IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 1, 2017, accepted December 29, 2017, date of publication January 3, 2018, date of current version March 9, 2018.

Digital Object Identifier 10.1109/ACCESS.2017.2789178

The Modeling and Analysis of the Extensible Network Service Model

ZUQIN JI[1,2,3], JUN SHEN[1,2], DELIN DING[1,2], AND XIAOWEI CUI[1,2]

[1]School of Computer Science and Engineering, Southeast University , Nanjing 210096, China
[2]Key Laboratory of Computer Network and Information Integration of Ministry of Education, Southeast University, Nanjing 210096, China
[3]School of Computer Science and Engineering, Yancheng Teachers University, Yancheng 224002, China

Corresponding author: Jun Shen (junshen@seu.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61370206.

ABSTRACT With the increasing number of new application requirements in network, the capability of providing services in traditional network is being challenged constantly. The network service model, which is the core of the network architecture, directly determines the capability of providing network services. However, there lacks the explicit service model about the research of the next generation network architectures. We propose the extensible network service model (ENSM). In this paper, the basic principle of the ENSM is summarized. Then the mathematical modeling of the ENSM is provided and, based on the mathematical modeling, we analyze the functional description and the performance calculation for composite services when a service composition instance is implemented. Finally, through the experimental modeling and the simulation, the performances of the ENSM are compared with those of the traditional network when providing the same service. The experimental results show that the services can be customized or composed flexibly without sacrificing the performances in the extensible network service model. It can also be seen that the model can implement service extension better and the model idea is correct.

INDEX TERMS Network performance, network service model, service composition, next generation network, service extensibility.

I. INTRODUCTION

In the early stage of network technology, more research focused on how to avoid the complex design of communication protocols so as to provide a guarantee of better network performance. With the rapid development of the computer network [1], a variety of new network technologies [2] and user application requirements [3] emerge in an endless stream [4]. The capability of customizing and extending services becomes an obvious problem in the traditional network systems: First, communication can be simplified through the layered design in the traditional network, but new service functions can only be generated through developing and deploying new protocols, which is a static extension method. And, it is a long period to develop and deploy new protocols. Second, since the layers are independent with each other, there are many functional redundancies (such as repetitive error detection in several layers, repetitive data fragmentation in several layers etc.) between the new protocols and the old protocols, which not only affect the network performance but also restrict the capability of providing services. Due to the current situation of network development, more and more research organizations began to pay close attention to the new network architectures and their service model, such as GENI [5], FIND [6], PlanetLab [7] programs which were launched by the United States, the FIRE project [8] which was launched by the European Union, and the ''New Generation of Trustworthy Network'' [9] which was launched by China. Similar research programs were also carried out by Japan, Australia and South Korea.

Based on these studies, there are a number of research results about new network architecture models, such as the design of no layer, the design of cross layers, the uniform design in each layer. These results are aimed at improving the network service extensibility. Network service extensibility refers to the capability of providing new service functions on the premise that the network performance does not obviously decrease. But, there are no explicit service model in these new research results. The research results can't fundamentally solve the core problem of service extensibility in network. RBA [10] breaks down the idea of hierarchical architecture and takes the ''role'' as a communication entity. Hence, RBA can avoid the redundancies among the layers in traditional network. But the size of the "role" which is as a service function unit is too lager to do well in servicer composition. So, it's still poor in service extensibility. SILO [11] is also a non-hierarchical architecture, and service in the network model can be customized according to the needs. But the partial ordering of the service units is hard to define. And it is difficult to customize service units. Therefore, it is still weak in the aspect of service extensibility. RNA [12], [13] gives the idea that the protocols among layers can be reused. And they propose the concept of meta protocol which can be deployed in the protocol stack. The number of RNA's layers can be changed dynamically. And RNA can eliminate redundant services of the traditional network. However, the granularity of the meta protocol is so large that it is not conducive to service customization or service composition. Moreover, the dynamic stack is difficult to achieve. So, it's also weak in service extensibility. In the papers [14] and [15], service unit based network architecture is proposed. In this network model, the service units are independent and can avoid the redundancy of functions effectively. But, the granularity of service units is too large to realize the service extensibility easily. SDN [16] mainly consists of three layers: data plane, control plane and application plane. The communication between every two of the layers is through a unified interface (such as OpenFlow [17]). The research and application of SDN mostly is limited to the network and lower lays. The difference of network applications in upper layer makes SDN have no uniform northbound interface standard. OpenFlow is a universal Southern interface specification for SDN. However, its protocol rule is not mature and the version is still updated [18]. These problems have seriously affected the service extensibility of SDN. In short, the existing research results of new network architecture can't solve the problem of network service extensibility well; the design contents of a network architecture cover many aspects, while the body of the "contradiction" (service extensibility) should be focused on firstly.

Our team proposes an extensible network service model (ENSM) which does well in network service extensibility. In the research, we set "network service extensibility" as focus. In the paper [19], we known that services can be dynamically composed in an on-demand way with high performance in ENSM. However, the paper lacks modeling and some in-depth research which is based on modeling. In order to further study and analyze ENSM, we give the modeling of ENSM and analyze the functional attributes and the performance attributes for composite services in this paper. The key mathematical symbols used in this article are descripted in Table 1. The specific work of this paper is: first, the basic principle of the extensible network service model is summarized; and then, the mathematical modeling of the extensible network service model is provided; on the basis of the mathematical modeling, we analyze the functional description and the performance calculation for composite services; finally, through experimental modeling and simulation, the performances of ENSM are compared with those of the traditional network when providing the same service.

TABLE 1. The descriptions of the key mathematical symbols.

symbols	S	C_k	1,…,i,…, m	1,…,j,…,n	X
Description	service	the k-th service category	the amount of service	the amount of service attributes	a process in CSP
symbols	q	y	W_i	Z_i	x_j
Description	an event in CSP	an element of the functional input set	a waiting delay	a processing delay	a performance attribute
symbols	t	P	s	T	THP
description	the transition in SPN model	the probability	the place in SPN model	response time	throughput

The rest of this paper is organized as follows: in section 2, we introduce the theory of ENSM; then, we carry out modeling for ENSM in section 3; we describe the functional attributes for composite services in section 4; in section 5, the performance attribute functions for composite services are defined; in section 6, through experimental modeling and simulation, the performances of ENSM are compared with those of the traditional network when data transmission service is provided; finally, section 7 concludes the paper and points out the future research directions.

II. ENSM

Our team proposes ENSM which does well in service extensibility. We set "service extensibility" as focus in the research. ENSM breaks down the idea of hierarchical architecture. In ENSM, by decomposing and abstracting the basic functions of the traditional network protocols, each smallest network service function is modularized and all of these smallest network service function modular form a super large atomic service set. ENSM not only reduces the functional redundancies but also has good service extensibility by customizing atomic services and service composition.

The working mechanism of ENSM is described as follows: First, each smallest network service function is modularized by decomposing and abstracting the basic functions of the traditional network protocols, and all of these smallest network service function modular form a super large atomic service set. (Atomic services refer to the smallest entity unit with independent functions; The set of atomic service is growing along with the application requirements; Examples of categories and functions for atomic services are shown in Table 2); Second, the set of relations among the atomic services are obtained by analyzing, summarizing and abstracting the traditional network protocols (such as dependency relation, interactive relation etc.); Finally, based on the set of atomic services and the set of relations among the atomic services, dynamic expansion of services can be implemented (That is, the new atomic services can be customized and added into the atomic service set dynamically; and, atomic services can be selected and composed dynamically).

TABLE 2. The categories and the function of atomic services.

categories	functions
system services	buffer memory/ timer/…
Message services	Construct Message / message fragmentation / message reorganization / Revise message / Send message/ Receive message / extracting data from the message/…
routing services	Build neighbors / Keep the neighbors alive /Query the routing table/ Revise the entry of the routing table/ Delete the entry of routing table/ Add an entry to routing table/…
QoS services	error control/ jitter control/ congestion control/ priority processing/…
security services	packet encryption/ decryption/ packet filtering/…
…	…

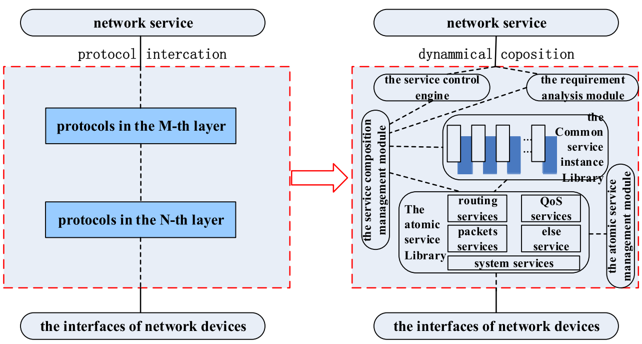

FIGURE 1. The comparison of the mode of providing services.

The comparison of the mode of providing services between ENSM and the traditional network is shown in figure 1: The traditional network provides services through the protocols in one or several layers While ENSM provides services by customizing atomic services and service composition. As shown in fig.1, ENSM is under network application and is above network hardware, which includes four functional modules (“the requirement analysis module”, “the service composition management module”, “the atomic service management module” and “the service control engine”) and two service libraries (“the common service instance Library” and “the atomic service Library”). The process of service extension in ENSM: (1) Admin users deploy “the atomic Service Library” and “the common service instance Library” to nodes and update service information in time. (2) Business users submit service requests according to their individual requirements. (3) The business users’ requirements will be converted into the semantic expression through “the requirement analysis module” and transmitted to “the service composition management module”. (4) According to the semantic expression, “the service composition management module” searches for a solution of the composition scheme from “the common service instance Library”. If there is no composition scheme, “the service composition management module” will select the appropriate atomic services from “the atomic service Library” (If there is no the required atomic services in “the atomic service library”, “the atomic service management module” will customize the atomic services and add them to “the atomic service Library”.) and generate a service composition scheme. And then, “the service composition management module” will send the composite scheme to “the service control engine”. (5) The composite service can be achieved after “the service control engine” obtain the service extension scheme and the corresponding atomic services. At last, the composite service instance is transmitted to the user.

III. SERVICE MODEL

In order to analyze the service functional description of ENSM and the service performance calculation of ENSM, the follow research on the characteristics of ENSM and on the service extensibility of ENSM, we need to model ENSM. In ENSM, all of the service functions can be achieved by customizing atomic services or composing services. Here, the modeling for ENSM is given, which includes the abstractions for atomic service and for composite service respectively.

A. THE ABSTRACTION FOR ATOMIC SERVICES

Atomic services refer to the smallest entity unit with independent functions. The abstraction for atomic services is a unified mathematical description, which includes the description of the functional attributes, the performance attributes, the service categories, ect. The abstraction for atomic services is given, as shown in formula (1).

$$S_i =< C_K, L, R, FA, PA, RIQ_K > \tag{1}$$

Where, S_i $(i = 1, 2, \ldots, m)$ indicates the name of the atomic service.

C_K represents the category of atomic services. There are different categories according to the difference of service functions. And they can be expressed as a set, $\{\mathrm{C}_1, \mathrm{C}_2, \mathrm{C}_3, \ldots, \mathrm{C}_\mathrm{K}, \ldots, \mathrm{C}_\mathrm{n}\}$.

L indicates the quantity of the network nodes which the atomic service is involved in. When it’s value is *0*, it represents the atomic service is involved to one network node; when it’s value is *1*, it represents the atomic service is involved to several network nodes.

R stands for the reliability of the atomic service.

FA denotes the functional attributes of an atomic service, which can be written as a tuple, namely $< IN, \psi >$. Where, IN is defined as the input functional parameters. It can be represented as a set; ψ is defined as functional operations of an atomic service. It can be expressed by using AS-CSP [20].

PA indicates the performance attributes of an atomic service. And it can be expressed as a tuple, namely $< tag, PV, WV, F_{WV}(x_j) >$. Where, tag expresses the mark vector of the performance attribute, namely $(tag_1, tag_2, \ldots, tag_n)$; PV denotes the vector of the value of the performance attributes, namely $(x_1, x_2, \ldots, x_n)$; *WV* means weight value vector of the performance attributes, and it will be set when the merits of the composite service is evaluated; $F_{WV}(x_j)$ denotes the function of weight value according to the preference for performance x_j $(1 \leq j \leq n)$ or personality demand of application examples.

RIQ_K represents a set of queues which stores the resource information for the atomic service. It can be written as $\{RIQ_1, RIQ_2, \ldots, RIQ_k\}$, where RIQ_k expresses the k-th kind of resource information.

B. THE ABSTRACTION FOR COMPOSITE SERVICES

Composite services mean that according to the different demand, the appropriate atomic services are selected and composed into a logic unit. The abstraction for composite services is a unified mathematical description, which includes the functional attributes of the composite service, the performance attributes of the composite service, the relationship among the atomic services which participate in the composition, ect. The abstraction for composite services is given, as shown in formula (2).

$$S_{Ci} =< SUL, L, R, CM, RE, FA, PA, RIQ_K > \tag{2}$$

Where, $S_{Ci}(i = 1, 2, \ldots, m)$ indicates the name of composite services.

SUL shows the list of atomic services which participate in the composition. And it can be expressed as a set, namely $\{S_1, S_2, \ldots, S_m\}$.

L indicates the quantity of the network nodes which the composite service is involved to. When the value is *0*, it represents the composite service is involved to one network node; when the value is *1*, it represents the composite service is involved to several network nodes.

R stands for the reliability of the composite service.

CM represents the composition approach of composite services and it can be expressed in a process algebraic expression, such as $S_1 \oplus (S_2 \| S_3)$.

RE indicates the relation among the atomic services which participate in the composition. Dependency relation and interactive relation are the most basic relationships among them. Dependency relation is used to describe the manner of dependence among services when there is a service interaction, namely $\mathrm{D\_SET} = \{< \mathrm{C}_1, \mathrm{C}_2 >, \vec{r}, \mathrm{v}\}$. C_1 and C_2 are categories of atomic services. $< \mathrm{C}_1, \mathrm{C}_2 >$ is the constraint for the two service categories in a dependency relation. $\vec{r}$ is a vector for dependency rules, and it is optional. If $\vec{r}$ exists, the dependency relation is directed. Otherwise it is undirected. v is the cost of dependency relation. Interactive relation is used to describe the services which are in one interaction, namely $\mathrm{I\_SET} = \{\mathrm{S}_1, \mathrm{S}_2, \ldots\ldots, \mathrm{S}_\mathrm{m}\}$.

FA denotes the functional attributes of a composite service, which is composed with atomic service functions. It can be expressed with AS-CSP.

PA indicates the performance attributes vector for a composite service, namely $(FP_1(x_1), FP_2(x_2), \ldots, FP_j(x_j), \ldots, FP_n(x_n))$, $(1 \leq j \leq n)$. Here, x_j represents the variable of the j-th performance attributes of m atomic services; $FP_j(x_j)$ denotes the function which can compute the j-th attribute of the composite service.

RIQ_K represents a set of queues which stores the resource information for the composite service. It can be written as $\{RIQ_1, RIQ_2, \ldots, RIQ_k\}$, where RIQ_k expresses the k-th kind of resource information.

IV. THE DESCRIPTION OF THE FUNCTIONAL ATTRIBUTES FOR COMPOSITE SERVICES

When a service composition instance is implemented, the functional construction relationship between the composite service and the atomic services can be expressed by the description of the functional attributes for the composite service. Hence, based on the mathematical modeling, we give the description of the functional attributes for composite services.

A. THE FORMAL DESCRIPTION OF THE FUNCTIONAL ATTRIBUTES

The input set of functional attributes for an atomic service varies along with the different composition scenarios. The input set of functional attributes for an atomic service is defined as follows.

Definition 1: the input set of the functional attributes for an atomic service. Assume that Y is an input set of the functional attributes for an atomic service, y is an element of the input set, X is a process, αX is the alphabet of the process. Then, Y can be expressed as $Y = \{y | y \rightarrow X, y \in \alpha X\}$.

The formal description of the composite service is based on the description of atomic service, while the input set of the functional attributes for an atomic service is different along with its different composition scenarios. For an example, the atomic service "construct message" is used in different scenarios to construct different messages.

Therefore, it is necessary to define the formal description method of atomic service. The definition is given as follows.

Definition 2: the formal description method of atomic service based on the input set (AS-CSP). Assume that the atomic service process P will be executed if the input set of an atomic service in a composite scene is $\{y_1, y_2, \ldots\}$ and the atomic service participates in the event q. Then, the formal description of the atomic service which is in the composite scene can be expressed as $q\{y_1, y_2, \ldots\} \rightarrow P$; When the atomic service doesn't take part in a composition, the formal description of the atomic service can be expressed as $q\{\} \rightarrow P$.

B. A FORMAL DESCRIPTION OF THE COMPOSITE SERVICES

Taking a simple service "dynamic routing" as an example, we give the formal description of the functional attributes for the composite service. The composite service "dynamic routing" uses the following atomic service functions: "extracting data from the message", "Revise the message", "Query the routing table", "Revise the entry of the routing table", "Delete the entry of the routing table", "Add an entry to the routing table'', ''Timer''. The formal description of these atomic services which is based on the input set is shown as the formulas (3-9).

$$q\{\} \rightarrow msg.Extract \tag{3}$$

$$q\{\} \rightarrow msg.Revise \tag{4}$$

$$q\{\} \rightarrow Router.Query \tag{5}$$

$$q\{\} \rightarrow Router.Revise \tag{6}$$

$$q\{\} \rightarrow Router.Delete \tag{7}$$

$$q\{\} \rightarrow Router.Add \tag{8}$$

$$q\{\} \rightarrow Timer \tag{9}$$

The service ''dynamic routing'' can be composed of several atomic services. As shown in fig.2, the formal description model of the service based on the input set can be divided into three modules: input, output and timer control.

```
Input = μ X.(left?Msg ⟶ DRequest ⋞msg = request ⋟
                    (DResponse  ⋞msg = response⋟ X))
DRequest=Initialmsg    ⟶    μ    X.(Initialmsg{msg.entry    ≠    <
          >} ⟶ msg.Extract;  //"extracting data from the message"
          e:=msg.Extract
          if e:= < >
          then right!response
          elseif (e.Ipid=0) and (e.metric=16)
             then InitialRouter{e.Ipid=0,e.metric=16} ⟶ Router.Query;
                          // "Query the routing table"
            Router.Query{ e.Ipid=0,e.metric=16} ⟶ msg.Revise; X)
             //"Revise message"
          else   (if   InitialRouter{e.IP  ≠  <   >,   Router.e.metric  ≠  <
>} ⟶ Router.Query;      // "Query the routing table"
               then e.metric:=Router.Query
               else e.metric:=16
            Initialmsg{e.IP=< >} ⟶ msg.Reivse))//"Revise the message"
DResponse= if msg.IP ∈( neighbourlist ∪ localIP )
              then ( μ X.(Initialmsg{msg.entry≠< >} ⟶ msg.Extract;
                         //"extracting data from the message"
                 e:=msg.Extract;
                 if (e.metric<16) and (e.symbol=2) and (e.version=1)
                      then (InitialRouter{e.IP} ⟶ Router.Query)
                          //"Query the routing table"
                if (Router.Guery)
                then e1:= Router.Query
                if e.sourceIP=e1.gateway
               then (InitialRouter{e1.metric=e.metric} ⟶ Router.Revise;
                  //"Revise the entry of the routing table"
                 Settime{ e.sourceIP=e1.gateway,e1 } ⟶ Timer;
                 //"Timer"
                 e.SetFlag;
                 Initial Router{e1.metric=16} ⟶  Router.Delete)
                 //"Delete the entry of the routing table"
          Else (if e.metric<e1.metric
               Then          InitialRouter          {e1.metric=e.metric,
                  e1.gate=e.sourcrIP} ⟶Router.Revise
                            //"Revise the entry of the routing table"
                SetTime{e.metric< e1.metric,e1 } ⟶ Timer;  //"Timer"
                E1.SetFlag;
                  Initial Router{e1.metric=16} ⟶  Router.Delete)
                 //"Delete the entry of the routing table"
else   (InitialRouter{e2.ConstructIP,   e2.ConstructMetric,   e2.GateWay:=
       e.sourceIP} ⟶ Router.Add)  //"Add an entry to the routing table",
       "Timer"
        SetTime{ e.metric⩾16,e2 } ⟶ Timer; E2.SetFlag; X)  //"Timer"
       ⟶ TriggerUpdate
Trigger  Update  =  Initialmsg ⟶  μ X.(if  (Initialouter{Router.Entry ≠ <
>} ⟶ Router.Query)  and e1:=Router. Query // "Query the routing table"
         then if (e1.flag = 1)
          then Initialmsg{e.id=e1.id, e.metric:=e1.metric} ⟶ msg.Revise;
                //"Revise the message"
             e1.flag:=0;X); right!response
Output= μ X.(SetTime{time=30s} ⟶  Timer ⟶  OutputResponse ⟶  X)
               //"Timer"
OutputResponse=Initialrouter{Router                  ≠                 <
                >} ⟶ Router.Query ⟶ right!response
                //"Query the routing table"
   Timer Out= (SetTime{time=180s} ⟶ Timer){e} ⟶ Router.Delete)
               //"Delete the entry of the routing table"
Dynamic routing= Input ∥ Output ∥ Timer Out
```

FIGURE 2. The service formal model.

V. COMPOSITE SERVICES

When implementing a service composition instance, the performance computation relationship between the composite service and the atomic services can be expressed by the operation of the functional attributes for the composite service. Hence, based on the mathematical modeling, we give the computation of the performance attributes for composite services.

Assume that m atomic services participate in the service composition. The j-th ($1 \leq j \leq n$) performance of an atomic service is x_j. Hence, the j-th performance of a composite service can be achieved by computing m atomic services' performances, i.e. $FP_j(x_j)$.

A. THE COMPUTATION EXAMPLE OF THE PERFORMANCE ATTRIBUTES FOR COMPOSITE SERVICES

The solution of $FP_j(x_j)$ varies due to different performances and different compositions. Take response time and throughput as examples, we analyze the solution of $FP_j(x_j)$.

The parameters can be set as follows: the response time of m atomic services (i.e. S_i, $i = 0, 1, 2, \ldots, m$) are m independent random variables, i.e. T_i; the throughput of m atomic services are m independent random variables, i.e. Y_i; the waiting delay of m atomic services are m independent random variables, i.e. W_i; the processing delay of m atomic services are m independent random variables, i.e. Z_i. In the selective execution flow of the atomic services, the probability of selective execution is $P_i(\sum_{i=1}^{m} P_i = 1)$, and in the loop execution flow of the atomic services, the probability of loop execution is P.

For an atomic service (S_i), it's response time (T_i) should be equal to the sum of the waiting delay (W_i) and the processing delay (Z_i), as shown in the formula (9); and it's throughput can be calculated with marking flow rate which is mentioned in the paper [21] and [22], as shown in the formula (10). Where, $R(t, s)$ indicates the average number of tokens which flow from transition t to the post place s per unit time, i.e. throughput; $W(t, s)$ indicates the weights between the transition t and post place s; $U(t)$ indicates the sum of the stable probability of all of the marking, which can let the transition t be enabled;

λ indicates the average firing rate of the transition t.

$$T_i = W_i + Z_i \tag{10}$$

$$R(t,s) = W(t,s) \times U(t) \times \lambda \tag{11}$$

When m atomic services are composed sequentially, the response time of sequential composition is achieved according to the analysis of serial workflow performance in the paper [23] and formula (10), as shown in formula (12); and the equivalent throughput of sequential composition should be the throughput of the last atomic service according to the formula (11), as shown in formula (13).

$$T_{total} = \sum_{i=1}^{m}(W_i + Z_i) = \sum_{i=1}^{m} T_i \tag{12}$$

$$THP_{total} = Max_{i=1}^{m}(THP_i) \tag{13}$$

When m atomic services are composed parallel, the response time of parallel composition is achieved according to the analysis of parallel workflow performance in the paper [23] and formula (10), as shown in formula (14); and the equivalent throughput of parallel composition should be the total throughput of m atomic services according to the formula (11), as shown in formula (15).

$$\begin{aligned}
T_{total} &= \sum_{i=1}^{m}(W_i + X_i) - \sum_{i=1}^{m-1}\sum_{j=i+1}^{m}\left(\frac{1}{W_i+Z_i} + \frac{1}{W_j+Z_j}\right)^{-1} \\
&+ \sum_{i=1}^{m-2}\sum_{j=i+1}^{m-1}\sum_{k=j+1}^{m}\left(\frac{1}{W_i+Z_i} + \frac{1}{W_j+Z_j} + \frac{1}{W_k+Z_k}\right)^{-1} \\
&+ \ldots + (-1)^{-1}\left(\sum_{i=1}^{m}\frac{1}{W_i+Z_i}\right)^{-1} \\
&= \sum_{i=1}^{m} T_i - \sum_{i=1}^{m-1}\sum_{j=i+1}^{m}\left(\frac{1}{T_i} + \frac{1}{T_j}\right)^{-1} \\
&+ \sum_{i=1}^{m-2}\sum_{j=i+1}^{m-1}\sum_{k=j+1}^{m}\left(\frac{1}{T_i} + \frac{1}{T_j} + \frac{1}{T_k}\right)^{-1} \\
&+ \ldots + (-1)^{-1}\left(\sum_{i=1}^{m}\frac{1}{T_i}\right)^{-1}
\end{aligned} \tag{14}$$

$$THP_{total} = \sum_{i=1}^{m} THP_i \tag{15}$$

When m atomic services are composed selectively, the response time of selective composition is achieved according to the analysis of selective workflow performance in the paper [23] and formula (10), as shown in formula (16); and the equivalent throughput of selective composition should be the total throughput of the atomic services which are executed according to the formula (11), as shown in formula (17).

$$T_{total} = \sum_{i=1}^{m}(P_i * (W_i + Z_i)) = \sum_{i=1}^{m}(P_i * T_i) \tag{16}$$

$$THP_{total} = \sum_{i=1}^{m}(P_i * THP_i) \tag{17}$$

When one atomic service is composed iteratively, the response time of iterative composition is achieved according to the analysis of loop workflow performance in the paper [23] and formula (10), as shown in formula (18); and the equivalent throughput of iterative composition should be the throughput of the atomic service before halting iterative composition according to the formula (11), as shown in formula (19).

$$\begin{aligned}
\mathrm{T}_{total} &= \frac{1}{1-P}((W_1 + Z_1) + p * (W_2 + Z_2)) \\
&= \frac{1}{1-P}(T_1 + p * T_2)
\end{aligned} \tag{18}$$

$$THP_{total} = THP_1 \tag{19}$$

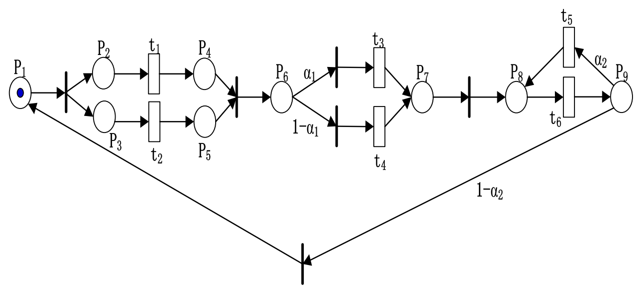

FIGURE 3. The model of the example.

B. THE VERIFICATION OF THE PERFORMANCE ATTRIBUTES COMPUTATION FOR COMPOSITE SERVICES

By means of calculation and experimental simulation respectively, response time and throughput of the composite service can be obtained. The Stochastic Petri net model is shown in fig.3. Where, 6 atomic services is expressed with 6 transitions, and the execution process includes sequence structure, parallel structure, selection structure and loop structure. At the place of P_6, the probability of selective execution (α_1) is set as 1/3; at the place of P_9, the probability of iterative execution (α_2) is set as 9/10; the execution rate of atomic services is expressed by the firing rate of transitions, i.e. $\lambda = \{\lambda_1 = 14, \lambda_2 = 7, \lambda_3 = 4, \lambda_4 = 8, \lambda_5 = 3, \lambda_6 = 1\}$.

By the experiment using SPNP [24], we can obtain that the average number of tokens in all places is 1.012; the average flow rate of the tokens is 0.074, namely, the throughput of the composite service is 0.074; the response time of the composite service is obtained by "Little rules" [25], and it is 13.493.

In the meantime, we can know the response time and the throughput of each atomic service after the probability of the stable state in the SPN model is obtained using SPNP. And we can calculate the response time and the throughput respectively with formula 12-19. Through calculation, we can obtain that the response time of the composite service is 13.333, and the throughput the composite service is 0.075.

It can be seen that the difference between the experimental result and the computation obtained with formula 12-19 is very small. Therefore, the computation of the performance attributes for composite services (formula 12-19) are correct and applicable.

VI. EXPERIMENTAL MODELING AND SIMULATION

Taking data transmission service for an experimental example, the performances of ENSM are compared with those of the traditional network to verify the correctness of the model idea.

A. EXPERIMENTAL MODELING

Here, modeling involves firing rates of transitions and conditional functions of transitions etc. What's more, system state space is large in the model. Hence, we select SHLPN [26] as the modeling tool. Data transmission service which has the mechanism of flow control and error retransmission is modeled with SHLPN in the traditional network and in ENSM respectively.

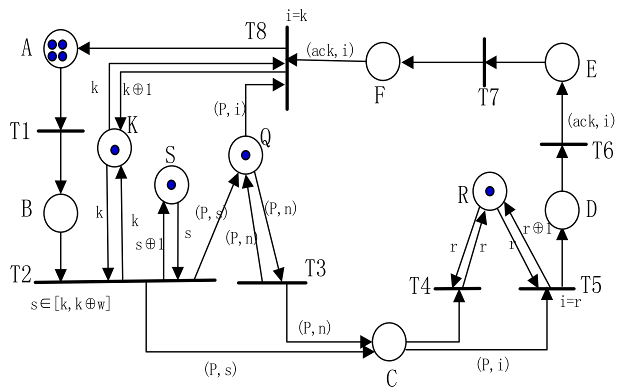

FIGURE 4. The instance model of data transmission service in traditional network.

Data transmission service of the traditional network [27] is modeled by using SHLPN, as shown in fig.4. Where, the places and the transitions are illustrated as follows: *A*: buffer for sending data, *B*: buffer for outputting data, *C*: interface at the sender, *D*: buffer for receiving data, *E*: buffer for outputting acknowledge, *F*: interface at the receiver, *K*: counter for the window of flow control, *S*: counter for sending serial numbers, *Q*: queue for resending data, *R*: counter for receiving serial numbers, *T1*: generate data and plus the checksum, *T2*: send data, *T3*: resend data, *T4*: discard data,*T5*: inspect the checksum and accept Data, *T6*: generate acknowledgements and plus the checksum, *T7*: send acknowledgements, *T8*: inspect the checksum and accept acknowledgements.

Data transmission service of ENSM is modeled by using SHLPN, as shown in fig.5 (The following categories of atomic services are required: system services, message services, QoS services, etc.). Where, the places and the transitions are illustrated as follows: *A*: buffer for sending data, *B*: buffer for sending packets, *C*: buffer for outputting packets, *D*: interface at the sender, *E*: buffer for inputting packets,

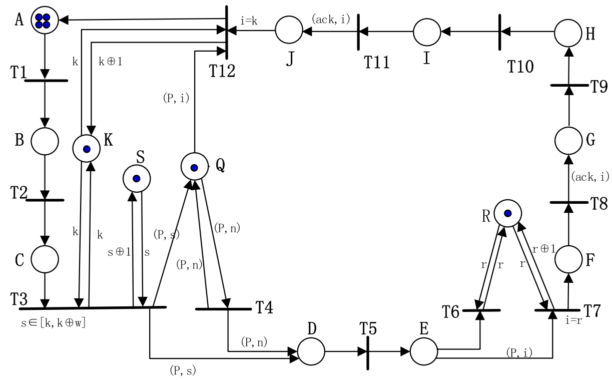

FIGURE 5. The instance model of data transmission service in ENSM.

F: correct packets, *G*: buffer for sending acknowledge, *H*: buffer for outputting acknowledge, *I*: interface at the receiver, *J*: buffer for inputting acknowledge, *K*: counter for the window of flow control, *S*: counter for sending serial numbers, *Q*: queue for resending data, *R*: counter for receiving serial numbers, *T1*: generate packets, *T2*: plus the checksum, *T3*: send packets, *T4*: resend packets, *T5*: inspect the checksum, *T6*: discard packets, *T7*: accept packets, *T8*: generate acknowledgements, *T9*: plus the checksum of acknowledges, *T10*: send acknowledgements, *T11*: inspect the checksum of the acknowledgements, *T12*: accept acknowledgements.

B. SIMULATION AND ANALYSIS

Data transmission service are simulated with SPNP [24] in the traditional network and in ENSM respectively. The difference of the performances between the two instance models is analyzed.

The parameters can be set as follows [23]: Newly arrived data in the two instance models conforms to the Poisson process; in the two instance models, the average arrival rate is λ_1; error rate of the transmission is 0.05; the size of the control packets is 128 bits, while the size of the data packets is 1024 bits. The firing rate of the transitions in the instance model of the traditional network is set as: $\lambda = \{\lambda_2 = 9.0, \lambda_3 = 100.0, \lambda_4 = 5.0, \lambda_5 = 95.0, \lambda_6 = 100.0, \lambda_7 = 100.0, \lambda_8 = 100.0\}$; while the firing rate of the transitions in the instance model of ENSM is set as: $\lambda = \{\lambda_2 = 100.0, \lambda_3 = 9.0, \lambda_4 = 100.0, \lambda_5 = 5.0, \lambda_6 = 95.0, \lambda_7 = 100.0, \lambda_8 = 100.0, \lambda_9 = 100.0, \lambda_{10} = 100.0, \lambda_{11} = 100.0\}$.

The offered load is scale linearly with the throughput in the two instance models, as shown in fig.6. When the offered load is 5 packets/s, the throughput in the instance model of the traditional network is 5 packets/s, while the throughput in the instance model of ENSM is 4.98 packets/s. And when the offered load is 25 packets/s, the throughput in the instance model of the traditional network is 25 packets/s, while the throughput in the instance model of ENSM is 24.98 packets/s. In short, the throughput in the instance model of ENSM is similar to that in the instance model of the traditional network, and the difference of the throughput between the

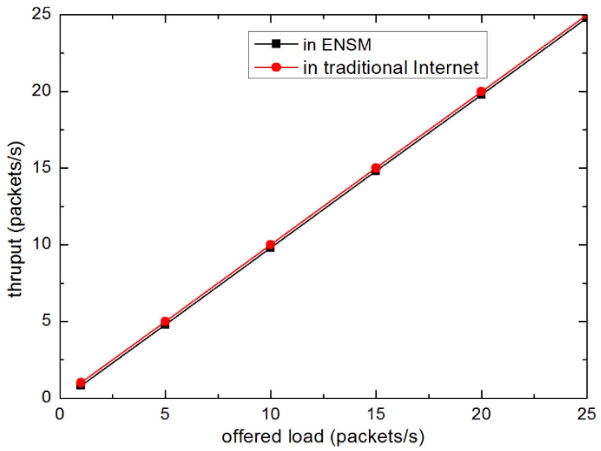

FIGURE 6. The comparison on throughput with different offered load between the two instance models.

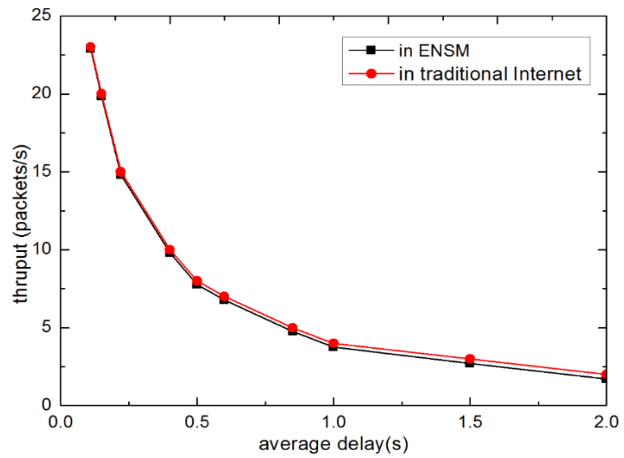

FIGURE 7. The comparison on throughput with different values of average delay between the two instance models.

two instance models is slightly smaller. The reason can be concluded as follows: the functional granularity (the atomic services) in the instance model of ENSM is smaller, thus the associations of the parameters among the atomic services are more. Therefore, with the same supply load, the throughput in the instance model of ENSM is slightly smaller, but the difference is very small.

The average delay is inversely proportional to the throughput in the two instance models, as shown in fig.7. When the average delay is 0.11 s, the throughput in the instance model of the traditional network is 23 packets/s, while the throughput in the instance model of ENSM is 22.99 packets/s. And when the average delay is 2 s, the throughput in the instance model of the traditional network is 2 packets/s, while the throughput in the instance model of ENSM is 1.950 packets/s. In short, the throughput in the instance model of ENSM is similar to that in the instance model of the traditional network. The difference of the throughput between the two instance models is small when the average delay is small. But, the difference will slightly increase along with the increase of the average delay. The reason can be concluded as follows: When the average delay increases, the associations of the parameters among the atomic services increase too. Therefore, with the same average delay, the throughput in the instance model of ENSM is slightly smaller than that in the instance model of the traditional network, and the difference will slightly increase along with the increase of the average delay.

From the experimental simulation and analysis, we can draw the conclusion: data transmission service can be composed using a number of atomic services according to certain rules in ENSM; compared with those in the instance model of the traditional network, the performances in the instance model of ENSM don't decrease significantly. It can be seen that services can be customized or composed flexibly in ENSM while not sacrificing the performances. Therefore, ENSM can implement service extension better and the model idea is correct.

VII. CONCLUSION

Along with the increasing development of network technologies and applications [28]–[30], Internet is facing many problems and challenges which are mainly reflected in the capability of providing service. The reason can be concluded that the traditional network architecture can't support the dynamic deployment of the new protocols well and there are functional redundancies among the layers of the traditional network architecture. Thus, many research organizations began to focus on how to improve the network architecture and network service model. However, the existing research results of new network architecture can't solve the problem of network service extensibility well.

Our project team propose ENSM which does well in service extensibility. In the research, we set "service extensibility" as the focus. We devote to modeling and analyze ENSM from the theoretical point of view in this paper. On the basis of the mathematical modeling, we give the description of the functional attributes and the computation of the performance attributes for composite services. Through the experimental modeling and the simulation, the performances of the extensible network service model are compared with those of the traditional network when providing the same service. The difference of performances between the two instance models is very small. Therefore, services can be customized or composed flexibly in ENSM while not sacrificing the performances. It can be seen that ENSM can implement service extension better and the model idea is correct.

ENSM can provide recursive service capability and is a new exploration of the future network. There are more theoretical analysis and demonstration work to do so that ENSM can be designed, implemented and popularized better in the future. And it is our next step to research on the operation of service composition.

REFERENCES

[1] B. M. Leiner *et al.*, "A brief history of the Internet," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 5, pp. 22–31, 2009.

[2] Q. Zhang, C. Zhu, L. T. Yang, Z. Chen, L. Zhao, and P. Li, "An incremental CFS algorithm for clustering large data in industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 13, no. 3, pp. 1193–1201, Jun. 2017.

[3] Q. Zhang, L. T. Yang, X. Liu, Z. Chen, and P. Li, "A tucker deep computation model for mobile multimedia feature learning," *ACM Trans. Multimedia Comput., Commun. Appl.*, vol. 13, no. 3s, pp. 39:1–39:18, 2017.

[4] J. Wu, S. Lin, K. Xu, Y. Liu, and M. Zhu, "Advances in evolvable new generation Internet architecture," *Chin. J. Comput.*, vol. 35, no. 6, pp. 1094–1108, 2012.

[5] *Geni: Global Environment for Network Innovation [EB/OL]*. [Online]. Available: http://www.geni.net

[6] *FIND: Future Internet Network Design [EB/OL]*. [Online]. Available: http://find.isi.edu/

[7] *PlantLab [EB/OL]*. [Online]. Available: http://www.plantlab.org/

[8] *FIRE: Future Internet Research and Experimentation [EB/OL]*. [Online]. Available: http://cordis.europa.eu/fp7/ict/fire/

[9] *New Generation of Trustworthy Network [EB/OL]*. [Online]. Available: http://www.most.gov.cn/tztg/200711/t20071107_57004.html

[10] R. Braden, T. Faber, and M. Handley, "From protocol stack to protocol heap: role-based architecture," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 33, no. 1, pp. 17–22, 2003.

[11] R. Dutta, G. N. Rouskas, I. Baldine, A. Bragg, and D. Stevenson, "The SILO architecture for services integration, control, and optimization for the future Internet," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2007, pp. 1899–1904.

[12] J. Touch, "Next steps in enabling a virtual Internet," in *Proc. IEEE Comput. Commun. Workshop*, 2009, pp. 35–41.

[13] J. Touch *et al.*, "A dynamic recursive unified Internet design (DRUID)," *Comput. Netw.*, vol. 55, no. 4, pp. 919–935, 2011.

[14] Y. Fa-Sheng, C. Gui-Hai, L. Ming, G. Hai-Gang, and Z. Jia-Zhi, "Design and implementation of service-based network architecture," *Chin. J. Softw.*, vol. 19, no. 12, pp. 3179–3195, 2008.

[15] H. Yu-Rong, Y. Fa-Sheng, and Z. Jia-zhi, "Delay-based performance test of micro communication element mainframe prototype," *Comput. Eng.*, vol. 35, no. 20, pp. 121–122, 2009.

[16] G. Wang, Y. Zhao, J. Huang, Q. Duan, and J. Li, "A K-means-based network partition algorithm for controller placement in software defined network," in *Proc. IEEE Int. Conf. Commun. (CC)*, May 2016, pp. 1–6.

[17] X.-N. Nguyen, D. Saucez, C. Barakat, and T. Turletti, "Rules placement problem in OpenFlow networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1273–1286, 2nd Quart., 2016.

[18] Q.-Y. Zuo, M. Chen, G.-S. Zhao, C.-Y. Xing, G.-M. Zhang, and P. Jiang, "Research on OpenFlow-based SDN technologies," *Chin. J. Softw.*, vol. 24, no. 5, pp. 1078–1097, 2013.

[19] Z. Ji, J. Shen, X. Zhou, D. Ding, and Y. Zhang, "The analysis of service extensibility in extensible network service model," in *Proc. 35th IEEE Int. Perform. Comput. Commun. Conf. (IPCCC)*, Las Vegas, NV, USA, Dec. 2016, pp. 1–8.

[20] Q. Ran, X. Wu, X. Li, J. Shi, J. Guo, and H. Zhu, "Modeling and verifying the TTCAN protocol using timed CSP," in *Proc. Theor. Aspects Softw. Eng. Conf. (TASE)*, Changsha, China, Sep. 2014, pp. 90–97.

[21] R. Korkikian, S. Pelissier, and D. Naccache, "Blind fault attack against SPN ciphers," in *Proc. Workshop Fault Diagnosis Tolerance Cryptogr. (FDTC)*, Busan, South Korea, 2014, pp. 94–103.

[22] K. Hiraishi, "Simulating Markovian stochastic Petri Nets by difference equations with interval parameters," *Discrete Event Dyn. Syst.*, vol. 25, no. 3, pp. 365–386, 2015.

[23] L. Chuang, *Stochastic Petri Net and System Performance Evaluation*. Beijing, China: Tsinghua Univ. Press, 2001.

[24] G. Giaodo, J. Muppala, and K. S. Trivedi, "SPNP: Stochastic Petri net package," in *Proc. 3rd Int. Workshop Petri Nets Perform. Models*, Kyoto, Japan, 1989, pp. 142–151.

[25] K. S. Trivedi, *Probability and Statistics With Reliability, Queuing, and Computer Science Applications*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1982.

[26] L. Lei, H. Wang, C. Lin, and Z. Zhong, "Wireless channel model using stochastic high-level Petri nets for cross-layer performance analysis in orthogonal frequency-division multiplexing system," *IET Commun.*, vol. 8, no. 16, pp. 2871–2880. 2014.

[27] L. Chuang, *Stochastic Petri Net and System Performance Evaluation*. Beijing, China: Tsinghua Univ. Press, 2005.

[28] Q. Zhang, L. T. Yang, Z. Chen, and P. Li, "A survey on deep learning for big data," *Inf. Fusion*, vol. 42, pp. 146–157, Jul. 2018.

[29] Q. Zhang, L. T. Yang, Z. Chen, P. Li, and M. J. Deen, "Privacy-preserving double-projection deep computation model with crowdsourcing on cloud for big data feature learning," *IEEE Internet Things J.*, to be published, doi: 10.1109/JIOT.2017.2732735.

[30] Q. Zhang, L. T. Yang, Z. Chen, and P. Li, "High-order possibilistic c-means algorithms based on tensor decompositions for big data in IoT," *Inf. Fusion.*, vol. 39, pp. 72–80, Jan. 2018,

ZUQIN JI received the B.S. degree from Southeast University, Nanjing, China, in 2003, and the M.S. degree from Changzhou University, Changzhou, China, in 2010. She is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, Southeast University. Her research interests include peer-to-peer systems, network architecture, and service extensibility.

JUN SHEN received the B.S. and Ph.D. degrees from Southeast University, Nanjing, China, in 1986 and 2007, respectively. He is currently a Professor with the School of Computer Science and Engineering, Southeast University. He presided over a number of national projects. His research interests include peer-to-peer systems, network architecture, service scalability, software model and architecture, and collaborative computing and network teaching.

DELIN DING received the B.S. and M.S. degrees from Southeast University, Nanjing, China, in 2012 and 2015, respectively, where he is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering. His main research interests include network architecture and software model and architecture.

XIAOWEI CUI received the B.S. degree from Southeast University, Nanjing, China, in 2017, where he is currently pursuing the master's degree with the School of Computer Science and Engineering. His research interests include network architecture and service extensibility.