CNGI-CERNET2 SAVI Deployment Update

China Education and Research Network (CERNET)
/Tsinghua Univ.

IETF77, Anaheim

March 23, 2010

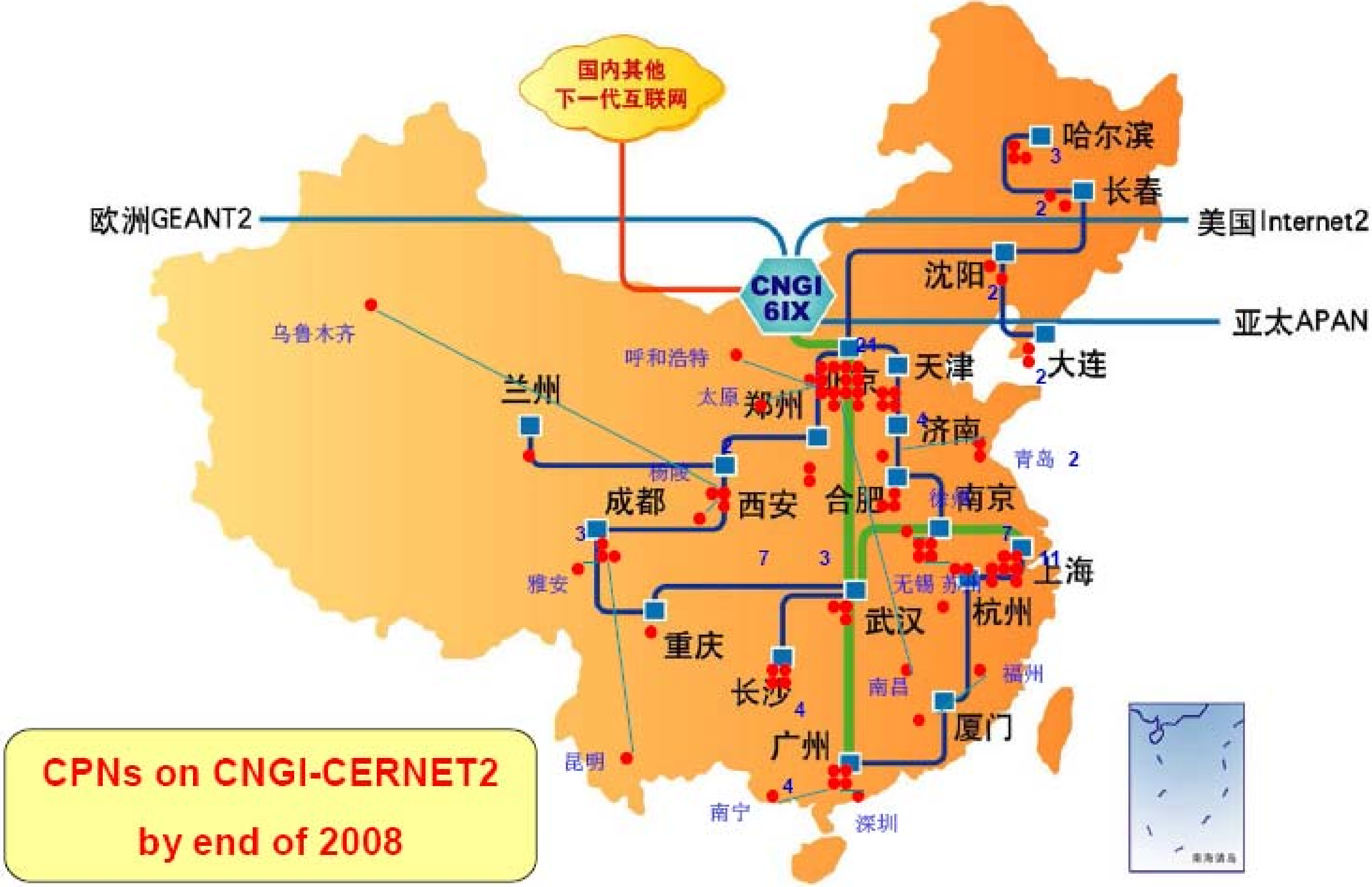
Outline

- SAVI Deployment in CNGI-CERNET2
- SAVI Switches Testing
- SAVI Management System and MIB Design
- Discussion on SAVI-SLAAC
- Conclusion

Brief Introduction

- CNGI is China Next Generation Internet
- CNGI-CERNET2
  - CERNET: was the 2nd Large ISP in China, 2000+ university campus networks, 20M+ users
  - CERNET2 is the largest IPv6 network
- CNGI-CERNET2 SAVI Deployment Plan
  - 100 universities campus networks nationwide
  - 1 Million users
  - Time frame: 2008-2010
  - SAVI software upgrade at about 20K+ access switches
  - SAVI management system installation in 100 campuses
- China Telecom signed collaboration agreement with Tsinghua Univ. on IPv6 SAVI collaboration recently

SAVI switches installation:100 Univ. campus net (red dot)

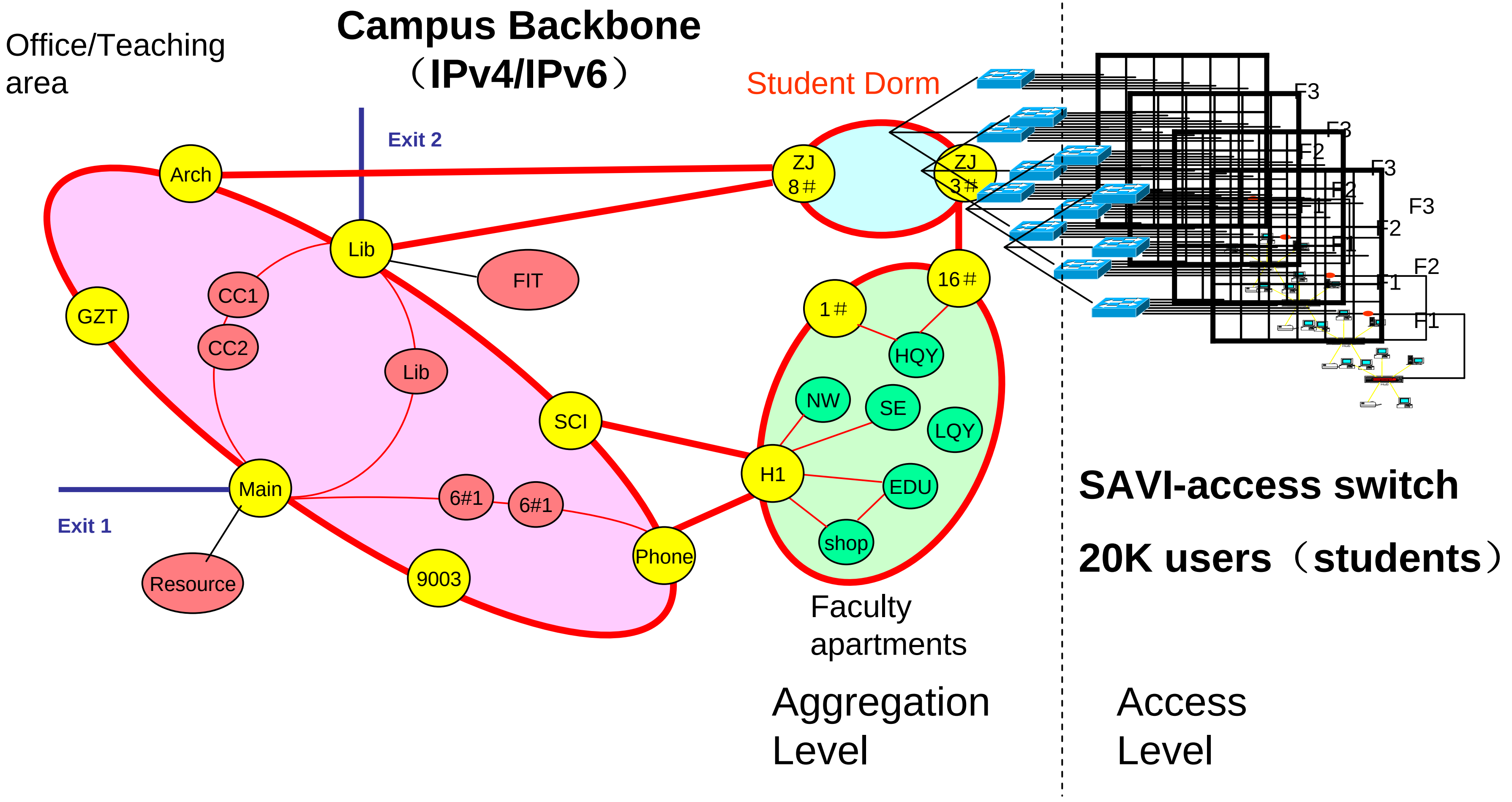


CPNs on CNGI-CERNET2

by end of 2008

Example: Tsinghua Univ. campus network is being deployed (software upgrade at access switch)

subnets	switches	port	hosts	users
114	1018	23414	22644	20280



SAVI-access switch

20K users（students）

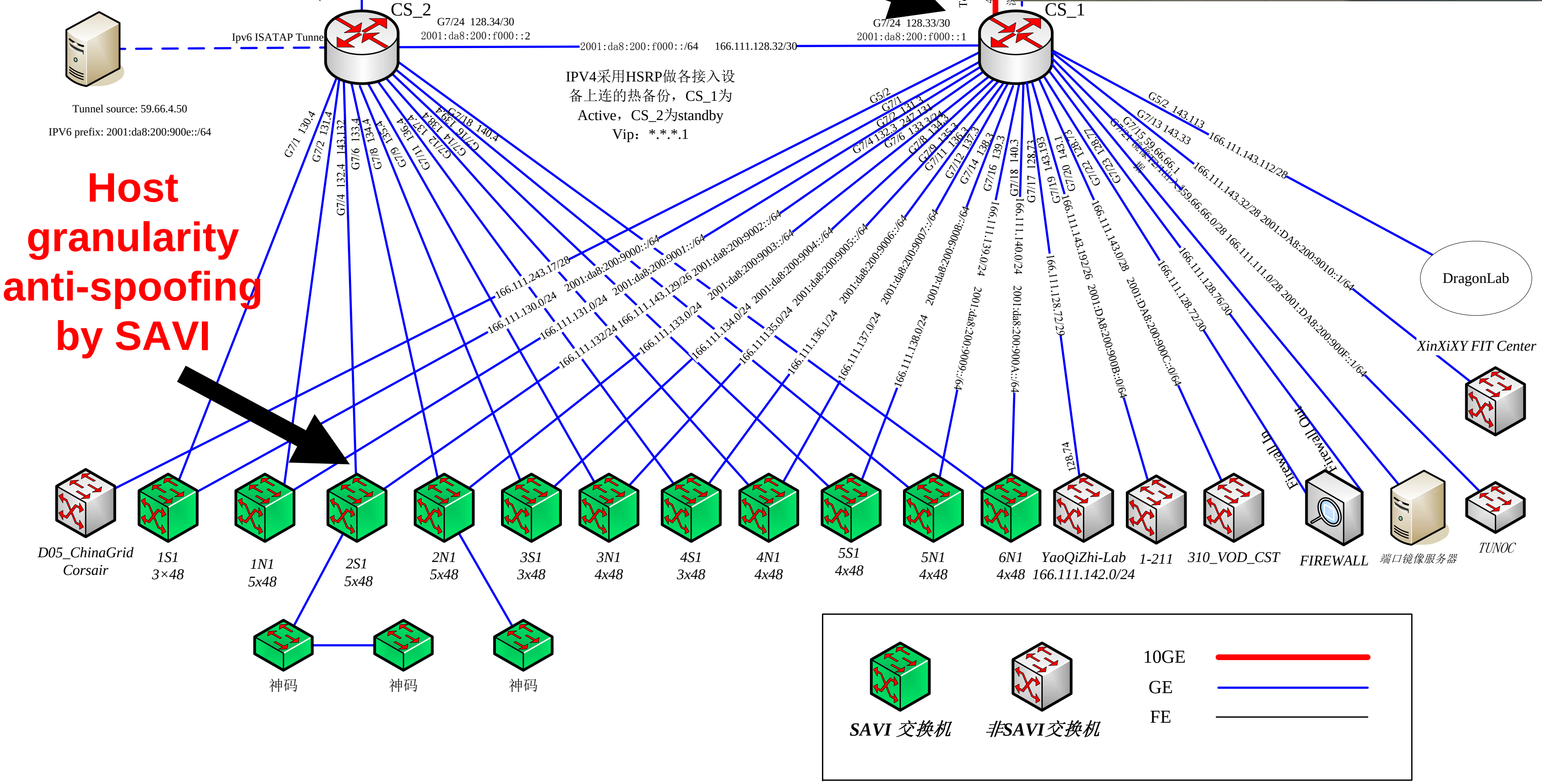
Aggregation Level

Access Level

Example: SAVI deployment in Tsinghua FIT building

Prefix granularity anti-spoofing by RPF

Host granularity anti-spoofing by SAVI



Scenarios in Deployment

- DHCP-only
  - Only DHCP and link local address are allowed.
  - DHCP and link local address snooping are enabled.
- SLAAC-only
  - Only SLAAC address is allowed.
  - SLAAC snooping is enabled.
- DHCP-SLAAC-Mixed
  - DHCP and SLAAC address are allowed.
  - DHCP snooping and SLAAC snooping are enabled.
- Static addresses (usually for servers) are manually configured in the above scenarios.

Scenarios in Deployment

- Each administrator selects the address assignment scenario in its subnet
  - E.g. Tsinghua uses dhcp-slaac-mixed
- SEND is considered the same as SLAAC
- dhcp-snooping implementation in switch conforms to draft-savi-dhcp-02 (without optional functions)
- slaac-snooping implmentation in switch conforms to draft-bi-stateless-00
  - Will be discussed in the last part of this ppt
- All SAVI-switches have been tested
  - Will be discussed in the next part of this ppt

Prioritization

- Static address has the highest prior
  - The administrator make sure the static address won’t be assigned by dhcp server
  - Only the administrator can remove
- Stateless and DHCP addresses are treated equally.
  - Once bound, always bound during lifetime (unless the host is off-link)
  - A host has to detect conflict after assigned an address by DHCP (in dhcp-slaac-mix scenario)

Command Line Design

- **Snooping**
  - Enabled at global view or vlan view
- Command line: *XXX Snooping enable*
  - Start snooping and binding
  - Drop the server-end message(DHCP reply, RA) by default, except for packets from anchor with attribute XXX-Trust
- For example, in DHCP-only senario:
  - *Dhcp snooping enable*
  - *NDP snooping link-local enable*
- Undo XXX snooping
  - Stop snooping
  - Stop filter server-end message
- SHOULD write memory if snooping is enabled, and enable snooping automatically after reboot.

Command Line Design

- **Verification**
  - Enabled at port view
  - *IP check source IP-address*

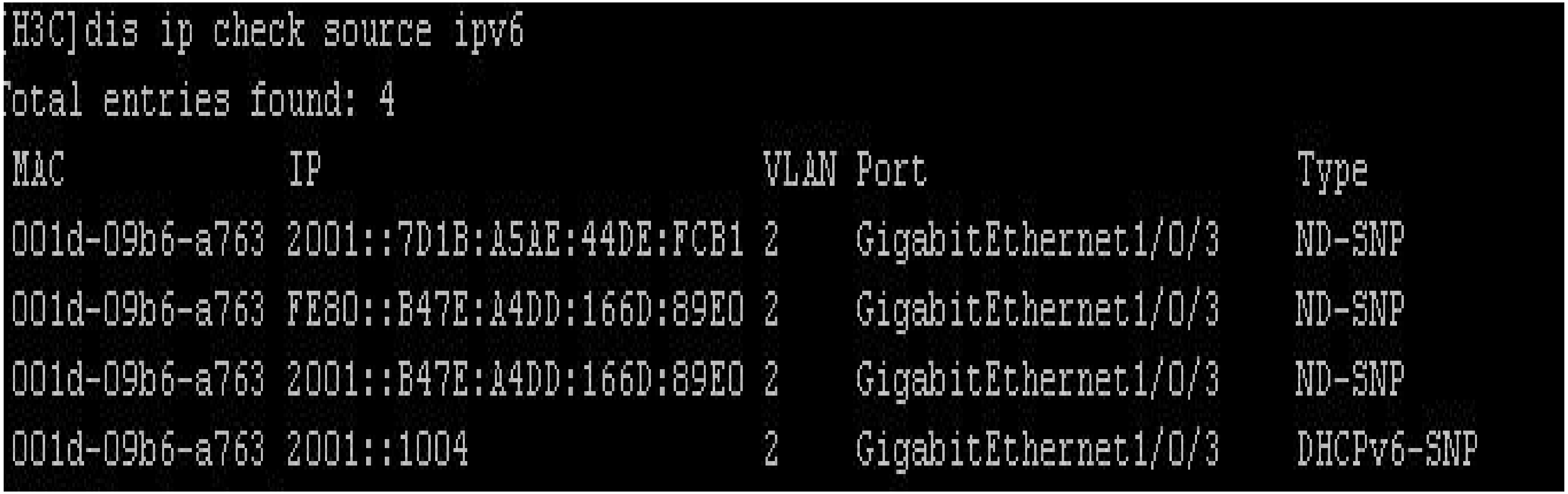
Command Line Design

- Port configuration
- Attached to monitored host
  - *IP check source IP-address*
- Attached to router or DHCP server/relay
  - *RA trust* or *DHCP trust*
- Fully trusted port
  - *RA trust* and *DHCP trust*
- Default port
  - No configuration

Command Line Design

- **View & Modification**
  - **At global view**
- **View:** show all the IPv6 bindings
  - *display ipv6 check source binding table*
- **Modification:** add or del bindings manually
  - *ipv6 check source binding table add IP XXX MAC XXX PORT XXX TYPE XXX [LIFETIME XXX]*
  - *Ipv6 check source binding table del IP XXX PORT XXX*

Console Example



```
[H3C]dis ip check source ipv6
Total entries found: 4
 MAC            IP                           VLAN Port                 Type
 001d-09b6-a763 2001::7D1B:A5AE:44DE:FCB1    2    GigabitEthernet1/0/3    ND-SNP
 001d-09b6-a763 FE80::B47E:A4DD:166D:89E0    2    GigabitEthernet1/0/3    ND-SNP
 001d-09b6-a763 2001::B47E:A4DD:166D:89E0    2    GigabitEthernet1/0/3    ND-SNP
 001d-09b6-a763 2001::1004                   2    GigabitEthernet1/0/3    DHCPv6-SNP
```

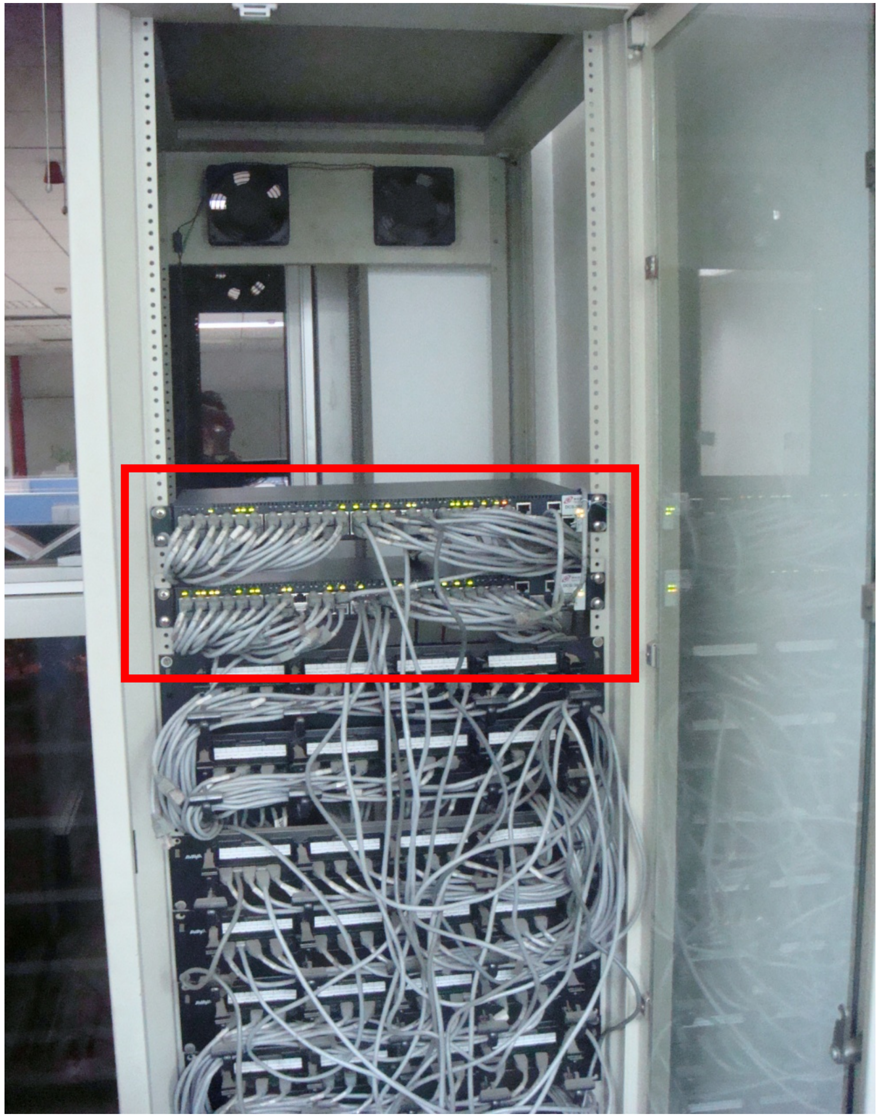
Binding State Table of *H3C S5500*

Entry:

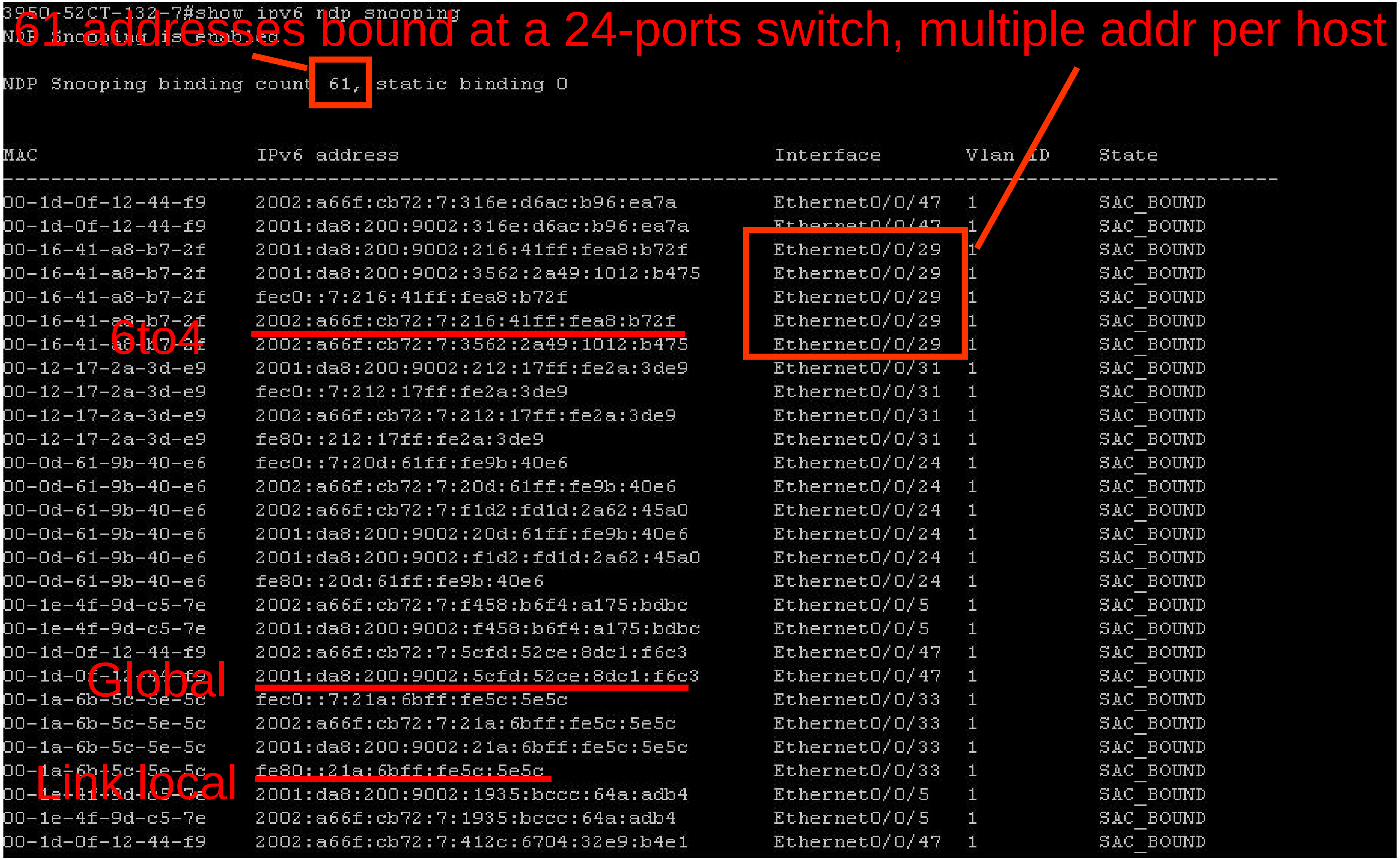
Source IP | Source MAC | Vlan ID | Type(DHCP or ND)

Real Deployment

- FIT Building of Tsinghua Univ
- From Oct 2009 (about 5 months)
- No initial DAD-NS loss observed (link local addr bound)
- *Digital China* *S3950* Switches



Real Deployment



```
3950-52CT-132-7#show ipv6 ndp snooping
NDP Snooping is enabled

NDP Snooping binding count 61, static binding 0

MAC                  IPv6 address                           Interface       Vlan ID    State
-----------------------------------------------------------------------------------------------------------
00-1d-0f-12-44-f9    2002:a66f:cb72:7:316e:d6ac:b96:ea7a    Ethernet0/0/47  1          SAC_BOUND
00-1d-0f-12-44-f9    2001:da8:200:9002:316e:d6ac:b96:ea7a   Ethernet0/0/47  1          SAC_BOUND
00-16-41-a8-b7-2f    2001:da8:200:9002:216:41ff:fea8:b72f   Ethernet0/0/29  1          SAC_BOUND
00-16-41-a8-b7-2f    2001:da8:200:9002:3562:2a49:1012:b475  Ethernet0/0/29  1          SAC_BOUND
00-16-41-a8-b7-2f    fec0::7:216:41ff:fea8:b72f             Ethernet0/0/29  1          SAC_BOUND
00-16-41-a8-b7-2f    2002:a66f:cb72:7:216:41ff:fea8:b72f    Ethernet0/0/29  1          SAC_BOUND
00-16-41-a8-b7-2f    2002:a66f:cb72:7:3562:2a49:1012:b475   Ethernet0/0/29  1          SAC_BOUND
00-12-17-2a-3d-e9    2001:da8:200:9002:212:17ff:fe2a:3de9   Ethernet0/0/31  1          SAC_BOUND
00-12-17-2a-3d-e9    fec0::7:212:17ff:fe2a:3de9             Ethernet0/0/31  1          SAC_BOUND
00-12-17-2a-3d-e9    2002:a66f:cb72:7:212:17ff:fe2a:3de9    Ethernet0/0/31  1          SAC_BOUND
00-12-17-2a-3d-e9    fe80::212:17ff:fe2a:3de9               Ethernet0/0/31  1          SAC_BOUND
00-0d-61-9b-40-e6    fec0::7:20d:61ff:fe9b:40e6             Ethernet0/0/24  1          SAC_BOUND
00-0d-61-9b-40-e6    2002:a66f:cb72:7:20d:61ff:fe9b:40e6    Ethernet0/0/24  1          SAC_BOUND
00-0d-61-9b-40-e6    2002:a66f:cb72:7:f1d2:fd1d:2a62:45a0   Ethernet0/0/24  1          SAC_BOUND
00-0d-61-9b-40-e6    2001:da8:200:9002:20d:61ff:fe9b:40e6   Ethernet0/0/24  1          SAC_BOUND
00-0d-61-9b-40-e6    2001:da8:200:9002:f1d2:fd1d:2a62:45a0  Ethernet0/0/24  1          SAC_BOUND
00-0d-61-9b-40-e6    fe80::20d:61ff:fe9b:40e6               Ethernet0/0/24  1          SAC_BOUND
00-1e-4f-9d-c5-7e    2002:a66f:cb72:7:f458:b6f4:a175:bdbc   Ethernet0/0/5   1          SAC_BOUND
00-1e-4f-9d-c5-7e    2001:da8:200:9002:f458:b6f4:a175:bdbc  Ethernet0/0/5   1          SAC_BOUND
00-1d-0f-12-44-f9    2002:a66f:cb72:7:5cfd:52ce:8dc1:f6c3   Ethernet0/0/47  1          SAC_BOUND
00-1d-0f-12-44-f9    2001:da8:200:9002:5cfd:52ce:8dc1:f6c3  Ethernet0/0/47  1          SAC_BOUND
00-1a-6b-5c-5e-5c    fec0::7:21a:6bff:fe5c:5e5c             Ethernet0/0/33  1          SAC_BOUND
00-1a-6b-5c-5e-5c    2002:a66f:cb72:7:21a:6bff:fe5c:5e5c    Ethernet0/0/33  1          SAC_BOUND
00-1a-6b-5c-5e-5c    2001:da8:200:9002:21a:6bff:fe5c:5e5c   Ethernet0/0/33  1          SAC_BOUND
00-1a-6b-5c-5e-5c    fe80::21a:6bff:fe5c:5e5c               Ethernet0/0/33  1          SAC_BOUND
00-1e-4f-9d-c5-7e    2001:da8:200:9002:1935:bccc:64a:adb4   Ethernet0/0/5   1          SAC_BOUND
00-1e-4f-9d-c5-7e    2002:a66f:cb72:7:1935:bccc:64a:adb4    Ethernet0/0/5   1          SAC_BOUND
00-1d-0f-12-44-f9    2002:a66f:cb72:7:412c:6704:32e9:b4e1   Ethernet0/0/47  1          SAC_BOUND
```

61 addresses bound at a 24-ports switch, multiple addr per host

6to4

Global

Link local

SAVI Switch Testing

SAVI-Software upgradable

- Savi-upgradable switches in our deployment
  - H3C (3Com): S5500EI, S5500SI, S5120EI、E126A, E152, E328, E352
  - ZTE: ZXR10 8900,5900,3900A
  - Digital China (spun off from Lenovo): DCRS-5950,3950
  - Ruijie: RG-S8600,S5750,S5760,S2900,S2600
  - Bitway: BitStream 7000, 6000, 3000
  - Centec: E600 and E300
- Cisco and Huawei are also interested to collaborate with CERENT2 to upgrade

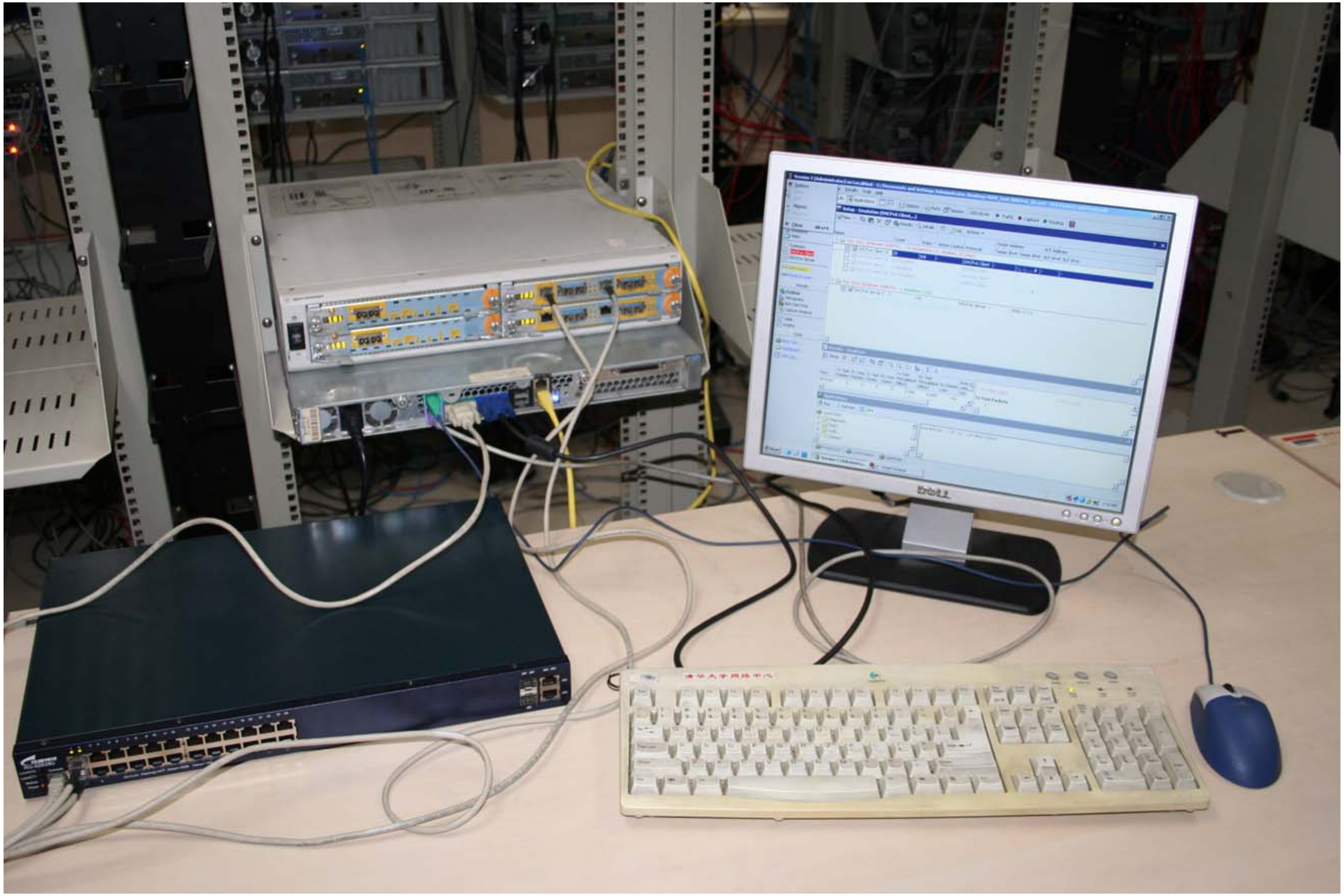
SAVI switch test for 100 campus networks

Catalogs of SAVI Testing

- Conformance testing
- Performance testing
- Test-bed (interoperability) testing

SAVI Switch under Test
(form difference vendors)

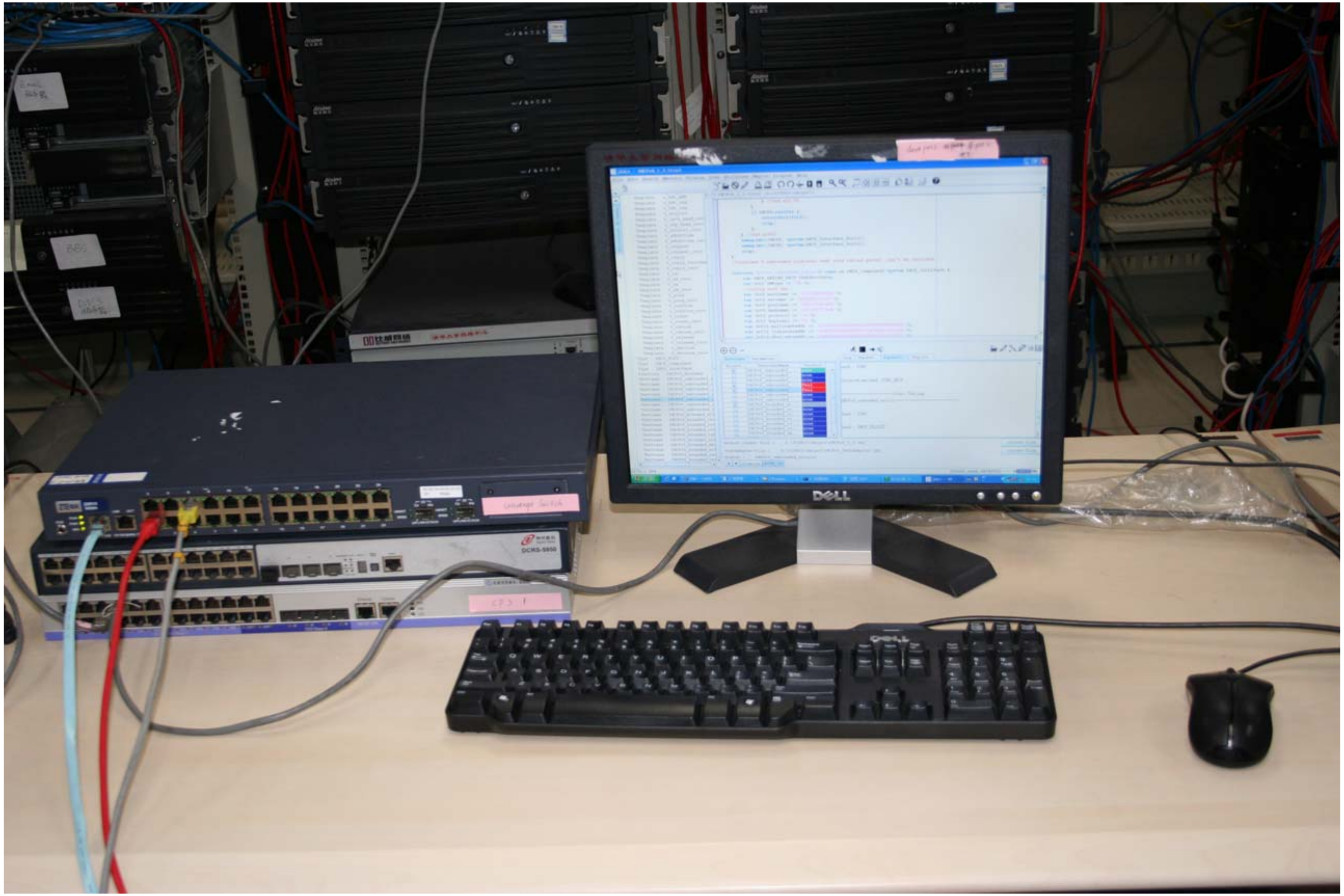
Performance Testing (AGILENT N2X)



Performance Testing: SAVI filtering enabled for dhcpv6/slaac/mixed/static

Throughput	78bytes	-
	79bytes	-
	512bytes	-
	1518bytes	-
Delay (Min/Average/Max)	78bytes	μs
	79bytes	μs
	512bytes	μs
	1518bytes	μs
Packet loss	78bytes	-
	79bytes	-
	512bytes	-
	1518bytes	-

Conformance Testing (TTCN3 based testing system developed by Tsinghua)



Conformance Testing: DHCP-only

2.1.1	DHCP Solicit	Use unbounded link-local addr send DHCP-Solicit
2.1.2	DHCP Solicit-Advertise	Use bounded link-local addr send DHCP-Solicit then receive Advertise
2.1.3	DHCP Request	Use unbounded link-local addr send DHCP-Request
2.1.4	DHCP Request-Reply	Use bounded link-local addr send DHCP-Request then received reply
2.1.5	DHCP Confirm	Use unbounded link-local addr send DHCP Confirm
2.1.6	DHCP Confirm-Reply	Use bounded link-local addr send DHCP Confirm then received reply
2.1.7	DHCP Decline	Use bounded and unbounded link-local addr send DHCP Decline
2.1.8	DHCP Release	Use bounded and unbounded link-local addr send DHCP Release
2.1.9	DHCP Rebind	Use bounded and unbounded link-local addr send DHCP Rebind
2.1.10	DHCP Renew	Use bounded and unbounded link-local addr send DHCP Renew

Conformance Testing: SLAAC-only

2.2.1	LinkLocalAddr_DAD-NS	Send DAD-NS Use LinkLocal Addr as Target
2.2.2	LinkLocalAddr_DAD-NS_NA	Send DAD-NS Use LinkLocal Addr as Target and received NA
2.2.3	LinkLocalAddr-RS	Use bounded and unbouneded link-local addr send SLAAC RS
2.2.4	Global Addr-DAD-NS	Use unbounded and bounded Global addr send DAD NS without receivd NA.
2.2.5	Global Addr-DAD-NS-NA	Use unbounded and bounded Global addr send DAD NS then receivd NA

Conformance Testing: DHCP-SLAAC-MIX

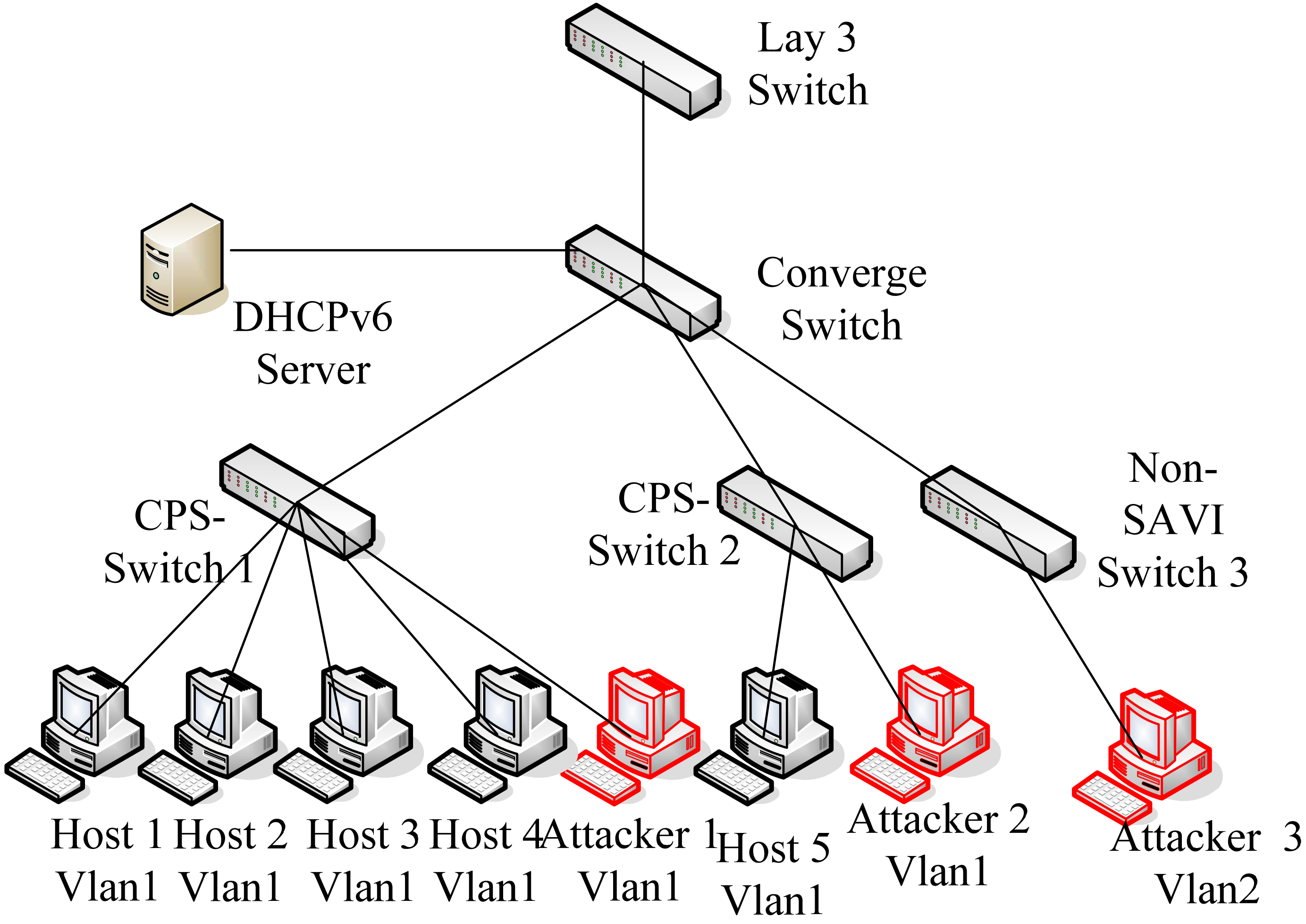
2.3.1	DHCP Request	Send DHCP Request use bounded and unbounded addr Under MIX
2.3.2	DHCP-DAD-NS	Send DHCP Request then send DAD NS use Bounded and unbounded addr without received NA
2.3.3	DHCP-DAD-NS-NA	Send DHCP Request then send DAD NS use Bounded and unbounded addr with received NA
2.3.4	DHCP-Confirm-NS	Send DHCP Confirm then send DAD NS use Bounded and unbounded addr without received NA
2.3.5	DHCP-Confirm-NS-NA	Send DHCP Confirm then send DAD NS use Bounded and unbounded addr with received NA

Conformance Testing: Static address

2.4.1	Static Binding	Check static Binding’s function

Test-bed (interoperability) testing

Test-bed (interoperability) testing



Testbed testing: DHCPv6-only

- Host movement (across ports in one switch)
- Host movement (across switches)
- Topology change (switch uplinks to another port of the upstream switch)
- Topology change (switch uplinks to another upstream switch)
- Switch reboot
- NDP can not setup binding
- Address conflict（within one switch）
- Address conflict (across switch)
- Static address binding in dhcp-only scenario

Testbed testing: SLAAC-only

- Host movement (across ports in one switch)
- Host movement (across switches)
- Topology change (switch uplinks to another port of the upstream switch)
- Topology change (switch uplinks to another upstream switch)
- Switch reboot
- DHCP can not setup binding
- Address conflict（within one switch）Address conflict (across switch)
- Static address binding in slaac-only scenario

Testbed testing: DHCP-SLAAC-mix

- Host movement (across ports in one switch)
- Host movement (across switches)
- Topology change (switch uplinks to another port of the upstream switch)
- Topology change (switch uplinks to another upstream switch)
- Switch reboot
- DHCP and SLAAC co-existence
- Address conflict（within one switch）Address conflict (across switch)
- Static address binding in dhcp-slaac-mix scenario

Interoperability test for host OS

- Windows XP with SP3
- Windows Vista
- Windows 7
- Linux
- MAC OS (to be tested)
- Some dhcpv6 client software

SAVI Management System and MIB Design

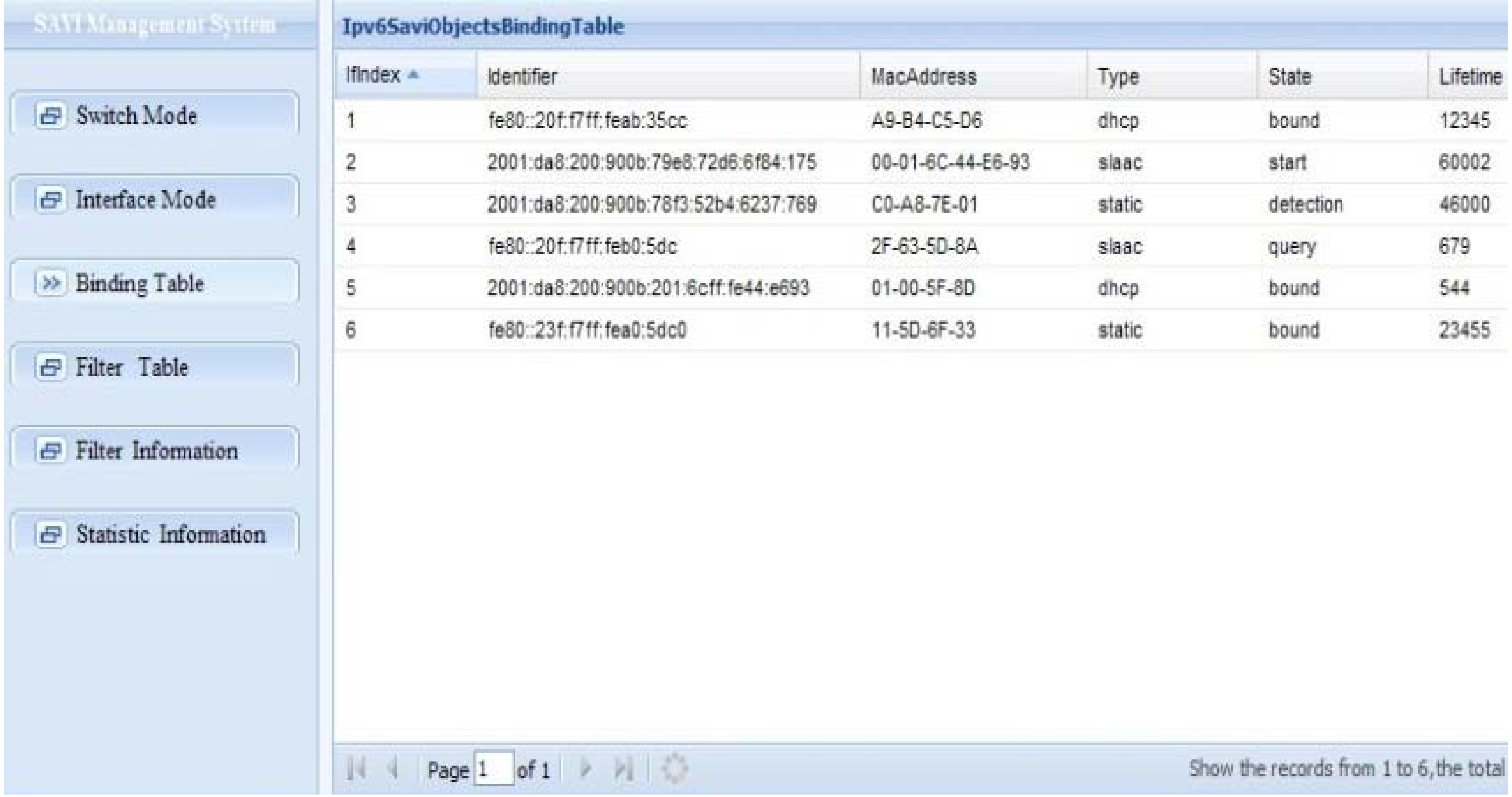
Motivation

- The CERNET Network Center is designing a Network management system for SAVI
- Set and Get SAVI status using standard management protocol like SNMP
- Provide standard operation interface for manager

Function

- Set :
  - SAVI-DHCP or SAVI-SLAAC function
  - Anchor (switch port) type
  - Binding limitation of anchor
- Get:
  - Binding State Table entries
  - Filtering Table entries
  - Statistics

CERNET2 SAVI Management System



Ipv6SaviObjectsBindingTable

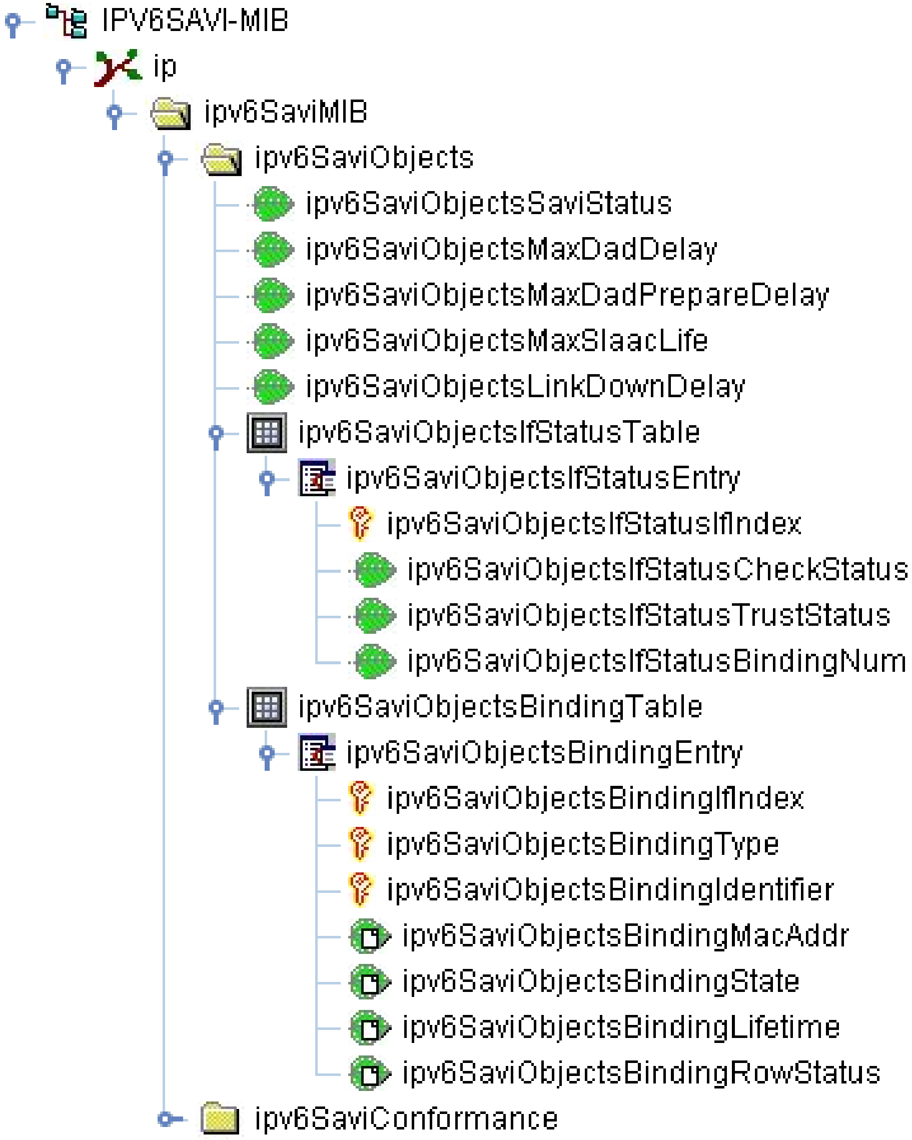
IfIndex ▲	Identifier	MacAddress	Type	State	Lifetime
1	fe80::20f:f7ff:feab:35cc	A9-B4-C5-D6	dhcp	bound	12345
2	2001:da8:200:900b:79e8:72d6:6f84:175	00-01-6C-44-E6-93	slaac	start	60002
3	2001:da8:200:900b:78f3:52b4:6237:769	C0-A8-7E-01	static	detection	46000
4	fe80::20f:f7ff:feb0:5dc	2F-63-5D-8A	slaac	query	679
5	2001:da8:200:900b:201:6cff:fe44:e693	01-00-5F-8D	dhcp	bound	544
6	fe80::23f:f7ff:fea0:5dc0	11-5D-6F-33	static	bound	23455

Show the records from 1 to 6,the total

Structure of SAVI-MIB

- Two separate MIB tree
  - IPV4SAVI-MIB for IPv4
  - IPV6SAVI-MIB for IPv6
  - They have Similar Structure
- Following we illustrate IPV6SAVI-MIB

MIB tree



- IPV6SAVI-MIB
  - ip
    - ipv6SaviMIB
      - ipv6SaviObjects
        - ipv6SaviObjectsSaviStatus
        - ipv6SaviObjectsMaxDadDelay
        - ipv6SaviObjectsMaxDadPrepareDelay
        - ipv6SaviObjectsMaxSlaacLife
        - ipv6SaviObjectsLinkDownDelay
        - ipv6SaviObjectsIfStatusTable
          - ipv6SaviObjectsIfStatusEntry
            - ipv6SaviObjectsIfStatusIfIndex
            - ipv6SaviObjectsIfStatusCheckStatus
            - ipv6SaviObjectsIfStatusTrustStatus
            - ipv6SaviObjectsIfStatusBindingNum
        - ipv6SaviObjectsBindingTable
          - ipv6SaviObjectsBindingEntry
            - ipv6SaviObjectsBindingIfIndex
            - ipv6SaviObjectsBindingType
            - ipv6SaviObjectsBindingIdentifier
            - ipv6SaviObjectsBindingMacAddr
            - ipv6SaviObjectsBindingState
            - ipv6SaviObjectsBindingLifetime
            - ipv6SaviObjectsBindingRowStatus
      - ipv6SaviConformance

Structure of IPV6SAVI-MIB

- ipv6SaviObjectsStatus
  - SAVI-DHCP/SAVI-SLAAC Status
- ipv6SaviObjectsMaxDadDelay, ipv6SaviObjectsMaxDadPrepareDelay,
  - constants of SAVI
- ipv6SaviObjectsIfStatusTable
  - Validation type of anchor
  - Trust type of anchor
  - Binding limitation of anchor
- ipv6SaviObjectsBindingTable
  - Binding State Table entries

Structure of IPV6SAVI-MIB

- ipv6SaviObjectsIfStatusTable
  - ipv6SaviObjectsIfStatusIfIndex InterfaceIndex,
  - ipv6SaviObjectsIfStatusCheckStatus Integer32,
  - ipv6SaviObjectsIfStatusTrustStatus Integer32,
  - ipv6SaviObjectsIfStatusBindingNum Unsigned32

Structure of IPV6SAVI-MIB

- ipv6SaviObjectsBindingTable
  - ipv6SaviObjectsBindingIfIndex   InterfaceIndex,
  - ipv6SaviObjectsBindingType   Integer32,
  - ipv6SaviObjectsBindingIdentifier InetAddressIPv6,
  - ipv6SaviObjectsBindingMacAddr   MacAddress,
  - ipv6SaviObjectsBindingState   Integer32,
  - ipv6SaviObjectsBindingLifetime   TimeInterval,
  - ipv6SaviObjectsBindingRowStatus   RowStatus

OID For SAVI-MIB

- Parent OID: IP
  - Because SAVI-MIB provide binding information at IP layer.
- sub-identifier
  - The sub-identifier of IP has been used up to 39.
  - 40 for IPV4SAVI-MIB
  - 41 FOR IPV6SAVI-MIB
- Need register a IANA NUMBER for the SAVI MIB

Discussion on SAVI-SLAAC

Solution Scope

- Solution for all stateless addresses, including
  - IPv6 SLAAC address
  - IPv4/v6 non-static manually configured address

Core problem for SAVI-SLAAC

- How to determine the ownership of an address when conflict happens?
- On the aspect of host:
  - DAD is unreliable: NS/NA loss, inactive node, malicious node
- On the aspect of SAVI-device:
  - It is hard or even impossible to determine who is the first to use an address without reliable DAD:
    - First sniffed $\neq$ First used
    - Detection is unreliable, and may be cheated

A Compromise Solution without Reliable DAD

- Principle:
  - RFC4862 allows host to configure an address after it finishes a DAD, without caring the address might be actually conflict with other hosts due to unreliable DAD (NS/NA loss, inactive node, etc.)
  - Then the goal of SAVI-SLAAC conforms to RFC4862, like “best effort” source address validation
  - Don’t try to fix problem of RFC4862 in SAVI, if necessary, fix it in SLAAC itself (re-chartering)

Binding Set-up Mechanism

- If SAVI switch detects an node finishes a successful DAD by Control plane snooping, then bind the address
- The initial DAD-NS might be loss, two options
  - Data-triggered probe (heavy cost to access switch but automatic), or
  - Host repairs the network connection (CERNET2 use this option, but really didn’t meet this problem)
- An address might be bound with multiple nodes due to the unreliable DAD (e.g. inactive node, NA loss), but RFC4862 allows

Binding Removal Mechanism

- Only remove a binding:
  - Lifetime expires (Lifetime equals prefix lifetime sniffed from RA)
  - After the savi-device detects the anchor turns off-link for a certain period (when savi-device directly connects to host)

Control Plane Snooping based action vs. Data Triggered action

- Control packet snooping MUST be enabled
- Data trigger action CAN be enabled on the required anchors to handle special cases
  - The trade-off between savi-swtich-automaticly or host-manually repairs for special cases is left to network administrator
  - CNGI-CERNET2 make it an optional function. if administrators need, then can ask higher-end switch to implement the optional function

Experience of CERNET2 SAVI-SLAAC deployment

- Make SAVI solutions as simple as possible
  - low end access switch can implement by simply software upgrade
- Then SAVI can be deployed widely at access switches directly connects with host
- Then get the better “best effort” results
  - single-host granularity anti-spoofing
  - easily handle the binding removal when host off-link or moving
  - easily handle switch rebooting

Experience of CERNET2 SAVI-SLAAC deployment

- Data triggered binding brings much cost to switch based on feedbacks from vendors
  - More temporal states to keep and memory occupation
  - Consume more CPU computation resource
  - Potentially DoS attacks
    - Hard to do rate limit in reality
    - If do rate limit for CPU slow path in a switch, then all slow path packets will be affected (high end router may be more intelligent), then more important control packets can’t be processed by CPU, will cause more serious problem

Conclusions

Conclusions

- SAVI drafts have been implemented by multiple vendors and being largely deployed in CERNET2
  - draft-ietf-savi-dhcp-02
  - draft-bi-savi-stateless-00
- SAVI switches in CNGI-CERNET2 have been fully tested
- SAVI management system and MIB have been designed
- A light-weight savi-slaac is necessary for low end access switch for large scale deployment
  - Currently, no major problem found
  - For details: draft-bi-savi-stateless-00

Thank You!

Q & A