Journal of Engineering, Project, and Production Management
2026, 16(6), 2026-215

A Basic Element Analysis Model for Network Service Extensibility

Zuqin Ji[1] and Yuan Cheng[2]
[1] Lecturer, School of Network Security, Jinling Institute of Technology, Nanjing, China, E-mail: jizuqin@jit.edu.cn (corresponding author).
[2] Associate Professor, School of Network Security, Jinling Institute of Technology, Nanjing, China, E-mail: chengyuan2018@jit.edu.cn

Project Management
Received February 9, 2026; revised April 12, 2026; accepted June 3, 2026
Available online June 17, 2026

Abstract: In order to meet the increasing demand of emerging network applications, the extensibility of network services has become a key issue in future network research. Existing research on future networks lacks consideration of the basic elements that networks should have to realize service extensibility, and of the degree to which networks possess these elements, and cannot provide a unified metric for assessing whether networks have the basis for service extensibility. Furthermore, it analyzes the basic elements of network service extensibility, proposes methods for analyzing these elements, and then constructs a basic element analysis model for network service extensibility. Then, the availability of the analysis model is verified by assessing the extent to which the "extensible network service model" possesses the basic elements of service extensibility. Finally, the analysis model is applied to two typical future networks in order to identify the root cause of their insufficient service extension capability.

Keywords: Network service, service extensibility, basic element, analysis model.

DOI 10.32738/JEPPM-2026-215

1. Introduction

The continuous emergence of new network applications (Noor et al., 2018; Chin et al., 2019; Senyo et al., 2018; Liu et al., 2025; Song and Sun, 2025) has led to the inadequacy of traditional network service capabilities. To effectively solve this problem, we need to study from the perspective of network service extensibility. Although the traditional network has achieved service extensibility to a certain extent by adding protocols, this comes at the cost of reduced network system processing efficiency. Given the network's development status, research began to focus on improving the network architecture to enhance its service expansion capability (Geni; FIND;; FIRE; MOST). On the basis of these studies, although there have been some research results on future networks (such as network based on virtualization technology, service-oriented network, open programmable network and reconfigurable basic communication network), these future network research results still have deficiencies in service extensibility. The network's typical architecture is based on virtualization technology. Nebula (Anderson et al., 2010; Nebula White Paper, 2010) does not have the capability to store new services (only infrastructure resource services can be directly reused, and the reuse of all types of services cannot be realized). Therefore, when implementing service expansion, Nebula cannot have good service reusability. The typical architecture of service-oriented network SONA (Cisco, 2006) does not provide an open programming interface. It needs to customize service functions in advance and deploy them to the service pool before providing services. Therefore, Service-Oriented Network Architecture (SONA) lacks service programmability when implementing service expansion. The typical architecture of open programmable network SDN (Flores-de la Cruz et al., 2019; Kasabai et al., 2019; Souza and Neto, 2025; Abdi et al., 2024; Tache et al., 2024) makes it difficult to customize a variety of complex upper layer applications because its north interface has no unified control protocol. Therefore, SDN does not achieve full service completeness when implementing service expansion. The research on the typical architecture of the reconfigurable basic communication network FARI (Wang et al., 2016; Wang et al., 2015) focuses on the basic communication network, making it difficult to customize a variety of complex upper-layer applications. Therefore, FARI does not have complete service coverage when implementing service expansion.

The common problem with existing research on future networks is that it is insufficient for service expansion. Many network researchers neglect to explore the basic elements and analytical models of network service extensibility from the perspective of the network itself in their studies of future networks and their service extensibility. Whether the network fully supports the basic elements of service extensibility will directly affect whether it can effectively expand services. To determine whether the network has the basic elements of service extensibility and identify the aspects that need improvement, an analysis model is needed to evaluate these elements. Therefore, a model of the basic elements for network service extensibility is proposed in this paper. The specific work of this paper is as follows. First, through analysis and demonstration, the basic elements of service extensibility are identified, and a model of the basic elements for network service extensibility is proposed. Then, the analysis model's availability is verified by assessing the extent to which the "extensible network service model" possesses the basic elements of service extensibility. Finally, using the analysis model of basic elements for network service extensibility, two typical future networks are analyzed, and the root cause of their insufficient service extension ability is found.

The rest of this paper is organized as follows: In section one, the basic elements of network service extensibility are proposed, and then the analysis model is constructed. In section three, the availability of the analysis model of extensible basic elements for network services is verified. In section four, the analysis model is applied to two typical future networks to identify the root cause of their insufficient service extension capability. Section five concludes the paper and outlines future research directions.

2. Introduction to the Basic Elements of Network Service Extensibility and Its Analysis Model

2.1. The Basic Elements of Network Service Extensibility

Network Service Extensible (SE) means that the network does not sacrifice performance when it provides on-demand, dynamic service expansion. Therefore, this paper defines an ideal goal: realizing service extensibility so that the network can dynamically and optimally expand services on demand. To realize service expansion on demand, dynamically and optimally, it is necessary to identify the basic conditions (i.e., the basic elements of network service extensibility) that must be met to enable service extensibility from the network's perspective.

Facing new application requirements, traditional networks need to develop and deploy new protocols to provide services (that is, they cannot have service deployment, service programmability and reusability). The typical future networks also have their own shortcomings in expanding services. SDN and FARI are difficult to customize for a wide range of complex upper-tier applications (i.e., they lack service completeness). SONA needs to develop service functions in advance, deploy them to the service pool, and then provide services through dynamic composition according to application requirements (that is, it lacks service programmability). Nebula cannot store new services (that is, it cannot have good service reusability). Looking at the shortcomings of traditional and future network research in terms of service extensibility, we can identify four basic elements of network service extensibility, including service deployment, service programmability, service reusability and service completeness. If a network has these four basic elements of service s extensibility completely and thoroughly, it can realize service extensibility in a real sense. The demonstration is as follows.

Service Deployment (SD) (Riera Ferrer et al., 2015; Haas et al., 2003; Bornholdt and Röbert, 2025) means that any newly generated service can be deployed to the network. The difficulty of service deployment largely affects the implementation of service extension, so the service deployment must be realized if the network can achieve service extensibility (that is, $SE \Rightarrow SD$). Service Programmability (SP) (Mizuno et al., 1997) means that the network has an open programming interface through which extended service functions can be programmed. Whether the network has service programmability will determine whether it can efficiently and intelligently extend services to meet application requirements. Therefore, if the network can achieve service extensibility, it must be able to achieve service programmability (that is $SE \Rightarrow SP$). Service Reusability (SR) (Feuerlicht et al., 2007) means that the network can reuse existing service units or service instances during the implementation of service extension. Whether the network can reuse the existing service units or service instances as much as possible will affect the energy consumption during service extension. Therefore, if the network can achieve service extensibility, it must be able to achieve service reusability (that is $SE \Rightarrow SR$). Service Completeness (SC) means that any extended service function can be achieved in the network. Whether the network can provide any extended service functions in line with application requirements is an important aspect of its service extensibility. Therefore, if the network can achieve service extensibility, it must be able to achieve service completeness (that is $SE \Rightarrow SC$). From the above analysis, it can be seen that service deployment, service programmability, service reusability and service completeness are the necessary conditions to achieve network service extensibility (that is $SE \Rightarrow SD \cup SP \cup SR \cup SC$).

Because network service extensibility means that the network can highly achieve Dynamic Service Extension (DSE), On-demand Service Extension (OSE) and Service Extension with Superior Performance (SESP) at the same time, it can be concluded: $SE \Leftrightarrow DSE \cup OSE \cup SESP$. At the same time, if the network is able to achieve the capability of service deployment, service programmability, service reusability and service completeness, it can achieve service extension which is on-demand, dynamic and excellent performance (that is, $SD \cup SP \cup SR \cup SC \Rightarrow DSE \cup OSE \cup SESP$), its analysis is shown in Fig.1: If service deployment, service programmability and service completeness are realized in the network, dynamic service extension will be achieved. That is $SD \cup SP \cup SC \Rightarrow DSE$. If service programmability and service completeness are realized in the network, on-demand service extension will be achieved. That is $SP \cup SC \Rightarrow OSE$. If service programmability, service reusability and service completeness are realized in the network, service extension with superior performance will be achieved. That is $SP \cup SR \cup SC \Rightarrow SESP$. Therefore, the inference shows that the network can achieve service extensibility if service deployment, service programmability, service reusability and service completeness are realized at the same time in the network. That is $SD \cup SP \cup SR \cup SC \Rightarrow SE$.

In summary, service deployment, service programmability, service reusability and service completeness are the necessary and sufficient conditions for network service extensibility. That is $SD \cup SP \cup SR \cup SC \Leftrightarrow SE$, as shown in Fig. 1.

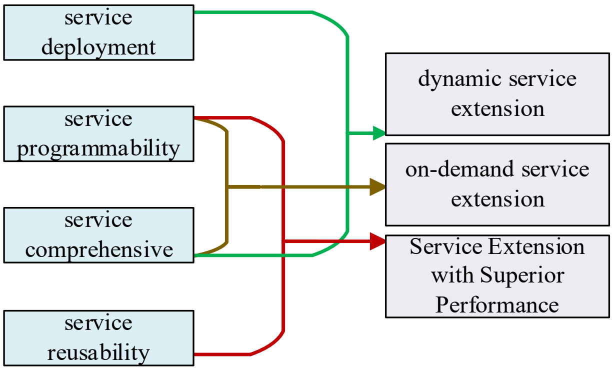


Fig. 1. Service extension features reflected by conditions of service extensibility

In Fig. 1, service deployment, service programmability, service reusability and service completeness are the necessary and sufficient conditions (i.e., basic elements) for network service extensibility. To achieve true service extensibility, the network should include the basic elements of service deployment, service programmability, service reusability and service completeness.

2.2. The Basic Elements of Network Service Extensibility

Whether the network has the basic elements of service extensibility will directly affect whether it can fully and effectively expand services. To assess whether existing networks possess the basic elements of service extensibility and identify areas for improvement, this paper proposes the Basic Elements Analysis Model of Network Service Extensibility (BEAM-NSE). The analysis model consists of four analysis methods of basic elements.

(1) Service deployment

When deploying network services, the cost of deployment time and energy consumption is called network service deployment overhead, and its size can reflect the degree of service deployment. The higher the cost of network service deployment is, the weaker the service deployment is; on the contrary, the stronger the service deployment is. Therefore, the deployment of network services can be evaluated by the magnitude of its deployment cost, which can be divided into the following situations: no update is required (no update is required in the actual deployment), and the magnitude of its cost is expressed as L_0; The configuration parameters need to be updated, and the cost level is expressed as L_1; The software needs to be updated, and its cost level is expressed as L_2; The hardware needs to be updated, and its overhead level is expressed as L_3. The magnitude diagram of service deployment is shown in Fig. 2, where $L_0 < L_1 < L_2 < L_3$.

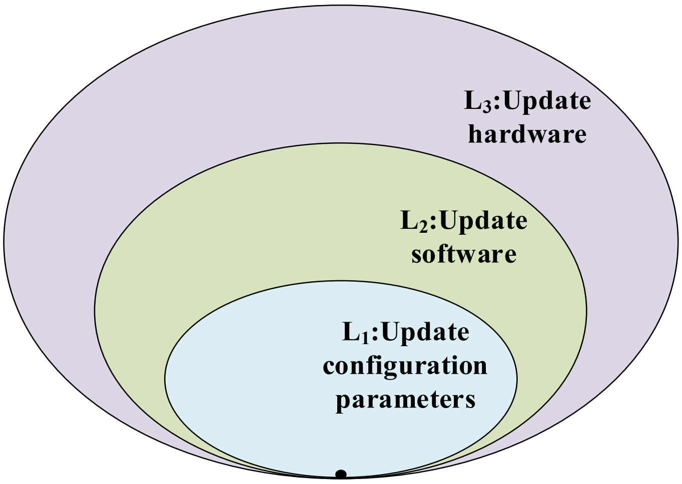


Fig. 2. Scale diagram of service deployment overhead

(2) Service programmability

There are many kinds of network applications, including “QoS”, “security”, “real-time”, and other complex business requirements, in addition to the basic communication business requirements. Therefore, the diversity and complexity of application business logic pose challenges to the standardization of service-programmable interfaces. The degree of unification of the service programmable interface will directly affect the realization of service programmability. There are many kinds of programmable control granularity (such as programming devices, protocols, service functions and minimum service function units). Different programming granularity will also directly affect service programmability. Therefore, service programmability can be evaluated using two indicators: the degree of unification of the service programmable interface and the control granularity of service programmability.

Since the unity degree of service programmable interface and the control granularity of service programmability are two related dimensions, the product of the two is used to express the degree of service programmability, which is expressed as Eq. (1). Where SP refers to service programmability; IU represents the unification degree of the service programmable interface, and its value is expressed by Eq. (2) (that is, if the interface is uniform, its value is 1, otherwise it is 0.); PG represents the control granularity of service programmability, and its value is expressed by Eq. (3) (that is, if the control granularity is the smallest functional unit, its value is 4. If the control granularity is the functional unit, its value is 3. If the control granularity is the protocol, its value is 2. Otherwise, its value is 1. From Eq. (1) to Eq. (3), it can be seen that the calculated value of SP is less than 4, and the closer it approaches 4, the better its service programmability.

$$SP = IU * PG \qquad \text{Eq. (1)}$$

$$IU = \begin{cases} 1 & \mathit{Unified} \\ 0 & \mathit{non-Unified} \end{cases} \qquad \text{Eq. (2)}$$

$$PG = \begin{cases} 1 & \text{equipments} \\ 2 & \text{protocols} \\ 3 & \text{functional unit} \\ 4 & \text{Minimum functional unit} \end{cases} \qquad \text{Eq. (3)}$$

(3) Service reusability

Services with small function granularity can be directly reused or instantiated; Services with large functional granularity are less likely to be directly reused and more likely to be reused after being instantiated. In instantiation reuse, the modifiability of service instances affects their reuse. Service function granularity affects the mode of service reuse; the maintainability of services, that is, the degree of easy modification, will directly affect the possibility of instantiation reuse. Therefore, the granularity of service functions and maintainability of services are chosen as the evaluation indicators of service reusability.

Since service function granularity and service maintainability are two different dimensions, the weighted sum of the two is used to express the degree of service reusability, as shown in Eq. (4), where SR refers to service reusability. SFG represents the granularity of service function, and its magnitude is expressed by Eq. (5) (that is, the granularity of service function is the smallest service unit, and its value is 2; otherwise, its value is 1). SM represents the service maintainability, and its magnitude is represented by Eq. (6) (that is, the value of service maintainability is 2 if it is good, and the value of service maintainability is 1 if it is bad, otherwise it is 0), and β_i represents the weight (i=1,2, $\beta_1 + \beta_2 = 1$). The weight is determined by their relative importance (if service function granularity and service maintainability are equivalent, set their weights to 0.5). From Eq. (4) to Eq. (6), it can be seen that the calculated value of SR is less than 2, and the closer it approaches 2, the better its service reusability is.

$$SR = \beta_1 * SFG + \beta_2 * SM \qquad \text{Eq. (4)}$$

$$SFG = \begin{cases} 1 & \mathit{Service\ Unit} \\ 2 & \mathit{Minimum\ Service\ Unit} \end{cases} \qquad \text{Eq. (5)}$$

$$SM = \begin{cases} 0 & \text{Non-maintainability} \\ 1 & \text{Poor maintainability} \\ 2 & \text{good maintainability} \end{cases} \qquad \text{Eq. (6)}$$

(4) Service completeness

When the network realizes service expansion, whether the service types it can provide are complete determines whether it has service integrity. Therefore, we use the service types of the existing traditional TCP/IP network as the reference benchmark. The evaluation index is whether the network can provide the service types specified in the TCP/IP model through service extension.

Taking the service types (4 types of services) of the traditional network TCP/IP model as the reference benchmark of service completeness. Assuming that *n* types of services can be provided when a network realizes service expansion. We can use SC to represent the complete service, and its judgment can be expressed by Eq. (7) (that is, if n<4, its value is 0; otherwise, its value is 1).

The evaluation index is whether the network can provide the service types defined by the TCP/IP model through service extension. Assuming that *n* types of services can be provided when a network realizes service expansion. We can use SC to represent the service completeness, and its judgment can be expressed in Eq. (7) (that is, if n<4, its value is 0; otherwise, its value is 1).

$$SC = \begin{cases} 0 & n < 4 \\ 1 & n = 4 \end{cases} \qquad \text{Eq. (7)}$$

Based on the above analysis, a model of the basic elements of network service extensibility can be derived, as shown in Table 1. The analysis model can provide a theoretical basis for designing future networks with service extensibility.

Table 1. Basic elements analysis model of service extensibility

Basic elements	Evaluation index	Evaluation measures
service deployment	deployment overhead	$SD=\begin{cases} L_0 & \text{No updates are required} \\ L_1 & \text{parameters need to be updated} \\ L_2 & \text{software needs to be updated} \\ L_3 & \text{hardware needs to be updated} \end{cases}$
service programmability	the unification degree of the service programmable interface; the control granularity of service programmability	$SP = IU * PG$
service reusability	service function granularity; service maintainability	$SR = \beta_1 * SFG + \beta_2 * SM$
service completeness	Four service types in TCP/IP model	$SC=\begin{cases} 0 & n<4 \\ 1 & n=4 \end{cases}$

3. Validation of BEAM-NSE

First, the BEAM-NSE theory is used to analyze the degree to which the network with ODSEM (Ji et al., 2019) has the basic elements of service extensibility. Then, the degree to which the network deploying the ODSEM prototype system possesses the basic elements of service extensibility is analyzed from the perspective of the network node's functional testing. Lastly, the availability of BEAM-NSE is verified by comparing the experimental and theoretical analysis conclusions.

3.1. Theoretical Analysis

The ODSEM has an atomic service library, and when new application requirements arise, it dynamically customizes atomic services and adds them to the library, enabling on-demand customization of service instances. Generating a new service involves only configuring service resources and related input and output parameters. Therefore, according to BEAM-NSE, its service deployment can be positioned at L1, and the network's ODSEM deployment is very good.

The ODSEM can pre statically customize atomic services and store them in the atomic service library, or dynamically customizes atomic services. The dynamic customization function can be implemented in the network using ODSEM, with a unified programming interface and a programming granularity of an atomic service (the smallest service function unit). Therefore, according to BEAM-NSE, SP=1*4=4, and the service programmability of the network with ODSEM is very good.

The ODSEM has two service libraries, namely, the atomic service library and the common instance service library. Atomic services can be directly reused due to the smallest granularity of service function units, and common instance services can be instantiated due to their large granularity. Since the importance of service function granularity and service maintainability is similar in ODSEM, their respective weights are set to 0.5. Since ODSEM supports dynamic programming (i.e., it provides a programming interface), its services are well maintained (i.e., they can be instantiated and reused). Therefore, according to BEAM-NSE: SR=0.5*2+0.5*2= 2, the service reusability of the network with ODSEM is very good.

The atomic service Library in ODSEM is formed by splitting, summarizing and abstracting the network protocol functions of each layer of the traditional network and then generating them statically in advance. When new application requirements arise, ODSEM dynamically customizes the missing atomic services and adds them to the atomic service library. Therefore, ODSEM can provide any type of service provided by the traditional Internet. According to BEAM-NSE, SC = 1, and the network with ODSEM has service integrity.

To sum up, the network with ODSEM shows a high degree of service extensibility, as demonstrated by BEAM-NSE analysis. The results are summarized in Table 2.

3.2. Experimental Analysis

(1) Experimental environment

The design of the experimental environment needs to consider the network scale and the hardware and software configurations of the network nodes. Network scale is primarily reflected in network topology and the number of network nodes. Changes in network topology and the number of network nodes will mainly affect the port traffic of network hardware, the size of the address table, and the utilization rate of network nodes. Because the basic elements testing whether the ODSEM prototype system has service extensibility is the function test on a single node, the network topology and the number of network nodes will not have a biased impact on the conclusion of the experiment (that is, the size of the network will not have a biased impact on the conclusion of the experiment). Therefore, the network topology shown in Fig. 3 is set in the experiment: Six PCs are selected as the network terminal nodes (n1-n6), and four servers are selected as the routers (R1-R4).

Table 2. Analysis of basic elements of ODSEM service scalability

Basic elements	Evaluation measures	Evaluation results
service deployment	$SD=\begin{cases} L_0 & \text{No updates are required} \\ L_1 & \text{parameters need to be updated} \\ L_2 & \text{software needs to be updated} \\ L_3 & \text{hardware needs to be updated} \end{cases}$	SD=L1, very good
service programmability	$SP = IU * PG$	SP=4, very good
service reusability	$SR = \beta_1 * SFG + \beta_2 * SM$	SR=2, very good
service completeness	$SC=\begin{cases} 0 & n<4 \\ 1 & n=4 \end{cases}$	SC=1, very good

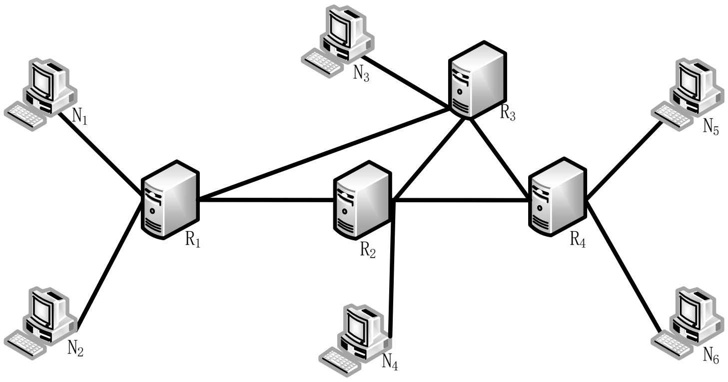


Fig. 3. Network topology of the experiment

Since the prototype ODSEM system cannot be deployed to the existing router or switch, the router and switch in the experiment use the servers to act as. All network nodes use WinPcap_4_1_3 and deploy the ODSEM prototype system. The main software and hardware configuration information of each node in the network topology shown in Fig. 3 is shown in Table 3. The network topology and node configuration of the experiment can meet the test requirements for performance and function.

Table 3. Main configuration information of network node software and hardware

network node	CPU	memory	operating system
N1-N6	Pentium(R4)	4G	Windows 7 Professional WinPcap_4_1_3 the prototype system of ODSEM
R1-R4	Intel Xeon E5-2600	16G	Windows server 2012 WinPcap_4_1_3 the prototype system of ODSEM

(2) Experiment contents and steps

The experiment content: using the prototype system of ODSEM in the network as shown in Fig. 4 to realize the customization and deployment of atomic services; the reuse atomic services and service composition are realized, then connection oriented transport services is customized and the connection establishment log of the service instance is checked; new atomic services is customized and service is composited, and then connection oriented data transmission service with encryption function is realized, the transmission data log of this service instance is checked lastly.

The specific steps of the experiment include:

<1> A customized routing service (Shen et al., 2019) is used to realize the routing between network nodes N2 and N4. Customize the atomic services required by connection oriented services in the ODSEM prototype system, and update the atomic service library of the ODSEM prototype system in each node of the network, as shown in Fig. 4. Customize

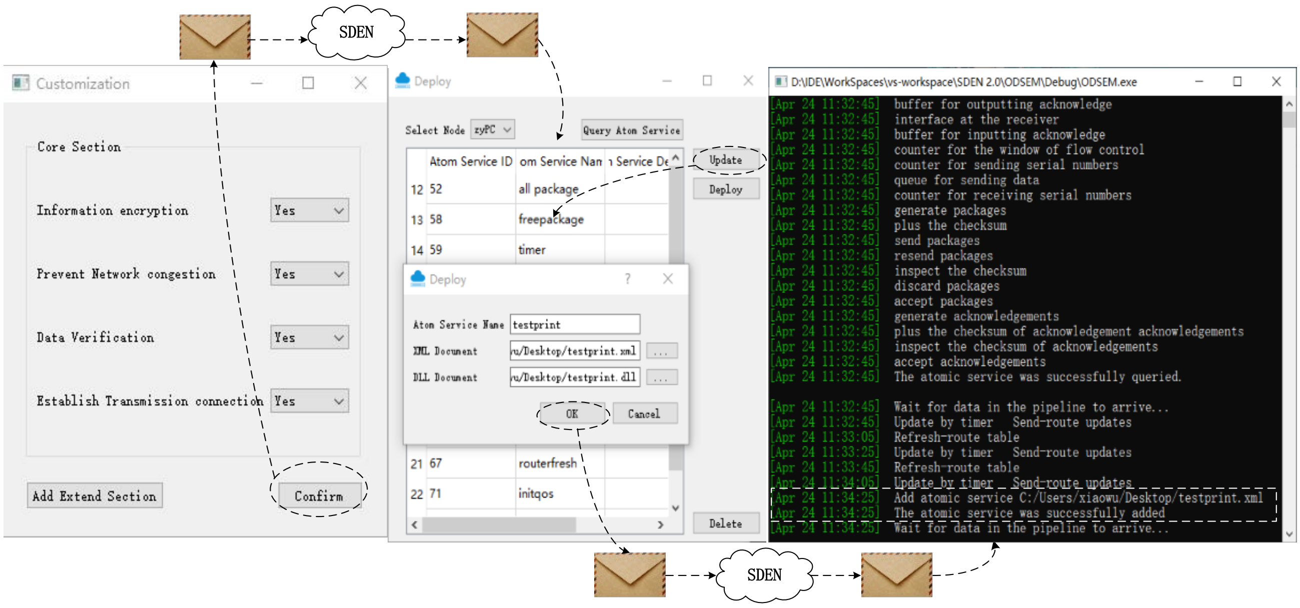
connection oriented network services on node N4. Next, combine services into a server to generate service instances and start monitoring. At last, observe changes in service status in system log records and record the service instance ID.

<2> At node N2, the customized connection oriented network service is realized by reusing atomic services, including construction message, message plus verification, sending message, verification message, receiving message and service combination and the service instance is generated in the form of a client. Use the name of node N4 and the service ID recorded in step 1 as the server-side connection details. Following this, start the customized service, establish a connection on the server side, and view the connection establishment log for the N2 node.

<3> Customize new atomic services with encryption functions such as key generation, key distribution, encryption and decryption in the ODSEM prototype system and update the atomic service library of the ODSEM prototype system in each node of the network, as shown in Fig. 4. The connection oriented transmission service between nodes N2 and N4 can be extended to secure transmission through service composition. Use the generated service instance on node N2 to send data to node N4, and view the data transmission log of node N4.

(3) Analysis of experimental results

According to Fig. 4, the network where the ODSEM prototype system is deployed has good service deployment and service programmability: The atomic service management module within the ODSEM prototype system supports on-demand programming for customizing atomic services, as well as their dynamic deployment to the atomic service repository. When an atomic service requires customization, it can be configured via the atomic service customization submodule. Upon completion of the customization process, the atomic service deployment submodule registers the newly defined atomic service into the atomic service library, and the ODSEM prototype system will then display a notification confirming successful addition of the atomic service.



Core Section

Information encryption: Yes

Prevent Network congestion: Yes

Data Verification: Yes

Establish Transmission connection: Yes

Select Node: zyPC

	Atom Service ID	om Service Nan	ı Service De
12	52	all package	
13	58	freepackage	
14	59	timer	
21	67	routerfresh	
22	71	initqos	

Atom Service Name: testprint

XML Document: u/Desktop/testprint.xml

DLL Document: u/Desktop/testprint.dll

```
[Apr 24 11:32:45]  buffer for outputting acknowledge
[Apr 24 11:32:45]  interface at the receiver
[Apr 24 11:32:45]  buffer for inputting acknowledge
[Apr 24 11:32:45]  counter for the window of flow control
[Apr 24 11:32:45]  counter for sending serial numbers
[Apr 24 11:32:45]  queue for sending data
[Apr 24 11:32:45]  counter for receiving serial numbers
[Apr 24 11:32:45]  generate packages
[Apr 24 11:32:45]  plus the checksum
[Apr 24 11:32:45]  send packages
[Apr 24 11:32:45]  resend packages
[Apr 24 11:32:45]  inspect the checksum
[Apr 24 11:32:45]  discard packages
[Apr 24 11:32:45]  accept packages
[Apr 24 11:32:45]  generate acknowledgements
[Apr 24 11:32:45]  plus the checksum of acknowledgement acknowledgements
[Apr 24 11:32:45]  inspect the checksum of acknowledgements
[Apr 24 11:32:45]  accept acknowledgements
[Apr 24 11:32:45]  The atomic service was successfully queried.

[Apr 24 11:32:45]  Wait for data in the pipeline to arrive...
[Apr 24 11:32:45]  Update by timer   Send-route updates
[Apr 24 11:33:05]  Refresh-route table
[Apr 24 11:33:25]  Update by timer   Send-route updates
[Apr 24 11:33:45]  Refresh-route table
[Apr 24 11:34:05]  Update by timer   Send-route updates
[Apr 24 11:34:25]  Add atomic service C:/Users/xiaowu/Desktop/testprint.xml
[Apr 24 11:34:25]  The atomic service was successfully added
[Apr 24 11:34:25]  Wait for data in the pipeline to arrive...
```

Fig. 4. Customizing and deploying atomic services

According to Fig. 5, the network with the ODSEM prototype system deployed has enhanced service reusability. Node N4 and node N2 customize connection oriented services in the form of server and client respectively, and multiple atomic services need to be reused in the compositions; From the connection establishment log of N2 node, it can be seen that the connection service is implemented between the two nodes, that is, the custom connection oriented service instance is implemented by reusing atomic services and service composition in this experiment.

According to the Fig. 6, the network where ODSEM prototype system is deployed has service completeness: by customizing new atomic services (atomic services with encryption function) and service composition (combining new atomic services with existing connection oriented services), connection oriented data transmission services with security function can be realized between node N4 and node N2, That is, ODSEM prototype system can realize various services by customizing atomic services and service composition.

Experiments verify that the network deployed with the ODSEM prototype system possesses the basic elements of service extensibility (i.e., service deployment, service programmability, service reusability and service completeness). The experimental analysis conclusion is consistent with the theoretical analysis conclusion of the network with ODSEM using BEAM-NSE, so the BEAM-NSE proposed in this paper is available.

```
[DEBUG]<StateTransLOG>@2020_04_24_19
:43:28: ClosedState ->
ActiveOpenState
[DEBUG]<StateTransLOG>@2020_04_24_19
:43:28: ActiveOpenState<InitState ->
NewBuffState>
[DEBUG]<StateTransLOG>@2020_04_24_19
:43:28: ActiveOpenState<NewBuffState
-> HeaderFillState>
[DEBUG]<StateTransLOG>@2020_04_24_19
:43:28:
ActiveOpenState<HeaderFillState ->
CheckSumState>
[DEBUG]<StateTransLOG>@2020_04_24_19
:43:29:
ActiveOpenState<CheckSumState ->
SendPacketState>
 [DEBUG]<StateTransLOG>@2020_04_24_19
      :43:29: ActiveOpenState ->
             SynSentState
[DEBUG]<StateTransLOG>@2020_04_24_19
:43:29: SynSentState<InitState ->
NewBuffState>
[DEBUG]<StateTransLOG>@2020_04_24_19
:43:29: SynSentState<NewBuffState ->
HeaderFillState>
[DEBUG]<StateTransLOG>@2020_04_24_19
:43:29: SynSentState<HeaderFillState
-> CheckSumState>
[DEBUG]<StateTransLOG>@2020_04_24_19
:43:29: SynSentState<CheckSumState
-> SendPacketState>
[DEBUG]<StateTransLOG>@2020_04_24_19
:43:29: SynSentState ->
EstablishedState
```

Fig. 5. Connection establishment log of N2 node

```
[DEBUG]<StateTransLOG>@2020_04_24_19:
43:29: EstablishedState ->
EstablishedState
[DEBUG]<StateTransLOG>@2020_04_24_19:
43:35: EstablishedState<RecieveState
-> NewBuffState>
[DEBUG]<StateTransLOG>@2020_04_24_19:
43:35: EstablishedState<NewBuffState
-> HeaderState>
[DEBUG]<StateTransLOG>@2020_04_24_19:
43:35: EstablishedState<HeaderState
-> InspectCheckSumState>
[DEBUG]<StateTransLOG>@2020_04_24_19:
43:35: EstablishedState<CheckSumState
-> AcceptPacketState>
[DEBUG]<StateTransLOG>@2020_04_24_19
:43:35:      EstablishedState       ->
EstablishedState
```

Fig. 6. Data transmission log of N4 node

4. Basic Elements Analysis of Service Extensibility of Typical Future Networks

In this section, we use BEAM-NSE to analyze two typical future networks and identify the root cause of the inadequate extensibility of typical network services.

4.1. SDN

SDN is a typical architecture of an open programmable network (Li et al., 2019; Singh et al., 2018). The initial goal of SDN is to change the communication infrastructure. SDN can centrally and flexibly control the network by separating the control and forwarding functions, and can quickly introduce new network services by providing programmable interfaces to expand network services.

Because SDN uses automation, newly generated services can be deployed dynamically. When new services are deployed in SDN, this involves updating the parameters of the SDN controller, SDN switches and SDN routers in the infrastructure layer. As newly generated services place higher demands on energy efficiency and security, a more robust dynamic deployment algorithm for an SDN controller and an SDN service migration algorithm will be needed to support service deployment. At this time, a software algorithm update will be involved. Deploying services in SDN rarely involves updating hardware devices. Therefore, according to BEAM-NSE, we can see that its service deployment can be positioned at the level of L_2, and SDN has good service deployment.

In SDN, there is a unified standard for the southbound interface (namely, OpenFlow technology), while there is no unified standard for the northbound interface. The programming granularity of the southbound interface in SDN can be as small as the smallest functional unit. In theory, the programming granularity of the northbound interface in SDN can be as small as the smallest functional unit. Because the unified degree and programming granularity of the programming interface of the north-south interface in SDN are inconsistent, and the unified degree and programming granularity of the programming interface are also inconsistent, the two interfaces should be analyzed differently, and the weight values of the two interfaces should be set to 0.5. Therefore, according to BEAM-NSE, we can see that: SP=0.5 * (1 * 4)+0.5 * (0 * 4)=2, SDN has good service programmability

SDN may add new services (including the smallest and largest functional units) to the control plane, enabling direct reuse and instantiated reuse. In SDN, the granularity of service function and service maintainability are two different dimensions, and their importance is similar, so the weight of each is set to 0.5. Since the smallest service granularity in SDN is the smallest functional unit, its service granularity is the smallest functional unit. Because SDN can open programmable interfaces, its services have better maintainability; that is, they can well realize instantiated reuse. Therefore, according to BEAM-NSE, we can see that SRU=0.5 * 2+0.5 * 2=2, and SDN has better service reusability.

Because SDN is originally designed to change the structure of the communication infrastructure to improve performance and energy efficiency, it is well suited to providing communication infrastructure services below the transport layer. In the face of a variety of upper layer network applications with complex logic (such as “QoS”, “security”, “real-time”), the open levels of northward interface in SDN (or from the application point of view, or from the user point of view, or from the operational point of view) are different, and there is no unified standard at present. This leads to SDN’s particularity in the implementation of upper-layer network applications, and it is not suitable for general network applications. Therefore, according to BEAM-NSE, we see that SG=0 and that SDN lacks service completeness.

4.2. SONA

SONA is a typical architecture of a service-oriented network (FIRE, n.d.). In SONA, the network is regarded as a huge service pool. It uses service addressing to enhance network intelligence, making the Internet a service system that integrates transmission, computing and storage. SONA uses a service composition strategy, a routing migration strategy, an intelligent network and a strategy for separating service identity from service location to realize the extension of network services.

Because SONA adds intelligent functions on the network side, it can dynamically deploy services when new services are generated. Because SONA clusters and encapsulates service logic modules with the same characteristics and designs collaborative interfaces for different cluster modules, deploying newly generated services will only change the parameters of infrastructure virtualization in the interactive service layer. Therefore, according to BEAM-NSE, we can see that its service deployment can be positioned at the level of L_1, and SONA has better service deployment.

Services are stored and deployed in the interactive service layer in SONA. Services are stored and deployed in the interactive service layer in SONA. SONA has no programming interface, which means new services are pre-generated and encapsulated according to application requirements, then added to the interactive service layer. Therefore, according to BEAM-NSE, we can see that SP=0 and that SONA does not have service programmability.

SONA can provide three types of services (basic, composite and processing), each with different granularity. Small-grained services (such as error checking, flow control and other services) in SONA can be reused directly. In SONA, services with higher granularity (such as data transmission services) are clustered and encapsulated (with poor maintainability), making it difficult to achieve instance reuse. In SONA, the granularity of service function and service maintainability are two different dimensions, and their importance is similar, so the weight of each is set to 0.5. Because the service granularity is not the smallest functional unit, SONA has only limited direct reuse capability; due to poor service maintainability, it is difficult to realize instantiation reuse. Therefore, according to BEAM-NSE, we can see that SR = 0.5*1 + 0.5*1 = 1, indicating that the service reusability of SONA is high.

SONA defines services as basic units and focuses on applying the principles of software design to design various service functions in the network to provide controllable, manageable and high-quality services. SONA can provide any service that the traditional Internet offers. Therefore, according to the basic elements analysis model of network service extensibility, we can see that SC=1 and SONA have service completeness.

In conclusion, by employing the BEAM-NSE framework to analyze the two typical future network architectures, we can derive the comparative information summarized in Table 4. According to the analytical results, although both network paradigms can achieve limited service extension, they do not adequately address the core technical bottlenecks in a targeted manner across several fundamental dimensions of service extensibility. Due to these inherent limitations, it remains challenging for such network designs to fully and effectively realize robust and reliable network service extensibility in practical deployment scenarios.

Table 4. Basic elements analysis of service extensibility for the two typical future networks

Basic element future networks	Service deployment	Service programmability	Service reusability	Service completeness
SDN	SD=L2, good	SP=2, good	SR=2, better	SC=0, poor
SONA	SD=L1, better	SP=0, poor	SR=1, general	SC=1, good

5. Conclusion

With the emergence of future technologies and applications in the network, the current Internet architecture faces many challenges, especially in extending network services. Given the current state of the network, many research institutes are devoting themselves to the research of future network architectures to improve the capability of service extension. And then, some research results have emerged. However, these research results do not effectively address the problem of weak service extension capability. The reasons are as follows: Researchers seldom realize that whether the network has the basic elements of service extensibility directly affects its ability to achieve high extensibility of network services.

Whether the network is fully equipped with the basic elements of service extensibility will directly affect whether it can fully and effectively realize service extension. Based on the analysis and demonstration of the basic elements of service extensibility, this paper proposes an analysis model of the basic elements of network service extensibility. Then, the analysis model is used to examine two typical future networks, and the root cause of their inadequate service extension capability is identified. The analysis model will play an enlightening role for future research on network service extension.

Existing network research results cannot address the problem of weak service extension capability. The analysis model of network service extensibility proposed in this paper systematically summarizes the key dimensions and core indicators that affect it. Based on the analytical model of the basic elements of service scalability developed in this paper, the next step will focus on the analysis and design of the future network architecture. By deeply exploring the core role of each scalable element in the model, combining with the needs of diverse application scenarios of future networks, we will optimize the functional modules, interaction logic and implementation path of the architecture, promote the in-depth adaptation between the network architecture and service scalability requirements and provide systematic technical support for the improvement of future network service scalability.

Author Contributions

Zuqin Ji contributed to conceptualization, methodology, software, validation, analysis, investigation, draft preparation, manuscript editing, supervision, project administration, and funding acquisition. Yuan Chen contributed to conceptualization, software development, validation, analysis, draft preparation, project administration, and funding acquisition.

Funding

This work is supported by Jinling Institute of Technology High-Level Talent Research Start-up Fund (jit-b-202108) Jiangsu Province Industry-University-Research Cooperation Project (BY2022588).

Institutional Review Board Statement

Not applicable.

Declaration of AI Tools

The authors confirm that no AI tools were used in the preparation of this manuscript.

References

Abdi, A. H., Audah, L., Salh, A., Alhartomi, M. A., Rasheed, H., Ahmed, S. and Tahir, A. (2024). Security control and data planes of SDN: A comprehensive review of traditional, AI, and MTD approaches to security solutions. *IEEE Access,* 12, 69941-69980.

Anderson, T., Birman, K., Broberg, R., Caesar, M., Comer, D., Cotton, C., Freedman, M., Haeberlen, A., Ives, Z., Krishnamurthy, A. and Lehr, W. (2010). Nebula-a future internet that supports trustworthy cloud computing. *White Paper*, 1-31.

Bornholdt, H., Röbert, K., Schulte, S., Edinger, J. and Fischer, M. (2025). Simplifying distributed application deployment at the edge through software-defined overlay networks. *Computer Communications*, 108294.

Chin, J., Callaghan, V., and Allouch, S. B. (2019). The Internet-of-Things: Reflections on the past, present and future from a user-centered and smart environment perspective. *Journal of Ambient Intelligence and Smart Environments,* 11(1), 45-69.

Cisco Systems. (2006). Service-oriented network architecture (SONA*)*.

Feuerlicht, G. and Wijayaweera, A. (2007). Determinants of service reusability. *FRONTIERS IN ARTIFICIAL INTELLIGENCE AND APPLICATIONS*, 161, 467.

FIND. (n.d.). Future Internet Network Design. http://find.isi.edu/

FIRE. (n.d.). Future Internet Research and Experimentation. http://cordis.europa.eu/fp7/ict/fire/

Flores-De La Cruz, A., Manzanares-Lopez, P., Muñoz-Gea, J. P., and Malgosa-Sanahuja, J. (2019). OpenFlow compatible key-based routing protocol: Adapting SDN networks to content/service-centric paradigm. *Journal of Network and Systems Management*, 27(3), 730-755.

GENI. (n.d.). Global Environment for Network Innovation. http://www.geni.net

Haas, R., Droz, P. and Stiller, B., 2003. Autonomic service deployment in networks. *IBM Systems Journal*, 42(1), 150-164.

Ji, Z. Q., Shen, J., Ding, D. L., and Cui, X. W. (2019). The analysis of service extensible capability for extensible network service model. *Journal of Communications,* 40(5), 38-46. (in Chinese)

Kasabai, P., Djmame, K., and Puangpronpitag, S. (2019). Priority-based scheduling policy for OpenFlow control plane. *KSII Transactions on Internet and Information Systems,* 13(2), 733-750.

Li, P., Guo, S., and Pan, C. (2019). Fast congestion-free consistent flow forwarding rules update in software-defined networking. *Future Generation Computer Systems,* 97, 743-754.

Liu, F., Farkiani, B., and Crowley, P. (2025). Large language models for computer networking operations and management: A survey on applications, key techniques, and opportunities. *Computer Networks,* 271(2), 148-159.

Mizuno, O., Shibata, A., and Okamoto, T. (1997). Models for service management programmability in advanced intelligent network. *IEICE Transactions on Communications,* 80(6), 915-921.

MOST (Ministry of Science and Technology of China) (2007). New generation of trustworthy network. http://www.most.gov.cn/tztg/200711/t20071107_57004.html (in Chinese).

Noor, T. H., Zeadally, S., and Alfazi, A. (2018). Mobile cloud computing: Challenges and future research directions. *Journal of Network and Computer Applications,* 115, 70-85.

Riera, J. F., Hesselbach, X., Zotkiewicz, M., Szostak, M., and Botero, J. F. (2015). Modelling the NFV forwarding graph for an optimal network service deployment. In *Proceedings of the 17th International Conference on Transparent Optical Networks* ,1-4.

Senyo, P. K., Addae, E., and Boateng, R. (2018). Cloud computing research: A review of research themes, frameworks, methods and future research directions. *International Journal of Information Management,* 38(1), 128-139.

Shen, J., Zhou, X., and Ji, ZQ. (2019). Implementation of service dynamic extended network and its node system model. *Journal of Jilin University (Engineering Edition),* 49(6), 2058-2067 (in Chinese).

Singh, D., Ng, B., and Lai, Y. C. (2018). Modelling software-defined networking: Software and hardware switches. *Journal of Network and Computer Applications,* 122, 24-36.

Song, B., Sun, B., Fu, Q., and Li, H. (2025). A protocol-independent in-network security service for cloud applications. *Journal of Network and Computer Applications*, 104368

Souza, C. H., Pascoal, T., Neto, E. P., Sousa, G. B., SL Filho, F., Batista, D. M., and Silva, F. S. D. (2025). SDN-based solutions for malware analysis and detection: State-of-the-art, open issues and research challenges. *Journal of Information Security and Applications*, 93,104145.

Tache, M.D., Păscuțoiu, O. and Borcoci, E., (2024). Optimization algorithms in SDN: Routing, load balancing, and delay optimization. *Applied Sciences*, 14(14), 5967.

Wang, P., Lan, J., and Hu, Y. (2015). Towards locality-aware DHT for fast mapping service in future Internet. *Computer Communications,* 66, 14-24.

Wang, P., Lan, J., and Zhang, X. (2015). Dynamic function composition for network service chain: Model and optimization. *Computer Networks,* 92(2), 408-418.

Zuqin Ji earned her PhD in Computer Science and Engineering (2020) from SEU, Southeast University. Presently, she is a Lecturer at the School of Network Security, Jinling Institute of Technology, Nanjing. She presided over several projects and published articles in internationally renowned journals and conference proceedings. Her research interests include peer-to-peer systems, network architecture, service scalability and information security.

Yuan Cheng obtained his PhD in Communication and Information Systems (2010) from SEU, Southeast University. He is currently an associate professor at the School of Cyber Security, Jinling Institute of Technology, Nanjing. He led several technology projects and published multiple academic papers in journals at home and abroad. His research interests include wireless communications, information security, and artificial intelligence.