



Enhanced Mechanism of Automated IPv6 Site Renumbering

Present by

Miss Kuljaree Tantayakul

**Centre for Network Research
Department of Computer Engineering
Faculty of Engineering
Prince of Songkla University**



Outline

- Introduction
 - Problem and Motivation
 - Neighbor Discovery Protocol
 - Autoconfiguration Protocol
 - Router Renumbering Protocol
- Implementation
- Experimental Setup
- Experimental Results
- Conclusions



Introduction

- IPv6 is a new version of Internet protocol.
- Repair limitation of IP address problem with 128 bit address.
- The small routing table of router.
- Security
- Renumbering



Problem and Motivation

- Changing the global network address
- Changing ISP
- Large the network topology



How Renumbering get done

- Renumbering include 4 part

Host

- Neighbor Discovery and Autoconfiguration Protocol

Router

- Router Renumbering Protocol

DNS Server

- Dynamic DNS Update

TCP wrapper and Firewall



Neighbor Discovery Protocol (ND)

- IPv6 nodes which share the same link physical medium (link) use Neighbor Discovery
 - Determine link layer address of their neighbor
 - Find router

ICMPv6 message concerning renumbering

- Router Solicitation (RS)
- Router Advertisement (RA)



Neighbor Discovery Protocol (cont.)

- **Router Solicitation Message (RS)**

The router solicitation message is transmitted by host to router to generate router advertisement message.



Neighbor Discovery Protocol (cont.)

- **Router Advertisement Message (RA)**

periodic advertisement (of availability of a router)
which contain

- list of prefix use on the link
- a possible value for Max Hop limit
- value MTU



Autoconfiguration Protocol

- A host generates own address using two elements of information
 - The host part is called interface identifier (MAC address)
 - The router part come from address prefix



Router Renumbering Protocol

- Router Renumbering protocol used to update the routers.
- Combines with Autoconfiguration and Neighbor Discovery

Format of router renumbering command

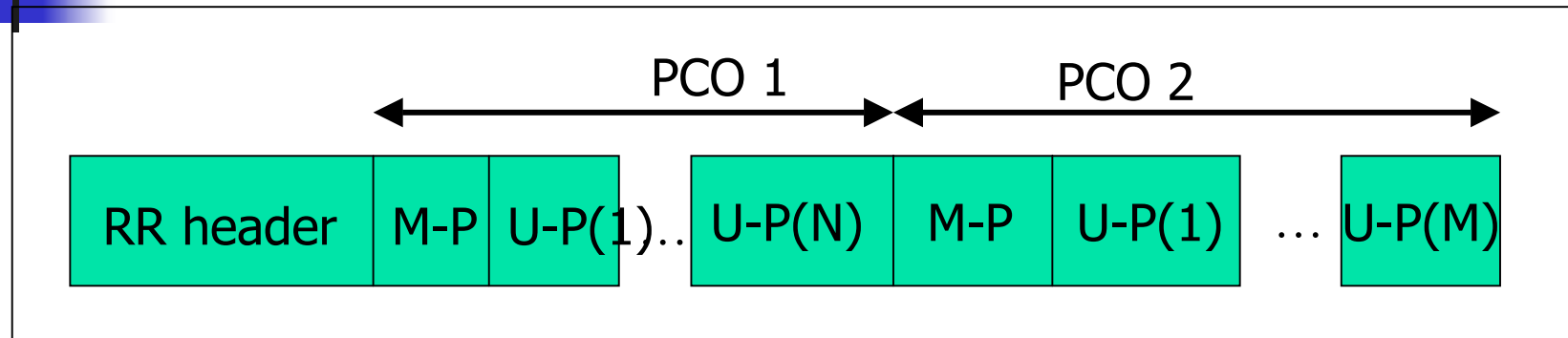


Figure1 Format of router renumbering command

Match-Prefix Part have 3 operations.

1. The ADD operation
2. The CHANGE operation
3. The SET-GLOBAL operation



Implementation

- Management station
 - Compile kernel
 - because the old kernel have not the rrenumd (router renumbering daemon)
 - Implement the function of operations.
- Router
 - enable to receive router renumbering request packets.
 - Implement the function of operations.



Implementation (Cont.)

- DNS Server
 - Configured AAAA Record type
 - Configured to support Dynamic DNS Update
- Firewall and TCP Wrapper
 - Host must be embedded the program for automatically update the filter rule.



The relationship between renumbering IP address and DNS, Firewall and TCP Wrapper

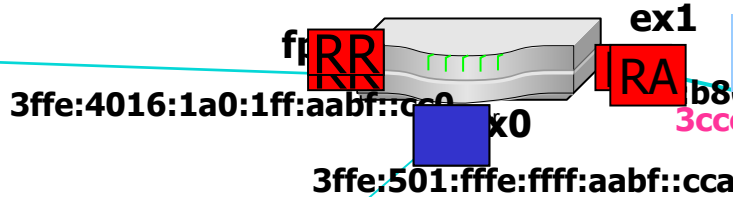
- Add a new address
- Update DNS (Filter rules)
- Set preferred and valid lifetime to be zero
- Delete a old address

Testing Re

Change operation
 Match-prefix: 3ffe:b80:53:1c4:: /64
 Use-prefix: 3ccc:bbbb:cccc:1234:: /64
 Keeper

Set preferred
 and valid lifetime
 Preferred and valid lifetime
 to be zero

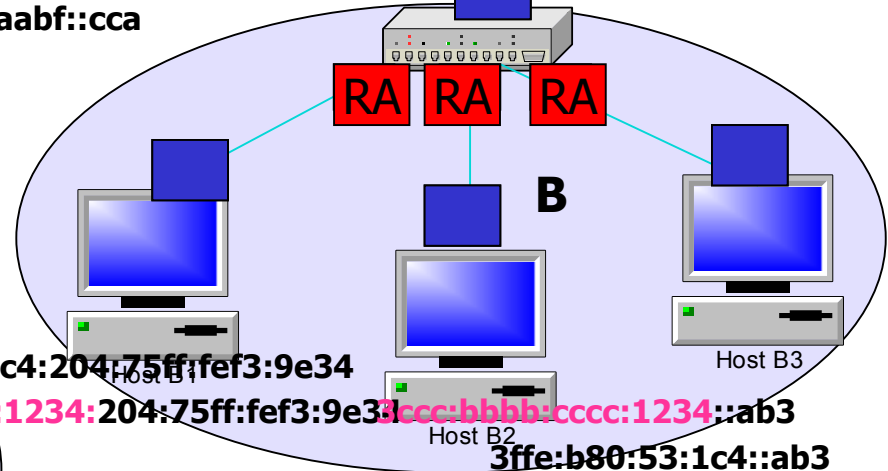
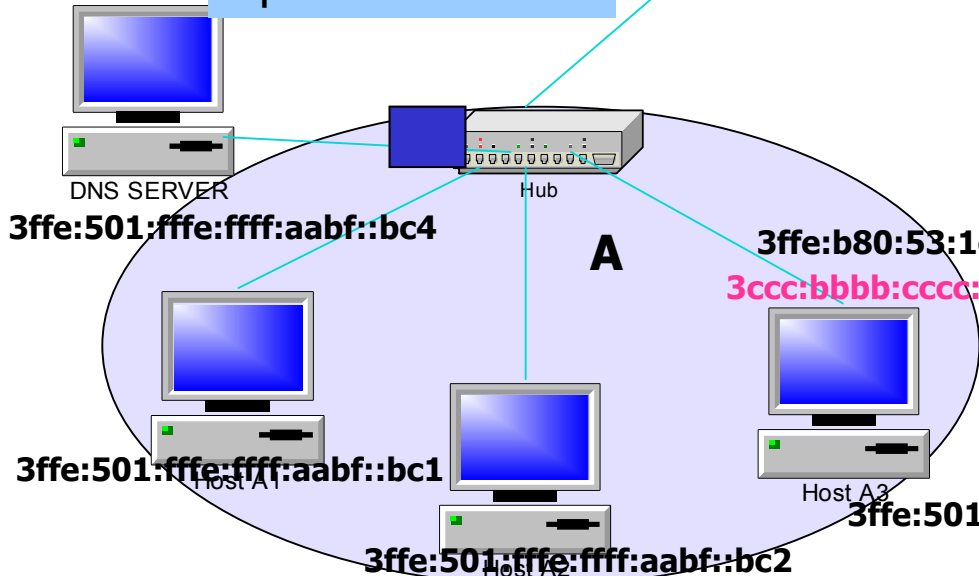
RR request Packet



Send dynamic update request

3ccc:bbbb:cccc:1234:210:5aff:fe68:82ff

Update database



Host B detect have renumbering

3ccc:bbbb:cccc:1234::ab2
 3ffe:b80:53:1c4:ab2

Testing Router Renumbering

```
rtadvd[327]: router renumbering
rtadvd[327]: <rr_command_check> seq.seq 0.0 (last 0 [0.0.0.0.0.0.0.0]) from fe80
::250:baff:fe62:eb46
rtadvd[327]: flags is 16
rtadvd[327]: checklen is 64
rtadvd[327]: interface ex0
rtadvd[327]: addr = fe80:0001:0000:0000:0201:03ff:fe40:8b4f
rtadvd[327]: addr = 3ffe:0b80:0053:ffff:0201:03ff:fe40:8b4f
rtadvd[327]: interface ex1
rtadvd[327]: addr = fe80:0002:0000:0000:0210:5aff:fe68:82ff
rtadvd[327]: addr = 3ffe:0b80:0053:01c4:0210:5aff:fe68:82ff
rtadvd[327]: matching addr = 3ffe:0b80:0053:01c4:0210:5aff:fe68:82ff prefix = 3f
fe-0b80-0053-01c4-0210-5aff-fe68-82ff
rtadvd[327]: address is 3ccc:bbbb:cccc:1234:210:5aff:fe68:82ff
rtadvd[327]: Add address = 3ccc:bbbb:cccc:1234:0210:5aff:fe68:82ff to interface
= ex1
rtadvd[327]: My address = 3ccc:bbbb:cccc:1234:0000:0000:0000:0000 to interface =
ex1
```

Adding address to interface ex1

Figure2 The procedure of the router for check match when renumber.

Testing DNS Renumbering

```
root@linuxtle246:/etc/init.d
:: AUTHORITY SECTION:
cnr.v6.          1M IN NS      kannazung.cnr.v6.

:: ADDITIONAL SECTION:
kannazung.cnr.v6. 1M IN A      172.30.130.132

:: res_findzonecut: add_addrs: 1
:: res_findzonecut: satisfy(kannazung.cnr.v6): 1
:: res_findzonecut: FINISH n=1 (OK)
:: res_nupdate: res_mkupdate -> 58
:: res_send()
:: ->>HEADER<<- opcode: UPDATE, status: NOERROR, id: 46091
:: flags: ZONE: 1, PREREQUISITE: 0, UPDATE: 1, ADDITIONAL: 0
::      cnr.v6, type = SOA, class = IN
host1.cnr.v6.    6d23h59m58s IN AAAA  3ccc:bbbb:cccc:1234:204:75ff:fef3:9
e34

:: Querying server (# 1) address = 172.30.130.132
:: new DG socket
:: got answer:
:: ->>HEADER<<- opcode: UPDATE, status: NOERROR, id: 46091
:: flags: qr ra; ZONE: 0, PREREQUISITE: 0, UPDATE: 0, ADDITIONAL: 0

bash-2.05b#
```

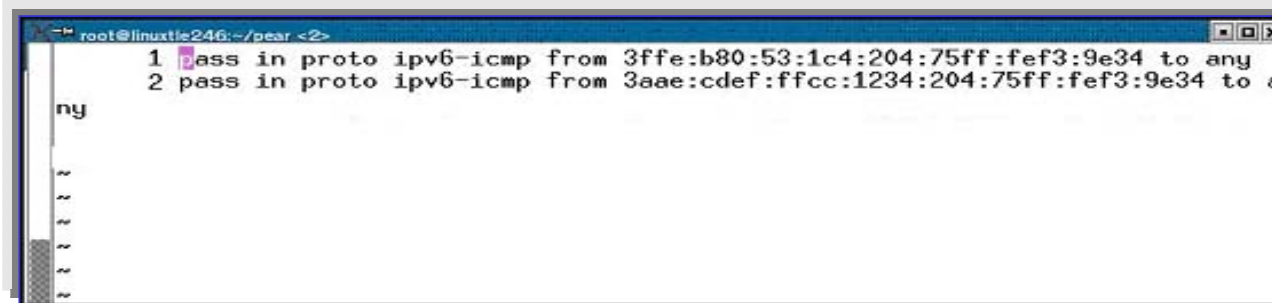
Figure3 The output of testing DNS renumbering.

Testing DNS Renumbering (cont.)

```
root@linuxtle246:/etc/init.d
:: got answer:
:: ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 2
:: flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 1, ADDITIONAL: 2
:: QUERY SECTION:
::   host1.cnr.v6, type = AAAA, class = IN
:: ANSWER SECTION:
host1.cnr.v6.      6d23h59m58s IN AAAA  3ccc:bbbb:cccc:1234:204:75ff:fef3:9
e34
host1.cnr.v6.      6d23h59m58s IN AAAA  3ffe:b80:53:1c4:204:75ff:fef3:9e34
:: AUTHORITY SECTION:
cnr.v6.           1M IN NS      kannazung.cnr.v6.
:: ADDITIONAL SECTION:
kannazung.cnr.v6. 1M IN A       172.30.130.132
kannazung.cnr.v6. 1M IN AAAA    3ffe:b80:53:ffff:204:75ff:fef3:9f46
:: Total query time: 7 msec
:: FROM: host1 to SERVER: default -- 0.0.0.0
:: WHEN: Tue Jan 27 10:35:59 2004
:: MSG SIZE sent: 30 rcvd: 154
bash-2.05b#
```

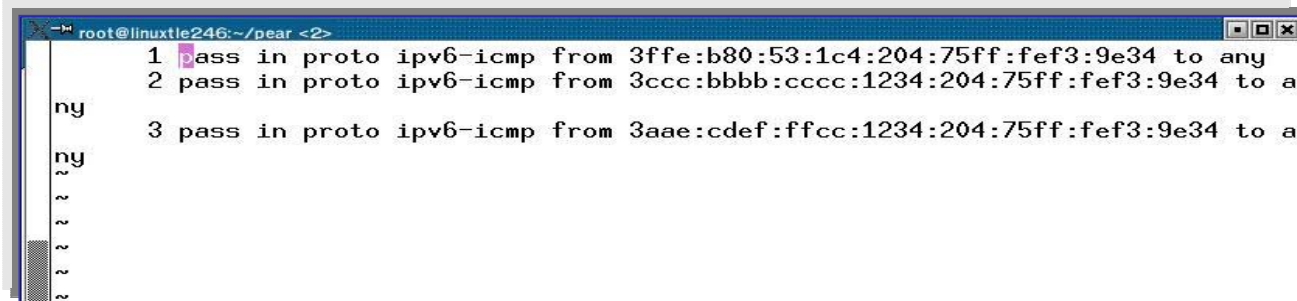
Figure4 The output when query to host1.cnr.v6

Testing Firewall and TCP Wrapper renumbering



```
root@linuxtle246:~/pear <2>
1 pass in proto ipv6-icmp from 3ffe:b80:53:1c4:204:75ff:fe3:9e34 to any
2 pass in proto ipv6-icmp from 3aae:cdef:ffcc:1234:204:75ff:fe3:9e34 to a
ny
~
~
~
~
~
```

Figure5 The rules of firewall before renumbering occurred



```
root@linuxtle246:~/pear <2>
1 pass in proto ipv6-icmp from 3ffe:b80:53:1c4:204:75ff:fe3:9e34 to any
2 pass in proto ipv6-icmp from 3ccc:bbbb:cccc:1234:204:75ff:fe3:9e34 to a
ny
3 pass in proto ipv6-icmp from 3aae:cdef:ffcc:1234:204:75ff:fe3:9e34 to a
ny
~
~
~
~
~
```

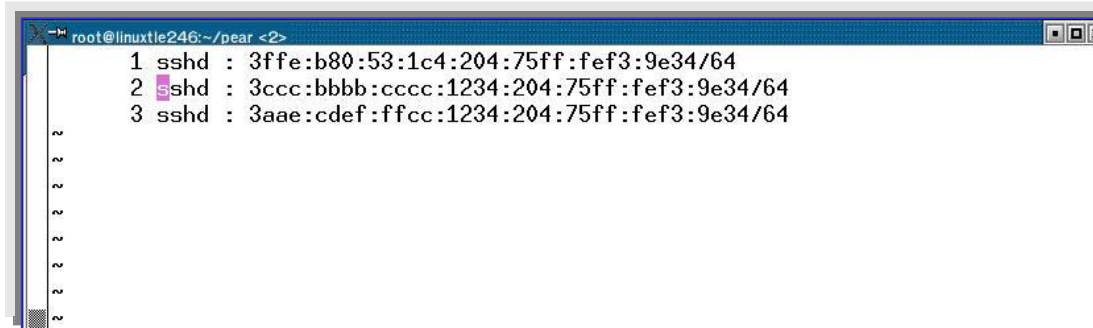
Figure6 The rules of firewall after renumbering occurred

Testing Firewall and TCP Wrapper renumbering (cont.)



```
root@linuxtie246:~/pear <2>
1 sshd : 3ffe:b80:53:1c4:204:75ff:fef3:9e34/64
2 sshd : 3aae:cdef:ffcc:1234:204:75ff:fef3:9e34/64
~
~
~
~
~
~
~
```

Figure7 The rules of TCP Wrapper before renumbering occurred



```
root@linuxtie246:~/pear <2>
1 sshd : 3ffe:b80:53:1c4:204:75ff:fef3:9e34/64
2 sshd : 3ccc:bbbb:cccc:1234:204:75ff:fef3:9e34/64
3 sshd : 3aae:cdef:ffcc:1234:204:75ff:fef3:9e34/64
~
~
~
~
~
~
~
```

Figure8 The rules of TCP Wrapper after renumbering occurred

Testing Firewall and TCP Wrapper renumbering (cont.)

```
root@linuxtle246:/etc/init.d
kannazung.cnr.v6.      1M IN AAAA      3ffe:b80:53:ffff:204:75ff:fef3:9f46

;; AUTHORITY SECTION:
cnr.v6.                1M IN NS         kannazung.cnr.v6.

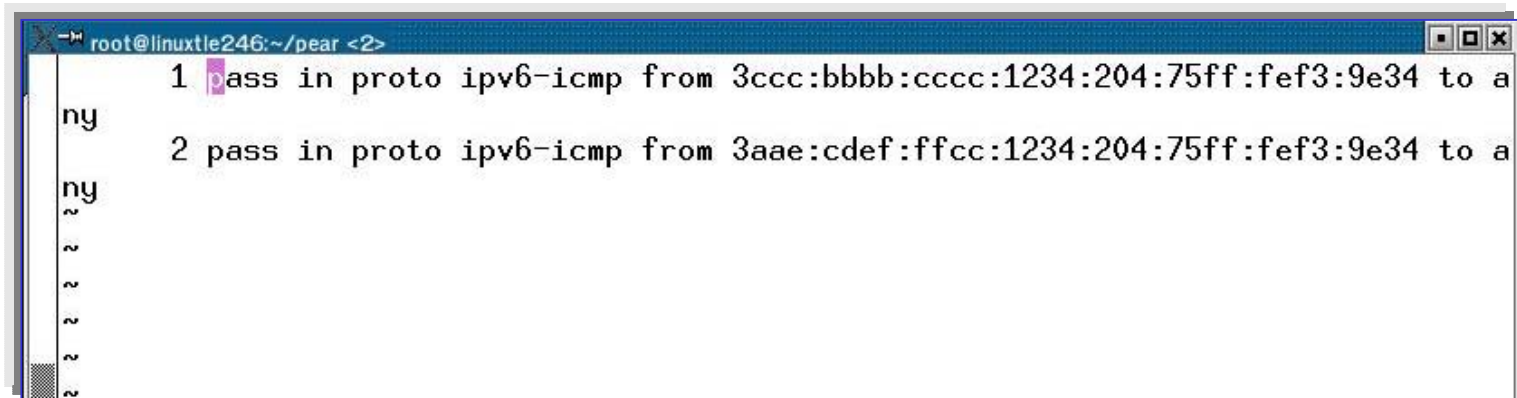
;; ADDITIONAL SECTION:
kannazung.cnr.v6.     1M IN A         172.30.130.132

;; res_findzonecut: add_addrs: 1
;; res_findzonecut: satisfy(kannazung.cnr.v6): 1
;; res_findzonecut: FINISH n=1 (OK)
;; res_nupdate: res_mkupdate -> 58
;; res_send()
;; ->>HEADER<<- opcode: UPDATE, status: NOERROR, id: 56701
;; flags:: ZONE: 1, PREREQUISITE: 0, UPDATE: 1, ADDITIONAL: 0
;;      cnr.v6, type = SOA, class = IN
host1.cnr.v6.         0S NONE AAAA    3ffe:b80:53:1c4:204:75ff:fef3:9e34
;; Querying server (# 1) address = 172.30.130.132
;; new DG socket
;; got answer:
;; ->>HEADER<<- opcode: UPDATE, status: NOERROR, id: 56701
;; flags: qr ra; ZONE: 0, PREREQUISITE: 0, UPDATE: 0, ADDITIONAL: 0

bash-2.05b#
```

Figure9 The information after update database, when life time to be zero

Testing Firewall and TCP Wrapper renumbering (cont.)



```
root@linuxtle246:~/pear <2>
1 pass in proto ipv6-icmp from 3ccc:bbbb:cccc:1234:204:75ff:fef3:9e34 to a
ny
2 pass in proto ipv6-icmp from 3aae:cdef:ffcc:1234:204:75ff:fef3:9e34 to a
ny
~
~
~
~
~
~
```

Figure10 Updating the filter firewall when the old rule was deleted.



Conclusions

- Discussed the router renumbering protocol and process of renumbering IPv6 network
 - Host and Router
 - DNS
 - TCP Wrapper and Firewall
- Decreasing the task of system administrator for renumbering



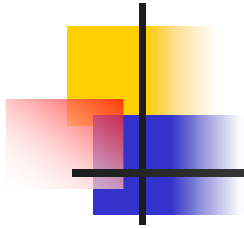
References

- Chritian Huitema, “The IPv6 Header Format”, *IPv6: The New Internet Protocol, Prentice Hall PTR, pp.60-62, 1996.*
- M.Crawford, “Router Renumbering for IPv6”, *RFC 2894, August 2000.*
- H. Berkowitz, “Router Renumbering Guide”, January 1997.
- KAME-snap kit, www.kame.net/snap-users.



References (Cont.)

- W. Simpson Daydream, “Neighbor Discovery for IP Version 6 (IPv6)”, *RFC2461*, December 1998.
- M. Crawford, “DNS extension to support IPv6 address aggregation and renumbering”, *RFC2874*, July 2000.
- S. Thomson, “IPv6 Stateless Address Autoconfiguration”, *RFC 2462*, December 1998.



Question and Comment

Experiment Setup

