

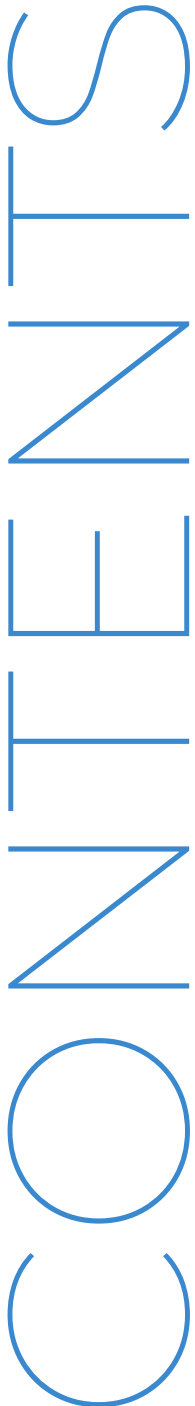
FINAL PROJECT

VLAN Implementation

吳弘叡

Class NSCAP

Table of Contents



01.

WHY and WHAT?

02.

List of implemented functions

03.

Network topology

04.

Demonstration

05.

Reference

WHY AND WHAT?

WHY VLAN?

I think VLAN is a grace technique in the network since it is able to do following goals **transparently**.

1. Divide A LAN into multiple pieces
2. Decrease the bandwidth occupation of broadcasting
3. Message privacy

WHAT I have done?

1. VLAN create
2. VLAN delete
3. VTP (VLAN Trunk Protocol)
 - Make sure all switches sychronized VLAN in a physical LAN
4. broadcasting

LIST OF IMPLEMENTED FUNCTIONS

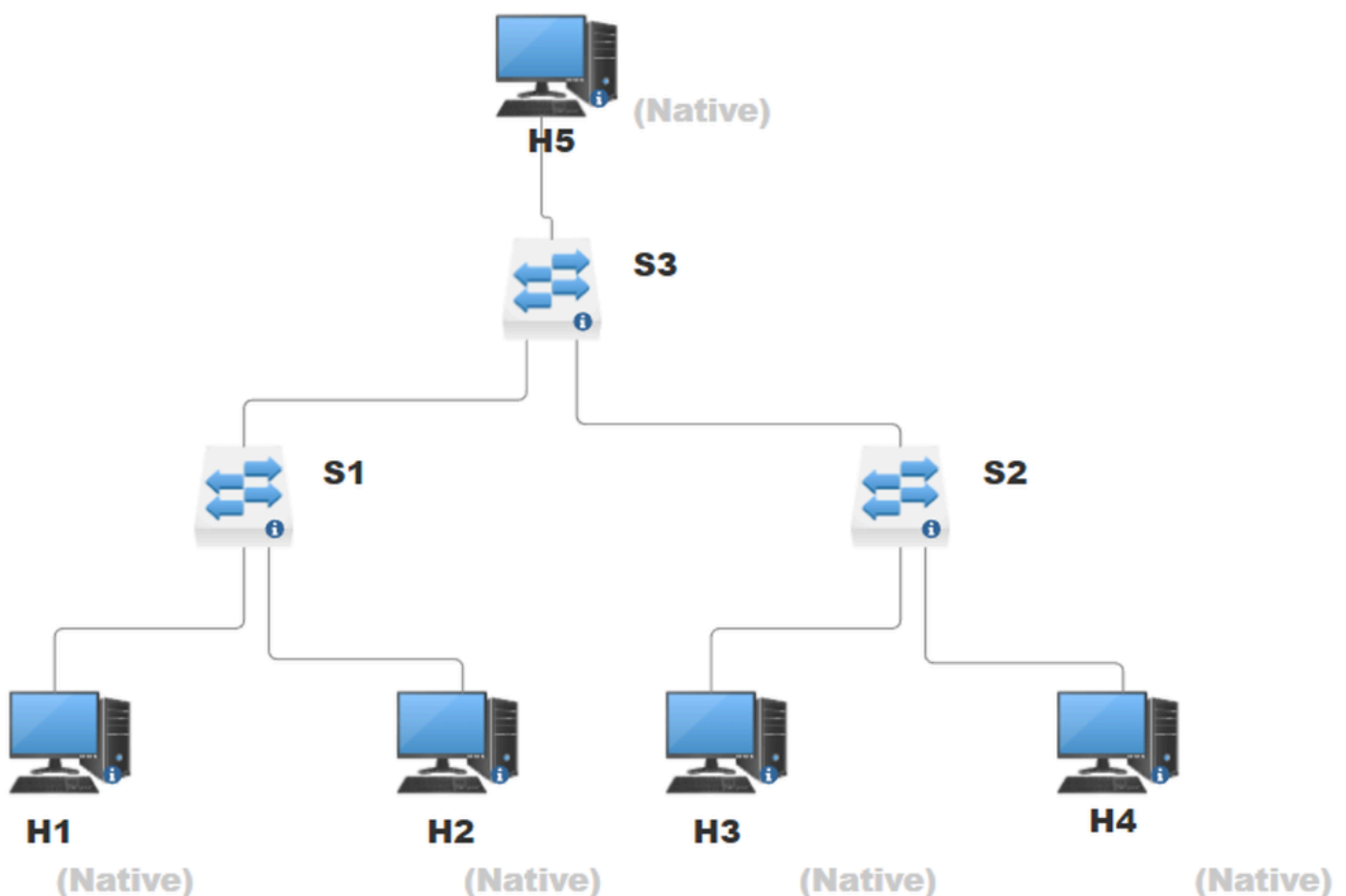
Function	Usage	Action
ping	host1 ping host2	<ul style="list-style-type: none">pinging from host1 to host2
createVLAN	createVLAN #VID	<ul style="list-style-type: none">create VLAN at the server switch
deleteVLAN	deleteVLAN #VID	<ul style="list-style-type: none">remove VLAN from the server switch
setAccess	setAccess #SID #port #VID	<ul style="list-style-type: none">Append #port at the switch #SID to VLAN #VID
send	host1 send host2 [message]	<ul style="list-style-type: none">host1 send [message] to host2
show_table	show_table [device.name]	<ul style="list-style-type: none">show kinds of table:switch: MAC table, VLAN table,host: ARP table

NETWORK TOPOLOGY

Since VLAN must handle **default case**; that is, no VLAN appended case.

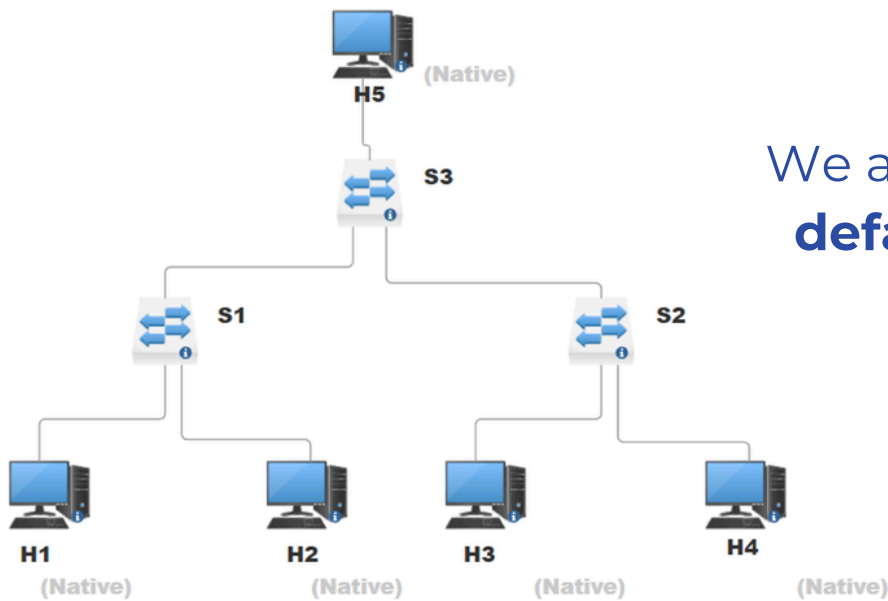
How? We have “Native VLAN”

Original Topology (All ports in native VLAN)



DEMONSTRATION

Original Topology (All ports in native VLAN)



We are going to demo the **default case** is handled

```
>> h1 send h2 hi
<h2> recv msg from h1ip: hi
>> h1 broadcast 大家今天好嗎
<h2> recv msg from h1ip: 大家今天好嗎
<h3> recv msg from h1ip: 大家今天好嗎
<h4> recv msg from h1ip: 大家今天好嗎
<h5> recv msg from h1ip: 大家今天好嗎
```

- h2 has received h1's message
- h2, h3, h4, h5 has received h1's **broadcast** message

After this two execution, we know that VLAN work well **without** any new added VLAN currently. Namely, “Native VLAN” works well in this state.

- send **Success**
- broadcast **Success**

DEMONSTRATION

What to do now?

1. Create 2 VLANs
 - VLAN10
 - VLAN20
2. We are going to work with
 - 2 different VLANs
 - Native

Let's establish VLANs

```
createVLAN 10
createVLAN 20
show_table all_switches #確保VLAN相同
```

Switch s1's table

```
-----s1:
MAC | Port
h1mac | 0
h2mac | 1
Port | VLAN
0 | Native
1 | Native
Trunk ports: [2]
VlanID | status | Configuration Revision
10 | active | 0
20 | active | 0
```

Switch s2's table

```
-----s2:
MAC | Port
h3mac | 0
h4mac | 1
h1mac | 2
Port | VLAN
0 | Native
1 | Native
Trunk ports: [2]
VlanID | status | Configuration Revision
10 | active | 0
20 | active | 0
```

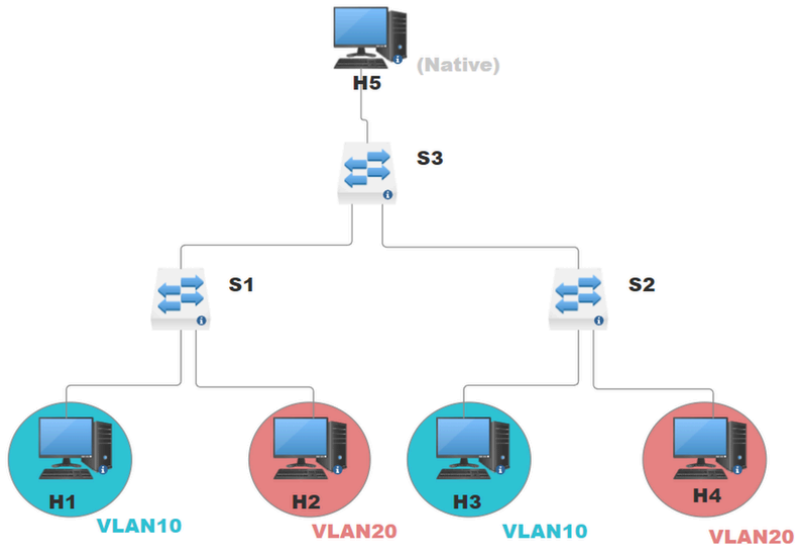
Switch s3's table

```
-----s3:
MAC | Port
h5mac | 2
h1mac | 0
Port | VLAN
2 | Native
Trunk ports: [0, 1]
VlanID | status | Configuration Revision
10 | active | 0
20 | active | 0
```

All switches synchronized by server switch.
s1, s2, s3 all have VLAN10 and VLAN20

DEMOSTRATION

Let ports become access ports



```
>> setAccess s1 0 10
Port0 is set to VLAN10 in s1
>> setAccess s1 1 20
Port1 is set to VLAN20 in s1
>> setAccess s2 0 10
Port0 is set to VLAN10 in s2
>> setAccess s2 1 20
Port1 is set to VLAN20 in s2
>> 
```

Set access ports for s1 and s2 respectively

check h1 -> port0 -> VLAN10
h2-> port1 -> VLAN20

```
-----s1:
MAC    | Port
h1mac  | 0
h2mac  | 1
Port   | VLAN
0      | 10
1      | 20
Trunck ports: [2]
```

check h3 -> port0 -> VLAN10
h4-> port1 -> VLAN20

```
-----s2:
MAC    | Port
h3mac  | 0
h4mac  | 1
h1mac  | 2
Port   | VLAN
0      | 10
1      | 20
Trunck ports: [2]
```

Now, the hosts are correctly in their groups(VLANs)

DEMOSTRATION

Test

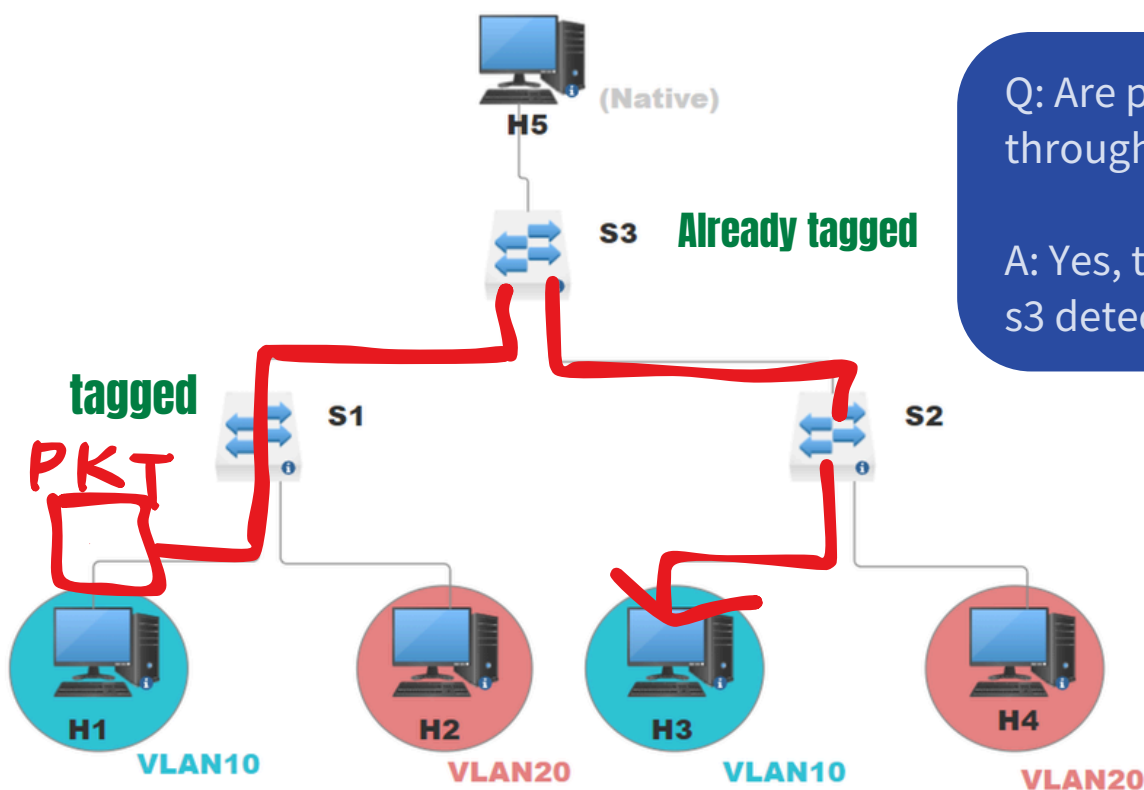
- communication within a VLAN
- trying to ping/send to another VLAN

Observe

- Are packets from specific VLAN port tagged when go through **trunk ports**?
- Who can receive h1's broadcast now?
- Will ARP table affect VLAN grouping?

>> h1 send h3 h3 我們是同夥了呢

```
>> h1 send h3 h3 我們是同夥了呢
<s1> tag Port0 pkt with VLAN10 and send to TrunkPort2
<s3> already tagged Port0 pkt with VLAN10 and send to TrunkPort1
<h3> recv msg from h1ip: h3 我們是同夥了呢
```



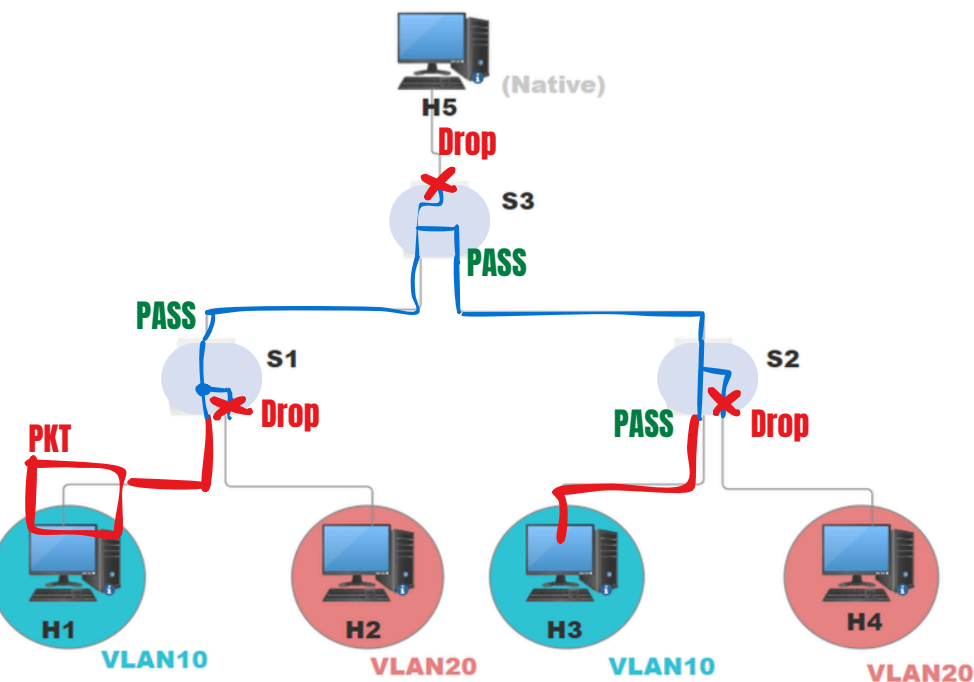
Q: Are packets when go through **trunk ports**?

A: Yes, tagged by s1 and s3 detected this tag.

DEMOSTRATION

>> h1 broadcast 有人在這嗎?理理我

```
>> h1 broadcast 有人在這嗎?理理我
<s1> Drop since 1's VLAN20 != 0's VLAN10
<s1> tag Port0 pkt with VLAN10 and send to TruncckPort2
<s3> already tagged Port0 pkt with VLAN10 and send to TruncckPort1
<h3> recv msg from h1ip: 有人在這嗎?理理我
<s2> Drop since 1's VLAN20 != 2's VLAN10
<s3> Drop since 2's VLANNative != 0's VLAN10
```



Q: Who can receive h1's broadcast now?

A: H3 only since they are in the same VLAN.

Notice switches on image. They dropped if the **VLAN of a port** didn't match.

h2's ARP table

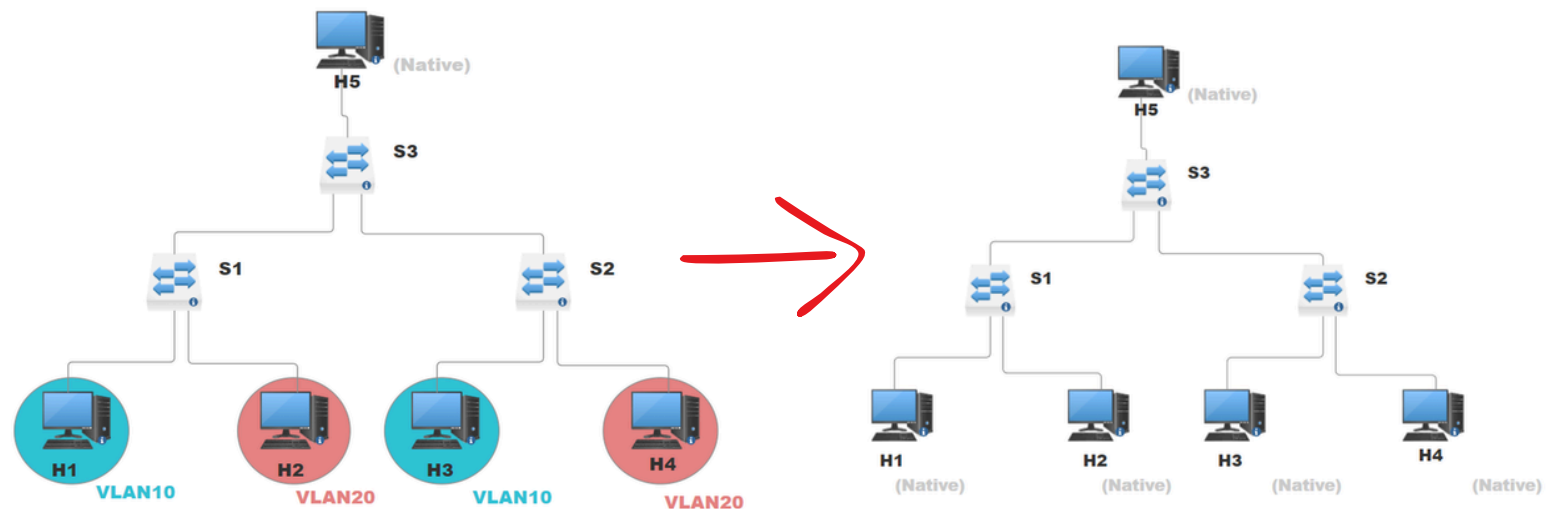
```
>> show_table h2
ip : mac
-----h2
h1ip : h1mac
```

Q: Will ARP table affect VLAN grouping?

A: No. Since switch is like a guard determine whether a packet go through to a port or not.

DEMOSTRATION

Finally, back to nature



>> deleteVLAN 10

>> deleteVLAN 20

>> show_table all_switches

```
>> deleteVLAN 10
<s3> Remove VLAN10
<s1> Remove VLAN10
<s2> Remove VLAN10
>> deleteVLAN 20
<s3> Remove VLAN20
<s1> Remove VLAN20
<s2> Remove VLAN20
```

```
-----s1:
MAC | Port
h1mac | 0
h2mac | 1
h3mac | 2
Port | VLAN
0 | Native
1 | Native
Trunk ports: [2]
VlanID | status | Configuration Revision
```

```
-----s2:
MAC | Port
h3mac | 0
h4mac | 1
h1mac | 2
Port | VLAN
0 | Native
1 | Native
Trunk ports: [2]
VlanID | status | Configuration Revision
```

```
-----s3:
MAC | Port
h5mac | 2
h1mac | 0
h3mac | 1
Port | VLAN
2 | Native
Trunk ports: [0, 1]
VlanID | status | Configuration Revision
```

We can observe that all ports on switches became “Native”

DEMOSTRATION

>> h1 broadcast 大家都回來了嗎?

```
>> h1 broadcast 大家都回來了嗎?  
<h2> recv msg from h1ip: 大家都回來了嗎?  
<h3> recv msg from h1ip: 大家都回來了嗎?  
<h4> recv msg from h1ip: 大家都回來了嗎?  
<h5> recv msg from h1ip: 大家都回來了嗎?
```

h2, h3 , h4, h5 can receive the broadcast
from h1 as default case

REFERENCE

[How VLAN work?] <https://www.etherwan.com/tw/support/featured-articles/brief-introduction-vlans>

[802.1Q] <https://blog.csdn.net/u013490557/article/details/37512045>