor the underlying conditions, for example, the QoS performance of services and the reliability conditions of routing paths. Once the QoS performance is detected not to be satisfied any more or the reliability guarantee is failed, the service customization and routing processes will be triggered. For the normal arriving requests, the service customization module will analyze them first to obtain users' requirements, because the services demanded by the users can actually reflect their requirement patterns. From the perspective of ISPs, the services provided by them can be used to form a general service classification model. Then, in order to make a balance between users and ISPs, a game theory based economic model is introduced in the service customization module. By reaching the Nash equilibrium between users and ISPs, the balance between them can be achieved. Once a service request arrives, it will then be handled by this economic model to calculate a customized service policy. Then, this customized service policy will be forwarded to the routing module, which will then be transformed to a set of routing instructions to finally implement the service customization and provision. After that, the methods of service customization and routing will be evaluated accordingly in the evaluation module. Throughout the whole process, the SAVI platform offers a basic secure environment for testing the proposed framework and mechanism.

As explained, the core parts of this work are service customization and routing. They are actually executed in strict order, that is, the routing is carried out after the service customization. Hence, we then comprehensively introduce and discuss them in sequence.

4 Service Customization

The service customization consists of three parts which are demand analysis, service classification and economic model. The main notations used for formulation are summarized in Table 1.

4.1 Demand Analysis

The service customization is implemented according to users' demands. Hence, we need to first analyze their demands which are closely related to the QoS and the

Table 1. Notations

Notation	Meaning
$maxB, minD, minJ, minP$	Expected performance of bandwidth, delay, jitter and packet loss rate of users respectively
$priceE, O$	Expected price and other value-added services of users respectively
$minB, maxD, maxJ, maxP$	QoS performance of bandwidth, delay, jitter and packet loss rate guaranteed by ISPs respectively
B, D, J, P	Actual service performance of bandwidth, delay, jitter and packet loss rate respectively
$U(B), U(D), U(J), U(P), U(price)$	User satisfaction on bandwidth, delay, jitter, packet loss rate and price respectively
$U(USR), \lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5$	Overall satisfaction of users and the weight parameters for the satisfactions on bandwidth, delay, jitter, packet loss rate and price respectively
$cost, price_1, price_2, price_3$	Service cost and the prices in terms of three different pricing strategies respectively
$U(ISP)$	Satisfaction of Internet service providers
$G(N, E)$	Network topology abstraction with nodes and links
$bandwidth(e), delay(e), jitter(e), loss(e)$	Bandwidth, delay, jitter and packet loss rate of the link $e \in E$ respectively
$cost(p), reliability(p), \Upsilon$	Cost and reliability of path $p \in PATH$, and the default reliability level
$F, f_{\rm B}, f_{\rm D}, f_{\rm J}, f_{\rm P}, \Phi(x)$	A fitness function and the functions used to calculate the punishment factors for different performance indicators of bandwidth, delay, jitter, and packet loss rate respectively
D, X, A, C	D is the distance between source and destination; X indicates the corresponding positions; A and C are the coefficient factors for the grey wolf algorithm
$\theta_{i,j}, \theta'_{i,j}$	Probability of selecting node j as the next move from node i with and without a mutation respectively

service level agreement (SLA). The basis indicators of QoS include the bandwidth, the delay, the jitter and the packet loss rate, based on which we implement the preliminary service customization. Usually, there are many kinds of services which include the file transfer, voice call, video conference, etc. Now, given a certain kind of services, from the perspective of users, they expect the best performance at a reasonable price. Assuming that one user gives the price denoted by $priceE$, then the corresponding service demand can be customized by the five-tuple $(maxB, minD, minJ, minP, priceE)$, where $maxB$ is the maximum bandwidth, $minD$ is the minimum delay, $minJ$ is the minimum jitter, and $minP$ is the minimum packet loss rate demanded by this user with the expected price of $priceE$ when accessing such kind of services. However, apart from these basic indicators, there are also many additional and optional services (e.g., security and reliability) that should be considered nowadays. For example, there are five security levels and the default one is level 1. If the customers do not explicitly state their security requirements, the service will be offered at the default security level. However, if the customers want a higher security level, they have to offer extra payment. In this regard, we extend the five-tuple to a six-tuple $(maxB, minD, minJ, minP, priceE, O)$, where O is a set notation including at least one additional service.

4.2 Service Classification

We classify the services into eight levels based on the basic QoS requirements which include the bandwidth, delay, jitter and packet loss rate. Generally, the higher the service level, the better the performance from the perspective of users. However, given the same price, a higher service level may damage the profits of ISPs. Thus, a balance between users and ISPs should be reached. Given the same kind of services, different ISPs may have different settings and approaches. The same point is that these ISPs all need to satisfy the users' demands as many as possible, while guaranteeing their own profits at the same time. In this condition, the service QoS parameters can be described by the four-tuple $(minB, maxD, maxJ, maxP)$, where $minB$ is the minimum bandwidth, $maxD$ is the maximum delay, $maxJ$ is the maximum jitter and $maxP$ is the maximum packet loss rate that can be guaranteed by ISPs respectively. ISPs provide the performance lower limit, while users desire the performance upper limit. Such a contraction results in that we cannot design the model and strategy only from the view of one side. In this regard, the game theory is introduced between users and ISPs to achieve a win-win service customization.

Now, jointly taking the above conditions into consideration, the actual service performance can be described as follows:

$$\begin{cases} minB \leqslant B \leqslant maxB, \\ minD \leqslant D \leqslant maxD, \\ minJ \leqslant J \leqslant maxJ, \\ minP \leqslant P \leqslant maxP, \end{cases} \tag{1}$$

where (B, D, J, P) in (1) indicates the actual service performance of bandwidth, delay, jitter and packet loss rate respectively. In particular, the additional value-added services are set to default when there is no explicitly statement.

4.3 Nash Equilibrium Based Economic Model

As explained, the user satisfaction (how satisfied the users are with the services provided by ISPs) conflicts with the ISP satisfaction (how satisfied ISPs are with their profits). In order to make a balance between them, we need to firstly calculate the utility functions for the four basic indicators based on the Gaussian membership function. Let us denote it by $U(*)$, and then we have:

$$\begin{aligned} &U(*) \\ &= \begin{cases} 1, \\ \quad \text{if } * \leqslant min*, \forall * \in \{D, J, P\} || * \geqslant maxB, \\ e^{\frac{(*-max*)^2}{(*-min*)^2}}, \\ \quad \text{if } min* \leqslant * \leqslant max*, \forall * \in \{B, D, J, P\}, \\ \varepsilon, \\ \quad \text{if } * = max*, \forall * \in \{D, J, P\} || * \geqslant minB, \\ 0, \\ \quad \text{if } * > max*, \forall * \in \{D, J, P\} || * < minB, \end{cases} \end{aligned} \tag{2}$$

where the value of ε approaches to 0, $* \in \{B, D, J, P\}$, $U(B), U(D), U(J), U(P)$ are the utility functions for bandwidth, delay, jitter and packet loss rate respectively, $max* \in \{maxD, maxJ, maxP\}$ and $min* \in \{minD, minJ, minP\}$.

The value of $U(*)$ is proportional to the user satisfaction, such that the higher the value of $U(*)$, the higher the user satisfaction towards this performance metric. However, the service price is another key aspect that should be focused, since it also affects user satisfaction as well. The expected price is denoted by

$priceE$ and we denote the maximum price the users can accept by $priceM$. Then, assuming that the actual transaction price is $price$, the user satisfaction on prices is formulated as follows:

$$U(price) = \begin{cases} 1, & \text{if } price \leqslant priceE, \\ \varepsilon, & \text{if } price = priceM, \\ 0, & \text{if } priceM < price, \\ 1 - \frac{price - priceE}{priceM - price}, & \text{otherwise.} \end{cases} \quad (3)$$

Combining (2) and (3), the user satisfaction can be calculated as follows:

$$\begin{aligned} U(USR) = {} & \gamma_1 U(B) + \gamma_2 U(D) + \\ & \gamma_3 U(J) + \gamma_4 U(P) + \gamma_5 U(price), \end{aligned} \quad (4)$$

where (4) should be under the constraints that $\gamma_1 + \gamma_2 + \gamma_3 + \gamma_4 + \gamma_5 = 1$ and $U(USR) \in [0, 1]$.

It is noted that the higher the user satisfaction, the more likely the users will accept the services. Then, denoting the cost for one kind of services by $cost$ and the profit desired by ISP by $profit$, three pricing strategies are designed. The first one is cost-oriented, which sets the price on the basis of the service cost, that is,

$$price_1 = \kappa cost, \quad (5)$$

where $\kappa > 1$ in (5) is the price coefficient.

The second one is profit-oriented, which means that the price is determined by the profit they want, that is,

$$price_2 = profit + cost, \quad (6)$$

where $profit$ in (6) is expected to be as high as possible by ISPs under the condition that $U(*) > 0$.

The third one is sales-oriented, which tries to maximize the service acceptance rate to finally increase the total profits. Generally, the higher the user satisfaction, the more likely they will accept the service; hence, it follows that

$$\begin{aligned} & profit = (price_3 - cost) \times U(USR), \\ \Longrightarrow \quad & price_3 = \frac{profit}{U(USR)} + cost, \end{aligned} \quad (7)$$

where $U(USR) \in [0, 1]$ in (7) is expected to be as high as possible by users.

According to the three pricing strategies, the ISP satisfaction is formulated as follows:

$$U(ISP) = \begin{cases} 0, & \text{if } price \leqslant cost, \\ \frac{price_i - cost}{price_i}, & \text{otherwise,} \end{cases} \quad (8)$$

where $price_i$ in (8) indicates the price under the i-th ($i \in [1, 3]$) pricing strategy.

During the service customization process, we cannot simply maximize the benefits of one side (i.e., user or ISP), but reach a balance between them. In this way, the game theory [5] is introduced. Specifically, given λ services and the three pricing strategies, we first present the corresponding strategy matrix $(\boldsymbol{U})_{\lambda,3}$, where any pair $(U(USR)_{i,j}, U(ISP)_{i,j}) (\forall i \in [1, \lambda], j \in [1, 3])$ indicates the user satisfaction and the ISP satisfaction when using the j-th pricing strategy for the i-th kind of services. Then, in order to make both sides satisfied, the Nash equilibrium between the user and the ISP should be reached. If $\exists (U(USR)_{i,j}, U(ISP)_{i,j})$, which makes

$$\begin{cases} U(USR)_{i,j} \geqslant U(USR)_{i',j'} \\ U(ISP)_{i,j} \geqslant U(ISR)_{i',j'} \end{cases}, \quad i \neq i', j \neq j', \quad (9)$$

satisfied, then, $(U(USR)_{i,j}, U(ISP)_{i,j})$ constrained in (9) is a Nash equilibrium solution and we regard it as the pricing policy between the ISP and the user to achieve the win-win situation.

5 Reliable Routing

5.1 Problem and Model

After the service customization, the corresponding routing should be carried out by transforming the constraints into the final service path. In this work, the network topology is abstracted as an undirected graph $G(N, E)$, where N is the set of nodes and E is the set of links. Given any request with customized service demand, we can find all the paths satisfying such customized requirements and store them in the set notation $PATH$. However, the selected path (denoted by p) for this customized service should satisfy the following objective and constraints:

$$\begin{aligned} \text{minimize: } & \frac{cost(p)}{reliability(p)}, \forall p \in PATH \\ \text{s.t. } & \min\{bandwidth(e) | e \in p\} \geqslant minB, \\ & \max\{jitter(e) | e \in p\} \leqslant maxJ, \\ & \sum_{e \in p} delay(e) \leqslant maxD, \\ & \sum_{e \in p} loss(e) \leqslant maxP, \end{aligned} \quad (10)$$

where e is a link of $p \in PATH$, and $bandwidth(e)$, $delay(e)$, $jitter(e)$, $loss(e)$ indicate the actual attributes of bandwidth, delay, jitter and packet loss rate of this customized service on e respectively. $cost(p)$ indicates the cost for building path p and it is determined based on the pricing strategy. $reliability(p)$ calculates the reliability of this path according to the full probability formula as follows:

$$\begin{aligned} & reliability(p) \\ = \ & \Pr(p|e_1)\Pr(e_1) + \Pr(p|e_2)\Pr(e_2) + \cdots + \\ & \Pr(p|e_m)\Pr(e_m) \\ = \ & \sum_{i=1}^{m} \Pr(e_i)\Pr(p|e_i), \end{aligned} \quad (11)$$

where m is the number of links of p and e_i is the i-th link belonging to p.

Let us assume that the default reliability level is denoted by Υ, and then the problem model is updated as follows:

$$\begin{aligned} & \text{minimize: } (10) \\ & \quad \text{s.t. } reliability(p) \geqslant \Upsilon. \end{aligned} \quad (12)$$

5.2 Grey Wolf Based Routing Algorithm

As explained, we design an optimized grey wolf algorithm to solve the routing problem. Generally, the grey wolf algorithm simulates the unique leadership structure and hunting mode of grey wolfs. This algorithm divides the wolf population based on the individual fitness to the environment. The higher the fitness, the higher the position of this wolf and the more likely it will capture the food. According to the leadership structure of grey wolfs, there are one α wolf, one β wolf and one δ wolf, while the rest of them are denoted by ω. The α wolf has the highest status and it is responsible for making hunting and management decisions, while the β and the δ wolf have the second and the third highest status respectively and they will assist the α wolf to manage the rest wolfs. Similarly, in this work, we regard the destination (e.g., contents) as the food and each path ($\in PATH$) to the destination as the route for a wolf to find the food. Hence, the routing process can be modeled as the grey wolf model.

The grey wolf algorithm generally has better performance in dealing with the models without constraints. Jointly taking this and (12) into consideration, we formulate the fitness function for paths to destination to initially construct the grey wolf leadership structure in our problem, as follows:

$$F(p) = \frac{reliability(p)}{(f_B + f_D + f_J + f_P) \times cost(p)}, \forall p \in PATH, \quad (13)$$

where

$$\begin{aligned} f_B &= \varPhi(\min\{bandwidth(e)|e \in p\} - B), \\ f_D &= \varPhi(D - \sum_{e\in p} delay(e)), \\ f_J &= \varPhi(J - \sum_{e\in p} jitter(e)), \\ f_P &= \varPhi(P - (1 - \prod_{e\in p}(1 - loss(e)))). \end{aligned} \quad (14)$$

In particular, f_B, f_D, f_J, f_P are the functions used to calculate the punishment factors for different performance indicators. It is noted that the core part of (14) is $\varPhi(x)$ which is defined as follows:

$$\varPhi(x) = \begin{cases} 1, & \text{if } x \geqslant 0, \\ 0, & \text{otherwise}, \end{cases} \quad (15)$$

where $\varPhi(x) = 1$ means that the performance of this selected path is not satisfied.

According to (13)–(15), the initial fitness values of all paths can be calculated and we sort them in descending order, where the first three paths are denoted by $\{p_\alpha, p_\beta, p_\delta\}$ and the rest are stored in p_ω. Then, the corresponding leadership structure is built. Based on it the initial positions of grey wolfs can be obtained and stored in vector $\boldsymbol{X}(0)$. Now, we introduce the optimized grey wolf algorithm: $\forall t \geqslant 0$, it follows that:

$$\begin{aligned} D &= |C \cdot \boldsymbol{X}_p(t) - \boldsymbol{X}(t)|, \\ \boldsymbol{X}(t+1) &= \boldsymbol{X}_p(t) - A \times D, \end{aligned} \quad (16)$$

where t is the number of iterations, $\boldsymbol{X}_p(t)$ is the position of food after the t-th iteration (i.e., the optimal solution), $\boldsymbol{X}(t)$ is the position of grey wolfs after the t-th iteration (i.e., the potential solution), D is the distance between the food and the grey wolf, and A, C are coefficient factors calculated as follows:

$$\begin{aligned} A &= 2a \times r_1 - a, \\ C &= a \times r_2, \end{aligned} \quad (17)$$

where a decreases from 2 to 0 linearly with the number of iterations, and r_1, r_2 are random numbers between 0 and 1.

(16) and (17) mean that the wolfs have surrounded the food. After that, wolfs β and δ will hunt the food under the guidance of α, during which the positions of the wolf and the food would be changed and updated

as follows:

$$(9)\&(10) \Longrightarrow \begin{cases} D_\alpha = |C_1 \cdot \boldsymbol{X}_\alpha(t) - \boldsymbol{X}(t)|, \\ D_\beta = |C_2 \cdot \boldsymbol{X}_\beta(t) - \boldsymbol{X}(t)|, \\ D_\delta = |C_3 \cdot \boldsymbol{X}_\delta(t) - \boldsymbol{X}(t)|, \end{cases}$$
$$\Longrightarrow \begin{cases} \boldsymbol{X}_1 = \boldsymbol{X}_\alpha(t) - A_1 \times D_\alpha, \\ \boldsymbol{X}_2 = \boldsymbol{X}_\beta(t) - A_2 \times D_\beta, \\ \boldsymbol{X}_3 = \boldsymbol{X}_\delta(t) - A_3 \times D_\delta, \end{cases} \quad (18)$$
$$\Longrightarrow \boldsymbol{X}_p(t+1) = \frac{\boldsymbol{X}_1 + \boldsymbol{X}_2 + \boldsymbol{X}_3}{3},$$

where $D_\alpha, D_\beta, D_\delta$ in (18) indicate the distances among α, β, δ and ω respectively.

As we formulate the iteration process, now we should simulate the process of attacking, that is, the wolfs attack the food to get the optimal solution, which is implemented by the decreasing of a from 2 to 0 linearly as shown in (17). Accordingly, the value of A also ranges in $[-a, a]$. In particular, if $|A| \leqslant 1$, that is, within $[-1, 1]$, the next positions of wolfs will be closer to the food. Otherwise, when $1 < |A| \leqslant 2$, the wolfs will be far away from the food, which means getting away from the optimal solution, and this may eventually lead to a local optimum situation.

5.3 Optimization

As explained, we formulate the leadership structure as $\{p_\alpha, p_\beta, p_\delta, p_\omega\}$. Then, let us present the corresponding link probability matrix as follows:

$$\begin{pmatrix} 0 & \theta^e_{0,\,1} & \cdots & \theta^e_{0,\,|N|} \\ \theta^e_{1,\,0} & 0 & \cdots & \theta^e_{1,\,|N|} \\ \vdots & \vdots & \ddots & \theta^e_{1,\,|N|} \\ \theta^e_{|N|,\,0} & \theta^e_{|N|,\,1} & \cdots & 0 \end{pmatrix}, \quad (19)$$

where $\theta_{i,j} \in [0, 1]$ indicates the probability of selecting node j as the next move from node i.

Apparently, given any node i, $\theta_{i,j}$ is constrained by (20) as follows:

$$\sum_{j \in [1, |N|]} \theta_{i,j} = 1. \quad (20)$$

To avoid the local optimum situation, we introduce a mutation in (19) after each iteration, as follows:

$$\theta'_{i,j} = \begin{cases} 1 - e^{-\theta_{i,j}}, & \text{if } \theta_{i,j} > 0, \\ 0, & \text{otherwise.} \end{cases} \quad (21)$$

We mainly focus on the mutation of the α wolf, since it is responsible for deciding the optimal solution. The previous path for α is p_α. Now, after the mutation of the link probability matrix according to (21), the new path for α may be generated and we denote it by p'_α. Calculating the fitness value of both paths according to (13), if $F(p_\alpha) \leqslant F(p'_\alpha)$, then we replace p_α by p'_α as the new route for the α wolf to find the food. Otherwise, we will accept p_α as the solution with the probability of $\exp(\Delta f/t)$, where $\Delta f = F(p_\alpha) - F(p'_\alpha)$.

6 Performance Evaluation

6.1 Setup

The proposed algorithm is implemented by C++ language and evaluated on the SAVI platform with two real-world topologies, that is, the CERNET2 and the Internet2 respectively. As shown in Fig.3, the CERNET2 topology is used to simulate a small-scale network environment with 20 nodes and 22 links, while the Internet2 topology is used to simulate a large-scale network environment with 64 nodes and 75 links.

During the experiments, we offer eight kinds of services and each has its own basic QoS requirements. The size of traffic for each service request is set to 10k and every node is able to generate requests. In order to simulate the real-world workloads, the traffic generation follows the Poisson distribution with the

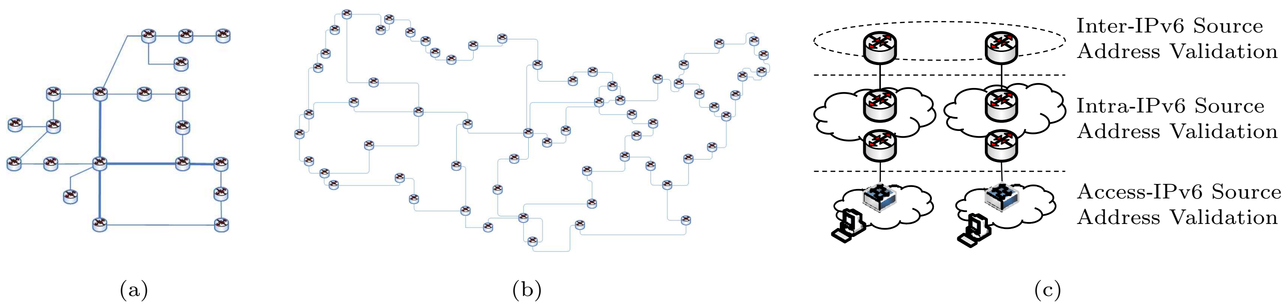


Fig.3. Topologies and the SAVI platform. (a) CERNET2. (b) Internet2. (c) SAVI.

mean parameter set to 10, that is, the average number of requests generated by each node is 10 in one second. Besides, in order to simulate the reliability environment, the nodes and links are separated into the high-failure and low-failure ones. In particular, the proportion of the high-failure node/link is set to 2.5% and the corresponding failure probability is randomly determined in [0.1%, 0.5%] following the power-law distribution with the index of −0.73. The failure probability of the low-failure node/link is randomly determined in [0.01%, 0.1%] following the power-law distribution with the index of −1.35.

The hardware environment used to support this simulation is an x64 PC platform with Inter® Core™ i7-6700 CPU at 3.40 GHz, 16 G RAM, on which a virtual machine with the Linux OS is generated to run the experiments. Besides, the simulation will run 100 times and each one will last for above three hours. After that, the average results are calculated and presented in Subsection 6.3.

6.2 Metrics and Benchmarks

In order to comprehensively evaluate the proposed work, four metrics are used for evaluation, which are the average delay, the service reliability, the user satisfaction and the ISP satisfaction, as follows.

• *Average Delay*. It indicates the average time spent on transmitting a packet and it is calculated by dividing the overall transmission time of the flow by the number of its packets.

• *Service Reliability*. It indicates the reliability of the whole service path and it is closely related to the reliability of each link belonging to this path as expressed by (11).

• *User Satisfaction*. It indicates the satisfaction degree that users have on the services provided by ISPs and it is influenced by the factors of bandwidth, delay, jitter, packet loss rate and price as expressed by (4).

• *ISP Satisfaction*. It indicates the satisfaction degree of ISPs and it is mainly reflected by the profit they make as expressed by (8).

Then, we compare the proposed algorithm with two benchmarks in terms of the above four metrics. Specifically, the first benchmark (i.e., benchmark 1) is the classical Open Shortest Path First (OSPF)[21] algorithm and the second one (i.e., benchmark 2) is a function learning based algorithm[29]. They are explained as follows.

• *Benchmark* 1. It operates on the basis of OSPF and selects the shortest path as the optimal one. In this work, the link weight for benchmark 1 is set according to the delay of this link.

• *Benchmark* 2. It is designed based on the multilayer feed-forward neural network which takes the history customized services as the training input and uses the back propagation algorithm for parameter adjustment and model optimization.

6.3 Results

6.3.1 *Average Delay*

The average delay actually evaluates the time spent on packet transmission (including the connection establishment time and the propagation time) and the results are shown in Fig.4, where Fig.4(a) illustrates the result achieved in the CERNET2 topology and Fig.4(b) illustrates the result achieved in the Internet2 topology.

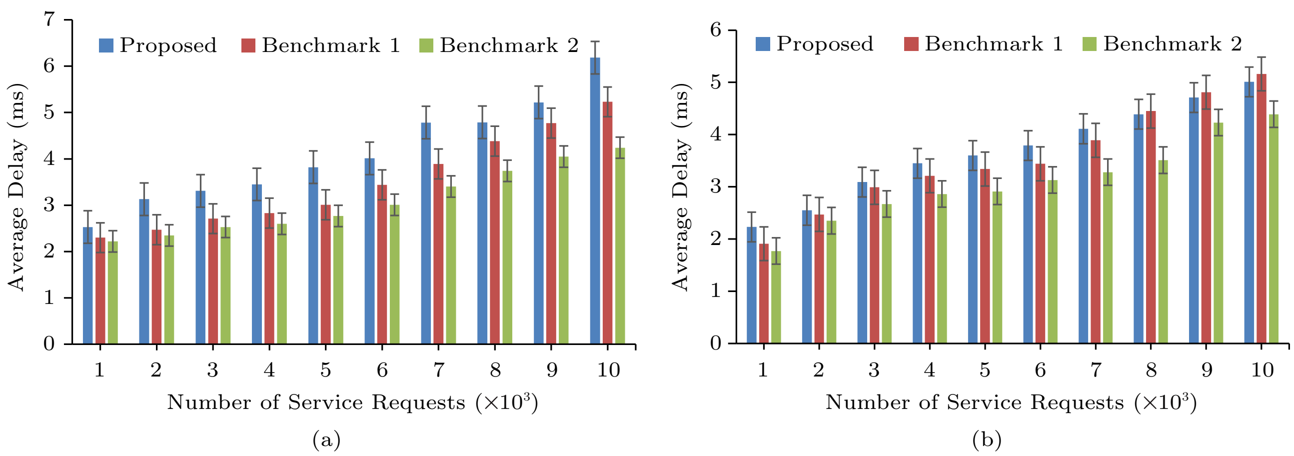


Fig.4. Results of average delay. (a) Average delay vs the number of service requests in CERNET2. (b) Average delay vs the number of service requests in Internet2.

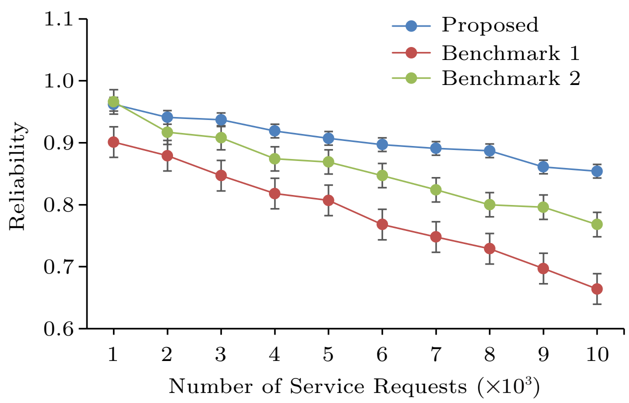
On the one hand, as shown in Fig.4(a), we can notice that the proposed algorithm has a higher average delay than the other two benchmarks. Benchmark 1 has the lowest average delay. One reason for this phenomenon is that we need to carry out the service customization process before routing, which consumes a lot of time at the beginning. On the contrary, benchmark 1 mainly needs to spend time on routing (e.g., path calculation and packet routing) without the service customization by using the shortest path first strategy, such that the overall time consumed by benchmark 1 is much lower. By optimizing the service model with a training process, benchmark 2 can also reduce the time spent on service customization. Despite this, the maximum delay of the proposed mechanism does not exceed 7 ms, which is under the acceptance. In addition, benchmark 2 performs weak when the service model is subjected to multiple constraints, since these constraints will naturally limit the training speed and effect. Such a conclusion is reflected in Fig.4(b), where the average delay of benchmark 2 exceeds that of the proposed algorithm when the number of services exceeds 8 $\times 10^3$. Then, the more the users, the more the requirements they will have, which leads to multiple constraints.

On the other hand, observing the results in both Fig.4(a) and Fig.4(b), it is discovered that with the increasing of the number of service requests, the average delay also increases. Nevertheless, the increasing rate of the proposed algorithm is slower, because the service customization process designed by the proposed algorithm would save a lot of history configuration information. Then, when the number of service requests becomes huge, we have a great probability to encounter the service requests having the same customization requirements and such a probability is proportional to the number of service requests. In this way, we do not need to re-implement the service customization process, but to reuse the history configuration, which promotes the calculation speed greatly and thus naturally slows down the average delay increasing rate of the proposed algorithm in the long run.

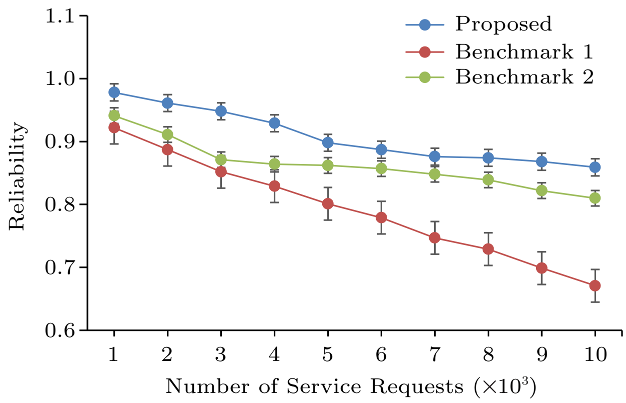
6.3.2 *Reliability*

The reliability evaluates the ability of algorithms on guaranteeing the service performance when suffering from network failures. The corresponding results are shown in Fig.5, where Fig.5(a) illustrates the results achieved in the CERNET2 topology and Fig.5(b) illustrates the results achieved in the Internet2 topology. It is easy to note that the proposed algorithm achieves better performance than the two benchmarks in both CERNET2 and Internet2. The main reason is that the proposed algorithm jointly takes the customer individual demands and the ISP profits into consideration when implementing the service customization. In particular, a balance between users and ISPs will be reached based on the Nash Equilibrium theory, such that the win-win solution can be finally provided. Under this condition, the resource is fully leveraged and more available resources will be used for reliability guarantee. Compared with the proposed algorithm, benchmark 1 offers the kind of best-effort services, which may waste the bandwidth sometimes and lead to unreliable situations. The function learning model used by benchmark 2 inevitably suffers from training errors, which directly leads to reliability problems.

In addition, another common phenomenon in both Fig.5(a) and Fig.5(b) is that the reliability decreases



(a)



(b)

Fig.5. Results of reliability. (a) Reliability vs the number of service requests in CERNET2. (b) Reliability vs the number of service requests in Internet2.

with the increasing number of service requests. The larger the number of the service requests, the more the bandwidth resource will be occupied. Once the available bandwidth resource cannot support the number of arriving requests, the performance bottleneck appears, which directly results in the decline in reliability. Moreover, by calculating the decreasing rate, we discover that the reliability of the proposed algorithm decreases slower than that of the other two benchmarks with the increasing of the number of service requests. In this regard, the proposed algorithm and benchmark 2 both introduce a reliability indicator when building their objective models, while benchmark 1 does not. Besides, the best-effort service model of benchmark 1 results in low resource utilization, which is another reason leading to this phenomenon. Meanwhile, the proposed algorithm and benchmark 2 both adapt the service path configuration according to users' preferences to maximize the reliability. However, due to the over-fitting issue of the learning model used by benchmark 2, its reliability is lower than that of the proposed algorithm.

6.3.3 User Satisfaction

The user satisfaction is related to many factors including the bandwidth, delay, jitter, packet loss rate and price as indicated by (4). We calculate this metric against different network loads and show the results in Fig. 6, where Fig. 6(a) illustrates the user satisfaction achieved in the CERNET2 topology and Fig. 6(b) illustrates the user satisfaction achieved in the Internet2 topology. In particular, we vary the network load from 0% to 100% to comprehensively test the performance of the proposed algorithm. As can be seen from Fig. 6(a) and Fig. 6(b), the proposed algorithm achieves the best performance. Meanwhile, the performance of benchmark 1 is the worst, since it mainly relies on using the integrated service model for service provision, and it simply rejects the service request once the amount of available resource is lower than that required. In contrast, the proposed algorithm and benchmark 2 use the differentiated service model to make a balance among different services and to ease the burden caused by the shortage of available resources. Despite this, there is a certain probability that benchmark 2 may lead to the over-fitting issue when using the back propagation algorithm to train the model. This situation is more obvious with the increasing of the network load, since the performance gap between the proposed algorithm and benchmark 2 becomes larger with the increasing of the network load.

Another two extreme conditions should also be noticed. Specifically, the first one is that when the network load reduces to zero (0%), the user satisfaction does not increase to 100%. This phenomenon is reasonable, because we need to make a balance between the users and the ISPs. In particular, the zero percent network load condition actually means that there are a lot of available resources which can be used to provide a better service quality for users. However, from the perspective of ISPs, they would want users to pay more for the high-quality services. Although the users are quite satisfied with the QoS provided, they may not be very happy about the price. QoS and the price are two critical factors affecting the user satisfaction. Hence, any one aspect not being fully considered will lead to the situation that the user satisfaction cannot reach 100% even when the network load becomes zero. On the other hand, when the network load is 100%,

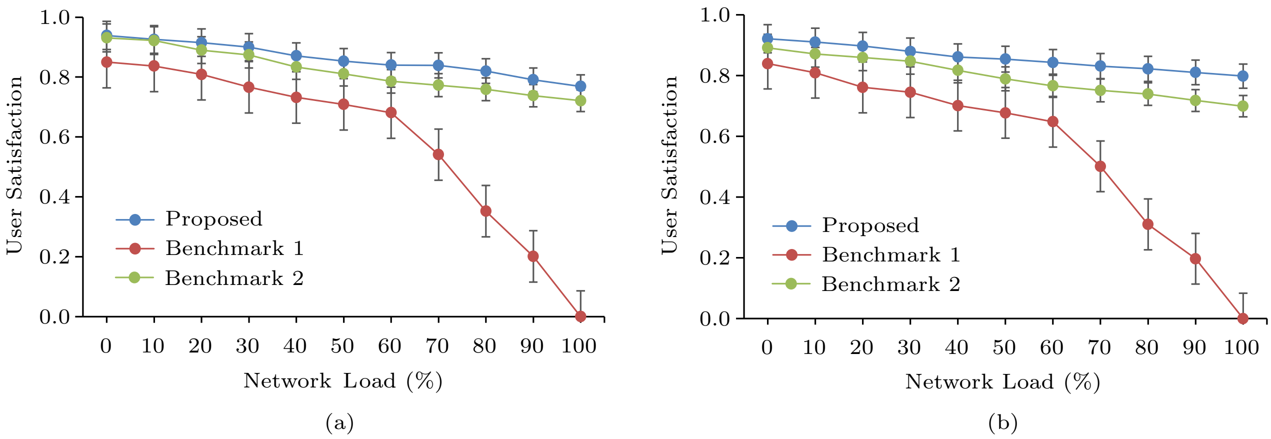


Fig. 6. Results of user satisfaction. (a) User satisfaction vs network load in CERNET2. (b) User satisfaction vs network load in Internet2.

the user satisfaction of benchmark 1 reduces to zero directly, since it will reject all the arrived service requests at the case. In contrast, due to the win-win solution offered by the proposed algorithm, its user satisfaction can remain stable (between 75% and 90%) when the network load changes from 0% to 100%.

6.3.4 ISP Satisfaction

The results of ISP satisfaction are shown in Fig.7, where Fig.7(a) illustrates the ISP satisfaction achieved in the CERNET2 topology and Fig.7(b) illustrates the ISP satisfaction achieved in the Internet2 topology. We can discover the similar phenomenon as indicated by Fig.6, that is, the proposed algorithm has the highest ISP satisfaction and benchmark 1 has the lowest ISP satisfaction. As explained, the proposed algorithm intends to achieve a win-win solution, which means that the proposed algorithm tries to not only maximize the profits, but also prevent users from rejecting the services of a high price. Such a design obviously improves the satisfaction of ISPs. However, different from the user satisfaction that goes down with the growth of the network load, the ISP satisfaction is actually on the opposite. Specifically, it increases with the growth of network load, because the higher the network load, the more the services offered by ISPs, and thus the more the profits that ISPs may gain. Besides, the proposed algorithm not only tries to maximize the profits, but also prevents users from rejecting the services due to the high price, which is another reason that the ISP satisfaction achieved by the proposed algorithm is higher than that by the benchmarks.

In order to clearly show the relationship between user satisfaction and ISP satisfaction, we evaluate the two metrics against the network load using the same algorithm. The corresponding results are shown in Fig. 8, where Fig. 8(a) compares the user satisfaction and the ISP satisfaction achieved by the proposed algorithm, Fig.8(b) compares the user satisfaction and the ISP satisfaction achieved by benchmark 1 and Fig.8(c) compares the user satisfaction and the ISP satisfaction achieved by benchmark 2. Apparently, it is noted that the changing trend of the user satisfaction is on the opposite of the ISP satisfaction, which means that we cannot achieve optimum results for both sides. Hence, a better compromise between them becomes important, which accords with the practical situation, since the user satisfaction and the ISP satisfaction are actually contradictory to a certain extent. Given the fixed network load, another phenomenon is that the sum of the user satisfaction and the ISP satisfaction does not equal 100%, which means that there is no clear relationship between them.

Furthermore, in order to show the stability of algorithms, we also calculate the difference between the user satisfaction and the ISP satisfaction achieved by these algorithms. Specifically, the difference fluctuation ranges of the proposed algorithm, benchmark 1 and benchmark 2 are [0.005, 0.167], [0.021, 0.801] and [0.02, 0.168] respectively. Hence, the stability of the proposed algorithm and benchmark 2 is similar and higher than that of benchmark 1, such that they are able to adapt to the dynamically changing network environments more easily.

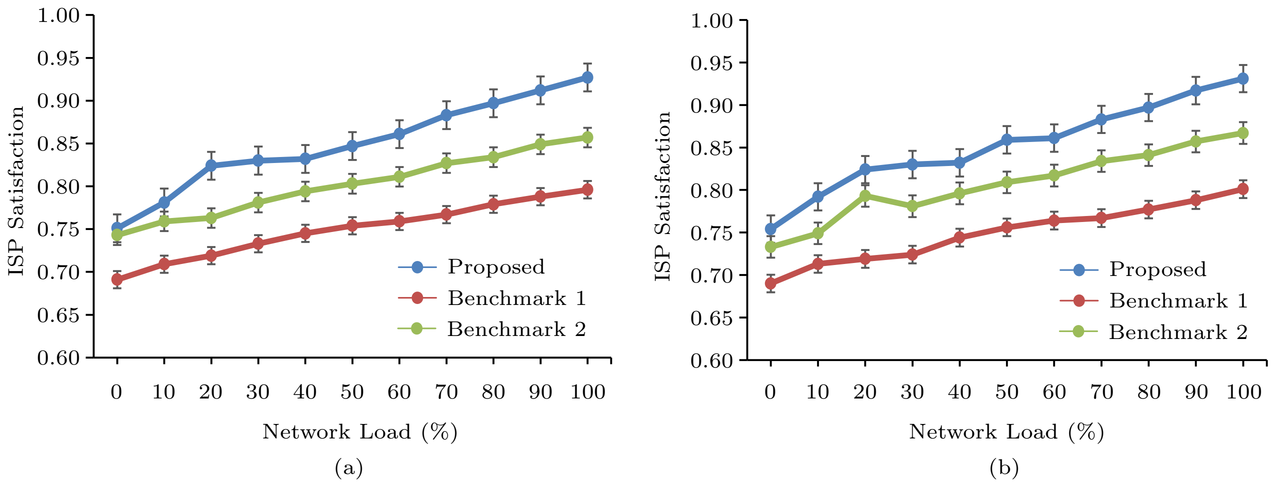
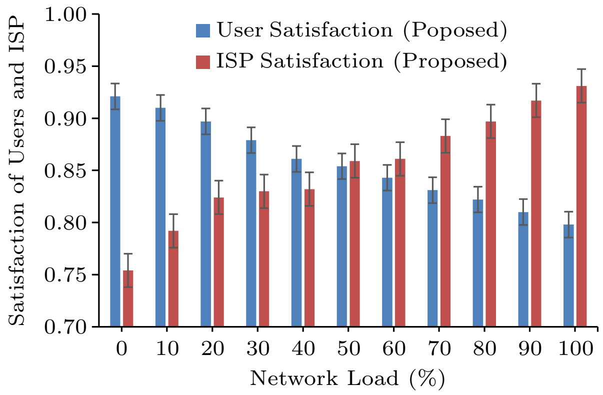
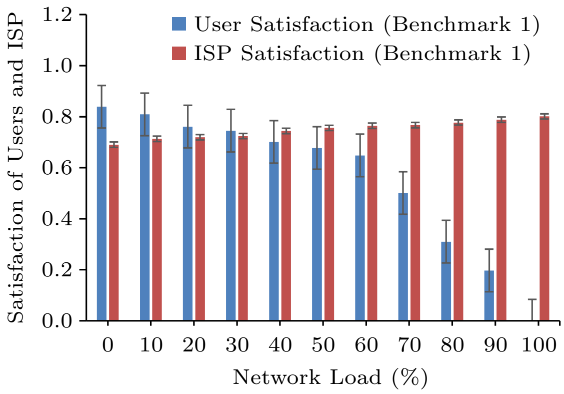


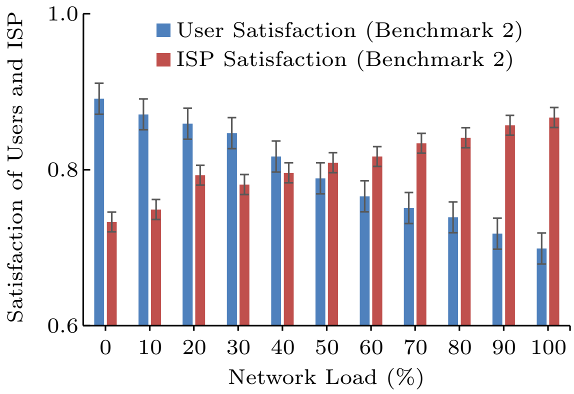
Fig.7. Results of ISP satisfaction. (a) ISP satisfaction vs network load in CERNET2. (b) ISP satisfaction vs network load in Internet2.



(a)



(b)



(c)

Fig.8. Comparison of user satisfaction and ISP satisfaction. (a) User satisfaction/ISP satisfaction vs network load of proposed algorithm. (b) User satisfaction/ISP satisfaction vs network load of benchmark 1. (c) User satisfaction/ISP satisfaction vs network load of benchmark 2.

7 Conclusions

This work intended to provide customized services for satisfying customers' individual demands. To fulfill this target, a service customization framework was designed, in which we proposed a service customization scheme and a grey wolf optimization based reliable routing mechanism. The system framework is responsible for providing the working environment and interactive interfaces for the service customization scheme and the reliable routing mechanism. In particular, the service customization scheme is responsible for classifying demanded services based on the analysis of user requirements, while the routing mechanism was proposed to transform the customized service into routing policies, during which the reliability is achieved. The experimental results indicated that the proposed mechanism could achieve better performance, improving the routing reliability by about 8.42%, the user satisfaction by about 15.5% and the ISP satisfaction by about 17.75% respectively. The future researches include exploring scalable and automatic service customization and routing by using intelligent technologies.

References

[1] Guck J W, Bemten A V, Reisslein M, Kellerer W. Unicast QoS routing algorithms for SDN: A comprehensive survey and performance evaluation. *IEEE Communications Surveys & Tutorials*, 2018, 20(1): 388-415. DOI: 10.1109/COMST.2017.2749760.

[2] Kreutz D, Ramos F M V, Veríssimo P E, Rothenberg C E, Azodolmolky S, Uhlig S. Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, 2015, 103(1): 14-76. DOI: 10.1109/JPROC.2014.2371999.

[3] Yi B, Wang X, Li K, Sajal D, Huang M. A comprehensive survey of network function virtualization. *Computer Networks*, 2018, 133: 212-262. DOI: 10.1016/j.comnet.2018.01.021.

[4] Sun G, Xiong K, Boateng G O, Ayepah-Mensah D, Jiang W. Autonomous resource provisioning and resource customization for mixed traffics in virtualized radio access network. *IEEE System Journal*, 2019, 13(3): 2454-2465. DOI: 10.1109/JSYST.2019.2918005.

[5] Liu S, Joe-Wong C, Chen J, Brinton C G, Zheng L. Economic viability of a virtual ISP. *IEEE/ACM Transactions on Networking*, 2020, 28(2): 902-917. DOI: 10.1109/TNET.2020.2977198.

[6] Li J, Shi W, Yang P, Shen X. On dynamic mapping and scheduling of service function chains in SDN/NFV-enabled networks. In *Proc. the 2019 IEEE Global Communications Conference*, Dec. 2019. DOI: 10.1109/GLOBECOM38437.2019.9013429.

[7] Gharbaoui M, Contoli C, Davoli G, Cuffaro G, Castoldi P. Demonstration of latency-aware and self-adaptive service chaining in 5G/SDN/NFV infrastructures. In *Proc. the 2018 IEEE Conference on Network Function Virtualization and Software Defined Networks*, Nov. 2018. DOI: 10.1109/NFV-SDN.2018.8725645.

[8] Yu X, Ye C, Li B, Zhou H, Huang M. A deep Q-learning network for dynamic constraint-satisfied service composition. *International Journal of Web Services Research*, 2020, 17(4): 55-75. DOI: 10.4018/IJWSR.2020100104.

[9] Tomovic S, Radusinovic I. Toward a scalable, robust, and QoS-aware virtual-link provisioning in SDN-based ISP networks. *IEEE Transactions on Network and Service Management*, 2019, 16(3): 1032-1045. DOI: 10.1109/TNSM.2019.2929161.

[10] Hu G, Xu K, Wu J, Cui Y, Shi F. A general framework of source address validation and traceback for IPv4/IPv6 transition scenarios. *IEEE Network*, 2013, 27(6): 66-73. DOI: 10.1109/MNET.2013.6678929.

[11] Li T, He T, Wang Z, Zhang Y. An approach to IoT service optimal composition for mass customization on cloud manufacturing. *IEEE Access*, 2018, 6: 50572-50586. DOI: 10.1109/ACCESS.2018.2869275.

[12] Kumara I, Han J, Colman A, Heuvel W V D, Tamburri D. FM4SN: A feature-oriented approach to tenant-driven customization of multi-tenant service networks. In *Proc. the 2019 IEEE International Conference on Services Computing*, Jul. 2019, pp.108-115. DOI: 10.1109/SCC.2019.00028.

[13] Bu C, Wang X, Zhang S, Huang M. Data-driven routing service composition via requirement chain. In *Proc. the 9th IEEE International Conference on Communication Software and Networks*, May 2017, pp.202-206. DOI: 10.1109/ICCSN.2017.8230106.

[14] Masruroh S U, Robby F, Hakiem N. Performance evaluation of routing protocols RIPng, OSPFv3, and EIGRP in an IPv6 network. In *Proc. the 2016 International Conference on Informatics and Computing*, Oct. 2016, pp.111-116. DOI: 10.1109/IAC.2016.7905699.

[15] Han L, Qu Y, Dong L, Li R. Flow-level QoS assurance via IPv6 in-band signalling. In *Proc. the 27th Wireless and Optical Communication Conference*, April 30-May 1, 2018. DOI: 10.1109/WOCC.2018.8372726.

[16] Bhattacharya A, Sen S, Sarkar A, Debnath N C. Hierarchical graph based approach for service composition. In *Proc. the 2016 IEEE International Conference on Industrial Technology*, March 2016, pp.1718-1722. DOI: 10.1109/ICIT.2016.7475022.

[17] Xiao A, Liu Y, Li Y *et al*. An in-depth study of commercial MVNO: Measurement and optimization. In *Proc. the 17th Annual International Conference on Mobile Systems, Applications, and Services*, Jun. 2019, pp.457-469. DOI: 10.1145/3307334.3326070.

[18] Li Y, Zheng J, Li Z, Liu Y, Qian F, Bai S, Liu Y, Xin X. Understanding the ecosystem and addressing the fundamental concerns of commercial MVNO. *IEEE/ACM Transactions on Networking*, 2020, 28(3): 1364-1378. DOI: 10.1109/TNET.2020.2981514.

[19] Moravejosharieh A, Ahmadi K, Ahmad S. A fuzzy logic approach to increase quality of service in software defined networking. In *Proc. the 2018 International Conference on Advances in Computing, Communication Control and Networking*, Oct. 2018, pp.68-73. DOI: 10.1109/ICACCCN.2018.8748678.

[20] Yan L, Mei Y, Ma H, Zhang M. Evolutionary web service composition: A graph-based memetic algorithm. In *Proc. the 2016 IEEE Congress on Evolutionary Computation*, Jul. 2016, pp.201-208. DOI: 10.1109/CEC.2016.7743796.

[21] Yang Y, Xu M, Li Q. Fast rerouting against multi-link failures without topology constraint. *IEEE/ACM Transactions on Networking*, 2018, 26(1): 384-397. DOI: 10.1109/TNET.2017.2780852.

[22] Bera S, Misra S, Vasilakos A V. Software-defined networking for Internet of Things: A survey. *IEEE Internet of Things Journal*, 2017, 4(6): 1994-2008. DOI: 10.1109/JIOT.2017.2746186.

[23] Papán J, Segeč P, Drozdová M, Mikuš L, Moravčık M, Hrabovský J. The IPFRR mechanism inspired by BIER algorithm. In *Proc. the 2016 International Conference on Emerging eLearning Technologies and Applications*, Nov. 2016, pp.257-262. DOI: 10.1109/ICETA.2016.7802053.

[24] Nobakht N, Kashi S S, Zokaei S. A reliable and delay-aware routing in RPL. In *Proc. the 5th Conference on Knowledge Based Engineering and Innovation*, Feb. 28-Mar. 1, 2019, pp.102-107. DOI: 10.1109/KBEI.2019.8734996.

[25] Behinfaraz R, Ghiasi A R. A survey on reliability analysis in controller design. In *Proc. the 14th IEEE International Colloquium on Signal Processing & Its Applications*, Mar. 2018, pp.198-202. DOI: 10.1109/CSPA.2018.8368712.

[26] Martínez-Peñas U, Kschischang F R. Reliable and secure multishot network coding using linearized reed-solomon codes. In *Proc. the 56th Annual Allerton Conference on Communication, Control, and Computing*, Oct. 2018, pp.702-709. DOI: 10.1109/ALLERTON.2018.8635644.

[27] Vignesh V, Premalatha K. Multi-QoS and interference concerned reliable routing in military information system. In *Advances in Big Data and Cloud Computing*, Rajsingh E B, Veerasamy J, Alavi A H, Peter J D (eds.), Springer, 2018, pp.351-360. DOI: 10.1007/978-981-10-7200-0_32.

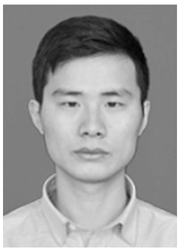
[28] Milolidakis A, Fontugne R, Dimitropoulos X. Detecting network disruptions at colocation facilities. In *Proc. the 2019 IEEE Conference on Computer Communications*, April 29-May 2, 2019, pp.2161-2169. DOI: 10.1109/INFOCOM.2019.8737615.

[29] Wang H, Gu M, Qi Y, Fei H, Li J, Yong T. Large-scale and adaptive service composition using deep reinforcement learning. In *Proc. the 15th International Conference on Service-Oriented Computing*, Nov. 2017, pp.383-391. DOI: 10.1007/978-3-319-69035-3_27.

Bo Yi received his B.S. and M.S. degrees in computer science from the South-Central University for Nationalities, Wuhan, in 2012 and 2015, respectively, and his Ph.D. degree in computer science from Northeastern University, Shenyang, in 2019. He is currently a lecturer with the College of Computer Science and Engineering, Northeastern University, Shenyang. His research interests include routing and service function chain in software-defined networking, network function virtualization, deterministic networking, and cloud computing.

Xing-Wei Wang received his B.S., M.S., and Ph.D. degrees in computer science from Northeastern University, Shenyang, in 1989, 1992, and 1998, respectively. He is currently a professor with the College of Computer Science and Engineering, Northeastern University, Shenyang. He has authored or coauthored more than 100 journal articles, books and book chapters, and refereed conference papers. His research interests include cloud computing and future Internet.

Min Huang received her B.S. degree in automatic instrument, M.S. degree in systems engineering, and Ph.D. degree in control theory from Northeastern University, Shenyang, in 1990, 1993, and 1999, respectively. She is currently a professor with the College of Information Science and Engineering, Northeastern University, Shenyang. She has authored or coauthored more than 100 journal articles, books, and refereed conference papers. Her research interests include modeling and optimization for logistics and supply chain system.



Qiang He received his Ph.D. degree in computer application technology from Northeastern University, Shenyang, in 2020. From 2018 to 2019, he was a visiting Ph.D. researcher with the School of Computer Science and Technology, Nanyang Technical University, Singapore. He has authored or coauthored more than 10 journal articles and conference papers. His research interests include social network analytic, machine learning, data mining, and software defined networking.