Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/080640

Modeling and Analysis of an IPv4-IPv6 Address Translation System

Zheng Hong[1,*], *Sun Nigang*[2] *and Pan Li*[3]

[1] School of Information Science and Engineering, East China University of Science and Technology, Shanghai 200237, China
[2] School of Information Science and Engineering, Changzhou University, Changzhou 213000, China
[3] Department of Information and Communication Engineering, Hunan Institute of Science and Technology, Yueyang 414006, China

Received: 10 Nov. 2013, Revised: 8 Feb. 2014, Accepted: 9 Feb. 2014
Published online: 1 Nov. 2014

Abstract: With the development of IPv6, several IPv6-IPv4 translation systems for communicating between IPv6 networks and IPv4 networks have been proposed. The paper presented an IPv4-IPv6 translation mechanism by using network Address. Also, the IPv4-Pv6 translation system is modelling based on Petri nets. By analyzing the boundedness, liveness and reversibility of the model, the proposed IPv4-IPv6 translation method is feasible and satisfies system security requirements.

Keywords: Modelling, IPv4, IPv6, Translation, Petri nets

1 Introduction

The recent concerns about IPv4 address space exhaustion increased the attention given to IPv6 deployment. Available globally-addressable space on the IPv4 Internet is decreasing. It is difficult to measure the rate of decrease, and even one or two very large-scale applications that require global address space could exhaust most of the space that can be allocated without disruption to existing users and applications. Even an expansion of dedicated Internet connections in many countries, if done using IPv4, could substantially exhaust the remaining IPv4 address space. IPv6 (Internet protocol, version 6) was developed by the Internet Engineering Task Force (IETF), starting in 1993, in response to a series of perceived problems, primarily regarding exhaustion of the current, IP version 4 (IPv4) address space. It arose out of an evaluation and design process that began in 1990 and considered a number of options and a range of different protocol alternatives.

Each entity on the network needs IP address to be used as a fundamental and unique identifier. IPv4 [1] was the first version of the Internet protocol that was widely deployed in order to provide unique global computer addressing to make sure that two computers (or any two network devices) can uniquely identify one another. Due to the fast growth of the network, a huge number of unique addresses are needed; the existing IPv4 Protocol exposes serious well-known flaws. The almost-exhausted IPv4 address with more than three-quarters of the 4 billion addresses occupied [2]. Thus, a new version of the Internet Protocol has been designed by the Internet Engineering Task Force (IETF), known as IPv6 [3]. The main goal for designing the new Internet Protocol (IPv6) is to increase the number of IP addresses (address spaces). The IPv6 address was designed with a 128-bit (16- bytes) address scheme instead of the 32-bit (4-bytes) address scheme in IPv4, which means IPv6 can express over 3.4x1038 possible unique addresses [4]. IPv6 will have enough to uniquely address each device (e.g. telephone, cell phone, mp3 player, automobile, etc) on the surface of earth with full end to end connectivity (about 32 addresses per square inch of dry land). In addition, IPv6 is designed to support security (IPSec), scalability, and multimedia transmissions. Overall, IPv6 was carefully thought out and was designed with future applications in mind.

Several countries have prepared a schedule for implementation the new Internet Protocol (IPv6) to meet their future deployment needs. However, IPv6 is not backward-compatible with the IPv4. The displacement of IPv4 by IPv6 is a fairly long process. So it is impossible to throw away the existing IPv4 network and to adopt IPv6 immediately. It is foreseen that the transition will

* Corresponding author e-mail: zhenghong@ecust.edu.cn

happen in stages with a few IPv6 nodes introduced into an IPv4 network and the number gradually increasing over time till sometime in the distant future when the entire network becomes IPv6 [5,6,7]. Thus, it is necessary to resolve IPv4/IPv6 address translation.

The remainder of the paper is structured as follows: In section 2, we introduce some related work and our research motivation. Section 3 proposed IPv4-IPv6 Translation System. In section 4, the translation process based on Petri nets is modelled and analyzed. Finally, in section 5, we conclude our paper.

2 Related Works and Research Motivation

IPv4 was widely deployed and uses 32-bit for both source and destination IP addresses which limit the address space to 4.3x109 possible unique addresses. Many of these addresses are reserved for special purposes (in private networks or multicast addresses). IPv4 has some limitations with a shortage of IP address spaces, the requirements for security at the Internet layer, and the increase on quality of services (QoS) demands. The improvement from IPv4 to IPv6 provides a platform for new Internet functionality that will be required in the near future in addition to simplification of the header format and size [8]. In addition, IPv6 addresses come in three different types: Unicast, Multicast, and Anycast; where each address type is used to determine if the sent packets are destined for one or many machines [9,10,11,12]. Thus, IPv6 has several features such as the new header format with minimum header overhead, large address spaces, built-in security, better support for QoS, and enhanced support for Mobile IP.

The problem of porting existing applications to IPv6 has been so far addressed by several researchers, including companies and academic institutes. A white paper by Microsoft [13] focuses on Windows applications, but at the same time offers some general guidelines that apply to any application for any operating system.

IPv6 provides many benefits over IPv4 technology, but IPv6 deployment requires co-existence with IPv4 for some period of time in order to enable IPv4 network users to communicate with their old applications [11,12]. A number of mechanisms have been developed for managing the transition from IPv4 to IPv6 and vice versa. There are three main mechanisms that have already emerged; Tunnelling, Dual-Stack, and Translation. The Dual-Stack Mechanism proposes to use the dual stack IP approach on the basis of IPv4 addresses assigned dynamically only when needed, and the use of IPv4 over IPv6 tunnelling in order to cross the local IPv6 network before accessing the outer IPv4 network. But this mechanism has a disadvantage that all the edge nodes need edge nodes upgraded to run IPv6 as well as IPv4 protocols [10,11]. These edge nodes need to be able to support dual addressing schemes, dual management routing protocols as well as having sufficient memory for both IPv4 and IPv6 routing tables. The tunneling mechanism can be used when two hosts that are located in two different IPv6-only zones want to communicate with each other by passing their packets through an IPv4-only zone, in this case the IPv6 packet will be encapsulated in an IPv4 packet to be passed through the IPv4-only zone. The tunnelling mechanism suffers from the increase of the network traffic overhead (as a result of encapsulating IPv6 packets in IPv4 packets).

A variety of translation mechanisms are proposed such as Stateless IP/ICMP Translation (SIIT), Network Address TranslationCProtocol Translation (NAT-PT), and Network Address and Port Translation CProtocol Translation (NAPT-PT) [12].The translation mechanism has several limitations such as the number of simultaneous connections and the capacity of the translator. In addition, some security protocols such as IPSec are not compatible with the translation device [13].

The paper proposes a translation method, which depends on identifying two public addresses (IPv4 and IPv6) for each communicating session, understanding the received datagram, capturing and identifying the header, converting the header, transformation of the datagram to the destination environment and then transmitting the datagram to the destination address. This method is inspired by the fact that a host only use a small amount of port numbers to connect with others (Some mainstream operating systems also limit the maximum number of concurrent TCP connections of a host). In addition, the translation method deals with the bi-directional operation that converts the received packet into the destination environment depending on identifying two public addresses for the two different environments (IPv6 and IPv4 environments).

3 The IPv4-IPv6 Translation System

The transition phase from IPv4 to IPv6 has raised many discussions among the Internet community, as a lot of companies and network administrators are reluctant, facing what they perceive as a great challenge with large costs. Apart from the network and hardware part of the issue, a very important aspect is the modification (porting) of existing applications so that they become IPv6 enabled. It is a necessary step in the wider adoption of IPv6, not only because without them the new infrastructure becomes useless for the user, but also because applications have the ability to clearly demonstrates the advantages of IPv6. The majority of network applications in existence today presume the use of the IPv4 protocol, so the transition to IPv6 has to be accompanied by the development of new applications and/or the modification of the existing ones, so that they can be used in IPv6 environments. It has often been demonstrated that the difficulty of modifying existing applications varies significantly from one case to another.

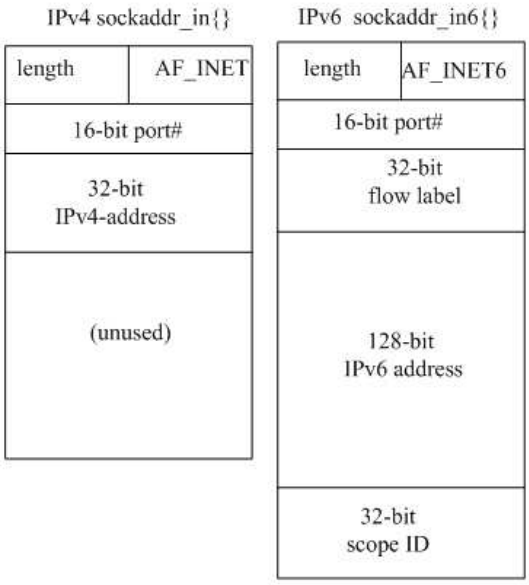

Fig. 1: IPv4 and IPv6 address structures

The principle rule of address translation is to establish the mapping mechanism between sender and receiver, which can map message sequence on a particular protocol to another protocol sequences. Protocol conversion is a process of the mapping from one sequence to another. The basic operation of IPv4-IPv6 translation is the mapping of the IP header between the two protocols, replacing the header from sender to receiver; whether the higher layer of the two protocols conducts the similar replacement is depended on differences between them [13,14,15]. In RFC6052 [15], a fundamental framework of stateless address translation is defined. The IPv4 and IPv6 address structures are illustrated by Fig. 1.

The IPv6 address header is simplified by ignoring or setting default for some information fields in the IPv4 protocol. In general, the header of the two protocols is quite similar that some fields can be directly copied between two protocols. Of course, others need to conduct translation.

If the IPv6 data packets are required to address the neighbour discovery protocol, it is addressed; otherwise, IPv6 packets and IPv4 packets proceed along the same way to determine the translation. Then according to the different direction of the translation, packets are transmitted to the corresponding translation processing module. The translation processing module processes converted data packets and generates new packets. Finally, the processed data packets are directly sent to the network interface. Table I is IPv4-IPv6 Transition Mapping.

The system work flow can clearly describe the work flow of the IPv4 and IPv6 translation process (as shown in Fig.2).

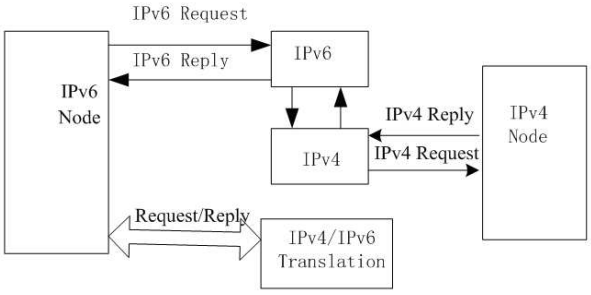

Fig. 2: The IPv4-IPv6 address translation System

First, the system listens on the network interface to obtain a data frame, and then reads the header of data frame, and determines the protocol type according to the type of header fields. If the packet type value is equal to 0x0800, it is an IPv4 packet, which can use to determine the way of the IPv4 packet's translation and to process according to the configuration as follow. If the packet type value is equal to 0x0806, it is an ARP packet. The packet is transmitted to the ARP module to convert ARP request and response and to update the ARP table. If the packet type value is equal to 0x08DD, it is an IPv6 packet [16].

IPv4 to IPv6 transition detailed process describes the details of IPv4 and IPv6 translation, as shown in Fig.3. These steps as follow:

(1)To determine whether the IPv4 packet TTL field value is equal to 1. If the value is equal to 1, the data packet's lifetime has been used up. The packet will be discarded. The system will wait to receive next data packet.
(2)According to the address mapping table, the IPv4's source address and destination address are transformed into IPv6's address, and a new IPv6 packet source address and a new destination address are generated. If there is no address mapping table to find the corresponding IPv6 address during translation, an error return: the address is not available.
(3)According to translation algorithm, the IPv4 packet is converted into to an IPv6 data packet one by one field in accordance with each.
(4)According to the type of higher-layer protocol, call appropriate functions to deal with TCP, UDP, or ICMP packets, and a new checksum is calculated.

In addition, if the data packet is a fragment, and, it also has an additional fragmentation extension header, then, the domain settings are not changed basically, except for the following differences:

(1)Payload Length: Plus 8 on the above basis calculation results.
(2)Next Header: 44 (Fragment extension header).
(3)The setting of fragment extension header as follows:

Table 1: IPv4-IPv6 address Transition Mapping

	All 0	All 0
IPv4 Total length-IPv4 Header Length x 4	IPv4 Protocols value. If it is 1(ICMPv4), must be replaced by 58(ICMPv6)	IPv4 TTL value-1. If it is 0, report error: "TTL Exceeded"
The source address and destination address to corresponding IPv6 addresses by converting		

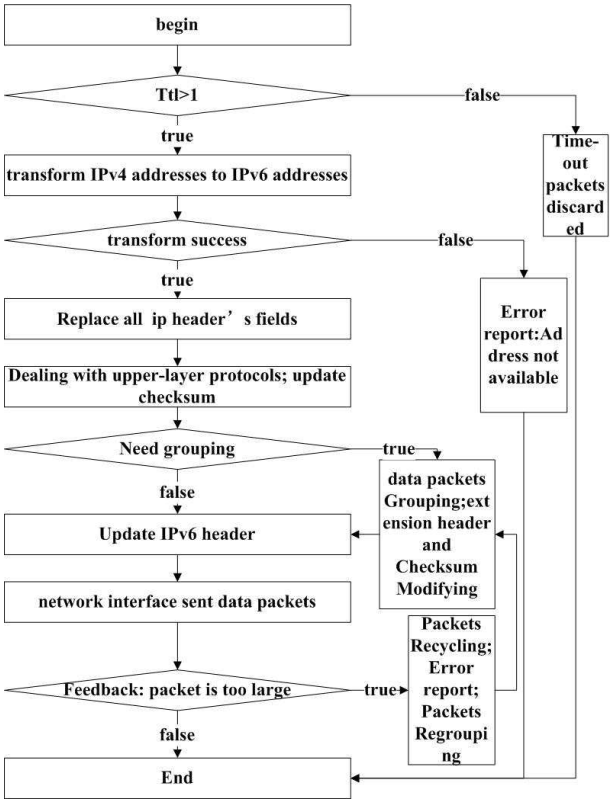

Fig. 3: IPv4-IPv6 Transition Process

–Next Header: Fill IPv4 Protocol value, if its value is 1(ICMPv4) , it must be replaced by 58(ICMPv6).
–Reserved: 0.
–Fragment Offset: Replaced by IPv4 Fragment Offset.
–MF Flag: Replaced by IPv4 MF Flag.

(4)Identification: The low-order 16 bits are replaced by IPv4 Flags; the high-order 16 bits set to 0.

4 Modelling and Analysis of IPv4-IPv6 Address Transition Process

4.1 Petri nets

Petri nets are widely used in various application domains for its simplicity and flexibility in depicting dynamic system behaviours. Their inherently asynchronous concurrent semantics matches that of many physical systems of interest. For example, they are very suitable to describe a networks architecture, services, and protocol. Petri nets have advantages in modelling, analysis, and verification because of their intuitive graphical representation and rigorous mathematical theory and their wealth of analytical techniques and tools [17,18,19].

Definition 1. A Petri net is defined as a 4-tuple, $PN = (S,T,F;M_0)$, and $S \cap T = \varnothing, S \cup T \neq \varnothing, F \subseteq (S \times T) \cup (T \times S), dom(F) \cup cod(F) = S \cup T$. S and T are two disjoint sets, known as the basic elements of a Petri net Σ. Elements of S named $S-$ or Place; Elements of T named $T-$ or transition. F is the net flow of Σ. $dom(F)$ and $cod(F)$ are Pre-domain and post-domain of F. M is a marking of Σ. Each resource of Place is called marking.A transition t is enabled in state M if and only if $\forall p \in^{\bullet} t$, $M(p) \geq 1$. If the transition t is enabled in the state M, then t is fired to the new state M, Denoted by $M[t > M'$.

Definition 2. Suppose that $\Sigma = (S,T,F;M_0)$ is a Petri net, where M_0 is the initial marking of Σ. Reachable marking set of Σ is $R(M_0)$ which is defined as a minimum set meets the following conditions: (1) $M_0 \in R(M_0)$; (2) if $M \in R(M_0)$, and $t \in T$ satisfy that $M[t > M'$, so we say that $M \in R(M_0)$.

Definition 3. Suppose that $\Sigma = (S,T,F;M_0)$ is a Petri net, and $s \in S$. If there exists a positive integer named B, such that $\forall M \in R(M_0) : M(s) \leq B$, then place s is bounded. The smallest positive integer B that meets this criteria is the bound of s, denoted by $B(s)$. When $B(s) = 1$, we say this place s is safe. If every $s \in S$ are bounded, we define Σ as a bounded Petri net, and $B(\Sigma) = max\{B(s)|s \in S\}$ is the bound of Σ. Only when $B(\Sigma) = 1, \Sigma$ is safe.

Definition 4. Suppose that $\Sigma = (S,T,F;M_0)$ is a Petri net, M_0 is the initial marking of Σ, and $t \in T$. Only if every $M \in R(M_0), M' \in R(M)$, and $M'[t >$, the transition t is fireable. If and only if every $t \in T$ is fireable, the Σ is a live Petri net [16].

4.2 IPv4 to IPv6 Translation Model Based on Petri Nets

According to the above system flow chart and the detailed flow chart, the corresponding Petri translation model was established (as shown in Fig.4 and Fig.5).

Besides describing the function of IPv4-to-IPv6 converter by the established Petri net model, it is also

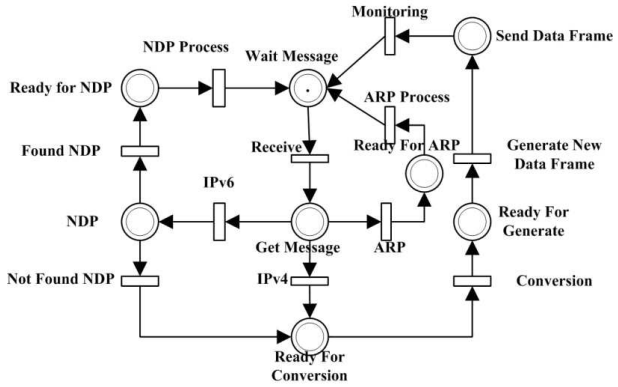

Fig. 4: The System Model Based on Petri Nets

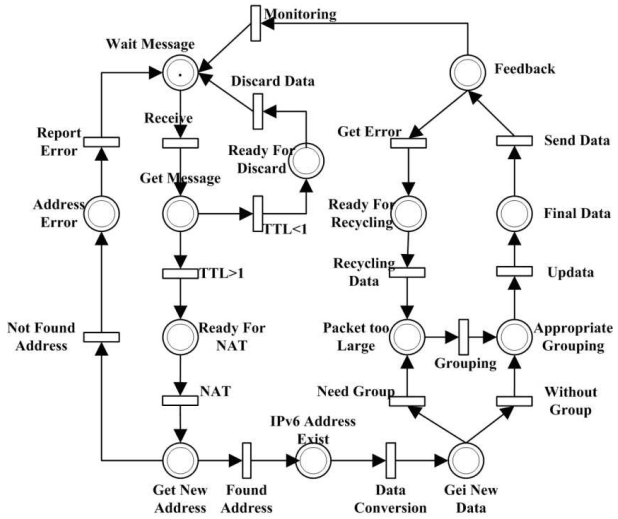

Fig. 5: The Translation Process Model based on Petri Nets

important to prove that this is a safe system, rather than to design flaws and system error during the translation process. Therefore, we will analyze dynamic natures of the model to verify the correctness and security of the system.

According to Fig.4, properties of the IPv4/IPv6 translation are as follows:

Property 1. The model can complete a basic IPv4 to IPv6 translation.

Starting from the model's initial state, through the following transition sequences: $S_1 = T_{receive} -> T_{IPv4} -> T_{(IPv4\ \ to\ \ IPv6)} -> T_{(generate\ \ new\ \ data\ \ frame)}$ and $S_2 = T_{receive} -> T_{(TTL>1)} -> T_{NAT} -> T_{(found\ \ Address)} -> T_{(data\ \ Translation)} -> T_{Without\ \ group} -> T_{(data\ \ Translation)} -> T_{send}$, these two firing sequences make Me(send data frame)=1, Me(feedback)=1. Converters have completed a basic IPv4 to IPv6 transition, which achieves our design purpose.

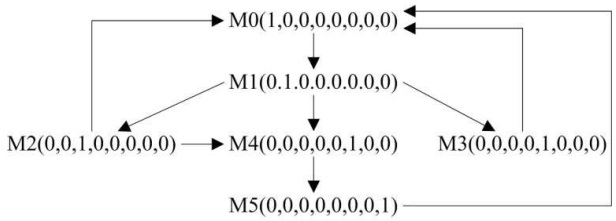

Fig. 6: Reachable Marking Graph of the System Model

Property 2. The error report can be presented according to different problems.

Other transition firing sequences $S_{error1} = T_{receive} -> T_{(TTL<1)} -> T_{(discarddata)}, S_{error1} = T_{receive} -> T_{(TTL>1)} -> T_{NAT} -> T_{(not\ \ found\ \ Address)} -> T_{(report\ \ error)}, S_{error3} = T_{receive} -> \cdots -> T_{(get\ \ error)} -> T_{(recycling\ \ data)} -> \ldots$ will be triggered, after achieving system basic functions, which suggests that the conversion failed due to external cause.

Transition firing sequence $S_g = T_{(need\ \ grouping)} -> T_{grouping} -> \ldots$, if that IPv6 data packet is too large to complete the treatment due to translation generated.

Transition firing sequence SARP: $T_{receive} -> T_{ARP} -> T_{(ARP\ \ process)}$ and $S_{NDP} = T_{IPv6} -> T_{(found\ \ NDP)} -> T_{(NDP\ \ process)}$ deals with other types of data frames.

Besides basic functions, the model must also meet the reversibility, boundedness, the liveness and other dynamic properties in order to guarantee system stability and accuracy.

The translation is transformed data frame one by one, which requires that the system can return to the initial state after the translation for one data frame is completed. That is, the initial state must be a home state. Also, the corresponding model based on Petri nets must be a recoverable system. So, the model is a reversible network system.

Based on the Petri nets model, the Petri nets reachable graph can be drawn (as shown in Fig.5). It shows that the reachable graphic is strongly connected, because any two states can reach each other through a transition firing sequence in the reachable graph. So there is $S_i \in R(S_0)$ and $S_0 \in R(S_i), i = 1,2,3 \ldots N, N$ respectively 7 and 12 in Fig.6). Petri nets are reversible network systems and have S_0 as home state at the same time. It proves that the system always waits to receive new data frame, while handles all of data frames after a series of transition firing.

In the process, we adopt FIFO to handle only one packet in a conversion. Because the number of resources packet is 1, in order to resume normal operation, obviously the model designed to satisfy the bounded and the bound number is equal to 1.

The reachable marking graph of every vertex of the vector shows that the state vector of each vertex are all

0-1 Vector, and any state S satisfies $B(S) = 1$. Therefore, It can be concluded that the Petri net model is safe according to the inference of safety reachable marking. It is a bounded Petri net and the bound is equal to 1. This model is consistent with our requirements.

S_0 is an initial marking of this model. For any marking S, there is $S \in R(S_0)$ and $S \in R(S)$(Features of strongly connected graph). For any transition t, there is $|^\bullet t| = |t^\bullet| = 1$.So for all $S \in ^\bullet t$, there is $S[t>$. Transition t is fireable, the Petri net is live. This means that this models dynamic operation can be achieved. In any place, a marking can start and generate follow-up marking, and the system cycles up. All of the above prove that this model and the corresponding system are safe and effective during the run-time.

5 Conclusion

The demand for Internet IP addresses is rapidly growing with large information explosion, but the current IPv4 protocol cannot meet the needs of the Internet. It is impossible to expect a fast, centrally coordinated cutover. To make the whole transition concept feasible, the coexistence of both IPv4 and IPv6 must be arranged in a practical and simple way.

The IPv4-IPv6 translation is proposed in this paper by network address. This method uses a small amount of port numbers to connect with others and deals with the bi-directional operation that converts the received packet into the destination environment depending on identifying two public addresses for the two different environments (IPv6 and IPv4 environments). In addition, we analyzed and verified more exactly the correctness and dynamic properties of transition by modelling the translation system. To verify the model can repeat the operation after the successful implementation, and data is complete and orderly, this paper analyzes three dynamic natures (reversibility, liveness and boundedness) of the model, and achieves the desired conclusion with reachable marking graph and vector. It is very helpful to design and specify IPv4-IPv6 converter.

Acknowledgement

The work was supported in part by the National Natural Science Foundation of China under Grant 61103115 and 61103172, Hunan Provincial Natural Science Foundation of China under Grant 11JJ4058, and Scientific Research Fund of Hunan Provincial Education Department under Grant 11A041.

References

[1] B. Carpentere, K.Moore, Connection of IPv6 Domains via IPv4 Clouds. RFC3056, Feb. (2001).

[2] R.Gilligan, E.Nordmark, Transition Mechanisms for IPv6 Hosts and Routers. RFC 2893, Aug. (2000).

[3] A. Azcorra, M. Kryczka , Garcia-Martinez A. Integrated Routing and Addressing for Improved IPv4 and IPv6 Coexistence[J]. IEEE Communications Letters, **14**, 477-479 (2010).

[4] K.C. Claffy, Tracking IPv6 Evolution: Data We Have and Data We Need, Computer Communication Review, **41**, 43-48 July (2011).

[5] D. Green, Fiuczynski ME. IPV6 Translation for IPV4 Embedded Systems, Proceedings of the IEEE Military Communications Conference: Oct 17-20, 2005, Atlantic City, NJ, USA, 2519-2525 (2005).

[6] J.Govi, On the Investigation of Transactional and Interoperability Issues between IPv4 and IPv6, Proceedings of the 2007 IEEE Electro/Information Technology Conference, Chicago, IL, USA, May 17-20, 604-609 (2007).

[7] C.Bouras, A.Gkamas, D.Primpas, K.Stamos, Porting and performance aspects from IPv4 to IPv6:The case of OpenH323. International Journal of Communication Systems, **18**, 847-866 (2005).

[8] J.Curran., An Internet Transition Plan, IETF Draft, Jul. (2008), [Online]Available: http://tools.ietf.org/html/rfc5211.

[9] X.H. Huang, Y. Ma, Flow Label-Based IPv6 Packet Classification Algorithm with Dimension Reduction Capability, China Communications, **9**, 1-9 (2012).

[10] G. Tsirtsis and P. Srisuresh, Network Address Translation-Protocol Translation (NAT-PT), RFC 2766, (2000).

[11] R. Haines, G. Clemo, A.Munro, Toward formal verification of 802.11 MAC protocols: Verifying a Petri-net Model of 802.11 PCF, Proceedings of the IEEE 64th Vehicular Technology Conf., Montreal, Que, Canada, Sep 25-28, 1-5 (2006).

[12] Fang, Xianwen; Hao, WenJun; Fan, Xiaoqin; et al. The Analysis Method about Change Domain of Business Process Model Based on the Behavior Profile of Petri Net. Applied Mathematics and Information Sciences, **6**, 943-949 (2012).

[13] S. Narayan, S. Sodhi, P. Lutui, et al. Network Performance Evaluation of Routers in IPv4/IPv6 Environment A Tested Analysis of Software Routers, Proceedings of the IEEE International Conference on Wireless Communications, Networking and Information, 2010 Beijing, China, Jun 25-27, 707-710 (2010).

[14] Grosse E, Lakshman Y. N. Network processors applied to IPv4/IPv6 transition, IEEE Network, **17**, 35-39 (2003).

[15] http://tools.ietf.org/html/rfc6052

[16] W.E. Chen, Y.B. Lin, A.C. Pang, An IPv4CIPv6 translation mechanism for SIP overlay network in UMTS All-IP environment. IEEE Journal of Selected Areas in Communications, **23**, 2152-2160 (2005).

[17] D. Aloini, R.Dulmin, V. Mininno, Modelling and assessing ERP project risks: A Petri Net approach, European Journal Of Operational Research, **220**, 484-495 (2012).

[18] A. L. Feller, T. Wu, D. L. Shunk, and J. Fowler, Petri net translation patterns for the analysis of ebusiness collaboration messaging protocols, IEEE Trans. Syst., Man, Cybern. A, Syst., Humans, **39**, 1022-1034 (2009).

[19] X.M. Zhu, J.X. LIAO, J.L. CHEN, Petri Nets Model of Protocol Conversion for CTF Service: Its Universal Coupling Criteria and Property Analysis. International Journal of Communication Systems, **20**, 533-551 (2007).

Zheng Hong received her Ph.D. degree in computer software and theory, Chinese Academy of Science, 2003. She joined East China University of Science and Technology in 2003, where she is currently an associate professor in the department of the Computer Science and Engineering. She is a visiting scholar with the Department of Computer Science and Engineering, University of California, Riverside, USA. Her research interests include formal methods, system modelling and analysis, Petri net theory and applications.

Sun Nigang received his Ph.D. degree in information security, University of Chinese Academy of Sciences, 2007. He joined East China University of Science and Technology in 2007 and Changzhou University in 2010. And now he is an associate professor in the department of the Computer Science and Technology. His research interests include formal methods, system modelling and analysis, information security.

Pan Li received his Ph.D. degree in computer applied technology from Tongji University, China in 2009. He is an associate professor with the Department of Information and Communication Engineering, Hunan Institute of Science and Technology, China. His research interests are in Petri nets, workflows and computational intelligence.