Networking as a Service: a Cloud-based Network Architecture

Tao Feng, Jun Bi, Hongyu Hu and Hui Cao
Network Research Center, Tsinghua University
Department of Computer Science, Tsinghua University
Tsinghua National Laboratory for Information Science and Technology (TNList), Beijing, China
fengt09@mails.tsinghua.edu.cn, junbi@tsinghua.edu.cn, huhongyu@cernet.edu.cn and cao-h06@mails.tsinghua.edu.cn

***Abstract*—With the rapid development and integration of the Internet, wireless communication network and the Internet of Things, the Internet faces many challenges as a bearer network: a large volume of information exchange, multi-level QoS and smoothly switching multiple access protocols. The Internet should be able to provide a variety of network capacities in a more dynamic and on-demand way, not just limited network resource provision through virtualization. The elastic network is expected to adapt to network changes by enabling network protocols selection and combination dynamically. Cloud computing illustrates a new Internet-based model of IT resources (hardware, software, data) provision, delivery and consumption as a service. Therefore, networking as a service can provide guaranteed quality of service and good quality of experience to users who do not care about any network configuration and network management. In this paper, we propose a novel idea of networking as a service by combining the service provision model of cloud computing with the openness of the network protocol. The related conception and stakeholders of networking as a service is depicted. Cloud-based network architecture is design to present the provision, delivery and consumption of networking as a service and discuss the key features of cloud-based network. Finally, a prototype of cloud-based network is implemented by extending OpenFlow architecture.**

***Index Terms*—network capacity; networking as a service; cloud computing; network architecture**

I. Introduction

With the rapid development and integration of the Internet, wireless communication network and the Internet of Things, the Internet faces many challenges as a bearer network in the future: a large volume of information exchange, multi-level QoS, smoothly switching multiple access protocols, mobility and management. The design philosophy of the current Internet [12] limits the flexibility of the network architecture to meet new requirements. For instance, the end-to-end argument proposes that a network simply forwards packets between end-systems while complex data processing function is implemented on the end-systems [33]. At present, the Internet is expected to handle more complex and customized forwarding capacity in the company of more and more mobile devices connected and new client/server paradigm of cloud computing. Even the current Internet provides a lot of new features or services that go beyond forwarding in order to deal with more tussles [13], such as heterogeneous network resources, personalized delivery service, trust network access and low-cost network maintenance, and this shift towards more network capacities will continue.

New network features or services are difficult to be introduced into the current network because a network protocol is locked in a vendor device. Network service should be decoupled from specific data transport technologies so that new features or services can be deployed freely. In addition, the Internet should be able to provide a variety of network capacities in a more dynamic and on-demand way, not just limited network resource provision through virtualization [7]. The elastic network is expected to adapt to network changes by enabling network protocols selection and combination dynamically. Cloud computing [1] illustrates a new Internet-based model of IT resources (hardware, software, data) provision, delivery and consumption as a service. Therefore, network capacity on demand can provide guaranteed quality of service and good quality of experience to users who do not care about any network configuration and network management.

There will be little place either for static network configurations like the current network stack or for manual optimization and tuning as enforced at the inter-layer boundaries of the current network in such a dramatically dynamic network operational circumstance. On the opposite end, the revolution that removes such constraints and at the same time maximizes the on-demand capacity of network will play a key role in the evolution towards future network.

In this paper, we propose a novel idea of networking as a service by combining the service provision model of cloud computing with the openness of the network protocol. The related conception and stakeholders of networking as a service is depicted. Cloud-based network architecture is design to present the provision, delivery and consumption of network protocol as a service and

This is the extended version of our paper at ICISCI'10.
Corresponding author: Tao Feng, fengt09@mails.tsinghua.edu.cn

doi:10.4304/jnw.6.7.1084-1090

discuss the key features of cloud-based network. Finally, a prototype of cloud-based network is implemented by extending OpenFlow architecture.

II. RELEATED WORK

The design of the current Internet architecture has been rethinking and some architectural principles for new Internet architectures have been proposed for some years on structuring a new generation of network protocols [14], adding mechanisms to the core of the Internet [15] and exploring specific architectural issues [16]. The trend of the Internet towards a commercial development has changed the underlying hypotheses of trust and economic incentives [17].

Several new features have been proposed and implemented in the Internet with the development of the Internet, which was not considered in the initial design of network architecture. Since the Internet does not support the dynamic deployment of new protocols, on-demand composition of network protocols and pay-as-you-go business model of network capacity, these features needed to be added as special processing functions to future network. Some of the current research gives a ray of hope for network capacity on demand. The following list highlights such features and functions:

A. The openness of the network protocol

The openness of the network protocol refers to the future that the introduction, deployment and operation of network protocol or service can be achieved on minimum cost by standardizing network interfaces and enhancing interoperability of network protocol. The openness of the network protocol is a prerequisite for network protocol as a service. The realization of the openness future makes the network protocol custom and flexible adoption according to different application scenarios and user requirements. Open architectures and analogous work on the openness of networks have contributed ideas for the programming behavior of a node [29, 30] and flow-driven modification of the data plane services [31]. To manage the complexity of new protocol in the network, a working group of IETF has attempted to define Open Pluggable Edge Services (OPES) [19]. In such architecture, a set of data flow operations that are implemented on nodes throughout the network can be specified in end-systems. Currently, there are two ways to achieve the openness of the network protocol in the control plane: the out-box openness and the in-box openness. OpenFlow [2] is one of the implementations of the out-box openness. OpenFlow provides a way to control network device by network protocols running outside of a network device. OpenFlow achieves a variety of network behaviors on the switch by controlling the flowtable such as routing, firewall, and so on. On the other hand, the JUNOS SDK [3] is another way to open network protocol. The JUNOS SDK enables developers to innovate on top of JUNOS and Juniper Networks platforms, so developers can create, deploy, and validate innovative applications tailored to specific needs.

B. The modularity of the network protocol

Modularity is central tenets in the design and implementation of hardware and software system. In the paper of [16], the modularity of the network architecture is defined that breaks a network system into parts, normally to permit independent construction and replacement, reuse of parts, and so on. Early works on modular protocols have provided some solutions on protocol decomposition, configurable frameworks and process model. [8] proposed an x-kernel environment and mechanisms for communication between micro-protocols. [9] is a configurable communication framework that provides a runtime platform of protocols consisting of standard, reusable services. [10] proposed a process-per-protocol model, a process or thread shepherds a message through the protocol stack. Some work [11] has proved the success of a modular platform with widespread deployment for reconfiguration of the entire data plane of a router system. Active networks [18] provided a powerful and very general approach to module packet processing function.

C. Service-oriented network protocol and network architecture

The research on service-oriented network protocol composition and network architecture enables dynamic adjustment of network features possible according to the requirement of users and applications. Service-oriented network architecture provides mechanisms for composing custom protocol stack [23], such as the SILO architecture [22]. The key technologies on service-oriented network protocol include abstraction of network service, network protocol composition and service path selection. A number of previous research projects have addressed some general thinking about how to specify a network service. [20] emphasize specifically on the middle boxes in the network such as traversing firewalls and network address translators, it can be seen as a step towards managing connections involving general services. [21, 35] provided a more general method that specifies services very similar to pipeline abstractions. Service socket [28] is a user-level abstraction that has implemented some networking applications and services in networks. Some approaches focus on the decentralization of service composition. The SpiderNet project [24] provides the ability of service composition by a decentralized approach in P2P networks. A similar research about service composition [25] discuss the challenges that how to support for the service compositio on top of the Internet Indirection Infrastructure (i3). In [26, 27], the path selection is also done in a distributed manner, but end-systems and other entities along the path may specify specific service requirements.

D. The appropriate mechanism of the provision, delivery, and consumption of network protocol

The appropriate mechanism of the provision, delivery, and consumption of network protocol is becoming an important foundation for network protocol as a service. A network protocol developed by JUNOS SDK developers needs to deploy and run on each network device of

Juniper. The development can gain development fees based on a software license. Cloud computing is different from JUNOS SDK. The devices and services are centralized deployment and running in data centers. Service developers can publish and sell their own software services to the cloud service provider.

III. Networking as a Service

A. Conceptualisation

Networking as a service refers to a new Internet-based model that communication service provider (CSP) can deliver network protocols on-demand and reliably to the user based on SLA. The service consumer can use the service as pay as you go and achieve a good quality of experience.

From the perspective of service, the abstraction of network function and the layer of network protocol stack will be re-organized and divided into three layers: service specification, network capacity, network behavior. In this vision, network service in different abstract forms will regard as middle ground for the continuous resolution of tussles between providers and users. At the top level of abstraction the service specification which defines data transmission parameters of user information needs to satisfy end user requirements. At the intermediate level of abstraction network capacity in accord with service specification is set up with network protocol composition. The dynamics of network capacity construction provides a utility function for the composition of horizontal service across the network and vertical service within a node. In this case, it is the service that utilizes the network and drives the customization of network capacity. As we move to lower levels this customization process is mapped to network behavior, access technologies and resource management policies such as forwarding, filtering, dropping, and so on.

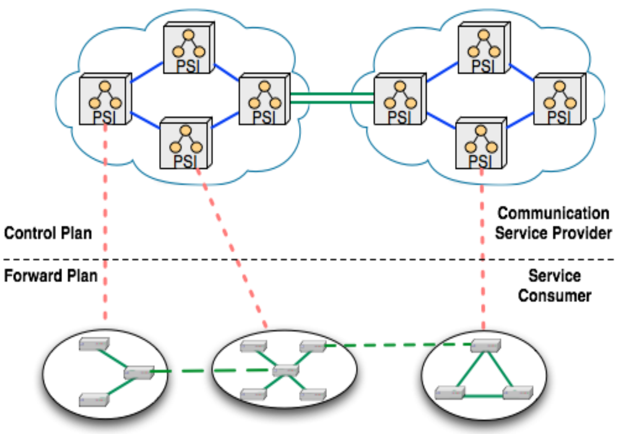

Figure 1. Cloud-based network.

Cloud-based network (CBN), as shown in Fig. 1, is a form of implementation of networking as a service, which learns from concepts and ideas of cloud computing and service-oriented architecture. CBN provides the ability to deploy and run network protocols in the cloud, configure network resources and compose network protocol dynamically according to the user's service requirements and SLA, accordingly generate network control rules to manipulate forwarding behavior of a network device. CBN transforms network protocol to network service with zero-configuration [4] and zero-maintenance for network users. Each CSP may set up a CBN or the federation of CBNs to serve for the costumers.

Protocol service instance (PSI) is a set of network protocols corresponding to each service requirement. PSI is the minimum unit of a network service in CBN.

B. Stakeholders in Cloud-based Network

In the current Internet business model, network-related stakeholders consist of end users, communication service providers and network equipment providers formed. In this case, end users pay for communications service providers to apply for network access services by communications service providers. Communications service providers are regard as a “pipeline” manager since it is almost impossible that communication service providers can deploy a new protocol to provide customized or value-added network services because the protocols are embedded in the device by network equipment providers. The business model of the Internet will be changed with the emergence of networking as a service. Communications service providers will enhance the ability to control the network. The role of network equipment providers will be subdivided. Users will apply for appropriate network services according to their need and consume the service in the way of pay-as-you-go with business development, thereby reduce the cost of network investment and maintenance.

Communication Service Providers: Communication service providers are responsible for the management and maintenance of network-based cloud and the provision of a guaranteed quality of network services to service consumers. CSP can gain service revenue from service consumers according to the period, quality, quantity and scale of network service.

Protocol Developers: Protocol developers can develop various network protocols with API and specification of CBN. After passed a test, a network protocol can be published to the CBN. Protocol developers can gain license fees from CSP according to the scale of deployment and frequency of running of the network protocols.

Network Equipment Providers: In the CBN, the various components and interfaces of network devices will be standardized. Thus, the function of network equipment providers will be refined and divided into network components providers and network equipment integrators. Network component providers will focus on improving the performance, capacity of network components, while network equipment integrators will focus on improving the stability and reliability of network equipment composed by network components provided by network component providers.

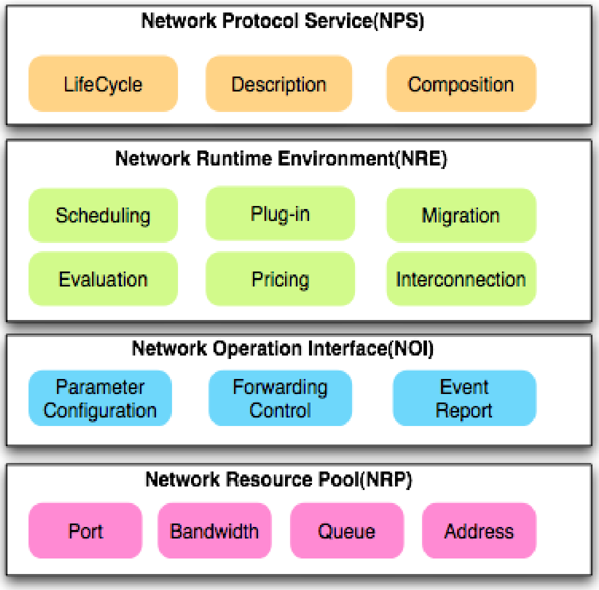

Figure 2. The reference architecture of Cloud-based network

Network Service Consumers: Network service consumers, including personal and business users, purchase network services in a "pay as you go" model. They do not need to purchase expensive network equipment, do not need to care about the network configuration and maintenance. They do only need to put forward the requirements of network services according to the development of the business. And then use it.

IV. Cloud-based Network Reference Architecture

Cloud-based network reference architecture, as shown in Fig. 2, is divided into four layers: network resource pool, network operation interface, network runtime environment and network protocol service. Network operation interface is implemented in network device to manipulate network resource. Network runtime environment is a platform for network protocol deployment and operation. Network protocol service generates control rules and call for network operation interface to control and manage network resource.

A. Network Resource Pool

Network resource pool (NRP) is the network resource such as ports, bandwidth, queue, address, which can be as a basic service related with packet forwarding. Examples are Amazon EC2 for IP address and bandwidth assignment. Instead of some components of raw network hardware, NRP typically offers the combination of these resources as a service through unified configuration and management.

B. Network Operation Interface

Network operation interface (NOI) is open and standardized API in order to configure and manage NRP. NOI provides three types of operating functions: parameter configuration for the network resource, forwarding control and event report. Parameter configuration function provides to set or get the max or minimum bandwidth limit, the numbers of queues, and IP address of a port, which can construct a user-oriented network topology. Forwarding control function provides abilities to output, drop, and filter packets according to the rules. Event report function provides an alarm or trap information when some network resource is down or overload.

C. Network Runtime Environment

Each of protocol set is called protocol service instance (PSI) which can be set up and running as a plug-in in network runtime environment (NRE). There is always a daemon running as a default PSI that provides a basic network layer protocol such as IP. NRE is responsible for billing, resource allocation, assessment, interconnect and reliability assurance for each PSI.

Scheduling: According to the network service requests and current SLA state of the user, the scheduling function, firstly, will search related network resources and network protocols. If the requests are satisfied, the scheduling function will reconfigure network resource properties and generates PSI to control the rule of packet forwarding by issuing to the network equipment. Meanwhile, the scheduling function will inform the user the service is working and start the billing.

Plug-in: Plug-in function enables network protocols to deploy, start, stop, upgrade and uninstall without reboot the system, just like OSGi, which is a module system and service platform for the Java programming language that implements a complete and dynamic component model, something that does not exist in standalone Java/VM environments. Thus, network protocol in PSI can be dynamically adjusted and smoothly switched by Plug-in feature.

Pricing: To regard network capacity as a service, there will be a new billing model instead of bit-per pricing or online-time pricing: network capacity-based pricing and flow-per pricing. Pricing function enables to calculate the cost of service consumers according to the amount of involved network protocols and run time of each PSI.

Evaluation: Evaluation function provides the ability to monitor the operational status and network resource usage of PSI. The evaluation feature can determine whether the service provided by the PSI match the SLA through checking the service request.

Interconnection: Interconnection function provides a communication mechanism and interface between multi PSIs. A PSI can send and receive the status information of a protocol through an interconnection interface such as JSON when it needs access or negotiates some information of protocols in another PSI. Eventually, network clouds can be interconnected by the multi PSIs interconnection.

Migration: Migration function provides the mobility of network services in the PSI level against network failures and high load. The PSI migration process includes: PSI state capture, marshaling PSI state and PSI service relocation.

D. *Network Protocol Service*

With the openness of network, a variety of new network protocol will be designed and implemented. How to identify and manage the new network protocols is a new problem in the future. Network protocol service consists of three functions: the description, management and composition of network protocol.

Service Description of Network Protocol: Service description of network protocol is a structured language such as XML that provides a model for describing the capabilities of a network protocol. The NRE can choose appropriate network protocols to set up a PSI that can meet the user's demands. Therefore, a network protocol should be able to accurately express properties and forward capacities of network protocol.

Service Lifecycle Management of Network Protocol: The feature of service lifecycle management of network protocol provides the functions to manage network protocol versions, registration, certification and licensing.

Service Composition of Network Protocol: The feature of service composition of network protocol provides the ability to generate new, more powerful network protocol service by composing protocols with different functions. The service composition of network protocols may learn from context aware service composition [5] and semantic web service composition [6].

V. Implementation

OpenFlow is an ideal way to build a network cloud. OpenFlow [36] is an open standard that allows network researchers to run experimental protocols in production network. It provides an open protocol to program the flowtable in a network device. The protocols implemented in a server outside control network devices by OpenFlow protocol, which is embedded in a device currently. It is in the process of being implemented by major switch vendors and used today by universities to deploy innovative networking technology. Thus, openness of network protocol in OpenFlow provides the possibility of network protocols as a service.

NOX [34] is an open-source OpenFlow controller intended to simplify the development of software for controlling or monitoring networks composed of OpenFlow switches. Programs written within NOX (using either C++ or Python) have flow-level control of the network. This means that they can determine which flows are allowed on the network and the path they take. In addition, NOX provides abilities to access to the network state including the network topology and the location of all detected hosts.

Apache Hadoop is an open source distributed processing framework. The framework split dataset into manageable blocks in order to compute large datasets. It is in charge of the whole process by launching protocol instances, processing the protocol messages across many machines where the protocol is physically deployed and, at the end, aggregating the set of forwarding rules output into a final result [32].

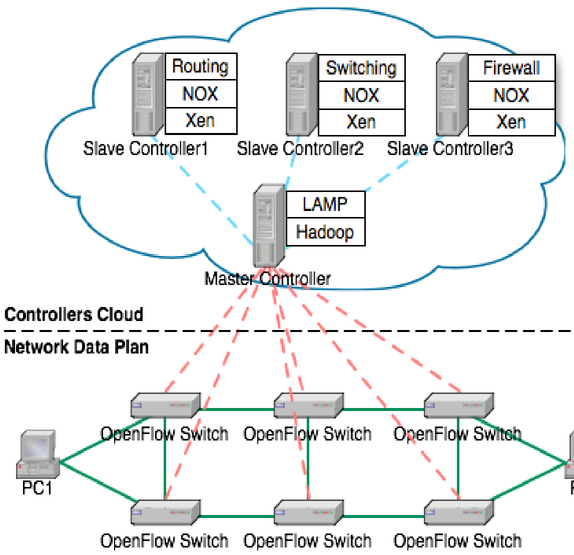

Figure 3. An initial prototype of Cloud-based network.

In our laboratory, an initial prototype of Cloud-based network, as shown in Fig. 3, has been implemented by extending OpenFlow architecture. The implementation is divided into two levels: controllers cloud plan and network data plan. Controllers cloud plan provides the functions of NRE and NPS in Cloud-based Network Reference Architecture. A master controller with Apache Hadoop is responsible for distributing the data stream to three slave servers with different protocols. The LAMP on the master controller is responsible for network protocol registration and lookup. Xen deployed on each slave server make multi slave server as a slave server cluster. Network data plan consists of six OpenFlow switches to receive control information of flows from the master server in controllers cloud plan.

VI. Future Work

In this article, we have demonstrated the conception and role of on-demand provision of network capacity in order to achieve networking as a service. We have designed a novel future network architecture leaned from cloud computing and service-oriented architecture: cloud-base network.

Based on the cloud-based network, we have built a prototype by extending the OpenFlow architecture and virtualization technology to verify on-demand provision of network capacity.

Moving forward, there are some future works in cloud-based network architecture. First of all, based on the above prototype implementation, we will evaluate the performance and latency of network protocol as a service in the cloud-based network. And we will research an accurate expression of the demand for network services. A formal network protocol description language will be designed to configure network services automatically according to an accurate expression of network service requirements. Then, we will extend the current prototype implementation to multi-clouds interconnection by designing a cloud interconnection communication protocol. In the multiple clouds based network prototype, we will research the capability of service optimization to provide a service consumer with the nearest service delivery. Finally, we will research the migration ability of PSI in multiple clouds to improve the reliability of the cloud-based network.

ACKNOWLEDGMENT

This work was supported by National Science Foundation of China under Grant 61073172, Program for New Century Excellent Talents in University, and National Basic Research Program ("973" Program) of China under Grant 2009CB320501.

REFERENCES

[1] NIST Definition of Cloud Computing v15, csrc.nist.gov/groups/SNS/cloud-computing/cloud-def-v15.doc
[2] N. McKeown, T. Anderson, H. Balakrishnan,G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. Openflow: enabling innovation in campus networks. SIGCOMM Comput. Commun. Rev.,38(2):69–74, 2008.
[3] J. Kelly, W. Araujo, and K. Banerjee, “Rapid service creation using the JUNOS SDK,” in ACM PRESTO, 2009.
[4] S. Cheshire and D. H. Steinberg, “Zero Configureation Networking: The Definitive Guide,” O’Reilly, 2006,.
[5] Vukovic M Context Aware Service Composition. PhD thesis, University of Cambridge, 2006
[6] RAO, J. Semantic Web service composition via logic-based program synthesis. PhD thesis. Department of Computer and Information Science, Norwegian University of Science and Technology, 2004.
[7] N. M. K. Chowdhury and R. Boutaba, “A Survey of Network Virtualization”, Technical Report, David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, Tech. Rep. CS-2008-25, Oct 2008.
[8] N. Hutchinson, L. Peterson, The x-kernel: An architecture for implementing network protocols, IEEE Transactions on Software Engineering 17 (1991) 64–76.
[9] M. Zitterbart, B. Stiller, A. Tantawy, A model for flexible high-performance communication subsystems, IEEE Journal on Selected Areas in Communications 11 (1993) 507–518.
[10] D. Schmidt, T. Suda, Transport system architecture services for high-performance communications systems, IEEE Journal on Selected Areas in Communications 11 (1993) 489–506.
[11] R. Morris, E. Kohler, J. Jannotti, M. Kaashoek, The click modular router, SIGOPS Operating Systems Review 33 (1999) 217–231.
[12] D. D. Clark, “The design philosophy of the DARPA internet protocols,” in Proc. of ACM SIGCOMM 88, Stanford, CA, Aug. 1988, pp. 106–114.
[13] D.D. Clark, J. Wroclawski, K.R. Sollins, and R. Braden, “Tussle in Cyberspace: Defining Tomorrow’s Internet,” Proc. ACM SIGCOMM 2002, pp. 347-356.
[14] D. D. Clark and D. L. Tennenhouse, “Architectural considerations for a new generation of protocols,” in Proc. of ACM SIGCOMM 90, Philadelphia, PA, Sept. 1990, pp. 200–208.
[15] M. S. Blumenthal and D. D. Clark, “Rethinking the design of the internet: the end-to-end arguments vs. the brave new world,” ACM Transactions on Internet Technology, vol. 1, no. 1, pp. 70–109, 2001.
[16] D. Clark, K. Sollins, J. Wroclawski, D. Katabi, J. Kulik, X. Yang, B. Braden, T. Faber, A. Falk, V. Pingali, M. Handley, and N. Chiappa, “New Arch: future generation internet architecture,” Tech. Rep., Dec. 2003.
[17] S. Shenker, D. Clark, D. Estrin, and S. Herzog, “Pricing in computer networks: reshaping the research agenda,” SIGCOMM Computer Communication Review, vol. 26, no. 2, pp. 19–43, 1996.
[18] Tennenhouse, D. L., and Wetherall, D. J. Towards active network architecture. ACM SIGCOMM Computer Communication Review 26, 2 (Apr. 1996), 5–18.
[19] Barbir, A., Reinaldo, P., Chen, R., Hofmann, M., and Hilarie, O. An architecture for open pluggable edge services (OPES). RFC 3835, Network Working Group, Aug. 2004.
[20] Guha, S., and Francis, P. An end-middle-end approach to connection establishment. In SIGCOMM ’07: Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications (Kyoto, Japan, Aug. 2007), pp. 193–204.
[21] Keller, R., Ramamirtham, J., Wolf, T., and Plattner, B. Active pipes: Program composition for programmable networks. In Proc. of the 2001 IEEE Conference on Military Communications (MILCOM) (McLean, VA, Oct. 2001), pp. 962-966.
[22] Rudra Dutta, G. N. R., Baldine, I., Bragg, A., and Stevenson, D. The SILO architecture for services integration, control, and optimization for the future internet. In Proc. of IEEE International Conference on Communications (ICC) (Glasgow, Scotland, June 2007), pp. 1899–1904.
[23] Baldine, I., Vellala, M., Wang, A., Rouskas, G., Dutta, R., and Stevenson, D. A unified software architecture to enable cross-layer design in the future internet. In Proc. of Sixteenth IEEE International Conference on Computer Communications and Networks (ICCCN) (Honolulu, HI, Aug. 2007).
[24] Gu, X., Nahrstedt, K., and Yu, B. SpiderNet: An integrated peer-to-peer service composition framework. In Proc. of Thirteenth IEEE International Symposium on High-Performance Distributed Computing (HPDC) (Honolulu, HI, June 2004), pp. 110–119.
[25] Lakshminarayanan, K., Stoica, I., and Wehrle, K. Support for service composition in i3. In Proc. of the 12th annual ACM international conference on Multimedia (New York, NY, Oct. 2004), pp. 108–111.
[26] Huang, X., Ganapathy, S., and Wolf, T. A scalable distributed routing protocol for networks with data-path services. In Proc. of 16th IEEE International Conference on Network Protocols (ICNP) (Orlando, FL, Oct. 2008).
[27] Fu, X., Shi, W., Akkerman, A., and Karamcheti, V. CANS: composable, adaptive network services infrastructure. In Proc. of the 3rd USENIX Symposium on Internet Technologies and Systems (USITS) (San Francisco, CA, Mar. 2001), pp. 135–146.
[28] Schmitt, M., Acharya, A., and Ibel, M. Service Sockets: A uniform user-level interface for networking applications. Tech. Rep. TRCS99-39, University of California, Santa Barbara, Santa Barbara, CA, Dec. 1999.
[29] D.J. Wetherall, J. Guttag, and D.L. Tennenhouse, “ANTS: A Toolkit for Building and Dynamically Deploying Network Protocols,” Technical Report, MIT, 1997, in Proc. OPENARCH’98.
[30] T. Meyer, L. Yamamoto, C. Tschudin, An artificial chemistry for networking, in: Bio-Inspired Computing and

Communication, First Workshop on Bio-Inspired Design of Networks (BIOWIRE 2007), pp. 45–57.
[31] D. S. Alexander, W. A. Arbaugh, M. W. Hicks, P. Kakkar, A. D. Keromytis, J. T. Moore, C. A. Gunter, S. M. Nettles, and J. M. Smith, "The SwitchWare active network architecture," IEEE Network, vol. 12, pp. 29–36, Aug. 1998.
[32] VARIA, J.2009. Cloud Architectures. Amazon Web Services.
[33] T. Wolf, "Service-centric end-to-end abstractions in next-generation networks," in Proc. of Fifteenth IEEE International Conference on Computer Communications and Networks (ICCCN), Arlington, VA, Oct. 2006, pp. 79–86.
[34] NOX: An OpenFlow Controller. http://noxrepo.org/wp/.
[35] S. Shanbhag and T. Wolf, "Implementation of end-to-end abstractions in a network service architecture," in Fourth Conference on emerging Networking EXperiments and Technologies (CoNEXT), Madrid, Spain, 2008
[36] The OpenFlow Switch Consortium. http://www.openflowswitch.org/

Tao Feng was born in Shandong, China, in 1979. He received the M.S. degree in communication engineering in 2006. He is currently working towards the Ph.D. degree in network technology form Tsinghua University, Beijing, China.

He has been selected for the APAN'31 fellowship program. His research interests include future network, data center network, QoS and network management.

Jun Bi received the B.S., M.S., and Ph.D. degree in computer science from Tsinghua University, Beijing, China. His dissertation studied Internet routing protocols and high performance routers and won the best dissertation award from Tsinghua University.

From 1999 to 2000, he was a postdoctoral scholar of High Speed Network Department in Bell Laboratories Research, Lucent Technologies, New Jersey, USA. From 2000 to 2003, he was a research scientist of Bell Labs Research Communication Science Division and Bell Labs Advanced Communication Technologies Center. His research interests include Next Generation Internet Architecture and Protocols, High Performance Routers/Switches, Source Address Validation, Internet Routing, IPv4/IPv6 Transition, etc.

Prof. Jun Bi is a full professor and director of Network Architecture & IPv6 Research Division, Network Research Center of Tsinghua University.

Hongyu Hu was born in Hubei, China, in 1976. She received Ph.D.degree in Beijing Institute of Technology, Beijing, China. She current is a Post Doctor in Network Research Center, Tsinghua University, Beijing, China.

Her current research interests include future Internet, QoS routing and IP multicast.

Hui Cao was born in Shandong, China, in 1980. She received the M.S. degree in communication engineering in 2006. She is currently working towards the Ph.D. degree in trust computing technology form Tsinghua University, Beijing, China.

Her research interests include trust computing, social network and complex network.