Design and implementation of packet filtering algorithm in 6LoWPAN access Internet

Dapeng Zhu [a], Yanping Yu [b] and Qinming Zhang [c]

School of Automation, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

[a]daydptr@163.com, [b]707306408@qq.com, [c]864381983@qq.com

Abstract

Focused on multiple data flow disorder and collision problem of gateway in 6LoWPAN access Inter-net, a packet filtering algorithm based on Berkeley Packet Filter and Netfilter mechanism is presented. Packets are classified by this algorithm according to processing mode in which the gateway handles them. With IP address of packets as the filtering condition, only allowing the same type of packets to enter single space and preventing them entering other spaces to solve the problem of packet collision in communication between heterogeneous networks. Experimental results demonstrate that the algo-rithm can realize the effective scheduling of packets in the gateway and improve the security of gate-way by capturing and blocking multiple datastreams accurately according to the given filtering strat-egy.

Keywords

IPv6, packet filtering, 6LoWPAN, wireless sensor network, gateway.

1. Introduction

WSN (Wireless Sensor Network) is an integrated intelligent information system that contains information collection, processing, transmission and application; it is made up of numerous micro embedded devices with wireless communicating and monitoring ability being used to provide real-time environmental monitoring [1]. In terms of its purpose, it is important that WSN must be connected with extern networks to make sure that observers can monitor physical world by extern network. As the widest and most popular network, there is no doubt that Internet is the best accessing object of WSN [2]. The 6LoWPAN（IPv6 over Low Power Wireless Personal Area Network） workgroup in IETF(Internet Engineering Task Force) is dedicated to add IPv6 protocol to the network layer of WSN protocol stack, and thereby achieves Seamless integration of Internet base on IPv6 and WSN [3]. However, the kernel of Internet is IPv4, thus the edge gateway, by doing protocol translation between IPv4 and IPv6, be deployed to connect WSN with Internet [4]. Because the gateway has the function of 6LoWPAN edge router and traditional IPv4 router, multiple data flows are brought together in the gateway consequently, in addition the malicious data flows from illegal WSN node may access gateway, accordingly design and implementation of appropriate packet filtering algorithm is of more significance to improve the security and efficiency of forwarding data of gateway. In this paper, the research background and existing filtering algorithm were first illustrated, and then a filtering algorithm base on BPF(Berkeley Packet Filter) and Netfilter was proposed, the testing system be set up to verify the validity of the above algorithm.

2. Related Work

Recently, both foreign and domestic scholars have researched the interconnecting technology and packet filtering algorithm used for heterogeneous networks interworking. For example the NAT-PT (Network Address Translation-Protocol Translation) proposed in reference[5] can connect IPv6 Networks with IPv4 Networks, but the packet filtering was executed in application layer of protocol stack, the efficiency of this filtering method was nowhere near as the packet filter in data link layer. in

reference[6], Multiple data flows access embedded gateway, and then protocol translation technology was used, but the problem of the data collision in gateway was not been considered and solved. in reference[7], the NAT module and PT module was separated and mount in Netfilter framework to allows IPv6 devices to communicate with IPv4 hosts, but the solution disordered the integrality of NAT-PT and went against fault diagnosis, meanwhile limited the flexibility of inputting way of NAT-PT. If this approach was used, we can only mount the hook function in Netfilter framework to perform packet capturing; its efficiency is less than capturing packets by using filtering technology in data link layer. Focused on the above issue, proper technologies of packet filtering will be selected and used for implementation of the flowing packet filtering algorithm.

Table.1 the comparison of two filtering technology

name	language	pseudo-machine type	Filtering position	Code type
BPF	pseudo-code	Acyclic control flow graph	Kernel space	Register-based
DLPI	pseudo-code	Boolean expression true	Kernel space	Stack-based

The BPF and DLPI (Data Link Provider Interface) technology are compared in Table 1, both are within the kernel space to perform filtering, which minimizes the amount of data copied from kernel to the application. This copy, from kernel space to user space, is expensive. If every packet was copied, Filter could have trouble keeping up with fast data links [8]. BPF and DLPI buffer the data destined for an application and this buffer is copied to the application only when the buffer is full, or when the read timeout expires. This timeout value can be specified by the application. The purpose of the buffering is to reduce the number of system calls. The same number of packets are copied between Filter and the application, but each system call has an overhead, and reducing the number of system calls always reduces the overhead [9]. The BPF filter is a directed acyclic control flow graph(CFG) [10], while DLPI uses a Boolean expression tree. The former maps naturally into code for a register machine while the latter maps naturally into code for a stack machine. The CFG implementation used by BPF is normally 3 to 20 times faster than Boolean expression tree, depending on the complexity of the filter [11]. Consequently, BPF is selected to implement the packet filter algorithm which is designed in this paper.

Fig.1 shows the architecture of gateway for protocol translation, the gateway, which provides a platform for implementation of filter algorithm, supports two interconnecting methods: NAT-PT and ISATAP(Intra-Site Automatic Tunnel Addressing Protocol). The former is used to connect 6LoWPAN and Internet-IPv4,the BPF Filter provides the input to the NAT-PT; the latter is used to connect 6LoWPAN with Internet-IPv6 through the IPv4 Network, ISATAP is an address assignment and host-to-host, host-to-router, and router-to-host automatic tunneling technology defined in RFC4214 that provides unicast IPv6 connectivity between IPv6/IPv4 hosts across an IPv4 intranet [12], it is applied in network layer of kernel TCP/IP protocol, thus the hook function was mount in Netfilter framework to capturing packets as the inputting of ISATAP Tunnel.

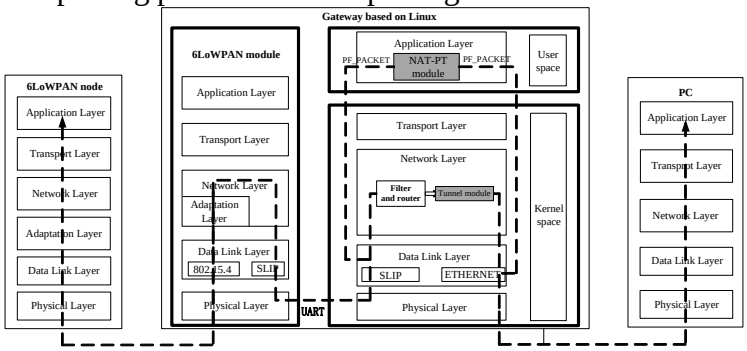

Fig.1 the architecture of gateway software system

Fig.2 shows five hooks located in network layer which are defined in Netfilter framework, the meanings of these five points are describes as follows, NF_IP_PRE_ROUTING: This hook will be triggered by any incoming traffic very soon after entering the network stack, it is processed before any routing decisions have been made regarding where to send the packet. NF_IP_FORWARD: This hook is triggered after an incoming packet has been routed if the packet is to be forwarded to another host. NF_IP_POST_ROUTING: This hook is triggered by any outgoing or forwarded traffic after routing has taken place and just before being put out on the wire. NF_IP_LOCAL_IN: This hook is triggered after an incoming packet has been routed if the packet is destined for the local system; NF_IP_LOCAL_OUT: This hook is triggered by any locally created outbound traffic as soon it hits the network stack. It is the basic framework of Netfilter that monitoring the network packets according to the callback functions registered by user and mount on one of the above five hooks [13].

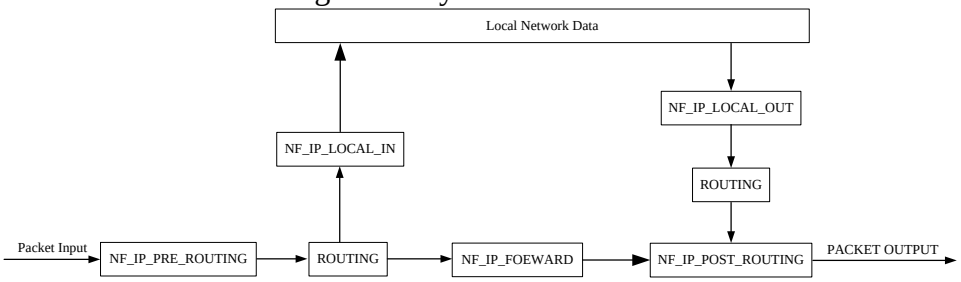

Fig.2 five hooks in Netfilter framework

The packet which originated from 6LoWPAN node and destined for dual-stack host, cross the hooks: NF_IP_PRE_ROUTING, NF_IP_FORWAED, NF_IP_POST_ROUTING. For improving the efficiency of capturing packets and the security of gateway system, the packet is checked immediately when it just enters to the network layer, so the NF_IP_PRE_ROUTING is selected to mount the hook function which is an implementation of filter algorithm to support packets as the input of ISATAP tunnel.

3. The Design of Algorithm for Packet Filter

There are complex and different types of multiple data flows in the edge gateway, so different filtering algorithm are designed According to the type of data flows. In order to facilitate the selection of filter algorithms, all data flows are introduced and then the filter conditions will be shown in Table 2. (6L is the abbreviation of 6LoWPAN sensor network, 4 is the abbreviation of IPv4 network and 6 is the abbreviation of IPv6 network.)

Table.2 the classification of multiple data flows

dataflow	gateway function	filter position	filter condition
6L→4	protocol translation	BPF module	dst-IPv6addr
4→6L	protocol translation	BPF module	dst-IPv6addr
6L→6	tunnel transmitting	Netfilter hook	dst-IPv6addr
6→6L	tunnel transmitting	Netfilter hook	dst-IPv6addr
6L↔6L	6LoWPAN routing	—	—
4↔4	IPv4 routing	—	—

3.1 6L→4 and 4→6L

As far the 6L→4 and 4→6L data flows, BPF filter is used in kernel space to capture the packets as the input of NAT-PT module, the packets are translated from IPv6 to IPv4 format or IPv4 to IPv6 format by NAT-PT module and then be transmitted to destination. Table 3 shows the BPF filter algorithm which is implemented in kernel space. (Filter_IPv6 stands for the IPv6 packet filter, Filter_ARP_Req stands for the ARP request packet filter, Filter_ARP_Rpl stands for the ARP reply packet filter, Filter_IPv4 stands for the IPv4 packet filter.)

Table.3 the filter algorithm based on BPF

Filtering Algorithm
Step1: BPF receives packet form data link driver and transmits the pointer of this packet to every Filter;
Step2: Filter_IPv6 determines whether it is an IPv6 packets, if it does, jump to Step3, otherwise continue check the packet; Filter_ARP_Req determines whether it is an ARP request packets, if it does, jump to Step 4, otherwise continue check the packet; Filter_ARP_Rpl determines whether it is an ARP reply packets, if it does, jump to Step 5, otherwise continue check the packet; Filter_IPv4 determines whether it is an IPv4 packets, if it does, jump to Step6; otherwise continue to step8;
Step3: Filter_IPv6 determines whether the prefix of the packet's IPv6 address is 3ffe::/64, if it does, jump to step7; otherwise, jump to step8;
Step4: Filter_ARP_Req determines whether the destination address of the ARP request packet is from NAT-PT address pool, if it does, jump to Step7, otherwise ,jump to Step8;
Step5: Filter_ARP_Rpl determines whether the destination address of the ARP reply packet is from NAT-PT address pool, if it does, jump to Step7, otherwise ,jump to Step8;;
Step6: Filter_IPv4 determines whether the destination address of this IPv4 packet is from NAT-PT address pool, if it does, jump to Step7, otherwise jump to Step8;
Step7: The packet passes through the filter, the packet is copied to appropriate buffer waiting for read by NAT-PT, jump to Step9;
Step8: Filter block this packet, jump to Step9;
Step9: BPF give the control back to data link driver.

3.2 6L→6 and 6→6L

The packets, captured by BPF Filter above, still enter the network layer of kernel protocol stack, in addition to other invalid data flows, if not properly regulated, they will enter ISATAP tunnel, as a result, the disorder and collision of data flows will occur. Focused on this situation, our filter method is that only allow the packets which follow the format of ISATAP tunnel packet to enter the tunnel, others are block. The filter function is mount in NF_IP_PRE_ROUTING hook to capture and intercept packets, Table 4 shows the specific filtering algorithm.

Table.4 the filter algorithm based on Netfilter framework

Filtering Algorithm
Step1: the hook function is mount in NF_IP_PRE_ROUTING, and then capture packets;
Step2: extracting the IPv6 destination address, if it is conforms with the ISATAP tunnel address format (::0:5efe:w.x.y.z[14]), the callback function executes for transmitting the packet to ISATAP module, and then it is processed in ISATAP module, finally the callback function return nf_stolen; otherwise, discard the packet.

4→4 and 6L→6L

In the process of communication between one IPv4 host and another through protocol-translation gateway, obviously the gateway, play a role of router, only routes and forwards packets. In the

process of communication between one IPv4 host and another through protocol-translation gateway, gateway only uses it 6LoWPAN protocol stack to routing and forwarding packets, the packets are not forwarding and routing in Linux kernel of gateway, therefore, we do not repeat their algorithm her.

4. Testing and Verification of Algorithm

The structure of the testing system is shown in Fig.3.

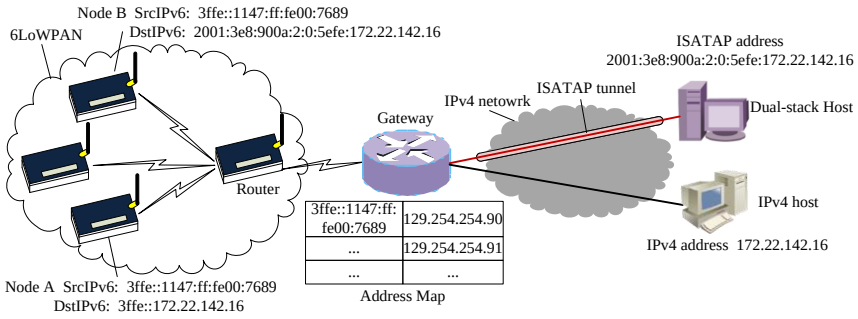

Fig.3 structure of the testing system

The sensor data packets sent by node A, which are captured by the 6LoWPAN protocol analyzer, are shown in Fig.4, and sensor data packets captured by the IPv4 host are shown as Fig.5. Conclusion can be drawn through packet payload and address information in Fig.4 and Fig.5 that IPv6 data packets sent by node A (IPv6 header + UDP header + Application layer data frame) are transformed into IPv4 data packets (IPv6 header + UDP header + Application layer data frame), and the source IPv4 address is the address in NAT-PT pool address; the phenomenon above shows that the kernel space BPF filtering algorithm successfully extracts data packets whose destination IPv6 address is 3ffe:: 172.22.142.16. If the sensor data packets sent by node A pass through the Netfilter filtering module simultaneously, and enter the tunnel. There are two possible results for the tunnel module to process the data packets: (1) Unable to process the packet, then, an ICMPv6 message must be returned, but the protocol analyzer shows no such message is captured, thus this assumption would not be the case. (2) Encapsulate the IPv6 packets as IPv4 packets and send to the IPv4 host, then the IPv4 host will receive the IPv4 packets, and the format should be: IPv4 header + IPv6 header + UDP header + Application layer data frame, however, Wireshark shows that no such packets are captured, this assumption does not hold. From above, it can be proved that the filtering algorithm in kernel space that is based on the Netfilter framework successfully blocks the data packets whose destination addresses are non-ISATAP tunnel addresses. If node A is an illegal node, then the filtering algorithm in kernel space that is based on the Netfilter framework can also filter the illegal malicious data packets.

The IPv6 echo request messages captured by the 6LoWPAN protocol analyzer are shown in Fig.6, Fig.4 and Fig.5 show that the IPv4 host is marked by IPv6 address: 3ffe:: 172.22.142.16 in WSN, the source IPv6 address of the message shown in Fig.6 is the above address, and the destination is node A, It can be known that the message is sent by the IPv4 host, and has been converted to IPv6 format from IPv4 format, from the destination IPv6 address and the address mapping table, we can know that the destination address of the corresponding IPv4 message is the address in address pool. Thus, the BPF filtering algorithm in kernel space successfully extracts data packets whose IPv4 addresses are addresses in NAT-PT module address pool.

The echo interactive process of the IPv4 host and gateway is shown in Fig.7, the echo request message (ICMPv4) sent by IPv4 host has been received and responded by the gateway itself (non-NAT-PT module). If the kernel space BPF filtering algorithm also extracts the echo request message (ICMPv4) from the IPv4 host, Then the processing result of NAT-PT module will be a return of a destination unreachable message (ICPMv4 Type:3 Code:0), however, Wireshark shows that no such packets are captured by the IPv4 host, from above, the kernel space BPF filtering algorithm successfully blocks the data packets whose destination IPv4 addresses are non-NAT-PT module address pool addresses.

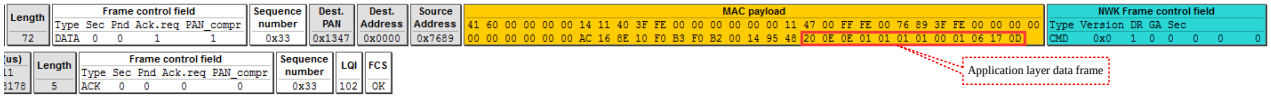

Fig.4 packets sent by node A

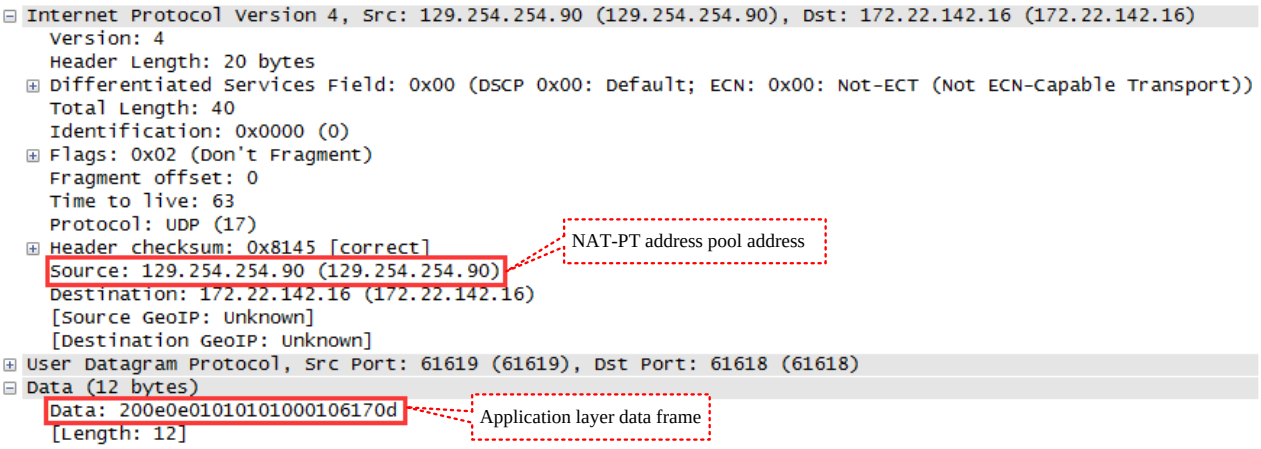

Fig.5 packets captured by IPv4 host

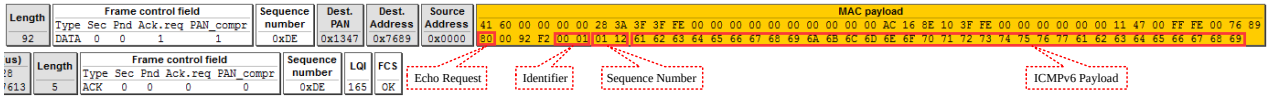

Fig.6 echo request captured by protocol analyzer

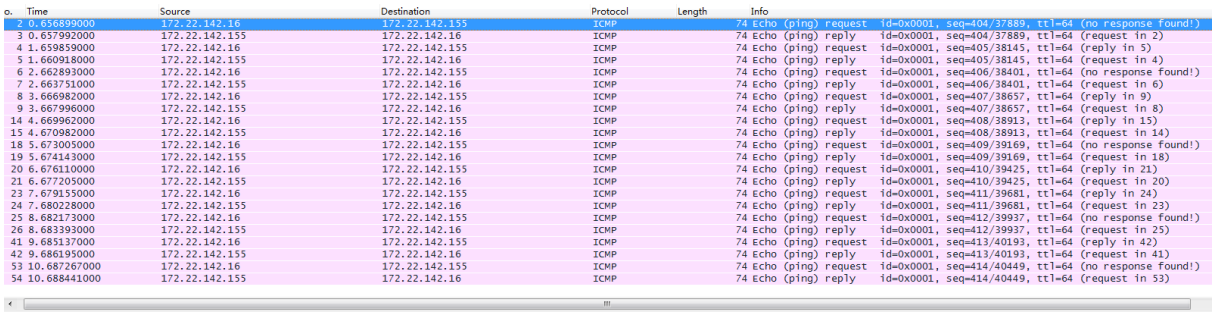

Fig.7 echo interactive process of the IPv4 host and gateway

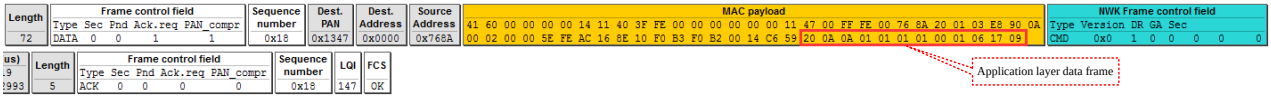

Fig.8 packets sent by node B

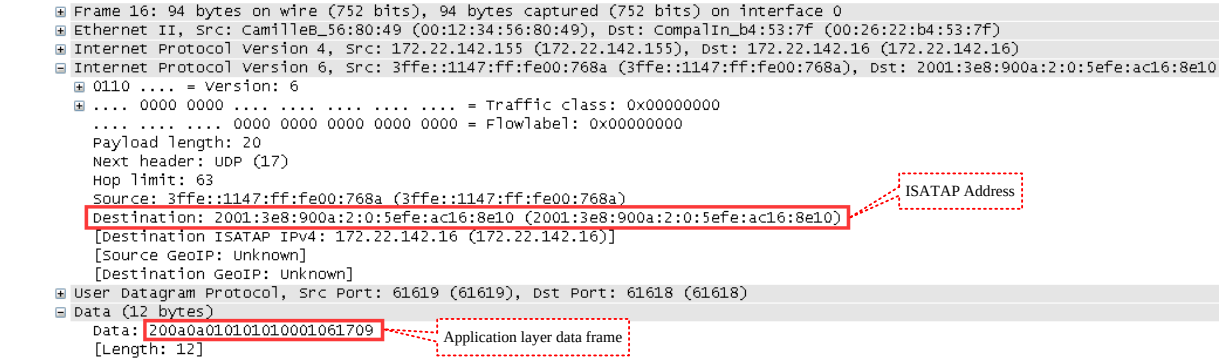

Fig.9 packets captured by dual stack host

The sensor data packets sent by node B are captured by the 6LoWPAN protocol analyzer and as shown in Fig.8. The sensor data packets captured by Wireshark through the dual stack host show in Fig.9, from the data packets load and address information shown in Fig.9 and Fig.8, The IPv6 data packets (IPv6 header + UDP header + Application layer data frame) sent by node B are encapsulated as IPv4 data packets (IPv4 header + IPv6 header + UDP header + Application layer data frame), and the IPv6 address is: 2001:3e8:900a:2:0:5efe:ac16:8e10 (ISATAP tunnel address), namely the IPv4 messages are ISATAP tunnel messages, the phenomenon above shows that the kernel space filtering algorithm based on the Netfilter framework successfully extracts the data packets whose destination addresses are ISATAP tunnel addresses. With the proof of the kernel space Netfilter filtering algorithm to block the process of non-ISATAP tunnel messages, it can be proved that the kernel space BPF filtering algorithm successfully blocks the data packets whose destination addresses are not 3ffe:: 172.22.142.16. If node B is illegal, the kernel space BPF filtering algorithm can also filter the illegal malicious node packets. The success of ARP interactive process is the premise of IP communication between network equipments, and thus the above conclusions can prove that the kernel space BPF filtering algorithm is able to achieve the ARP packets acquisition and blocking according to the established filtering strategies.

5. Conclusion

The experimental results show that the filtering algorithm designed in this paper can accurately extracts and blocks numbers of data flows gathered in the gateway, to achieve the effective scheduling of multiple data streams, and to improve the security of the gateway to a certain extent.

BPF packet filter also has its shortcomings, such as decreased multi-channel distribution performance, unable to handle IP fragment, etc. and need to be optimized; meanwhile, protocol adapter gateway faces increasingly serious security problems, and a secure access mechanism of the system is needed, to avoid malicious attacks and damage. The above two aspects will be the next focus of the research.

References

[1] B. da Silva Campos, J. J. P. C. Rodrigues, L. D. P. Mendes, et al: Design and construction of wireless sensor network gateway with IPv4/IPv6 support, IEEE International Conference on Communications (Kyoto, Japan, June 5-9, 2011), 2011, p.1-3.

[2] G. K. Jeong, A. Terzis, H. S. Dawson, et al. Connecting low-power and loss networks to the in-ternet, IEEE Communications Magazine, vol 49(2011), 96-99.

[3] Y. Xu, R.C. Tan, S. Wu, et al. Connect Internet with Sensors by 6LoWPAN, Journal of Networks, vol 8(2013), 8(7), 1480-1484.

[4] F. Wan, M. Li. Research of IPv4/IPv6 Transition Based on NAT-PT, Journal of Anhui Institute of Architecture & Industry, vol 18(2010), 73-75.

[5] Y. Lu, J. Shi, H. Huang, et al. A Secure Gatway of High preformance for IPv6/IPv4 Based on NAT-PT, Computer Applications and Software, vol 24(2007), 1480-1484.

[6] X. Y. Xie, C. Zhou. Design of Embedded Gateway Which Connect multi-channel Communication Port to Internet, Microcomputer Information, vol 23(2007), 40-42.

[7] S. Wang, L. Liu, Q. L. Chai. Implementation of Ipv4-IPv6 Translatio Gateway Based on NAT-PT with Netfilter Framework, Computer Engineering, vol 32(2006), 147-149.

[8] W. S. Richard, F. Bill, M. R. Andrew: UNIX Network Programming Volume 1 (Posts & Telecom Press, China 2010), p.623-627. (In Chinese)

[9] X. L. Wang, Q. Z. Shen, C. Zhu, et al. Research of Packet Capture Technology in IPv6 High Speed Network Environment, Telecommunication Science, vol 28(2007), 54-62.

[10]M. Zeng, R. C. Zhao. Research on BPF Implementation and Performance Improvement, Com-puter Engineering, vol 33(2007), 43-46.

[11]M. Steven, V. Jacobson: The BSD Packet Filter: A New Architecture for User-level Packet Cap-ture, the USENIX Winter 1993 Technical Conference (San Diego, USA, January, 1993), 1993, p.259-270.

[12]D. Joseph: Understanding IPv6 (Posts & Telecom Press, China 2014), p.244-245. (In Chinese)
[13]J. B. Song: Linux Network Programming (Tsinghua University Press, China 2014), p.492-493. (In Chinese)
[14]Internet Engineering Task Force. RFC5214 Intra-Site Automatic Tunnel Addressing Protocol (IETF 2008), p. 1-15.