

Automatic Network Service Discovery and Selection in Virtualization-Based Future Internet

Qiang Duan

Dept. of Information Science & Technology

The Pennsylvania State University

Abington, PA 19001

Tel: 1-215-881-7872 E-mail: qduan@psu.edu

Received: January 10, 2012

Accepted: May 16, 2012

Published: June 19, 2012

DOI: 10.5296/npa.v4i2.1275

URL: <http://dx.doi.org/10.5296/npa.v4i2.1275>

Abstract

The next generation Internet is expected to support a wide range of network applications with highly diverse requirements. Network virtualization, which will play a key role in the future Internet, allows a large number of service providers to offer various network services upon shared network infrastructure. Therefore, discovery and selection of the most appropriate network service for meeting the requirement of each application becomes an important issue in the virtualization-based future Internet. Service-Oriented Architecture (SOA) offers an effective architectural principle for heterogeneous system integration that may facilitate realizing the notion of network virtualization. The research reported in this paper explores application of SOA in network virtualization to address the network service discovery and selection problem. A broker system is proposed based on SOA for automatic network service discovery and selection, a scalable and efficient protocol is developed for updating network service information, and the multi-attribute decision making technique is employed for ranking and selecting network services to meet application requirements. Performance evaluation results are also presented in the paper to show the effectiveness of the developed protocol and algorithm.

Keywords: Network virtualization, network-as-a-service, Service-Oriented Architecture (SOA), network service discovery, network service selection

1. Introduction

Recently numerous network-based computing applications have been developed and deployed on top of the Internet. These applications have a wide spectrum of requirements for the underlying networking platform. The highly diverse network application requirements have motivated research on various new networking technologies and even alternative architecture for the next generation Internet. The idea of network virtualization was first employed as an approach to developing virtual test-beds for evaluating new network architecture. Then the network research community realized that fundamental changes must be made in the Internet architecture to allow networking systems with alternative architecture and different implementations to coexist upon a shared infrastructure platform. Toward this objective, the role of virtualization in networking has evolved from a research tool to a fundamental attribute of networking paradigm, which is expected to play a crucial role in the future Internet.

Network virtualization separates service provisioning from data transportation; thus decoupling the role of traditional Internet Service Providers (ISPs) into two entities: Service Providers (SPs) and Infrastructure Providers (InPs). Virtualization in the next generation Internet will have a significant impact on the service provisioning. SPs, released from the burden of managing and operating underlying network infrastructure, will be able to offer a wide variety of network services to meet different application requirements. Since a large number of service providers are expected in the virtualization-based future Internet, discovering available network services and select the most appropriate services that match application requirements become an important issue.

For example, the network services available in the vicinity of a user may include a free access service with no QoS guarantee and a service that guarantees delay and bandwidth performance but charges a usage fee. The user device may want to utilize the guarantee service only when running applications with QoS requirements, such as online video conference, but switch to the free service for other applications. For mobile users, the device may need to keep searching available services and selecting a service for each application when the user moves. Therefore, discovery and selection of the appropriate network services for applications must be performed automatically. This becomes particularly important when a large number of network services coexist for supporting diverse application requirements.

One of the key technical challenges to service discovery and selection in network virtualization environments lies in effective and flexible interaction between the network service customers, which are typically networking applications, and the SPs that offer various network services. The diversity of application and heterogeneity of network services in the future Internet makes such interaction more challenging. The SOA, as an effective architectural principle for heterogeneous system integration, offers a promising approach to addressing these challenges; thus may greatly facilitate service discovery and selection in network virtualization environments.

Research presented in this paper explores applications of the SOA principle in a network virtualization environment. The main contribution made in this paper focuses on the proposal

of a SOA-based broker system for automatic network service discovery and selection in network virtualization. Updating network service information and selecting the most appropriate network service for each application are two key issues that are particularly studied in this paper. An event-driven protocol is developed for service information update to enable real-time discovery and selection of network services. The multi-attribute decision making technique is employed at the service broker for selecting network services to meet application requirements.

The rest of this paper is organized as follows. Section 2 introduces the notion of network virtualization and its impact on service provisioning in the future Internet. Section 3 discusses applications of the SOA principle in network virtualization and the network-as-a-service paradigm. In Section 4 a SOA-based broker system is proposed for automatic discovery and selection of network services in the virtualization-based Internet. A scalable protocol is developed in Section 5 for updating network service information in large scale dynamic networking environments. Section 6 presents an algorithm based on multi-attribute decision making techniques for selecting network services. Performance evaluation of the proposed broker system, particularly about the developed updating protocol and selection algorithm is given in Section 7. Section 8 draws conclusions and discusses issues for future research.

2. Network Virtualization in the Future Internet

Essentially network virtualization follows a well-tested principle—separation of policy from mechanism—in the networking area. In this case, network service provisioning is separated from data transportation mechanisms; thus dividing the traditional role of ISPs into two entities: infrastructure providers who manage physical network infrastructure, and service providers who create virtual networks for offering end-to-end services by aggregating resources from network infrastructure [1]

Infrastructure providers (InPs) are in charge of operations and maintenance of physical network infrastructure and offer their resources through programmable interfaces to different service providers instead of providing direct services to end users. Service providers (SPs) lease networking resources from multiple InPs to create virtual networks and deploy customized protocols in the virtual networks by programming the resources in network infrastructure. Each virtual network is composed and managed by a single SP, which synthesizes the networking resources allocated in the underlying infrastructure to offer network services to the end users. A virtual network is a collection of virtual nodes connected together by a set of virtual links to form a virtual topology, which is essentially a subset of the underlying physical topology. Each virtual node could be hosted on a particular physical node or could be a logical abstraction of a networking system. A virtual link spans over a path in the physical network and includes a portion of the networking resources along the path. Figure 1 illustrates a network virtualization environment, in which the service providers SP1 and SP2 construct two virtual networks by using resources provided by the infrastructure providers InP1 and InP2.

Recently network virtualization has attracted extensive research interest from both academia and industry. New network architecture is proposed in [2] for diversifying the Internet, which enables various meta networks built on top of a physical substrate. The CABO Internet architecture proposed in [3] decouples network service providers and infrastructure providers to support alternative network architecture upon the shared infrastructure. Some standard organizations are also embracing the notion of network virtualization into their standard specifications. For example, the Next Generation Network (NGN) architecture defined by ITU-T follows a key principle of separating service-related functions from underlying transport-related technologies. An overview of more research efforts and the state of the art of network virtualization can be found in [1].

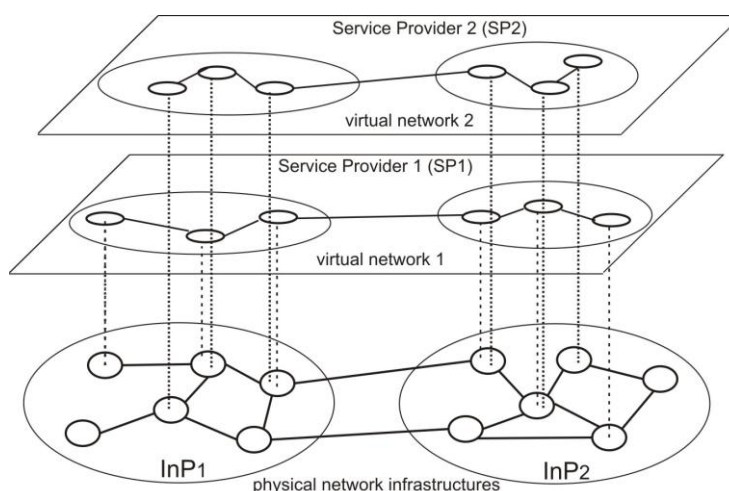


Figure 1 Illustration of a network virtualization environment.

Network virtualization will bring a significant impact on network service provisioning in the future Internet. The best-effort Internet today is basically a commodity service that gives network service providers limited opportunities to distinguish themselves from competitors. Network virtualization offers a rich environment for innovations that can stimulate the development and deployment of a wide spectrum of new Internet services. Network virtualization enables a single SP to obtain control over the entire end-to-end service delivery path across physical networks that may belong to different autonomous systems of the Internet, which will greatly facilitate end-to-end QoS provisioning.

In a network virtualization environment, SPs are released from the requirement of purchasing, deploying, and maintaining physical network equipments, which significantly lowers the barrier to entry of the Internet service market. It is reasonable to expect a large number of SPs coexisting upon the shared network infrastructure to offer various Internet services. On the other hand, a key requirement to the next generation Internet is to support a wide range computing applications with highly diverse requirements on network services. Therefore, discovering available network services and selecting the most appropriate service for each application that meet the application requirements becomes an important issue in the virtualization-based future Internet, which is referred to as the service discovery and selection problem in network virtualization.

3. Application of SOA in Network Virtualization

SOA is a system architecture initially developed in the distributed computing community as an effective solution to coordinating computational resources in multiple heterogeneous systems to support various application requirements. SOA is described in [4] as “an architecture within which all functions are defined as independent services with invokable interfaces that can be called in defined sequences to form business processes.” A *service* in SOA is a computing module that is self-contained (i.e., the service maintains its own states) and platform-independent (i.e., the interface to the service is independent of its implementation platform). In essence, services in SOA are modular, interoperable, composite entities that can cooperate with other services in the shape of service level agreements. Services can be described, published, discovered, orchestrated, and programmed through standard interfaces and messaging protocols.

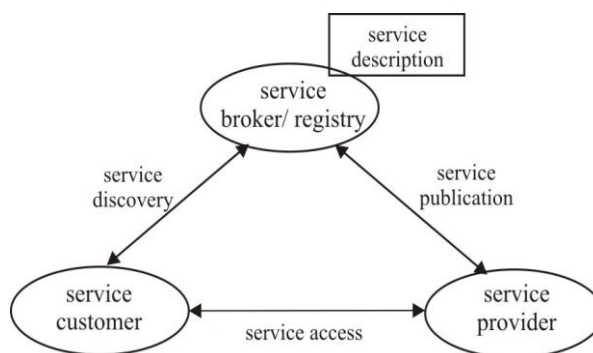


Figure 2 Key elements and their interaction of a Web services-based SOA implementation.

Although SOA can be realized by various technologies, currently Web services offer the main approach to implementing SOA. Figure 2 gives the key elements with the interaction among them for a Web service-based SOA implementation. A service provider publishes a service description at a service registry. The service description is a machine-readable document that gives descriptive information about the functions and accessing interfaces of the service. A service customer performs dynamic service location by querying the service registry for a service that matches its criteria. If such a service exists, the registry provides the customer with the information for accessing that service. Typically a service broker maintains the service registry and processes service discovery requests for customers.

Key Web service technologies include service description, service registration, service discovery, and message delivery protocol. The current standards for these technologies include Web Service Description Language (WSDL), Universal Description Discovery and Integration (UDDI), and Simple Object Access Protocol (SOAP), which are all based on the Extensible Markup Language (XML) as a standard for sharing data among computing systems.

A key attribute of SOA is the *loose-coupling* interaction among heterogeneous systems in the architecture. The term *coupling* indicates the degree of dependency any two systems have on each other. In a loosely coupled exchange, systems need not know how their partner systems behave or are implemented, which allows systems to connect and interact more freely. It is this feature that makes SOA an attractive solution to the problem of coordinating heterogeneous systems to support various computing applications, which is essentially the same challenge faced by the network service discovery and selection in the virtualization-based Internet. Therefore, SOA may provide a promising approach to enhancing the control intelligence in the future Internet for automatically discovering and selecting network services.

By applying SOA in the field of networking, the resources in various network systems can be encapsulated into SOA-compliant services. The capabilities of the network systems and the approach to accessing these systems are described in network service descriptions, which can be published at a network service registry. A layered structure of SOA-based network virtualization environment is shown in Figure 3. The network infrastructure is on the bottom layer, networking applications are on the top layer, and in the middle is a service provisioning layer that comprises virtual networks for meeting application requirements. Following the SOA principle, resources in network infrastructure can be encapsulated into *infrastructure services* and offered by InPs to SPs. SPs access underlying networking resources through an Infrastructure-as-a-Service paradigm and compose infrastructure services into end-to-end *network services*. Applications utilize underlying networking platform by accessing the network services offered by SPs, which is essentially a *Network-as-a-Service* paradigm.

Applying SOA in network virtualization makes loose-coupling a key feature of both SP-InP interaction and collaboration among heterogeneous network infrastructure. Therefore, such a network virtualization paradigm inherits the merit of SOA that enables flexible and effective collaboration across heterogeneous systems for providing services that meet diverse application requirements. SOA-based network virtualization provides a means to present abstracted networking capabilities to upper-layer applications. Because of the heterogeneity of network protocols, equipments, and technologies, exposing networking capabilities to applications without virtualization would lead to unmanageable complexity. The abstraction of networking resources through service-oriented network virtualization can address the diversity and significantly simplify the interaction between applications and the underlying network platform.

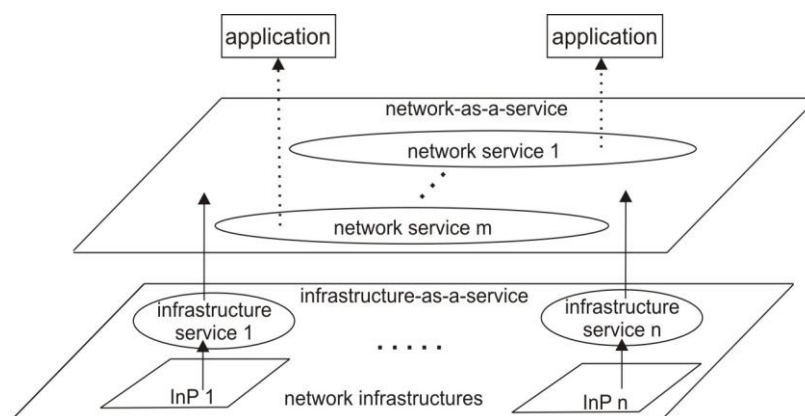


Figure 3 A layered structure for SOA-based network virtualization.

The SOA concept and Web services have been employed in telecommunications to enhance network service provisioning. For example, the Parlay X and OMA Service Environment (OSE) developed by the ETSI and 3GPP show their consolidated interest on applying SOA in network operation and management. TM Forum Service Delivery Framework (SPF) and the newly developed IEEE Next Generation Service Overlay Network (NGSON) specification both leverage SOA in network service provisioning. OASIS recently formed a technical committee (SOA-Tel) for investigating SOA in telecommunications. Surveys about applications of SOA in telecommunications can be found in [5] [6].

Service-oriented network virtualization has also become an active area for the future Internet research. In UCLPv2 (User Controlled Light Path), a Canadian research project for enabling user control and management of optical network infrastructure, Web service technologies were employed to expose resources in optical network infrastructure as services [7]. The framework of network infrastructure service developed in UCLPv2 then evolved into a number of different projects, including Argia as a commercial implementation for optical network infrastructure services, Ether for developing Ethernet and MPLS infrastructure services, and MANTICORE for supporting logical IP network as services [8]. In [9] the authors designed a transport stratum according to the SOA paradigm in order to expose transport functionalities as services to the service stratum in NGN. A service-oriented network virtualization architecture was developed in [10], which consists of physical infrastructure layer, virtual network layer, and service network layer from bottom to top. Modeling and performance analysis for end-to-end service provisioning in network virtualization-based future Internet was also investigated in the author's previous work [11] and [12].

Despite the aforementioned research progress for service-oriented network virtualization, this area is still at an early stage and facing various challenges. The problem of network service discovery and selection is still an open issue that needs further study. As the notion of virtualization widely adopted in the field of networking, various third party providers may construct their virtual networks to offer network services for supporting diverse applications. Therefore, scalable and efficient protocol and algorithm must be developed for automatic discovery and selection of network services in a network virtualization environment.

4. SOA-Based Broker System for Network Service Discovery and Selection

In this section we propose a service-oriented broker system for network service discovery and selection in the virtualization-based future Internet. The high-level structure for this system, as shown in Figure 4, consists of a network service broker, a network service registry, and a set of virtual networks offered by SPs as SOA-compliant network services. In this system, every SP publishes a service description for each of its network service at the network service registry (Step (1) in Figure 4). A network service description is a machine-readable document that provides all the necessary information for discovering and selecting the network service. When a service user, typically an application, needs to utilize the underlying network platform, it sends a request message to the network service broker (Step (2) in Figure 4). This request message specifies the application's demand on a network service and the criteria for network service selection. On receipt of this request, the service broker first searches the service registry for available network services (Step (3) in Figure 4). Then the broker examines the service descriptions of these candidate services to select a network service based on a set of decision criteria, which may include performance requirement specified by the application, current status of the available network services, and also network management policies (Step (4) in Figure 4). After the selection process, the service broker sends the user a reply message (Step (5) in Figure 4), which contains identification of the selected network service and all the information that the user needs for service access. Then the user starts utilizing the network service using the information provided by the broker (Step (6) in Figure 4).

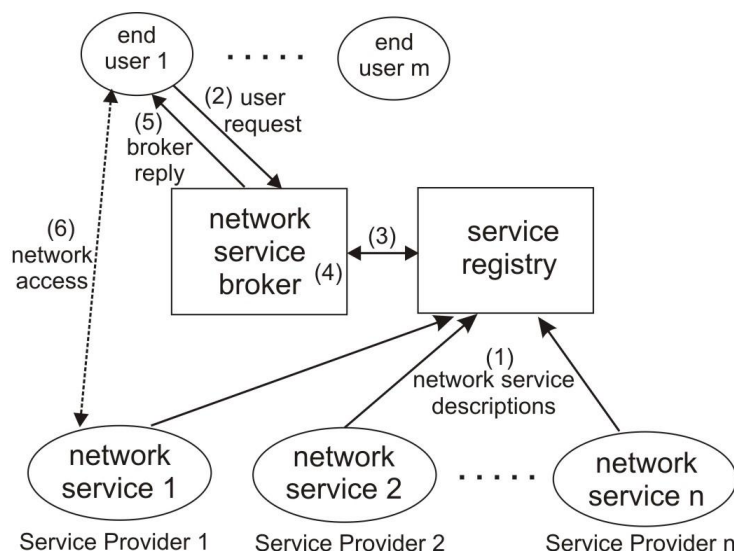


Figure 4. A SOA-based broker system for network service discovery and selection

The proposed SOA-based broker system serves as a mediator between network service providers and end users that can simplify interaction between these two key players in a network virtualization environment. The broker releases users' load for contacting service providers, collecting service information, and analyzing the information for service selection. The broker/registry allows service providers to publish their services without exposing the

internal implementations of virtual networks. Network services are only accessed through the interfaces that providers specify in the service descriptions, which can protect the underlying network platform from unauthorized access. The service registry, as a central repository of service descriptions, can be hosted on a server with strengthened security mechanisms that authenticate and authorize service publications and discovery requests. Since the broker system is based on SOA, currently available security technologies for Web services, including WS-Security, WS-SecurityPolicy, WS-Trust, WS-Privacy, WS-SecureConversation, WS-Federation, and WS-Authorization., can be employed in this system.

The proposed broker system can be realized by leveraging and extending Web service technologies. The network service description can be implemented based on the Web Service Description Language (WSDL). The network service registry can be built by employing the Universal Description, Discovery, and Integration (UDDI) standard. Message delivery among the service broker/registry, end users, and network service providers can be realized based on the SOAP protocol. However currently available service standards and technologies for service description and discovery were developed mainly for Web services and need to be evolved to meet the special requirements of network-as-a-service.

Recently researchers tend to apply established semantic web tools in the network realm and develop ontology specifically for describing network services. The Network Description Language (NDL) developed in [13] is an ontology designed based on Resource Description Framework (RDF) to describe network services. NDL is basically an RDF vocabulary defined for describing network elements and topologies. For facilitating description for abstract networking resources, authors of [14] propose the Network Resource Description Language (NRDL), which is also based on RDF but focuses on expressing the communication interaction relationship between network elements. Networking capabilities and QoS properties are important aspects of network service description. World Wide Web Consortium (W3C) developed WS-Policy specification as a general-purpose framework for expressing non-functional service characteristics such as performance requirements, preference, and capabilities as policies. OASIS is developing a Web Service Quality Model (WSQM) that defines taxonomy of QoS and a wide range of QoS properties. A general model for describing network service capabilities was also proposed in [15].

Although progress has been made toward network service descriptions, automatic discovery and selection of network services still need further investigation. A key expectation for a network service is to meet the performance requirements of the application that utilize the service; therefore discovery and selection of network services must be based on the capability and performance attributes of the services, which is essentially a multi-attribute decision making problem. In order to make a right decision for service selection, the broker must use the service information that precisely present the latest network states (i.e., the network attributes such as available bandwidth, delay, packet loss rate, etc.). However, keeping the latest state information at the service registry for all network services in a large scale dynamic networking environment is a very challenging issue that must be carefully examined. In the next two sections, we will particularly study these two key issues by proposing a protocol for service information update and developing an algorithm for network

service selection.

5. Network Service Information Update Protocol

In a large scale dynamic networking environment such as the future Internet, state and capability information of various network service change frequently. Keeping the latest service information available at the registry is significant for discovering and selecting appropriate network services for meet application requirements. However, if the entire service description document for a network service is republished at the registry whenever a state changes in the network service, it may cause a large amount of communication and management overheads between SPs and the registry thus forming performance bottleneck for the broker system. Therefore, efficient and scalable protocol for updating network service information becomes a key to achieve high-performance network service discovery.

An event-based protocol that employs a subscription-notification mechanism is developed in this section to solve this problem of network service information update. Event-driven processing introduces a notification pattern (also referred to as the event pattern) for SOA implementations. In this pattern an information producer sends one-way notification messages to one or more information consumers. The Organization for the Advancement of Structured Information Standards (OASIS) provides a standard approach to notification using a topic-based publish/subscribe mechanism [16]. By following this pattern, the registry in the network service broker system can subscribe to an SP and specify a set of network states as subscription topics. Then, the service registry will receive a notification message from the SP whenever a subscribed topic changes with a variation more than a pre-specified threshold. In this way the service registry can obtain the latest network information for supporting real-time networking service discovery and selection. Since the notification message is only generated when variation goes beyond a threshold and the message contains only the changed network states instead of the entire description, this updating mechanism greatly reduces the communications and management overhead.

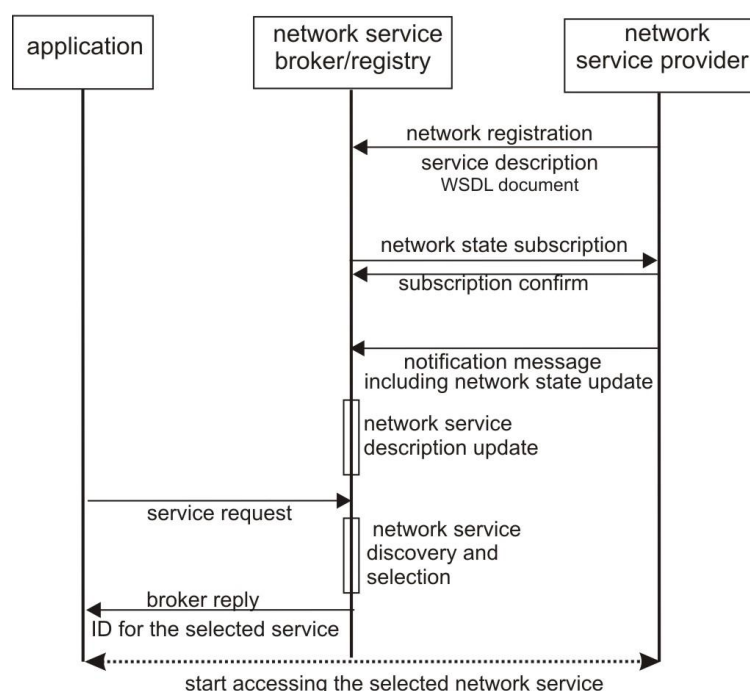


Figure 5. An event-driven protocol for network service information update

The interaction procedure for network service discovery and selection with the event-driven service information update protocol is shown in Figure 5. An SP makes a network service available by publishing a service description at the service registry. The service description document includes resource property items for network service information. Whenever an SP publishes a network service, the registry will subscribe to the network service in order to receive notification for any network state change in the service. The registry can specify what network states it is interested in as subscription topics. After this registration-subscription procedure is completed, the service registry will be notified whenever a network state specified as a subscription topic changes. Then the service description for this network will be updated at the registry accordingly. In this way, the service broker can always obtain the latest network information from the service registry. When an application needs to access the Internet, it sends a request message to the service broker with a specification of networking demand and performance requirements. The service broker searches the latest network service descriptions published at the service registry and discovers available network services. Then the broker selects the most appropriate one according to the requirements specified in the service request and sends a response message back to the application with service discovery and selection result.

6. Network Service Selection Algorithm

The service broker must perform a selection process if there are multiple available network services that meet the application requirements. The problem of network service selection can be solved by employing multi-attribute decision making (MADM) technique, which evaluates a set of alternatives (representing available network services) using a set of

attributes. The decision process ranks these alternatives in order of preference using the set of attributes that provide different aspects by which the alternatives can be viewed.

Each network service description includes a set of attributes. Though different services may have different attributes, these attributes in general can be classified into two categories: non-QoS-related attributes and QoS-related attributes. Examples of non-QoS-related attributes include service provider name, geographic location, coverage area, authentication mechanism, access technology, etc. The values of attributes in this category do not change very often, therefore can be published at the service registration phase and do not need to be updated frequently. Examples of QoS-related attributes include total and currently available service capacities (network bandwidth), service capacity utilization, networking delay and delay jitter, packet loss ratio, etc. These attributes are typically dynamic and need to be updated frequently. Considering a general case where n available network services described by m attributes, each network service S_i , $i=1, 2, \dots, n$, can be represented by a vector with m elements $S_i = (a_{i1}, a_{i2}, \dots, a_{im})$, where each element a_{ij} gives the value of the j -th attribute of this service. Therefore, for a given service request, the set of all candidate services and their attributes can be represented by the following matrix

$$S = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,m} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,m} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,m} \end{pmatrix} \quad (1)$$

where each element a_{ij} gives the value of the j -th attribute of the network service S_i .

Each service request submitted by an application specifies a set of attributes, which can also be classified into the two categories (non-QoS-related and QoS-related attributes). Due to the diversity of application requirements, different attributes may be included in different service requests. Even the same set of attributes, especially QoS-related attributes, may have different levels of relative importance to different applications. The relative importance of each attribute can be represented by a weight parameter assigned to the attribute. Different applications may have different requirements on network services, and also different levels of relative importance on different service attributes. The weight assignment to the set of attributes plays a key role in network service selection. When an application submits a service request, in addition to specify a set of required attributes, the request will also include a weight vector

$$W = (w_1, w_2, \dots, w_m), \quad (2)$$

where each element w_i , $i = 1, 2, \dots, m$, gives the weight assignment for the i -th attribute used for selecting network service. Such a weight vector could be saved as part of application profile at the service broker, and used by the broker each time for selecting a service for this application.

TOPSIS (Technique for Order Preference by Similarity to Ideal Solution), commonly used MADM algorithm, can be applied to the matrix S to objectively decide on the best service. The TOPSIS algorithm is based on the assumption that the chosen solution has the shortest distance from the best solution but the longest distance from the worst solution. The following steps are involved in application of TOPSIS to network service selection.

1. Normalize the value of each attribute in matrix S .
2. Adjust the normalized values of matrix S according to the weight assignment given in the vector W .
3. Determine the best and worst values for each of the attribute. Depending on the attribute, the best (or worst) value can be either the maximum or minimum value. For example for the bandwidth utilization attribute the best value will be the minimum value, while for the attribute of available capacity the best value is the maximum value.
4. For each candidate service, calculate the measures of Euclidean distance for the best and worst cases, which are respectively denoted as d_{best} and d_{worst}
5. For each candidate service, calculate the preference P based on the relative closeness to the best and separation from the worst solution as $P = d_{worst} / (d_{best} + d_{worst})$
6. The network service with the highest preference value p is selected for the application

The majority of computational cost involved in the TOPSIS-based algorithm comes from steps 3 and 4, namely searching the best/worst attribute values and calculating the Euclidean distance, which is minimal. Therefore, the programming effort, computing time, and needed resources for implementing this algorithm should not be significant. The developed algorithm for network selection is based on the network attributes that service providers have published at the registry and the service request specified by users (applications). Therefore, the same algorithm is applicable to select heterogeneous network services to support applications with diverse requirements.

7. Performance Evaluation

In this section performance of the proposed SOA-based broker for network service discovery and selection is evaluated. The effectiveness of the event-based protocol for updating network service information is evaluated. We particularly examined the communication overhead between the registry and SPs for updating network state changes because it is the key factor of the system scalability. Assume that m services, $S_j, j=1, 2, \dots, m$, are published at the registry. Each network service has k states whose changes must be updated at the service registry. Without the event-based update protocol, an SP must send an entire service description document (assumed to have L unit length) to the registry if any one of the k states changes. With the event-based updating protocol, the service registry subscribes to an SP for receiving notification of changes of these k states. We assume that the length for a notification message of any single state change is $f \cdot L$ units, where $0 < f < 1$ implies that the update message is just a fraction of the entire service description document. Since the

notification message only contains the identification of the changed state and the new value of this state, it is typically much shorter than a service description document.

Suppose the probability that a single state changes is p . Without the event-based updating protocol, the probability that a network service has a state change thus sending a new service description document to the registry, is $1-(1-p)^k$. Therefore the average communication overhead for updating information for a single network service is $L \cdot (1-(1-p)^k)$. With the event-based updating protocol, one notification message is generated for each changed state in a network service. The probability that j out of k states change in each time unit has a Binomial distribution with an expectation kp . Therefore, the average communication overhead between a network service and the registry is $fL \cdot kp$.

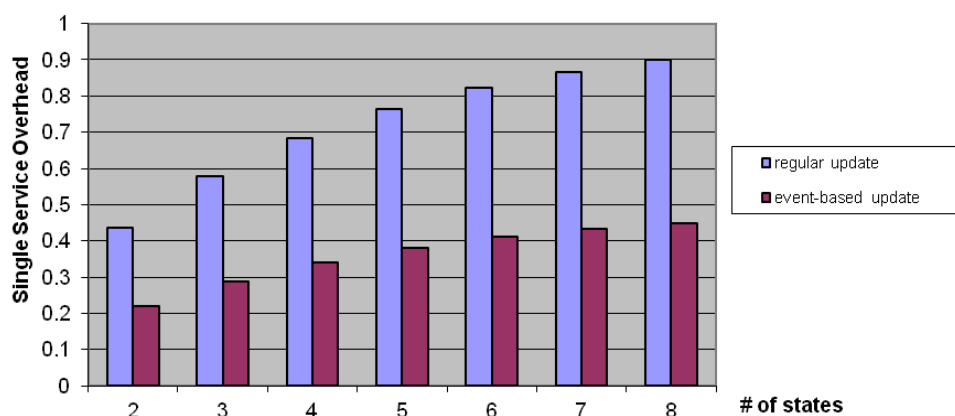


Figure 6 Communication overhead of a single network service with different number of states.

We analyzed a networking scenario in which each state in a network service changes with a probability $p=0.25$ and assume that an event notification message has a $1/8$ length of a service description document, i.e., $f=1/8$. Figure 6 gives the average communication overhead for updating a variety of numbers of states in a network service. These two curves in this figure respectively represent the update overhead (measured in the unit of L) with and without the event-based protocol. From the figure we can see that given a certain level of network dynamic (the state change probability p), the amount of communication overhead for keeping service information current at the registry always increase with the number of states in a network services. Comparing the two load curves we can see that the event-based protocol greatly reduces the amount of communication overhead for updating service information.

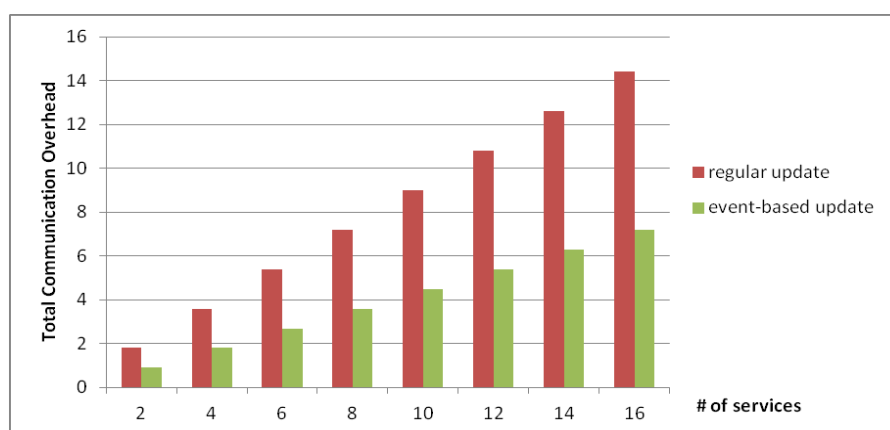


Figure 7 Total communication overhead for different number of network services.

Figure 7 gives the relationship between the total amount of overhead for network state update and the number of published network services. This figure indicates that the event-based update protocol significantly reduces the speed in which the total amount of overhead increases with the number of services. Therefore, the proposed protocol allows the broker system to support more network services with dynamic states without generating overwhelming communication overhead; thus enhancing the scalability of the broker system for network service discovery and selection.

In order to show the effectiveness of the developed algorithm for automatic network selection, a networking scenario with 5 different network services for supporting three types of applications was analyzed. These five network services are constructed based on typical network infrastructure that is widely available to regular users, including WiFi network, WiMAX network, and Ethernet network. The QoS-related network service attributes considered for service selection (listed in Table 1) include the total bandwidth (TB) offered by a service, available bandwidth in the service, bandwidth utilization, delay, delay jitter, packet loss rate, and transmit cost per byte charged by the service provider.

Three types of applications considered in this networking scenario are VoIP, video stream, and Web applications. Network service attributes have different relative importance to each application, for example cost per byte is much more important to Web applications than to the VoIP applications while delay and delay jitter are crucial to VoIP but only have a minor impact on Web applications. Therefore, each application will specify its own weight assignment for network service attributes when submitting a service request. Table 2 gives the weight assignment for each type of application.

The developed selection algorithm was employed for selecting a network service for each type of application. The preference values (P-value) of each network service obtained by the algorithm for these three types of applications are shown in Figure 8. From this figure we can see that each network service achieves quite different P values for different types of applications, which implies that applications with diverse requirements should be supported by different network services. Figure 8 shows that network service #5 has the largest P value for VoIP and video stream applications while network service #4 has the largest P value for

Web application.

Table 1 Network service attributes.

	Total bandwidth	Available bandwidth	Bandwidth utilization	Delay (ms)	Delay jitter (ms)	Packet loss rate	Cost per byte
Network service #1	2	1.8	10	400	50	100	100
Network service #2	11	8.8	20	200	25	20	20
Network service #3	54	37.8	30	100	15	15	10
Network service #4	100	60	40	150	30	20	5
Network service #5	100	80	20	100	20	15	30

Table 2 Weight assignment to network service attributes.

	Total bandwidth	Available bandwidth	Bandwidth utilization	Delay (ms)	Delay jitter (ms)	Packet loss rate	Cost per byte
VoIP application	0.05	0.05	0.15	0.3	0.3	0.1	0.05
Video stream application	0.1	0.2	0.2	0.1	0.1	0.2	0.1
Web application	0.05	0.15	0.1	0.05	0.05	0.1	0.5

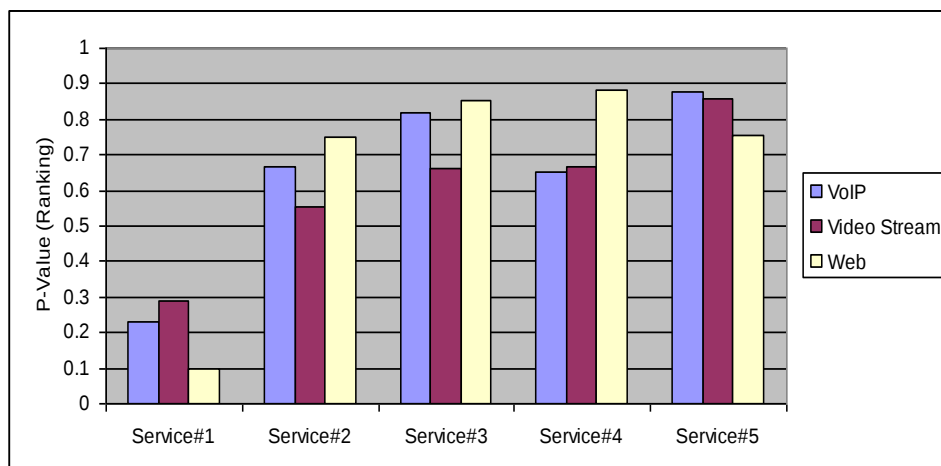


Figure 8 Network service selection for different types of applications.

Figures 9 and 10 respectively show the preference values of the five network services obtained by the selection algorithm for VoIP and Video stream applications with 10%, 20%, and 30% of variation in service attributes. From these two figures we can see that dynamic network states, reflected by the variation of network attributes, always have an influence on

the preference values of network services. However, a certain level of attribute variation may or may not change the result of network selection, which is also dependent on the types of applications and their weight assignments on different attributes. For example, in the analyzed cases, as shown in Figures 9 and 10, selection result for the video stream application is changed by 10% variation while the network selected for VoIP does not change even with 30% variation.

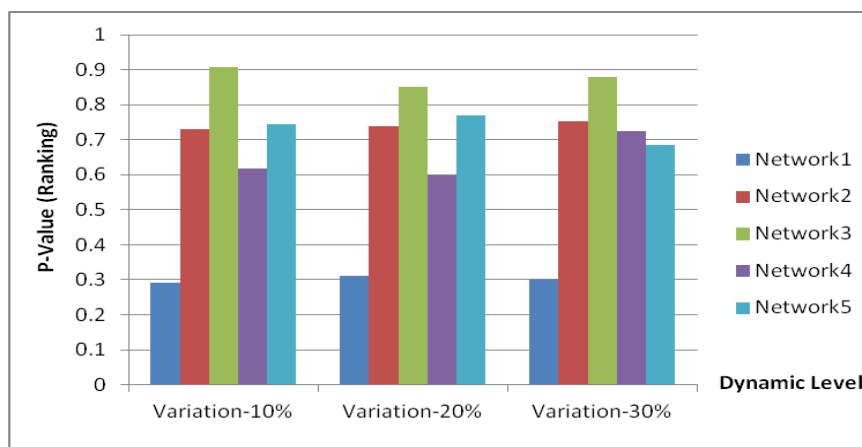


Figure 9 Network service selection for VoIP with variations in attribute values.

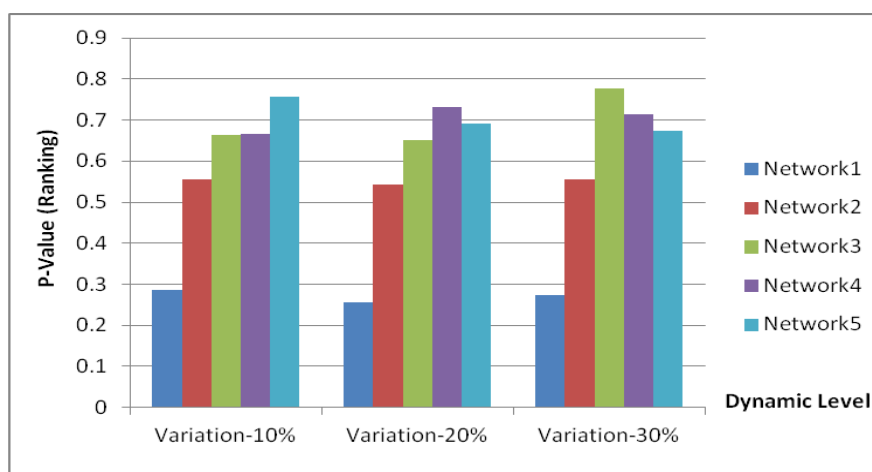


Figure 10 Network service selection for Video stream with variations in attribute values.

8. Conclusion and Discussion

Network virtualization is expected to play a crucial role in the future Internet. By de-coupling service provisioning from network infrastructure, network virtualization significantly lowers the barrier to entry of Internet service market; thus allowing a large number of service providers coexisting in the future Internet. Therefore, discovering and selecting the appropriate network services for supporting applications with diverse requirements becomes an important research problem for constructing virtualization-based future Internet. The research work presented in this paper tackles this challenging problem by

applying the SOA principle in a network virtualization environment. The main contributions made in this paper include a SOA-based broker system for automatic network service discovery and selection, a service information update protocol that improves the broker performance in large scale dynamic networking environment, and an algorithm based on multi-attribute decision making technique that can be used at the broker for network service selection. Performance evaluation is also conducted to show the effectiveness of the broker system with the developed protocol and algorithm.

Though simplify the interaction between network service providers and users, a centralized service broker/registry also introduces a point of failure. Therefore, the broker and registry should be hosted on a highly reliable server or duplicated servers operating in parallel for redundancy. To further address the reliability issue, a SOA-based broker system can be implemented with a distributed structure consisting of multiple brokers and registries that cooperate on service discovery and selection. This will improve the system reliability and may also further enhance scalability. The event-based update protocol and the network selection algorithm developed in the paper are applicable to a distributed broker system. However, a distributed broker system also brings in some new challenges. For example, the collaboration among brokers and information partition/synchronization among registries are important issues that must be fully addressed in future research.

References

- [1] N. M. Chowdhury and R. Bourtaba, "Network virtualization: state of the art and research challenges," IEEE Communications Magazine, Vol. 47, Issue 7, pp. 20-26, July 2009.
<http://dx.doi.org/10.1109/MCOM.2009.5183468>
- [2] J. Turner and D. E. Taylor, "Diversifying the Internet," Proc. of the 2005 IEEE Global Communication Conference, Nov. 2005. <http://dx.doi.org/10.1109/GLOCOM.2005.1577741>
- [3] N. Feamster, L. Gao, and J. Rexford, "How to lease the Internet in your spare time," ACM SIGCOMM Computer Communications Review, Vol. 37, Issue 1, pp. 61-64, Jan. 2007.
<http://dx.doi.org/10.1145/1198255.1198265>
- [4] K. Channabasavaiah, K. Holley, and E. Tuggle, "Migrating to a Service-Oriented Architecture," IBM DeveloperWorks, Dec. 2003.
- [5] T. Magedanz, N. Blum, and S. Dutkowski, "Evolution of SOA Concepts in Telecommunications," IEEE Computer Magazine, Vol. 40, Issue 11, pp. 46-50, Nov. 2007.
<http://dx.doi.org/10.1109/MC.2007.384>
- [6] D. Griffin and D. Pesch, "A Survey on Web Services in Telecommunications," IEEE Communications Magazine, Vol. 45, Issue 7, pp. 28-35, July 2007.
<http://dx.doi.org/10.1109/MCOM.2007.382657>
- [7] E. Grasa, G. Junyent, S. Figuerola, A. Lopez, and M. Savoie, "UCLPv2: A Network Virtualization Framework Built on Web Services," IEEE Communications Magazine, Vol. 46, Issue 4, pp. 126-134, April 2008. <http://dx.doi.org/10.1109/MCOM.2008.4463783>

- [8] S. Figuerola and M. Lemay, "Infrastructure Services for Optical Networks," IEEE/OSA Journal of Optical Communications and Networks, Vol. 1, Issue 2, pp. A247-257, July 2009. <http://dx.doi.org/10.1364/JOCN.1.00A247>
- [9] G. Branca, P. Anedda, and L. Atzori, "Transport Stratum Services in NGN: a SOA-oriented Design," Proc. of the 2010 IEEE Global Communication Conference, Dec. 2010. <http://dx.doi.org/10.1109/GLOCOM.2010.5683382>
- [10] M. E. Barchi, N. Kara, and R. Dssouli, "Toward a Service-Oriented Network Virtualization Architecture," Proc. of the 2010 ITU-T Kaleidoscope Conference, Dec. 2010.
- [11] Q. Duan, "End-to-End Service Provisioning in Network Virtualization for the Next Generation Internet," Chapter 12 in book *Future Internet Services and Service Architectures*, River Publishers, June 2011.
- [12] Q. Duan, "End-to-End Modeling and Performance Analysis for Network Virtualization in the Next Generation Internet," *International Journal of Communication Networks and Distributed Systems*, Vol. 8, Issue 1/2, pp. 53-69, Jan. 2012. <http://dx.doi.org/10.1504/IJCND.2012.044322>
- [13] J. van der Ham, P. Grosso, R. van der Pol, A. Toonk, and C. de Laat, "Using the Network Description Language in Optical Networks," Proc. of the 10th IFIP/IEEE International Symposium on Integrated Network Management, May 2007. <http://dx.doi.org/10.1109/INM.2007.374784>
- [14] A. Campi and F. Callegai, "Network resource description language," in Proc. of the 2009 IEEE Global Communication Conference, Dec. 2009. <http://dx.doi.org/10.1109/GLOCOMW.2009.5360708>
- [15] Q. Duan, "Network Service Description and Discovery for High-Performance Ubiquitous and Pervasive Grids," ACM Transactions on Autonomous and Adaptive Systems, Vol. 6, Issue 1, article 3, February 2011. <http://dx.doi.org/10.1145/1921641.1921644>
- [16] Organization for the Advancement of Structured Information Standards (OASIS), "Web Services Base Notification (WS-BaseNotification) v1.3," Oct. 2006

Copyright Disclaimer

Copyright reserved by the author(s).

This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).