SCIENCE CHINA
Information Sciences

• **RESEARCH PAPER** •

November 2014, Vol. 57 112101:1–112101:24
doi: 10.1007/s11432-014-5134-4

Towards evolvable Internet architecture-design constraints and models analysis

XU Ke[1*], ZHU Min[1], HU GuangWu[1], ZHU Liang[1], ZHONG YiFeng[1], LIU Ying[2], WU JianPing[1,2] & WANG Ning[3]

[1] *Tsinghua National Laboratory for Information Science and Technology (TNList), Department of Computer Science and Technology, Tsinghua University, Beijing* 100084*, China;*
[2] *Network Research Center, Tsinghua University, Beijing* 100084*, China;*
[3] *Centre for Communication Systems Research, University of Surrey, Guildford* GU2 7XH*, UK*

Received November 10, 2013; accepted December 15, 2013

Abstract There is a general consensus about the success of Internet architecture in academia and industry. However, with the development of diversified application, the existing Internet architecture is facing more and more challenges in scalability, security, mobility and performance. A novel evolvable Internet architecture framework is proposed in this paper to meet the continuous changing application requirements. The basic idea of evolvability is relaxing the constraints that limit the development of the architecture while adhering to the core design principles of the Internet. Three important design constraints used to ensure the construction of the evolvable architecture, including the evolvability constraint, the economic adaptability constraint and the manageability constraint, are comprehensively described. We consider that the evolvable architecture can be developed from the network layer under these design constraints. What's more, we believe that the address system is the foundation of the Internet. Therefore, we propose a general address platform which provides a more open and efficient network environment for the research and development of the evolvable architecture.

Keywords evolvable architecture, design constraints, evolvability evaluation model, economic adaptability, general address platform

Citation Xu K, Zhu M, Hu G W, et al. Towards evolvable Internet architecture-design constraints and models analysis. Sci China Inf Sci, 2014, 57: 112101(24), doi: 10.1007/s11432-014-5134-4

1 Introduction

The emergence of the Internet is a revolution to human communication. The report released by the "Internet World Stats"[1)] has showed that there were more than 2.4 billion Internet users in the world till June, 2012, which means about one-third of the world's population now has access to the Internet. There is no doubt that the Internet has become a global communication infrastructure.

The Internet's open architecture is one important factor contributing to its success. The architecture is a set of layers and protocols [1] which constitute the basis of Internet communication. It determines the function of each component of the Internet and their relationships. Therefore, it plays a central role

*Corresponding author (email: xuke@mail.tsinghua.edu.cn)
1) http://www.internetworldstats.com/stats.htm.

© Science China Press and Springer-Verlag Berlin Heidelberg 2014

on the Internet, and it is always becoming the primary study we should make in the research of the Internet.

The early Internet design aims to provide simple and low-cost communication services such as file and Email transfer. In 1974, Cerf and Kahn [2] proposed the TCP protocol to provide reliable end-to-end packet-switched communication on unreliable paths, which promoted the development of the Internet architecture. The flexibility and openness of the architecture bring a huge success to the Internet. But the application environment of the Internet has changed. Besides, new access technologies such as Wi-Fi and wireless networks, and new computing technologies such as P2P and Cloud appear constantly. They put forward new requirements to the architecture. Actually, the role of the Internet has shifted from a "communication channel" to a "communication and data processing repository". However, it is difficult for the existing Internet architecture to adapt to these changes. The architecture faces great challenges in the following aspects.

Scalability. On February 3, 2011, the Internet Assigned Numbers Authority (IANA) officially announced that the last five blocks of IPv4 addresses were evenly distributed to five Regional Internet Registries (RIRs) including AfriNIC, APNIC, ARIN, LACNIC and RIPE NCC. In the same year on April 5, RIR APNIC announced the completion of its IPv4 address allocation. That means the "Global Internet Address Pool" is empty, which limits the Internet scalability[2),3)]. But this is inconsistent with the requirements coming from the continuously growing number of users and the diversification of access methods. The defects of Internet addresses are becoming more and more serious even with the help of the CIDR and NAT technologies.

Router also faces scalability challenges. The routing table of an IPv4 addresses based core router has more than 450000 entries currently[4)]. This brought enormous burdens to routers, yet there is no good way to suppress the explosive growth of the routing table. The lack of aggregation in IPv4 addresses is the fundamental reason of routing table explosion.

Security. The original intention of the Internet is to provide research communication between mutual-trust users and groups, especially universities and government research institutions. No security problem occurs in that circumstance, so the security of the Internet architecture did not catch attention and consideration at that time. With the commercialization of the Internet and its explosive growth worldwide, user groups have become more and more complicated and different in scales, objectives and qualities. Trust no longer exists among users. According to Cisco 2013 Annual Security Report[5)], network threat increased 19.8% in 2012 over 2011. The Symantec 2013 Internet Security Threat Report[6)] showed that 5291 new vulnerabilities have been discovered in 2012. Ubiquitous viruses, Trojans, worms, and Spam are constantly threatening the user's privacy and property.

Mobility. The successful combination of mobile phones and online application stores induces large of mobile devices accessing to the Internet. Mobile Internet has become one of the most hot topics in Internet research. ITU World Telecommunication announced that the number of active mobile broadband subscriptions grew by 40% between 2010 and 2011. And there were almost twice as many mobile-broadband as fixed-broadband subscriptions[7)]. This further increased the scalability pressure to the Internet. Furthermore, mobile protocols have brought new problems to the Internet such as mobile node identity and the switching between access networks.

Internet has gradually moved from a single data transmission network to an integrated information network involving data, voices and videos. Different businesses have different QoS (Quality of Service) needs in transmission. For example, video conferences and multimedia applications such as IPTV have strict requirements on transfer delay jitter and bandwidth. They need appropriate QoS assurance. However, the destination-based addressing mode and the Best-Effort-based transfer features in the existing architecture cannot guarantee the priority of different services.

2) http://en.wikipedia.org/wiki/File:Ipv4-exhaust.svg.
3) http://en.wikipedia.org/wiki/File:Huston_rir_ipv4_exhaustion_projection.png.
4) http://bgp.potaroo.net/as2.0/bgp-active.html.
5) http://www.cisco.com/en/US/prod/vpndevc/annual_security_report.html.
6) http://www.symantec.com/security_response/publications/threatreport.jsp.
7) http://www.itu.int/en/ITU-D/Statistics/Pages/stat/default.aspx.

Controllability and manageability. The emergence of a wide variety of new applications and technologies, as well as the dynamic changes of network environments, greatly increase the complexity of network management and control. The traditional Internet architecture lacks necessary information used for network control and management.

Numerous strategies have been put forward up to now, offering influential methods such as NAT (Network Address Translation) technology, IPsec, and DiffServ (Differentiated Service) to address the above mentioned issues. But most of these proposals can only address part of the problem in the Internet. What is worse, these methods may make the Internet more complicated and vulnerable because of the penetration of middlewares and new protocols. Architecture innovation becomes an urgent and state-of-the-art research subject recently.

Many countries in the world including United States, European Union and China have launched the future Internet research program. How to improve the existing architecture to meet all kinds of requirements is the key topic in the future Internet research. Although the corresponding research has been conducted for more than ten years, different opinions on the evolution of the architecture in academia and industry still exist. The central debate is whether to redesign a new architecture without limitation or to patch on the basis of the existing architecture. Alternatively speaking, it is a debate on the choice between the clean-slate approach and the dirty-slate approach in designing future Internet architectures. Academia supports for the former approach more, but industry tends to modify the existing Internet and solve the problems gradually.

Both ideas of the clean-slate approach and the dirty-slate approach promote the development of future Internet architectures effectively. The innovative idea and flexible concept of the clean-slate approach, and the stability of the dirty-slate in transition are particularly important to the development of future Internet architectures. Nevertheless, we find that it is difficult for researchers to get quite appropriate architecture design and transit from the traditional Internet architecture to the new architecture smoothly under an un-restriction environment. However, relying solely on the improvement of patching cannot meet kinds of demands systematically either. We believe that the continuous development of Internet architecture should find a tradeoff between these two approaches. Combining the advantages of both ideas together and evolving without changing the core principal of the existing architecture could be an option. In the light of this idea, we propose an evolvable Internet architecture.

The rest of the paper is organized as follows: In Section 2, related work on the subject is reviewed, including the clean-slate and the dirty-slate approaches of the architectures. Then in Section 3, the conception, the design principles and the corresponding technology roadmap of the evolvable architecture are given. After that, three important design constraint models used to ensure the construction of the evolvable architecture, including the evolvability evaluation model, the economic adaptability analysis model, and the manageability constraint model, are introduced in detail in Section 4. Section 5 describes the address system of the evolvable architecture in-depth, that is, the general address platform, which is the corner stone of the evolvable architecture. Finally, Section 6 concludes the paper and discusses some future work.

2 Overview of future Internet architecture development

As we mentioned earlier, academia and industry have conducted a comprehensive exploration on the Internet architecture from a different angles, including the theory of protocols, the implementation of technologies, the establishment of test platforms for the future Internet, etc. These ideas can be divided into dirty-slate approach and clean-slate approach. Rexford and Dovrolis have summarized these two approaches in [3].

2.1 Dirty-slate approach

The Internet Engineering Task Force (IETF) has published a number of Request for Comments (RFC) standards to address the Internet problems all the time. For example, the CIDR protocol used for increasing the scalability of IP address, the CDN (Content Distribution Network) architecture which can provide data transport QoS warranty, the Mobile IP protocol applied by mobility users, as well as the IPSec authentication protocol used in security guarantee. These standards have added new functions and made incremental patches to the current protocols and architecture on the premise of backwards-compatible. They insist to meet the future needs of Internet applications without destroying the current Internet architecture and applications. This is called a dirty-slate approach, which is easy to deploy, implement and protect the existing investment on the Internet efficiently.

Nevertheless, due to the inherent drawbacks in the traditional architecture design, the dirty-slate approach can only solve a small range of local issues on the Internet. Furthermore, the introducing of middlewares damages the hourglass feature of the traditional Internet architecture. The architecture has been evolved into a complex and cumbersome structure which is full of "patches" and faced with more serious vulnerability, scalability, management and interoperability issues.

2.2 Clean-slate approach

The clean-slate approach deems that the current application requirements have exceeded the handling ability of the traditional Internet. It is necessary to break out the limitations of the existing architecture, and design a new architecture which can meet a variety of design goals. The clean-slate approach aims to solve the problems of the existing Internet architecture fundamentally in a systematic way. This innovation approach helps us think about the existing Internet issues from another perspective. It brings about great technological innovation on the Internet.

Currently, numerous research projects have been carried out over the world's developed countries under the guidance of the clean-slate idea, such as, the FIND (Future Internet Network Design) project funded by the U.S. government, the future Internet testbed GENI (Global Environment for Network Innovations) supported by top national institutions and universities in America, the FIRE (Future Internet Research and Experimentation) of EU. Pan and Koponen et al. made a comprehensive summary to these projects in [4, 5]. Under the guidance of these projects, researchers have proposed various future Internet architecture frameworks, such as the architectures FII [5], XIA [6], OPAE [7] based on the idea of openness; the protocols AIP [8], HIP [9] with direction of ID/Locator separation idea; data-oriented network CCN [10], NDN [11], DONA [12]; and safety-oriented hierarchical naming architecture[13, 14].

Despite the clean-slate study has solved many Internet problems theoretically, it gets out of the situation of current architecture. Since the new architecture may not be compatible with the existing one, there exist serious deployment and transition issues in the clean-slate approach. Furthermore, completely replacing the existing infrastructure may lead to high economic cost which exceeds far more than the benefits it may gain. The failure in deployment of protocols such as IP multicast and IPsec in the past few years shows that ISPs are not inclined to adopt new programs if they lack adequate incentives.

3 Evolvable Internet architecture framework

Given the rigidity of the dirty-slate approach and the radical of the clean-slate approach, some researchers begin to seek for compromise solutions. For example, IETF has supported a project named SDN (Software Driven/Defined Network) [15] which separates the control plane of the routing system from its forwarding plane by a centralized controller without changing the TCP protocol and IP protocol. The FIA (Future Internet Architecture) plans also launched an IP-based project called Nebula[8]. Nebula mainly focuses on providing a secure routing and transmission system for cloud computing. And in response to the development bottleneck of the current Internet architecture, Clark [16] reviewed the initial design principles of Internet architecture. He believed that the development of the architecture is a process of continuously seeking the new balance point, and it is inaccurate and unnecessary to predict the final state of the technology and the architecture. He insisted that many of the existing design principles of

8) http://nebula.cis.upenn.edu/.

the architecture should not be hastily discarded because they are fundamental for the innovation and development of the Internet.

Besides the researches conducted internationally, the studies on the future Internet architecture also have received more and more attention in China. In 2003, a National Basic Research Program (also called "973" program) project named "Theoretical Research on New Generation Internet Architecture" led by Tsinghua University was initiated. The main research contents of this project include the basic theoretical study of Internet architecture, the design and implementation of the architecture, and the experimental platform construction. The details of this project can be found in Ref. [17]. This project has achieved some meaningful research results. For example, the definition of multi-dimensional scalability of the Internet architecture has been proposed in Ref. [18], which can be employed to study the Internet scalability problem in performance, scale and service scalability. Given the pre-effective results, this project has got continuous support from "973" program, and a new project "the basic research on the Internet architecture and its protocols" was launched in 2009. This new project further inherits and develops the pre-project's preliminary findings. It focuses not only on the architecture's theoretical exploration, but also on the basic protocols to complete the evolution. The main research contents consist of: (1) the scalability and evolvability of the Internet architecture, (2) the dependability and polymerizability of the large-scale routing, (3) efficient network transmission of massive data; (4) real-time delivery in wireless network, (5) the management of complex and autonomous network among users in different domains. In this project, the evolvability of the architecture has been proposed formally as an important theoretical issue to be resolved. The other research results obtained by this project include: a general framework of source address validation and traceback for IPv4/IPv6 transition scenarios [19], a logarithm-barrier-based multipath protocol for Internet traffic management [20], optimal traffic engineering with OSPF [21], etc.

Relying on the "973" program and researchers' preliminary work, we put forward a theory of evolution [22] used to construct the new generation Internet architecture. The primary idea of an evolvable architecture is making reformation under the core design principle of the current architecture to achieve technology innovation as well as smooth transition from the existing architecture to a new one. This is a compromising approach between the clean-slate approach and dirty-slate approach in nature, as SDN and Nebula do. The main difference between our evolvable architecture and other architectures is the object and starting point that the architectures focus. For instance, SDN is more concerned about the evolution of the routing layer. It builds on the basis of OpenFlow [23], which can realize the centralized control of the data forwarding by servers to separate the routing control and data forwarding. Nebula is designed to achieve cloud computing, whose main goal is to ensure reliable and secure transmission and storage of large-scale data. Our architecture tries to address the technology roadmap problem in architecture evolution. We consider evolvability as the property of the architecture and discuss the design principle and constrians to guarantee the evolvability of it. In this section, we will describe these issues in detail.

3.1 The conception of evolvable architecture

We believe that the development of the Internet architecture should do some revolution in the process of improvement, so that it can not only support the new applications and technologies but also protect the stable development of existing applications and technologies. As Figure 1 shows, both the clean-slate approach and the dirty-slate approach have advantages and disadvantages. It would be safe to say that these two approaches are mutually complementary in some way. Inspired by the advantages of both approaches and the development requirements of Internet architecture, we define the evolvable architecture as follows:

Evolvable architecture. The evolvable architecutre is the architecture which has the feature of evolvability. The evolvability is the ability to relax the key constraints that limit the extension of the architecture with the aim of providing better support for application requirements while adhering to the core design principles of the Internet.

As shown in Figure 1, the evolvable architecture can be divided into the following three basic layers:

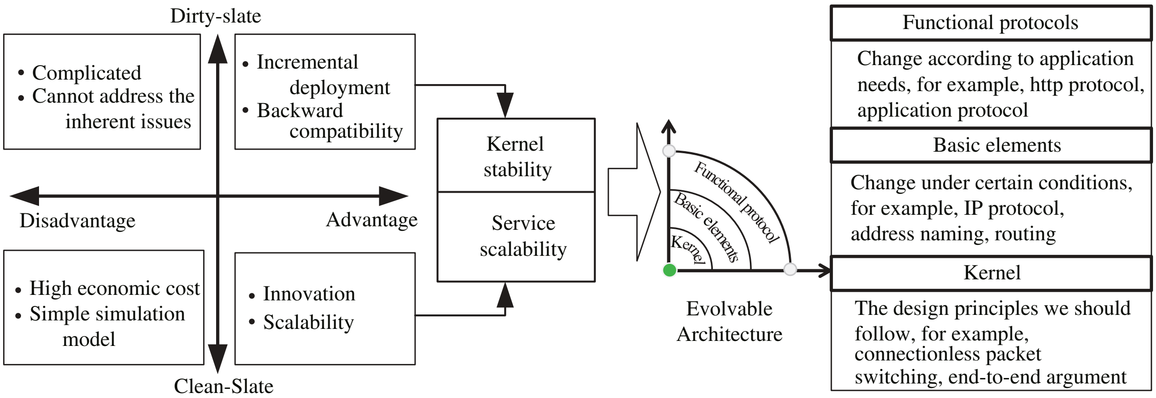

Figure 1 The framework of the evolvable architecture.

Kernel. The "kernel" refers to the core design principles such as the packet switching scheme of the traditional Internet architecture we should adhere to, which can ensure the stability of the architecture's development.

Basic element layer. The "basic element layer" contains some protocols and technologies that can be changed under some conditions, such as the IP protocol, address naming scheme, routing and transmission mechanisms.

Functional protocol layer. The "function protocol layer" includes some of application protocols on the top of the architecture protocol stack, for example, the Hypertext Transfer Protocol (HTTP). These protocols can be changed according to the application requirements without limitation.

As Figure 1 shows, the evolvable architecture we proposed combines the advantage of both the dirty-slate approach and the clean-slate approach. Compared with the dirty-slate approach, the evolvable architecture is more flexible. Moreover, the evolvable architecture is more stable compared with the clean-slate approach. It can be said that the evolvable architecture is a tradeoff between the clean-slate approach and the dirty-slate approach.

3.2 The basic design principle of evolvable architecture

Internet architecture is the channel that links the requirements and the technologies. For network technology, architectures play a guiding role for a longer period of time. As time goes on, the application requirements may change and develop, as well as the technologies. While once the architecture has been established, it will remain relatively stable. Therefore, the corresponding principles should be determined to ensure the sustainable and stable development of the evolvable architecture.

Kernel stability. Internet architecture has adapted to many new application requirements in the past 40 years' development. Especially over the past decade, facing the technology challenge posed by the cloud computing, P2P, social networking, and other new applications, the Internet architecture has met their demands well through its own adjustments. Reviewing the design principles [24] and the layering mechanism of the Internet architecture, we can find that they are not only the roots for the continuous development of the Internet, but also the successful experiences summarized from the long term and large-scale technology experiments. For example, the End-to-End argument, one of the most important concepts in the traditional architecture, not only promotes the innovation of kinds of protocols, but also reduces the complexity of data transmission. The connectionless packet switching protocol, another staple mechanism of the existing architecture, achieves the maximum of network resource reuse for the Internet. Therefore, although we need to rethink the existing architecture due to the application requirements, this does not mean that we have to completely abandon the original design principles. To protect the existing business and network infrastructure, the evolvable architecture should adhere to the core design principles and the protocol stack hierarchy relatively stable and not introduce new protocol hierarchies into the current layered architecture.

Service scalability. As we all know, the foremost reason of studies on the new generation Internet architecture is the mechanism defects of the current architecture both in function and performance. We have learned that it is not enough to improve these defects by means of patching. Kurose [25] claimed that the development of Internet architecture should emancipate the mind in order to overcome the current crisis. It is obvious that more innovative technology ideas should be introduced to meet the function and performance requirements of numerous applications. Existing studies show that the clean-slate approach can provide better service scalability since it gives full consideration to the possible future problems. Enlightened by the ideas of the clean-slate approach, we can relax some traditional constraint limitations to the basic element layer of the evolvable architecture shown in Figure 1 to achieve the service scalability.

Economic feasibility. Up to now, numerous protocols and mechanisms have been proposed to improve the existing architecture, but most of them have not been deployed finally although considerable resources have been consumed. These mechanisms were not applied to practice not because they cannot meet the applications' technical requirements, but because of their poor economic feasibility. If the cost for exploiting and deploying the technology is more than the utility obtained from it, it obviously cannot get sustainable development. The slow development of the clean-slate approach also lets us realize that the evolution of the Internet architecture depends on not only technological advances but also social and economic factors. Therefore, the construction of the evolvable architecture should ensure both performance and cost-awareness. Furthermore, the architecture can bring economic benefits for all the players on the Internet, and maintain the benefit balance between them.

3.3 The technology roadmap of evolvable architecture

In line with the definition of the evolvable architecture and the the corresponding design principles showed in Figure 1, we know that the evolvable architecture should be built on the existing architecture to keep the stability of evolution. At the same time, in order to change the status of the huge IP layer and corresponding challenges faced by the Internet, some innovations should be introduced into the traditional protocol stack. Under the support of "the Basic Research on Internet Architecture and its Protocols (973 Program)" and the guidance of the evolvable architecture design principles, we make thorough analysis to the existing architecture and build an evolvable architecture as shown in Figure 1.

Address. IP address is the basis of Internet communications, which determines the development of the architecture. After in-depth study on the issues in current architecture, we find that the naming and the use of "IP address" are the root cause of most Internet issues, such as scalability and security issue. Therefore, starting from the naming and the use of the "IP address", we make a general address platform in the traditional address layer.

The IP address performs the dual role of endpoint identifier and the network node's routing locator currently, which makes it difficult to support several new applications such as mobility and multi-homing. Furthermore, the current IP address faces serious scalability, aggregation and security issues. Different kinds of address representation strategies have been proposed in the research of future architectures, for example, CGA [26]. These strategies have their own advantages and it is difficult to say which one is better. We can construct a general address platform which supports all kinds of address representation. This is in conformity with the demands of long-term evolution since any address scheme can be deployed on it. In the general address platform, we summarize and extract some general properties of existing address systems, and formulate them into uniformed format. The details of the general address platform will be described in Section 5.

Routing. The routing system gets into three huge troubles in the traditional Internet architecture. Firstly, it runs high risks of being attacked due to the lack of authentication in the process of data transmission. Secondly, the increasing number of multi-homing prefix and un-aggregation prefix introduces continued growth in the routing table and the forwarding table, which brings in serious routing scalability issue. Thirdly, it is essential to exploit efficient transit scheme since multiple routing mechanisms may be deployed on the future Internet. Motivated by the above-mentioned consideration, a scalable, trusted and aggregation routing system has been proposed in our evolvable architecture. Some significant research results have been achieved in our "973" project team, including network monitoring [27], scalable and aggregation routing schemes [28–31], security routing schemes [32,33], and 4over6 transition

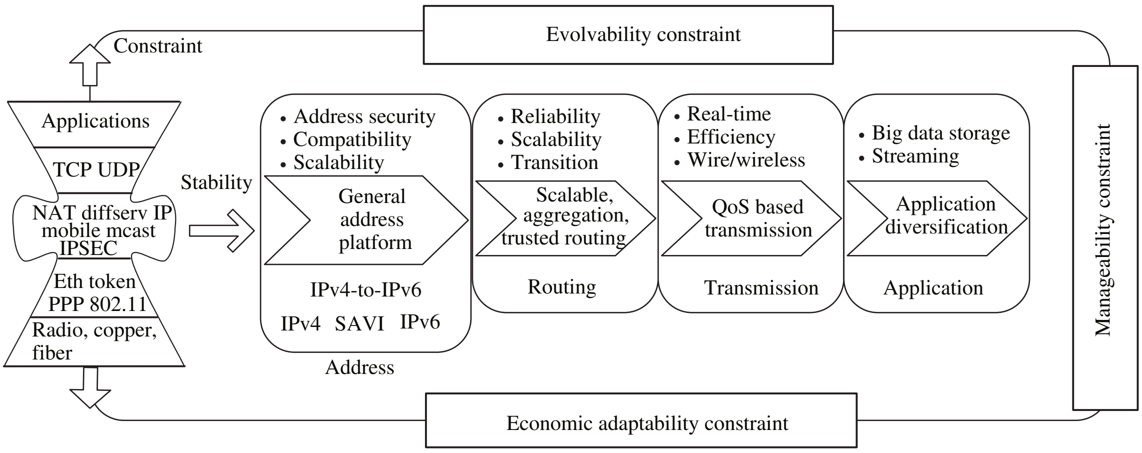

Figure 2 The technology roadmap realization of the evolvable architecture.

solution [34, 35].

Transmission. The Cisco Visual Networking Index (VNI) Forecast (2012–2017) report[9)] published in 2013 predicted that the sum of all kinds of video traffic would constitute 80% to 90% of global consumer Internet traffic by 2017. The change of data composition on the Internet has triggered reconsideration of the transmission model of Internet infrastructure due to the increase of QoS requirements. An experimental and analytical performance study over BitTorrent-like P2P networks for accelerating large-scale content distribution over Internet has been made in our project team [36]. In Ref. [37], a "streaming while downloading" service was proposed to enable BitTorrent support VoD service in existing swarms while maintaining the download efficiency of file-sharing users. What's more, as we mentioned in the Introduction, the mobile application has surpassed the static application with the improvement of wireless circumstance and the increasing number of access devices and access methods. Then, it is necessary for the evolvable architecture to transfer the data efficiently in both the wire and wireless circumstances. The performance of wireless opportunistic schedulers in multiuser systems has been studied under a dynamic data arrival setting in Ref. [38] by our project team.

Application. The Internet has effectively become a distributed repository of massive data and digital media content. How to store and manage these data is what application layer should resolve. A distributed cache infrastructure for P2P application has been proposed by Huang et al. [39] in the evolvable architecture. The traffic engineering issue, the traffic management and control method, and content distribution scheme can also get in-depth research at the application layer.

Design constraints. To ensure the construction of the evolvable architecture in compliance with the design principles, there are three constraints we need to insist during building the evolvable architecture. They are evolvability constraint, the economic adaptability constraint and management constraint as shown in Figure 2.

(1) Evolvability constraint: Evolvability is the foundation of the architecture we proposed, which represents features such as scalability, stability and economic feasibility. The evolvability constraint is responsible for evaluating the evolvability of the protocols, the technologies in the evolvable architecture and the architecture itself.

(2) Economic adaptability constraint: The economic constraint aims to capture and analyze the economic adaptability and features of the new network services such as the income of ISPs and users.

(3) Manageability constraint: The manageability constraint aims to solve the network management and control issues such as the identity authentication of users and user-flow control with high-performance.

The technology constraints are the theoretical basis of the evolvable architecture which can guarantee its sustainable development. They ensure that the operators and researchers in the Internet select the

9) http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360_ns827_Networking_Solutions_White_Paper.html.

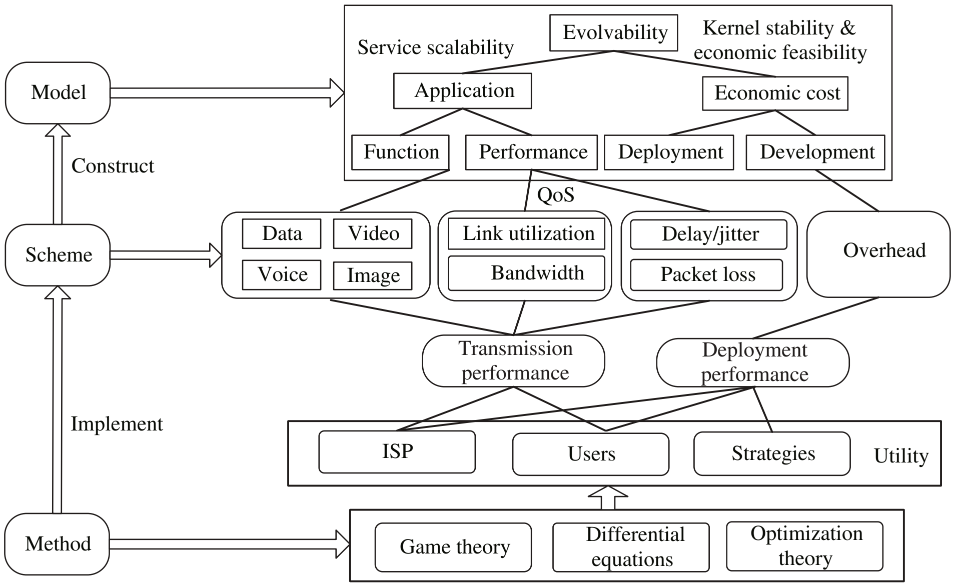

Figure 3 The evolvability evaluation framework of the Internet architecture.

most appropriate protocols or mechanisms to construct the architecture. We will introduce the corresponding constraint models in the next section in detail.

4 Design constraints and models analysis of the evolvable Internet architecture

In this section, we will introduce three important constraint models used in the development of the evolvable architecture, including evolvability framework and model, economic adaptability model and manageability model.

4.1 Evolvability framework and model

According to the definition and design principles of the evolvable architecture, we construct an evolvability evaluation framework and build two corresponding evaluation models from the application technology perspective and the user utility perspective to ensure the evolution of the architecture.

4.1.1 *Evolvability evaluation framework*

The model is the most effective method used to evaluate the complex Internet architecture. Although the means and strategies for the construction of the model may be different due to the requirements, the process to build the model is usually similar. In Ref. [40], we have summarized three issues involved in the construction of the evaluation model, including the evaluation model, the evaluation scheme and the evaluation method. The model describes the evaluation object. The scheme represents the criteria used in the evaluation. The method refers to the tools and theories used in the evaluation, such as game theory and differential equations. According to this, we construct the evolvability evaluation framework as shown in Figure 3.

Evaluation model. As we know, the design principles of the evolvable architecture include kernel stability, service scalability and economic feasibility. Since these principles have close relationship with the applications and the architecture's deployment, we determine the objects we should focus on in the evolvability evaluation as function and performance of applications and deployment economic cost of the architecture.

(1) Application function and performance. The applications deployed on the Internet, which provide great essential services for human beings, are one of the most important reasons for the popularity of

the Internet. The motivation of the architecture innovation and development comes from the needs for supporting higher application requirements and new applications. Therefore, it can be envisaged that a scalable architecture must have powerful ability to meet the application demand change both in function and performance aspects.

(2) Economic cost of deployment. No matter what distinct features it will realize, the future architecture should be built based on the current architecture for the sake of incremental migration. The slow development of IPv6 has witnessed that the success of a new technology relies on not only the superiority of the technology itself, but also some other objective factors, such as the profit of the ISPs, the development cost, the management and the maintenance overhead. We call them economic cost of deployment in this paper. For users, too high deployment cost is likely to increase the price of accessing appropriate services in the Internet market. For network equipment providers and network operators, the increasing cost would definitely affect their profits as well as their decision-making on the choice of infrastructure ultimately. As a result, these types of deployment costs would bring about important effect on the evolvability of the architecture.

Evaluation scheme. The Internet includes a variety of media entities, such as users, ISPs, and network providers. Although different entities may have different choices because of the difference in interests, we find that there is a uniform criterium that can efficiently combine most of the entities. That is the utility provided by the architecture, which includes application utility and deployment utility.

Since the basic function of the Internet is to deliver the application data or packets to destination nodes in the network, we reckon that the data forwarding capacity of the architecture directly reflects its utility to provide Internet-based services. It is easy to see that different data delivery techniques and the resulting routing protocols will incur different data transmission performance, and then provide different service utility in the architecture. To ISPs, it appears as network performance such as link utilization and throughput. To customers, it denotes service performance which mainly refers to the quality of service (QoS) assurance, with the key metrics of delay/jitter and packet loss.

The deployment utility is the cost used to support the applications. The reason is that there are some high-level technical and economic problems that need to be addressed before and after the applications are deployed on the architecture. Then if too much cost is needed to pay for the development of a new network equipment or the upgrading of existing network products in the architecture, we don't consider it as the evolvable architecture.

Evaluation method. The various performance parameters and utilities are criteria and theory used to construct the evaluation models. Besides, some other methods are needed to realize the function of the utility and the performance parameters. That is what the evaluation method should be. The integration of multi-disciplinary and multi-domain promotes the development of Internet. For example, the physics, biology genetics, statistics and other disciplines have been widely applied to computer science. With the aid of game theory, differential equations and optimization theory in these disciplines, the evolvability evaluation can be realized effectively.

Based on the framework we proposed in Figure 3, we can build various models from different angles, under different requirements, and based on different methods. For example, the following three models:

(1) Application adaptability model. The expansion and evolution of the Internet architecture is to better support the ever-increasing demand for new applications. Therefore, we can build the evolvability evaluation model from the perspective of applications utility according to their function, performance and overhead provided by the architecture.

(2) The user utility model. Whether one technology is more competitive than the others can be determined by the choice of users. Therefore, we can build evolvability evaluation model based on user utilities provided by the architecture.

(3) The ISPs utility model. ISPs is the operators and implementers of the architecture. Whether the architecture has incentive effects on the ISPs also determines which architecture will win in the competition. Therefore, it is a good choice to build the evaluation model from the utility of ISPs with the aid of game theory and other mathematical tools.

4.1.2 *Case study*

Not only the architecture, but also the technologies and protocols can be evaluated according to the evaluation framework we have proposed. Taking the evaluation of in-network caching mechanism and tri-network convergence for example, we talk about how to build the evolvability evaluation model according to the framework as shown in Figure 3.

In-network caching mechanism evaluation. The explosion of online video has triggered the reconsideration of the fundamental communication model for Internet infrastructure. Given limited network and server resources, how to distribute the heterogeneous data in a more efficient way has attracted more and more attention. The evolution of the hardware and the reduction at the cost of storage and CPU processing make in-network storage and data replication possible on a large scale. This provides the opportunity for in-network caching design, which allows the content to be cached anywhere on the delivery path. However, the advantages of the in-network caching mechanism have not been evaluated in a quantitative manner yet. Since the change of the applications demands is one of the most important reasons for the innovation of the architecture, we believe that the sustainable development of some architecture or mechanism such as in-network caching mechanism has relationship with its function and performance under different application scenarios and application data compositions, and whether it is less costly to support the application requirements. Therefore, we use application adaptability model to determine the relationship among the proportion of application data, the economic cost and the development of in-network caching mechanism based architecture.

We put forward the following holistic 2ACT model [41] based on the evolvability evaluation framework from the application perspective:

$$U = \frac{\alpha}{m}\sum_{i=1}^{m}\sum_{j=1}^{n} p_{ij}(Perf)_{ij} + \beta U_e,$$
$$\text{s.t.} \sum_{j=1}^{n} p_{ij} = 1, \alpha + \beta = 1. \tag{1}$$

The first part of the equation represents the service scalability of the protocol such as in-network caching mechanism, and the second part represents the economic cost to deploy the protocol. In this equation, the parameter m denotes the number of application classification dimensions while n denotes the number of application types in a particular classification dimension. The parameter p_{ij} represents the proportion of application data under classification i and application type j in the network. The parameter $(Perf)_{ij}$ indicates the application data transmission performance of the architecture. The parameters α and β are used to weight the relative impact of service scalability and economic cost on the development of in-network caching mechanism.

We make some analysis and computation under the experimental system PlanetLab[10)] and the Internet2 Abilene network[11)]. The results are as shown in Figure 4.

The results show that the in-network caching mechanism does not perform efficiently in any case. There are some distinct constraints and limitations for in-network caching mechanisms to pose real significant effects on data delivery across the Internet. For example, to better play its advantage, the in-network caching mechanism need satisfy the following conditions at the same time as Figure 4 shows: 1) The cacheable traffic proportion in the network should reach a lower bound of 22%. 2) The economic cost of deploying and managing the in-network-caching-based network is not higher than that of the regular network without data cache capacity. 3) The router-level hop count for cacheable data delivery must be less than half of the regular network level. 4) The value of data transfer performance in router hop should be less than 1.1 times of that in regular network. More details of the 2ACT model and the evaluation of in-network caching mechanism can be found in paper [41].

Tri-network convergence analysis. The convergence of triple play services including voice, video and data, which are mainly forwarded by the telecommunication network, the TV broadcasting network,

10) http://www.planet-lab.org/.
11) http://www.internet2.edu/network/.

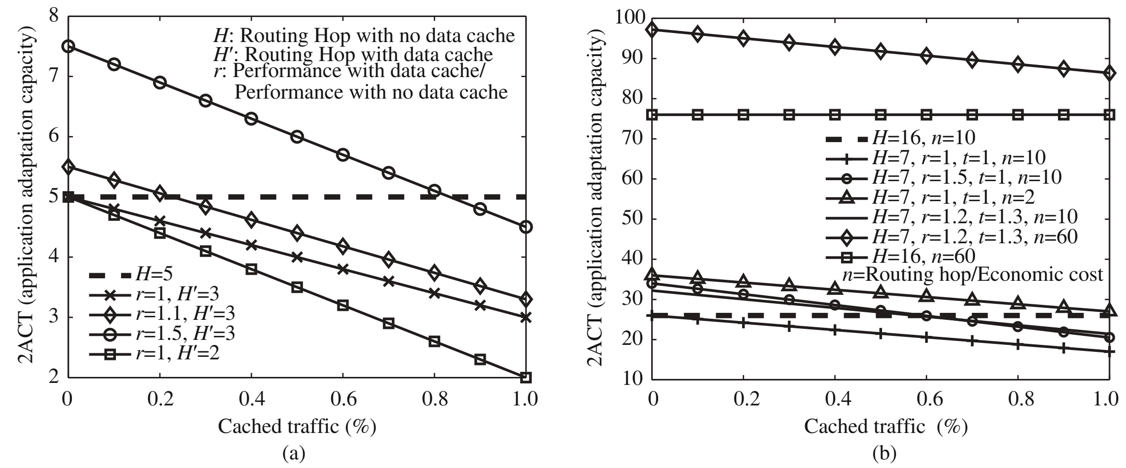

Figure 4 The results under different economic costs and application compositions.

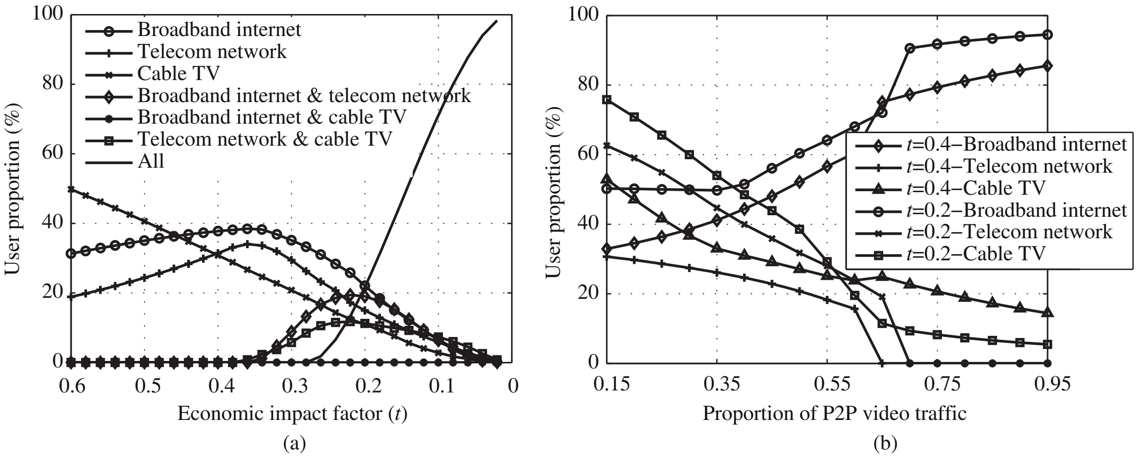

Figure 5 The results under different economic costs and application compositions.

and Internet respectively, has become an indisputable trend in the research of future architectures. This is called "Tri-Network Convergence". Several strategies have been proposed for tri-network convergence, but little has been known about the process and consequence of the convergence.

User is one of the biggest beneficiaries in tri-network convergence. It can be said that the tri-network convergence is the users and services game among Internet, telecommunications network, and cable TV network. In addition, user can connect all the elements that affect the convergence including ISPs, network technologies and application types. Therefore, the users network utility is an applicative method used to construct the tri-network convergence evolution model. We propose a user utility model based evolvable framework to examine the relationship between the evolvability of tri-network convergence and the economic cost, initial network market shares, application composition. The model is as follows:

$$U_k = \theta \sum_{j=1}^{4} \max_{1\leqslant i\leqslant 3} u_{ij} s_{ki} - t \sum_{i=1}^{3} s_{ki} C_i / L_i. \tag{2}$$

The first part of Eq. (2) represents the application values of the network provided to the users, and the second component of the equation denotes the costs that the users paid for the services. Part of the results are as shown in Figure 5. We can see from Figure 5(a) that the economic costs have significant impact on the convergence of the networks. With the decline of user economic level, the impact of economic cost on the user network utility will decline, and the competitiveness of the Internet and the telecommunication will increase more than cable TV because of their technical advantages. Furthermore, more and more

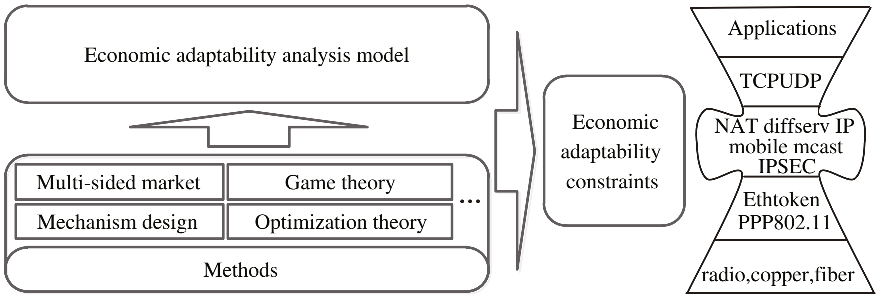

Figure 6 Methodology.

Table 1 Network players

Player	Description	Examples
Internet Service Provider (ISP)	Provide basic Internet services	China Telecom, AT&T
Network Developer (ND)	Develop network services	Companies such as Cisco, Research institutions such as Tsinghua University
Network Application Provider (NAP)	Develop and operate network applications	Tencent, Akamai
User (UR)	Consume network services and applications	End users, Youku
Government (GM)	Make policies	Ministry of Industry and Information Technology of P.R.C.

users may choose three networks simultaneously once the impact of economic cost impact declines to a certain level (when the economic impact factor is lower than 0.3, as Figure 5(a) shows). The application composition also has a significant impact on the evolution results, as Figure 5(b) shows. With the increase of the proportion of P2P data traffic in the network, advantages of the Internet have become more and more obvious.

4.2 Economic adaptability constraint model

To capture and analyze the economic constraints on the development, deployment and operation of new network services, we build an economic model to provide a framework to analyze the economic adaptability. The methodology is shown in Figure 6.

4.2.1 *Economic adaptability analysis model*

Economic issues are more and more widely considered in the design of networks. For example, ChoiceNet[12), one of the five NSF Future Internet Architecture (FIA) projects, takes choice as a major design principle by providing alternatives in each network level. Therefore, users can vote with financial encouragement for the alternatives to promote the development of superior services [42]. Players, targets and strategy sets, interactions and information are four key elements in the economic adaptability analysis model, which we will describe in detail, respectively.

Network players. We define the roles, the corresponding descriptions and examples of the network players in Table 1. Note that some types of players may overlap with each other. For example, the Internet Service Providers (ISPs) can act as Network Developers (NDs) to develop network services, while the Network Application Providers (NAPs) also act as Users (URs) to consume the Internet access service to support the operation of the applications. In addition, the Government (GM) always affect the player interactions by making policies, instead of directly participating in the interactions.

Targets and strategy sets. In economic models, targets and strategies are major properties of players. The targets and strategy sets of the players in this model are partly listed in Figure 7.

Based on targets, the players can be classified into three types. The first type includes ISPs, NDs and NAPs, who are generally chasing profits. However, they chase profits through different goals, such as

12) http://www.nets-fia.net/.

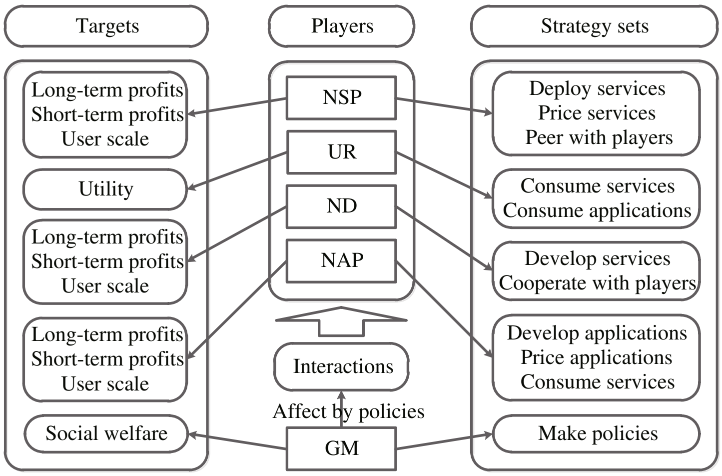

Figure 7 Targets and strategy sets of the players.

long-term profits, short-term profits and user scales. The second type includes end users, who try to maximize their own utilities. The third type includes governments, and their goals are to maximize social welfare, guarantee fairness, etc.

To achieve their targets, the players will dynamically adapt their own strategies based on that of other players. Now we discuss the players' strategies separately.

ISPs own the infrastructures and operate the Internet access services, so they can decide wether to deploy a new service in their networks. After deploying new services, they can set specific strategies to price users and NAPs for their consumption. To gain more revenue or reduce costs, they can cooperate with NDs or NAPs to take advantage of the integration of resources.

NDs can develop new network services, so as to support new kinds of applications, or improve the performance of existed services. The fund for the development of new services may come from both the NDs themselves and other players. To get fund from other players, the NDs should publish a new service proposal to attract their attentions. Even if the NDs themselves pay for the development, they need to contact the ISPs in advance to make sure that the ISPs would like to deploy the services. Therefore, the emergence of new services depends on all the players' choices.

NAPs develop network applications to attract users for profits or other targets. Similar to the ISPs, NAPs also set pricing strategies for the consumption of their applications. They can provide suggestions for NDs based on application-oriented demands. They can also cooperate with NDs or ISPs to reduce costs or attract more users.

URs consume both network services and applications. They dynamically adapt their consumption choices according to the pricing strategies set by the ISPs and the NAPs. They also play an important role in the acceptance of new services, because the goals of both development and deployment of the services are to give consumers a big incentive to spend or invest their money.

GMs set policies, which are the rules for the interactions of the players. The policies should guarantee fairness and the maximization of social welfare.

Since the players' choices of strategies are affected by that of all the other players, the interactions among the players are often modeled and analyzed based on game theory, which we will discuss in detail later.

Interactions. To derive the economic adaptability of new network services, game theory is used to model and analyze the interactions among network players based on their targets and strategy sets.

Different types of games should be used to model different scenarios. For example, in a monopoly market, a leader-follower game can model the pricing and consuming interaction between the ISP and users; while in an oligopoly market, a repeated game can be used to describe the competition interactions among multiple ISPs.

Information. Information is a key factor that affects the players' interactions and one of economic

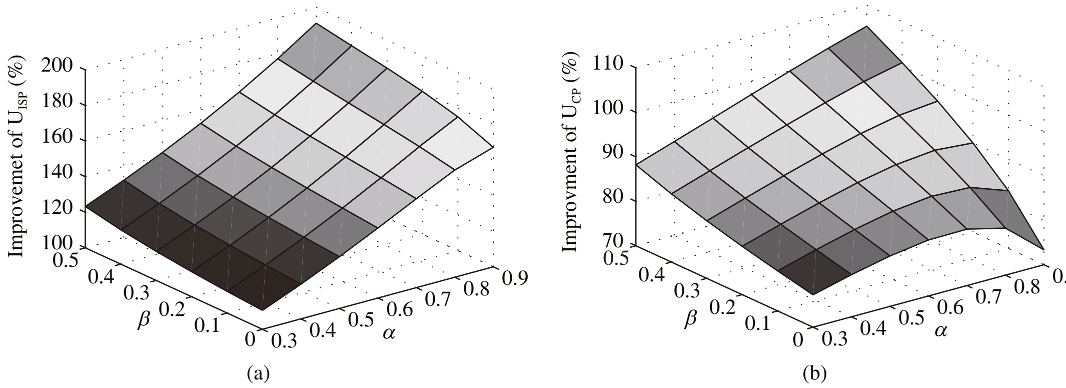

Figure 8 The Nash bargaining solution between the ISP and the CP [44].

goals is to solve the problem of information asymmetry. In game theory, information is also called knowledge.

In the evolution of network architectures, information should be fully used to promote the upgrade of existing services and the deployment of new services. For example, before NDs develop new services, they will publish the proposals to get fund or investigate the market. At the same time, NAPs are informed of the deployment plan of new services and they can start the development of corresponding applications in advance so as to shorten the waiting period for new services or new applications.

Another important type of information is the performance of existing services. With the development of infrastructures applications and user behaviors, the performance of existing services may change. Then regular evaluations are needed to find existing problems in time so as to promote the improvement of services.

4.2.2 *Case study*

The current Internet can also be modeled by the proposed economic adaptability model. For example, both Content Providers (CPs) and Cloud Service Providers (CSPs) can be seen as NAPs because CPs provide content viewing applications and CSPs provide cloud-based applications. Now we analyze several cases under the proposed framework.

The features and technical details of new services directly affect the feasible pricing strategies, which are of vital importance to the economic adaptability and the interactions among players. He et al. [43] surveyed some basic pricing models, pricing mechanisms in different network marketing environments.

New network services or applications may change the interactions among players. For example, Peer-to-Peer (P2P) applications have dramatically changed the traffic pattern, so the profit distribution in the Internet. The difference among terminals also generates different user-behaviors in mobile communication networks. Therefore, the economic issues in these specific scenarios should be considered. Xu et al. [44] introduced and categorized classical models in the Internet resource pricing and specifically present the pricing problems in P2P and mobile communication networks. To solve the unbalanced profit distribution in the P2P network, they propose a cooperative game based mechanism to distribute profit between ISP coalition and CP coalition, and a feasible mechanism to distribute profits within coalitions. They also analyzed the interactions among ISPs and users in the mobile communication network, and provided suggestions to the ISPs on their pricing strategies, so as to increase their profits. The improvements of the ISP's and the CP's profits compared with that on the starting point for different α and β are shown in Figure 8. We can see that cooperation increases the profit of both the ISP and the CP. Bargaining in the exclusive monopoly market and the dynamic game procedure in the oligopoly market have been analyzed by the non-cooperative game theory. They also provided suggestions for ISPs in the price-making process by the confirmation of ISP price monopoly position in the exclusive monopoly market through experiments.

The interactions among players in the proposed model are quite flexible. When traditional interactions rules cannot meet the new relationships, new ones should be introduced. Shi et al. [45] brought up an electronic auction platform for cloud, and a cloud Continuous Double Auction (CDA) mechanism is formulated to match orders and facilitate trading based on the platform. In competitive cloud computing markets, pricing policies are critical to market efficiency. While CSPs often publish their prices and charge users according to the amount of resources they consume, auction mechanism is rarely applied. In fact a feasible auction mechanism is the most effective method for allocation of resources, especially double auction is more efficient and flexible for it enables buyers and sellers to enter bids and offers simultaneously. Furthermore, they develop a novel bidding strategy for cloud CDA, BH-strategy, which plays a very important role for each player to maximize its own profit.

4.3 Manageability constraint model

In the network management aspect, since the IP source address can be spoofed and packets have no relation with their senders' identities, the future Internet faces the challenges such as authenticated IP source address, reliable user identification, and trusted outbound packets. These reasons seriously limited the evolution of the future Internet. In this section, we will give the model analysis at first, then present a detailed case study.

4.3.1 *Model analysis*

The most important design principle of the current Internet is "complexity in the center, but simplicity in the edge", which made the network resources are basically deployed in end-hosts but network devices such as switches, routers just do their best to deliver these packets to their next hop. However, the current Internet lacks of effective network management approaches in the aspects of IP address, user identity and dataflow control, which directly involved with a serials of issues such as IP source address spoofing, cyber-attack, and incorrect traceback. In order to solve these issues in the future Internet, we takes the methods of authenticated IP source address, reliable user identity and controllable dataflow to satisfy these requirements of network management and network programming. In the first place, we give the model analysis.

Assuming in a domain network named D, we use the sets $UID_D = \{uid^i | uid^i \in D\}, IP_D = \{ip^i | ip^i \in D\}, MAC_D = \{mac^i | mac^i \in D\}, Port_D = \{port^i | ip^i \in D\}$ to represent the collection of user identity, IP address, host MAC address and uplink port of layer2 switch for hosts accessing, respectively. In order to make sure every host's MAC address in its broadcast domain is unique, every element in the united collection $MAC\_Port_D = \{(mac^i, port^j) | mac^i \in MAC_D, port^j \in Port_D\}$ should be unique, or we say the two sets of MAC_D and $Port_D$ have the relationship of 1:1 reflection. Similarly, the united sets of $IP\_MAC_D = \{(ip^i, mac^j) | ip^i \in IP_D, mac^j \in MAC_D\}, UID\_IP_D = \{(uid^i, ip^j) | uid^i \in UID_D, ip^j \in IP_D\}$ express the relationship between the IP address and the host collections, the relationship between the user identification and the IP address sets, respectively.

IP address reliability. IP address reliability stands for packets sent out from any host should bring with the IP source address which belongs to the owner host, which is formulated as $\forall packet^i \in Packet_D, \exists (ip^{src} \leftarrow packet^i) \in IP_{(mac^j)}$ where $mac^j = f(ip^{src}, IP_{MAC_D})$.

User identification reliability. The user identification reliability refers to each packet should be sent out from the user whose identity is matched with what he(she) claimed, that is, $\forall packet^i \in Packet_D, \exists (ip^{src} \leftarrow packet^i) \in IP_{(uid^j)}$ where $uid^j = f(ip^{src}, UID_{IP_D})$.

User-flow controllability. In order to achieve the goal of user-flow controllability, every flow should own corresponding policies to control, namely, each element in the set $POLICY\_UID\_IP_D = \{(policy^i, uid^j, ip^k) | policy^i \in Policy_D, uid^j \in UID_D, ip^k \in IP_D^k\}$ should be unique.

4.3.2 *Case study*

Based on the analytical formulation in the previous section, in order to gain system reliability, we need those network identities to map with each other. Since the Source Address Validation Improvement (SAVI) [46] switch can achieve the IP source address reliability, naturally, we deploy them into the access layer of the network. Further, we need network users to authenticate them so that the system can combine

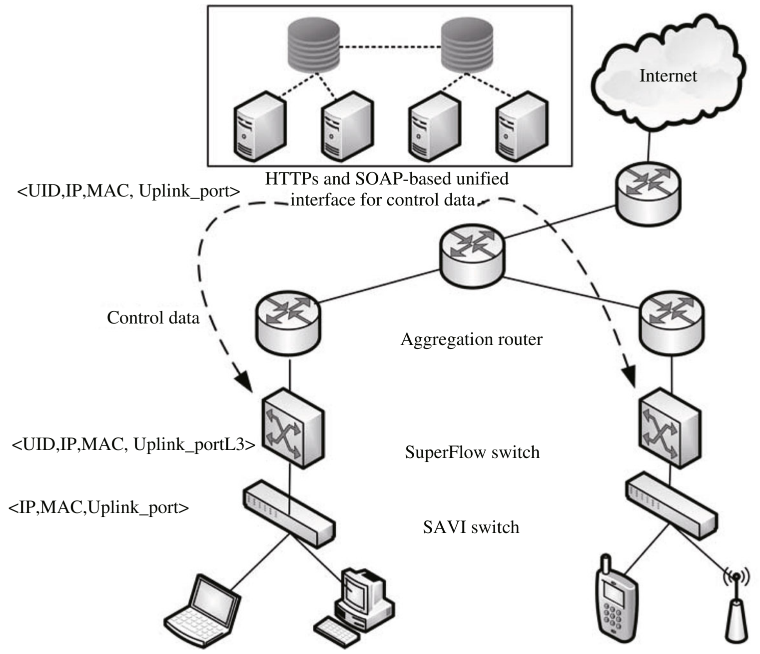

Figure 9 The multilayer-based identity mapping architecture.

IP binding entries from the SAVI switches with user identities. Therefore, the system can form the multiple identities mapping relationship from the aspect of global view, which also satisfies the requirements of user reliability and system reliability we mentioned. On the basis of system reliability, we need a layer-3 access switch to control user's data-flows to realize our goal of user-flow controllability. We named after this kind of switch SuperFlow switch. Certainly, SuperFlow switches have to accept the instructions from the central controller(s) to know how to deal with outbound flows. Eventually, we build the whole architecture named SuperFlow, which is illustrated in Figure 9. In the server section of it, we have more than one controllers to response to the numerous flow-control requests from the SuperFlow switches. The scalability in these controllers has been considered through a load-balancing designing. At the same time, controllers also possess an HTTPs and SOAP-based (Simple Object Access Protocol) unified interface to provide prompt and secure response to requests. In addition, databases are established to store the system's vital data, like policies, rules, user account, etc. There is not any change in the rest of parts, such as routers, hosts and firewall, which facilitates network assets retained in the greatest degree.

Authenticated IP address. In the aspect of the IP source address validation, the famous Source Address Validation Architecture (SAVA) [47] can provide transparent network service to ensure that every packet must hold an authenticated source IP address. It consists of three levels, the Inter-AS, Intra-AS and first-hop. In each level of this hierarchical architecture, it can get different granularities of the authenticity, and one or more mechanisms [46, 48, 49] are defined to solve this problem. In the user access subnet, SAVI proposal was approved by IETF to resolve this issue. Following the SAVI specification, a normal layer2.5 switch is named after SAVI switch, which can filter spoofing packets by establishing the triangle relationship of IP address, MAC address and uplink-port for each host. As to the binding relationship establishment and packet anti-spoofing fulfillment, it is accomplished by the IP address assignment protocols sniffing and the CPS protocol (Control Packet Snooping). Compared with the matured solution of unicast Reverse Path Forwarding (uRPF)[13], SAVI is more accurate because its effected point is the users' access switch rather than the access router. Besides, although it is very common and practice, the access control lists (ACLs) has been proved to be a double-edged sword because it is error-prone and expertise required. Compared with our normal layer2 switch, SAVI switch has two extra data-tables, the first one is Binding Status Table (BST) which can snoop its access hosts' IP requests (DHCPv6, SLAAC, DHCP-SLAAC-mix) and binding their IP address, MAC address with switch's

13) http://www.cisco.com/web/about/security/intelligence/unicast-rpf.html.

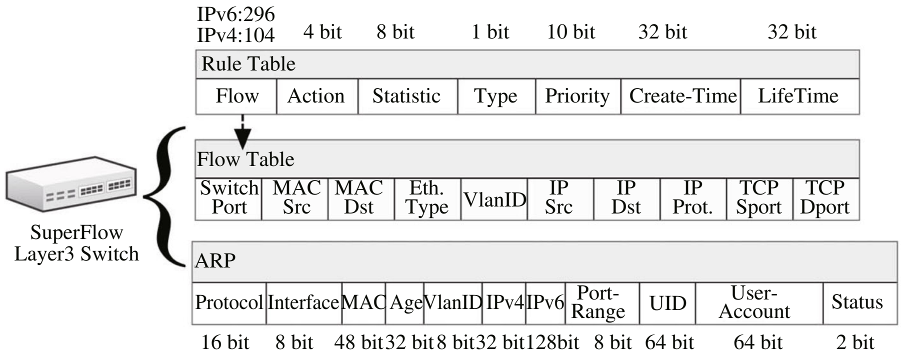

Figure 10 The diagram of the ARP Table, Flow Table and Rule Table in SuperFlow switch.

uplink-port together; yet another one is the Filtering Table (FT) which can filter the packets that their IP source addresses do not exist in BST. If we could fully deploy SAVI switch into our domain network, then eventually, we can achieve the goal of IP address authentication with host granularity.

Reliable user identity. Certainly, SuperFlow switch needs to build the corresponding data structure to save this mapping relationships. As Figure 10 illustrates, on the basis of ARP table in a normal layer3 switch, SuperFlow switch adds some extra fields, such as IPv4/IPv6 address, upper application port range (Port-Range, for adapting the situation of multiple hosts sharing one IP address), hashed user identification (UID), user account, current status (Status, binding status like request, binded), etc. Besides, to achieve the goal of user-flow controllability, SuperFlow switch adds extra two tables, the Flow Table and Rule Table, one for matching the user dataflows, while another is for storing corresponding process rules. Taking the ARP table as an example for storage analysis, one record of this table needs 410 bits storage space, therefore, the total storage cost of this table is $S(ARP - Table) = N * 410(b)$, where the N stands for the number of records. Imagine that if there are 1000 records, then the required storage space is 410000b= 51.25 KB, which means its storage cost is minor.

Globally identity mapping. As stated in the previous section, we use the SAVI switch to bind the host's IP address, MAC address and switch's uplink port with the distributed way to get the authentication of IP source address. Meanwhile, in the layer3 of the TCP/IP stack, we take the SuperFlow switch to store the relationship of user identity, IP source address and MAC address in order to keep the realiablity of user identification. While in the application layer, we use a database to collect all these binding data together via the SNMP protocol, and eventually the system forms the globally identity mapping relationship which directly faciliates the user-flow control, network control, etc.

5 General address platform

The identification scheme including naming and addressing is the most essential and important part of a network because the basic functionalities of a network tightly depend on it [50]. The Internet address system directly determines the sustainable evolutionary ability of Internet architecture to a large extent. For example, the IP address in the current architecture has dual semantic functions (indicating both the network node's routing locator and its endpoint identifier) which hindered the Internet from supporting routing scalability, mobility, multi-homing and security. The semantic overload of IP address, almost by design, lacks of consideration of long-term evolution support. Thus, we argue that it is significant to find the architecture evolvability begin with a general address platform which is able to support multiple address schemes natively.

Our goal is not to propose a new Internet address scheme; instead, we aim to construct an address platform which makes evolution possible. Multiple address system can be deployed on the platform with specific experiment and evaluation. In fact, the platform which can serve as the foot-stone for an evolvable Internet has the intrinsic capability to evolve; other parts of the network will develop naturally that will be illustrated by two examples at the end of this section.

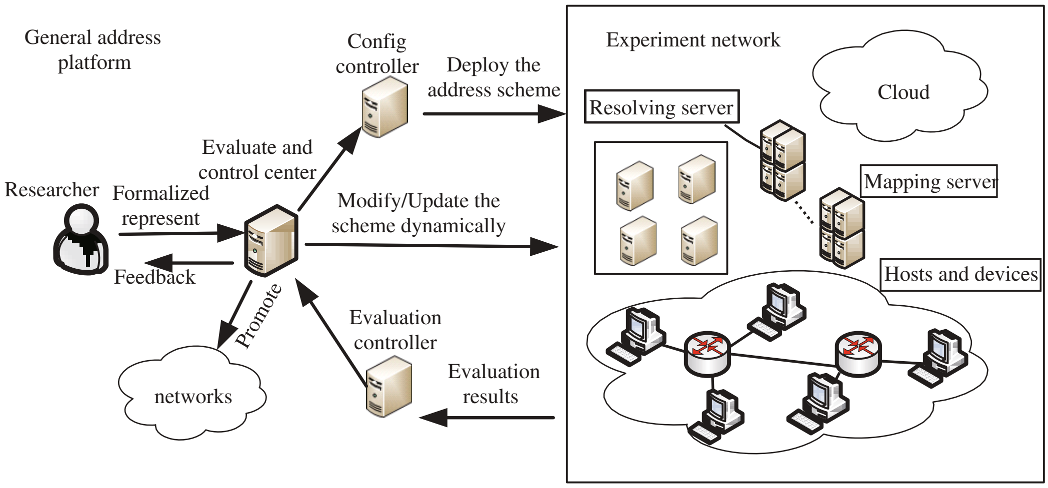

Figure 11 The composition and operation of the platform.

5.1 Introduction of the general address platform

The general address platform should be able to support multiple address schemes in the existing network, and evolve to support as-yet-unforeseen types as demand changes in addition. In order to ensure such flexibility and scalability, we plan to use formal description to represent an Internet address with various properties.

As shown in Figure 11, the platform mainly consists of the following resources. Evaluate and Control Center (ECC) is in charge of operation and management for the whole platform; Configure Controller (CC) translates the formalized description which submitted by researchers into a deployable address scheme; CC is responsible for collecting information from the experiment network in order to form a real evaluation outcome which would be reported to ECC. Our experiment network provides the physical resources for actual operation. From the perspective of experiment, the network can be only composed of hosts and forwarding devices, servers (such as database server), cloud (provide computing and storage capacity) and mapping system (for some address schemes such as ID/LOC split). Next, we will explain the whole running process of the platform briefly.

Researches who intend to evaluate a new address scheme formalize the scheme according to a given specification which would be submitted to ECC. ECC can help check the correctness of the description. Then, the address scheme can be deploy into the experiment network via CC. EC collects the information of measurement and report it to ECC. After receiving the information, ECC evaluates the address scheme based on our evaluation methods and inform researches the final results. What ensures scalability of the platform is that researches can change or update the test scheme only by revising the description dynamically. Moreover, if the address scheme proves beneficial, ECC can help promote it into the real network.

5.2 Multi-dimensional formal description

To break though limitations of the current address system, many new address schemes have been proposed, such as Cryptographically Generated Addresses (CGA) [26], ID/LOC split schemes (HIP [9]) and self-certification addresses (AIP [8]). These new address types change the semantics and attributes of the current address essentially. In order to support multiple even future address types, it is necessary to represent an address by using a general description which can cover core address attributes. For dynamic behavior in address system such as assignment, mapping and encapsulation, we also need to extract common primitives in the procedure and define proper interface to support above interactions to form a complete formal description of address system. Moreover, we should translate the network resources (such as hosts and servers) into a script file. In this case, we only need modify the script file to manage resources while not paying attention to the specific devices. It can be illustrated by Figure 12.

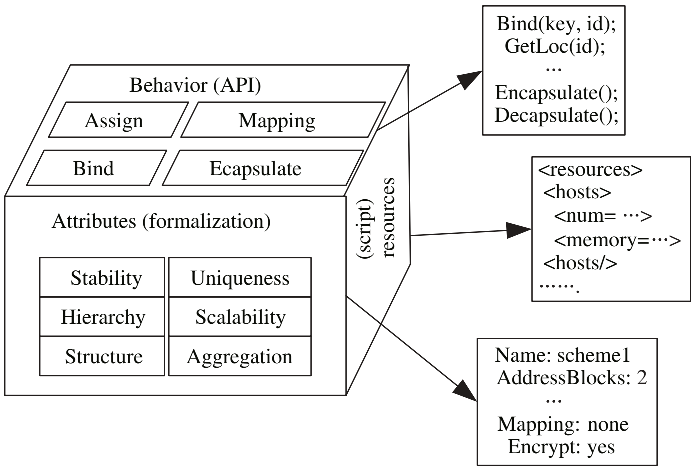

Figure 12 Multi-dimensional formal description.

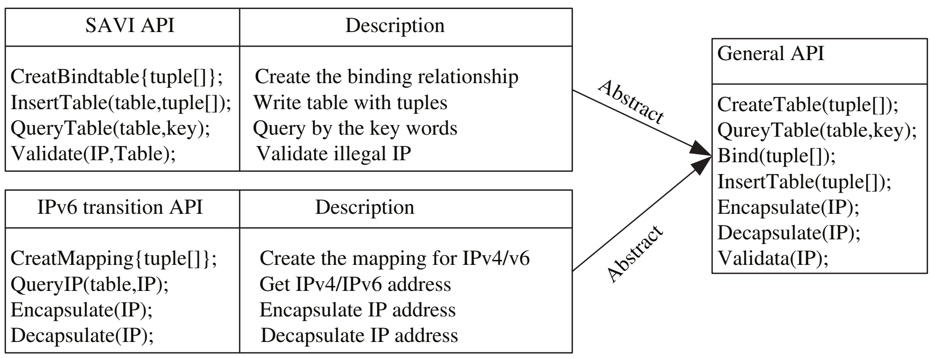

Figure 13 Abstract general API.

To combine the above three aspects, a complete, script-driven general address platform can be accomplished. Our multi-dimensional formal description provide a common abstract interpretation for different network address systems so that researchers can participate and customize address models they desired with powerful scalability.

5.3 Dynamical support for multiple address schemes

The general address platform is able to evolve to support various address schemes with high flexibility. In this section, it can be illustrated by the following two examples: SAVI (we have discussed above) and IPv6 transition.

SAVI [47] were developed to prevent nodes attached to the same IP link from spoofing each other's IP addresses, so as to complement ingress filtering with finer-grained, standardized IP source address validation. Its core idea is to bind IP address and network entities on IP layer, and validate when forwarding. For example, SAVI snoop the address assignment protocol to form a binding tuple $\langle IP\ address, MAC\ address, switch's\ port\rangle$.

IPv6 comprises multiple advanced network technologies, but it takes long to replace IPv4. Many IPv6 transition mechanisms which facilitate the transition of the Internet from its initial (and current) IPv4 infrastructure to the successor addressing and routing system of Internet Protocol Version 6 (IPv6) were proposed. As IPv4 and IPv6 networks are not directly interoperable, these technologies are designed to permit hosts on either network to participate in networking with the opposing network. From a technical perspective, it mainly consists of three types: tunnel, translation and dual-stack.

The interaction of the above two mechanisms is diverse from each other. However, we can abstract the general operation on the basis of multi-dimensional formal description we discussed. Figure 13 shows the basic API for the two mechanisms.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
        <xs:simpleType name="IPv6_Address">
                <xs:restriction base="xs:string">
                        <xs:pattern value="([A-Fa-f0-9]{1,4}:){7}[A-Fa-f0-9]{1,4}"/>
                </xs:restriction>
        </xs:simpleType>
        <xs:element name="address_structure" type="hierarchical"/>
        <xs:element name="address_blocks" type="xs:integer" fixed="8"/>
        <xs:element name="address_prefix" type="xs:integer"/>
        <xs:element name="separator" type="xs:string" fixed=":"/>
        <xs:element name="base_address" type="xs:hexBianry"/>
        <xs:element name="construct_method" type="append"/>
</xs:schema>
```

Figure 14 XML description of an IPv6 address.

Whether $CreateBindtable()$ in SAVI API or $CreateMapping()$ in transition API can be unified to $CreateTable()$ due to the same operation and purpose. Other APIs are similar to this.

When we have these general APIs, researches only need to modify the formal description for using a new mechanism which they desired. For example, people who want to add SAVI or tunnel transition mechanism into the platform would insert scripts into the formal description as follows, $\langle xs : securitymechanism = \text{“}SAVI\text{”}/\rangle$ and $\langle xs : transitionmethod = \text{“}tunnel\text{”}/\rangle$. The configuration controller will resolve the two scripts and call the corresponding APIs. Then, SAVI and tunnel mechanisms can be added into the platform flexibly. However, abstracting the general APIs for so many address schemes is a major challenge that we plan to explore an effective method to achieve that in future work.

5.4 A concrete example

In this section, We will explain how the general platform could support the deployment and experiment of multiple even future address types with a concrete example. In addition to support for existing address schemes, our address platform takes into account the support for the evolution of Internet address. Indeed, we cant forsee any address types in the future, but the platform provides maximum flexibility and possiblity to support a new scheme we cant predict today.

To provide a general way to construct diverse addresses, we need to summarize some core static address attributes. In other words, we can structure different address only based on the values of these properties. Thus, we abstract seven core attributes which can be used to explain structural characteristics of most address types. These core attributes are as follows:

address_name. Name of an address scheme, such as “IPv6”,“HIP”,“AIP”.

base_address. Base type of an address, its value is “binary” for IPv4 or “hexbinary” for IPv6.

address_structure. Address structurespecify which structure the address belongs to (flat or hierarchical).

address_blocks. Number of an address, its value is “four for IPv4 or “eight” for IPv6.

address_prefix. Length of network prefix.

separator. Separator between addressblocks, its value is “.” for IPv4 or “:” for IPv6.

construct_method. A method to construct an address. For HIP, its value is “hash”; For IPv4 and IPv6, its value is “append”.

Based on these core attributes, we give an example of using xml to describe an IPv6 address in Figure 14. Because the analytic functions for these attributes was built into the platform, users only need to give the values of attributes, then the platform will resolve these attribute values to construct the corresponding address.

Due to the great scalability of XML, users can extend the description to meet the actual demand of multiple even future address systems. Moreover, we realize that the construct method of an address type is evolving constantly. For instance, an IPv6 address is composed of four address blocks, an HIP address is derived from hash of public key and some new construct methods may arise in the future. Considering

Table 2 $\langle method - function \rangle$ mapping table

Construct method	Function
MD5_HASH	String MD5(string BaseAddress)...
SHA1_HASH	String SHA1(string BaseAddress)...
SHA1_HASH	String SHA1(string BaseAddress)...
DES_ENCRYPT	String DES(string BaseAddress)...
...	...

the diversity of future address construct method, a $\langle method - function \rangle$ mapping talbe which users can extend is built into our platform. It can be illustrated by Table 2.

If a future address type is constructed with a new method, users only need to bind the relationship between method and a corresponding function into the mapping table. When this new method is selected in the XML description, the platform will match the mapping table to call its concrete function. We argue that the xml description and the built-in mapping table provide great flexibility to support various even future address types.

We believe that it is essential to construct the general address platform which enables us to have a deep insight into the evolvability of Internet architecture. The general address platform which can support various even future address system facilitate us to put the evolvable principle into practice, and provide a solid foundation for further evolvability exploration of Internet architecture.

6 Conclusion and future work

With the fast growth of Internet and the challenges it faces, the problem of how to develop Internet architecture has attracted more and more attention in both academia and industry. In this paper, we summarize the challenges the Internet is facing, and analyze the features of two mainstream architecture development ideas, including the dirty-slate approach and the clean-slate approach. We find that bringing forth new ideas under the limitation of not changing the core principles of the existing architecture is more conducive to construct a scalable, stable and sustainable Internet infrastructure. We have put forward an evolvable architecture based on this idea, under the support of National Basic Research Program of China (973). The conception, the design principle, the corresponding technology roadmap, and the constructing constraints have got in-depth description in this paper. Beyond this, we construct a general address platform which serves as the foot-stone for an evolvable Internet architecture. We believe this platform can provide a more open and efficient network environment for the research and development of the evolvable architecture.

The basic idea of the architecture we proposed follows the core design principles of the Internet. It relaxes the constraints that limit the development of existing architecture. Therefore, the architecture we proposed is application oriented, and it can be compatible with the existing architecture and deployed incrementally. In fact, some of the protocols and mechanisms in the architecture have already been used in the practice, such as the source address validation and the scalable and aggregation routing schemes. And some of the mechanisms are still under development and deployment, for example, the address platform. We also take CERNET2, the next generation Internet testbed based on IPv6, as our experimental platform to demonstrate the theoretical idea that we have proposed.

Except for further improving the existing research work, our future work will mainly focus on the improvement of the general address platform and the implement of economic adaptability model. For general address platform, it is the solid foundation of the evolvable Internet architecture. We plan to complete the formal description and general APIs which can represent multiple address schemes in future work. For economic adaptability model, new interactive platforms and rules should be built to provide better incentives for the innovation and deployment of new network services. For example, we will evaluate the dominant strategies of the players in the Internet when multi-sided market model is applied. And the cooperations among players should be studied to redefine the ecosystem in the Internet.

Acknowledgements

This research was supported by National Basic Research Program of China (973 Program) (Grant Nos. 2009CB320501, 2012CB315803), New Generation Broadband Wireless Mobile Communication Network of the National Science and Technology Major Projects (Grant No. 2012ZX03005001), National Natural Science Foundation of China (Grant Nos. 61170292, 61161140454), National High-tech R&D Program of China (863 Program) (Grant No. 2013AA013302), National Science and Technology Support Program (Grant No. 2011BAK08B05-02), and EU MARIE CURIE ACTIONS EVANS (Grant No. PIRSES-GA-2010-269323).

References

1 Tanenbaum A S, Wetherall D J. Computer Networks, 5th ed. New Jersey: Prentice-Hall, 2010
2 Cerf V G, Kahn R E. A protocol for packet network interconnection. IEEE Trans Comm, 1974, COM-22: 627–641
3 Rexford J, Dovrolis C. Future internet architecture: Clean-slate versus evolutionary research. CACM, 2010, 53: 36–40
4 Pan J, Paul S, Jain R. A survey of the research on future internet Architectures. IEEE Comm Mag, 2011, 49: 26–36
5 Koponen T, Shenker S, Balakrishnan H, et al. Architecting for innovation. ACM SIGCOMM CCR, 2011, 41: 24–36
6 Anand A, Dogar F, Han D, et al. XIA: An architecture for an evolvable and trustworthy Internet. In: Proc HotNets, Cambridge, Massachusetts, USA. New York: ACM Press, 2011. No.(2)
7 Ghodsi A, Shenker S, Koponen T, et al. Intelligent design enables architectural evolution. In: Proc HotNets, Cambridge, Massachusetts, USA. New York: ACM Press, 2011. No.(3)
8 Andersen D G, Balakrishnan H, Feamster N, et al. Accountable internet protocol (AIP). ACM SIGCOMM CCR, 2008, 38: 339–350
9 Moskowitz R, Nikander P, Jokela P, et al. Host identity protocol. RFC 5201, 2008
10 Jacobson V, Smetters D K, Thornton J D, et al. Networking named content. In: Proc CoNEXT, Rome, Italy. New York: ACM Press, 2009. 1–12
11 Zhang L X, Estrin D, Burke J, et al. Named data networking (NDN) project. PARC Technical Report NDN-0001, Palo Alto, California, 2010
12 Koponen T, Chawla M, Chun B G, et al. A data-oriented (and Beyond) network architecture. ACM SIGCOMM CCR, 2007, 37: 181–192
13 Balakrishnan H, Lakshminarayanan K, Ratnasamy S, et al. A layered naming architecture for the internet. In: Proc ACM SIGCOMM, Portland, Oregon. New York: ACM Press, 2004. 343–352
14 Jung H Y, Koh S J. Mobile-oriented future internet (MOFI): Architecture and protocols (Release 1.2). Electronics and Telecommunications Research Institute (ETRI), 2010
15 Shin M K, Nam K H, Kang M, et al. Formal Specification Framework for Software-Defined Networks (SDN). IETF draft-shin-sdn-formal-specification-03, 2013
16 Blumenthal M S, Clark D D. Rethinking the design of the Internet: The end-to-end arguments vs. the brave new world. TOIT, 2001, 1: 70–109
17 Wu J P, Liu Y, Wu Q. Theoretical research progress in new-generation Internet architecture. Sci China-Ser F: Info Sci, 2008, 51: 1634–1660
18 Xu K, Xu M W, Li Q, et al. Analysis and case study on multi-dimensional scalability of Internet architecture. Sci China-Ser F: Info Sci, 2008, 51: 1661–1680
19 Hu G W, Xu K, Wu J P, et al. A general framework of source address validation and traceback for IPv4/IPv6 transition scenarios. IEEE Network, 2013, 27: 66–73
20 Xu K, Liu H Y, Liu J C, et al. LBMP: A logarithm-barrier-based multipath protocol for Internet traffic management. IEEE TPDS, 2011, 22: 476–488
21 Xu K, Liu H Y, Liu J C, et al. One more weight is enough: Toward the optimal traffic engineering with OSPF. In: Proc IEEE ICDCS Minneapolis, USA. Washington, DC: IEEE Computer Society, 2011. 836–846
22 Wu J P, Lin S, Xu K, et al. Advances in evolvable new generation internet architecture research (in Chinese). Chin J Comp, 2012, 35: 1094–1108
23 McKeown N, Anderson T, Balakrishnan H, et al. OpenFlow: Enabling innovation in campus networks. ACM SIGCOMM CCR, 2008, 38: 69–74
24 Clark D D. The design philosophy of the DARPA Internet protocols. ACM SIGCOMM CCR, 1995, 25: 102–111
25 Kurose J. Networking: Successes, new challenges, and an expanding waist, as the field approaches 40. IEEE Infocom 2004 Keynote Talk, 2004
26 Aura T. Cryptographically generated addresses (CGA). RFC 3972. 2005
27 Xiang Y, Yin X, Wang Z L, et al. Internet flattening: Monitoring and analysis of inter-domain routing. In: Proc IEEE ICC, Kyoto. New York: IEEE Press, 2011. 1–6
28 Li Q, Wang D, Xu M W, et al. On the scalability of router forwarding tables: Nexthop-selectable FIB aggregation.

In: Proc IEEE INFOCOM, Shanghai. Washington, DC: IEEE Computer Society, 2011. 321–325
29 Zhang W G, Yin X, Wu J P, et al. Real aggregation for reducing routing information base size. JCIT, 2010, 5: 47–53
30 Xu M W, Yang S, Wang D, et al. Two dimensional-IP routing. In: Proc ICNC, San Diego, 2013. 835–839
31 Li D, Li Y J, Wu J P. ESM: Efficient and scalable data center multicast routing. IEEE/ACM Trans Networking, 2012, 20: 944–955
32 Xiang Y, Shi X G, Wu J P, et al. Sign what you really care about-secure BGP AS paths efficiently. Comp Netw, 2013, 57: 2250–2265
33 Yao G, Bi J, Xiao P Y. VASE: Filtering IP spoofing traffic with agility. Comp Networks (Elsevier), 2013, 57: 243–257
34 Wu P, Cui Y, Wu J P, et al. Transition from IPv4 to IPv6: A state-of-the-art survey. IEEE Comm Surveys Tutor, 2012, 15: 1407–1424
35 Cui Y, Dong J, Wu P, et al. Tunnel-based IPv6 transition. IEEE Internet Comp, 2012, 17: 62–68
36 Chen Z J, Zhao Y, Lin C, et al. Accelerating large-scale data distribution in booming Internet: Effectiveness, bottlenecks and practices. IEEE Trans Consum Electr, 2009, 55: 518–526
37 Ma Z, Xu K, Liu J C, et al. Measurement, modeling and enhancement of BitTorrent-based VoD system. Comp Netw, 2012, 56: 1103–1117
38 Lei L, Lin C, Cai J, et al. Performance analysis of wireless opportunistic schedulers using stochastic Petri Nets. IEEE Trans Wirel Comm, 2009, 8: 2076–2087
39 Huang B, Sun Z G, Chen H, et al. BufferBank: A distributed cache infrastructure for peer-to-peer application. Peer-to-peer Netw Appl, 2012. 1–12
40 Xu K, Zhu M, Lin C. Internet architecture evaluation models, mechanisms, and methods (in Chinese). Chin J Comp, 2012, 35: 1–22
41 Xu K, Zhu M, Wang N, et al. The 2ACT model-based evaluation for in-network caching mechanism. In: Proc IEEE symposium on Computers and Communications (ISCC), Split, Croatia. 2013
42 Wolf T, Griffioen J, Calvert K L, et al. Choice as a principle in network architecture. ACM SIGCOMM CCR, 2012, 42: 105–106
43 He H , Xu K , Liu Y. Internet resource pricing models, mechanisms, and methods. Netw Sci, 2012, 1: 48–66
44 Xu K, Zhong Y F, He H. Internet Resource Pricing Models. New York: Springer-Verlag. 2013
45 Shi X L, Xu K, Wang H Y, et al. Online combinatorial double auction for mobile cloud computing markets. Technical Report. Beijing, China: http://www.thucsnet.org/uploads/2/5/2/8/25289795/auction.pdf
46 Wu J P, Bi J. Source Address Validation Improvement Framework. Draft-ietf-savi-framework-06, 2011
47 Wu J P, Ren G, Li X. Source address validation: Architecture and protocol design. In: Proc IEEE ICNP, Beijing. New York: IEEE Press, 2007. 276–283
48 Hu G W, Wu J P, Xu K et al. SAVT: A practical scheme for source address validation and traceback in campus network. In: Proc IEEE ICCCN, Maui, Hawaii. New York: IEEE Press, 2011. 1–8
49 Bi J, Liu B Y, Wu J P, et al. Preventing IP source address spoofing: A two-level, state machine-based method. Tsinghua Sci Tech, 2009, 4: 413–422
50 Choi J, Park C, Jung H, et al. Addressing in future internet: Problems, issues, and approaches. In: Proc 3rd International CFI, 2008