



Chapter 4: Network technologies

Lecturer: Nguyễn Thị Thanh Vân – FIT - HCMUTE

Object

- ▶ Understand bridges and bridging behavior
- ▶ Understand switches and switching behavior
- ▶ Understand about STP
- ▶ Understand about VLAN
- ▶ Understand about VTP
- ▶ Understand about routing: RIP, OSPF, BGP

Content

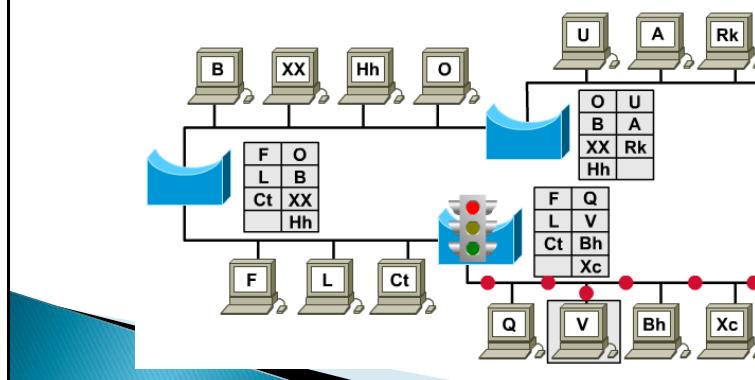
- Bridges & Bridging
 - Transparent Bridging
 - Source route bridging
 - Translational Bridging
- Switch & Switching
 - Circuit switching
 - Packet switching
- Redundancy & Broadcast Storm
- STP
- VLAN
- VTP
- Routing: RIP, OSPF, EIGRP

3

I. Bridges & Bridging

▶ Bridge:

- The main function is connect many segments together.
- Bridge actives base on Bridging mode, makes data traffic among segments effectively
- Bridge has memory, it can learn and storage of MAC



Bridging

- ▶ Bridging: can forward frames among different networks (Ethernet-Ethernet, Ethernet-TokenRing ...)
- ▶ Bridging operates base on:
 - Analyze incoming frames
 - Forward frames to the destination
- ▶ Bridging has many kinds :
 - Transparent bridging (Ethernet-Ethernet)
 - Source-route bridging (TokenRing-TokenRing)
 - Translational bridging (Ethernet and Token Ring)
 - Source-route transparent bridging (transparent bridging and source-route bridging)

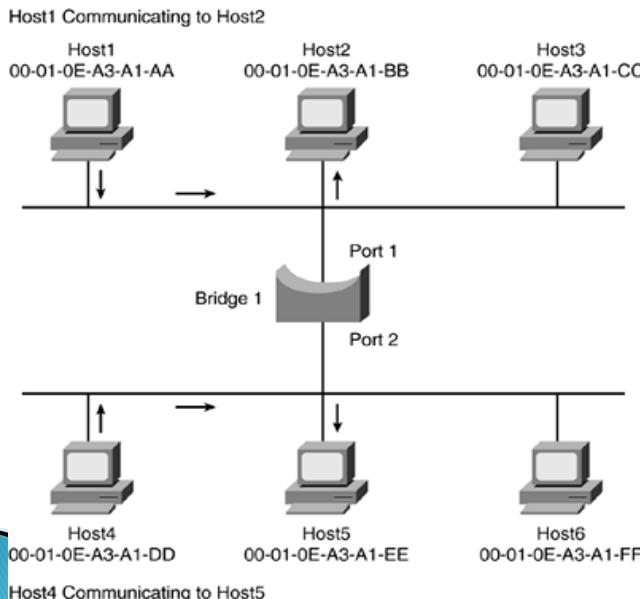
Java Simplified / Session 22 / 5 of 5

a. Transparent bridging

- ▶ The word transparent, as it is used here, means that the network devices are unaware of the presence of the bridge.
- ▶ So what does a transparent bridge do, and how does it accomplish its goal of segmenting network signals
- ▶ The 802.1D specification for transparent bridging defines **five** unique processes as part of transparent bridging:
 - Learning
 - Flooding
 - Filtering
 - Forwarding
 - Aging

Java Simplified / Session 22 / 6 of 6

Bridging, ex



Java Simplified / Session 22 / 7 of 45

Learning

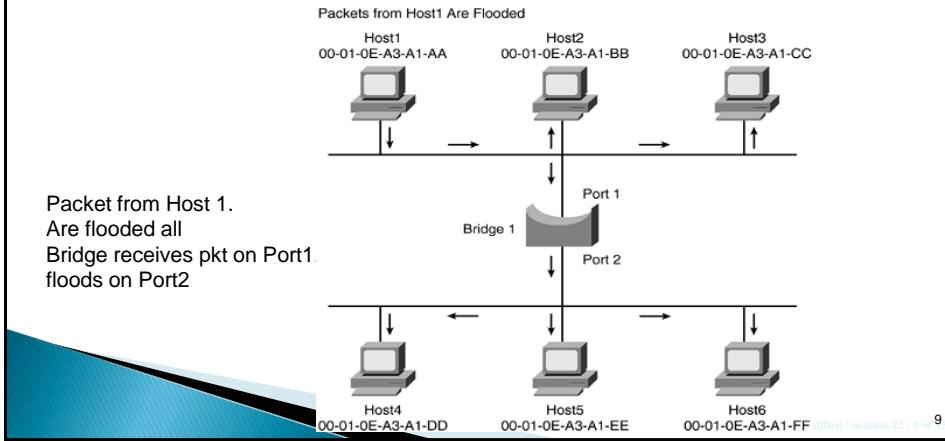
- When a bridge is first turned on, it has no entries in its bridge table.
- As traffic passes through the bridge, the sender's MAC addresses are **stored in a table** along with the associated port on which the traffic was received. (bridges learn by using the source MAC address) => Bridge table, MAC/CAM table.

Hosts	Port 1	Port 2
Host1/ 00-01-0E-A3-A1-AA	X	
Host2/ 00-01-0E-A3-A1-BB	X	
Host3/ 00-01-0E-A3-A1-CC	X	
Host4/ 00-01-0E-A3-A1-DD		X
Host5/ 00-01-0E-A3-A1-EE		X
Host6/ 00-01-0E-A3-A1-FF		X

Java Simplified / Session 22 / 8 of 45

Flooding

- ▶ Flooding is performed when a bridge does not have an entry in its bridge table for a specific address, it must transparently pass the traffic **through all its ports** except the source port. This is known as flooding.
- ▶ Flooding allows the bridge to learn, as well as stay transparent to the rest of the network, because no traffic is lost while the bridge is learning



Forwarding

- ▶ Forwarding is performed when the switch does have an entry for the frame's destination MAC address.
- ▶ **Forwarding** describes the process of **transmitting (or copying) a frame from one port to another using a known destination MAC address**.
- ▶ Forwarding is simply passing traffic from a known device located on one bridge port to another known device located on a different bridge port
- ▶ **A forwarding table** is used by a bridge to decide which port will receive a particular frame. ≥

Filtering

- ▶ When Bridge receives the packet from a Client. It looks up the destination MAC address of the packet, determines that the destination address is local, and does not forward the packet out of others Ports. This process is known as **filtering**.
- ▶ Filtering is performed when the switch has an entry for both the source and destination MAC address, and the MAC table indicates that both addresses are found off the same port.
- ▶ .≥

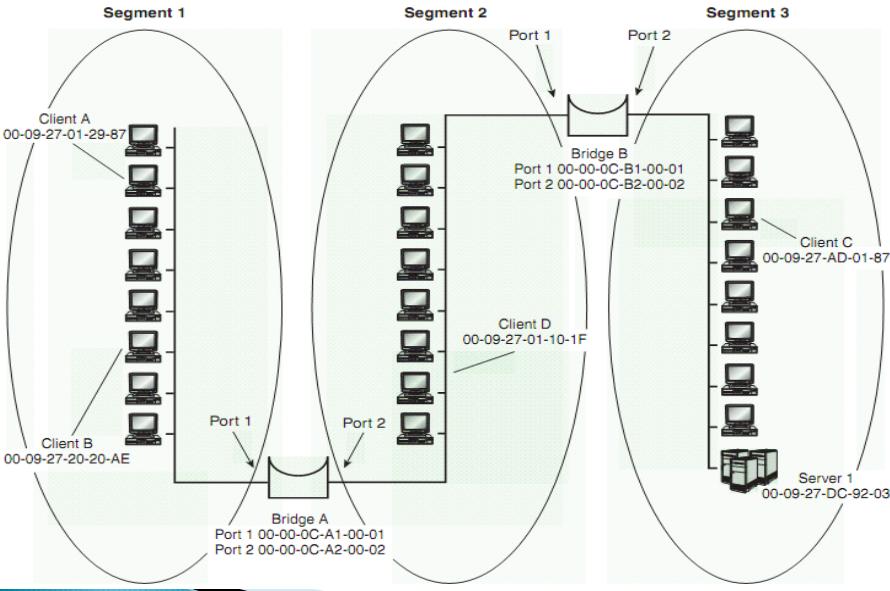
Java Simplified / Session 22 / 11 of 15 11

Aging

- ▶ **Aging** of learned MAC addresses allows the bridge to adapt to moves, adds, and changes of devices to the network. After a device is learned, the bridge starts an aging timer. Each time the bridge forwards or filters a frame from a device, it restarts that device's timer.
- ▶ If the bridge doesn't hear from a device in a preset period of time, the aging timer expires and the bridge removes the device from its table.
- ▶ **Buffering** occurs when the bridge receives frames of data and holds them in memory before forwarding or filtering

Java Simplified / Session 22 / 12 of 15 12

Transparent bridging



Java Simplified / Session 22 / 13 of 13

ex

- ▶ Client A -> Client B
- ▶ Client A -> Client D

Java Simplified / Session 22 / 14 of 13

exp

1. Client A sends out a broadcast request looking for the MAC address of Client B.
2. Client B receives the ARP request. In fact, all stations on network Segment 1 receive the broadcast, including Port 1 of Bridge A => Bridge A flooding every port except the port on which it was received (Segment 2)
3. All of the clients on Seg-ment 2 receive the packet, including Bridge B. Because it is a broad-cast, Bridge B floods the packet out of all ports except the port on which it was received
4. Bridge B finishes the initial process for Client A's search for Client B by flooding the packet out of Port 2 to Segment 3.
5. Client B answers the broadcast request for its MAC address. This is a unicast packet with a source address of Client B.

Java Simplified / Session 22 / 15 of 15

b. Source route bridging

- ▶ SRB is used in Token Ring.
- ▶ SRB discover the best path to a client in source route bridging networks (differ transparent bridging)
- ▶ SRB uses 2 frame types for finding route to network segments:
 - Frame Single-Route (SR): contains most of the network traffic and the destination
 - Frame All-Route (AR): discover the best path
- ▶ Bridge sends AR frame by:
 - dispersal frames to all paths and store all their routing
 - The first AR frame to destination, it is the best path for SR frames, others frame will be ignored

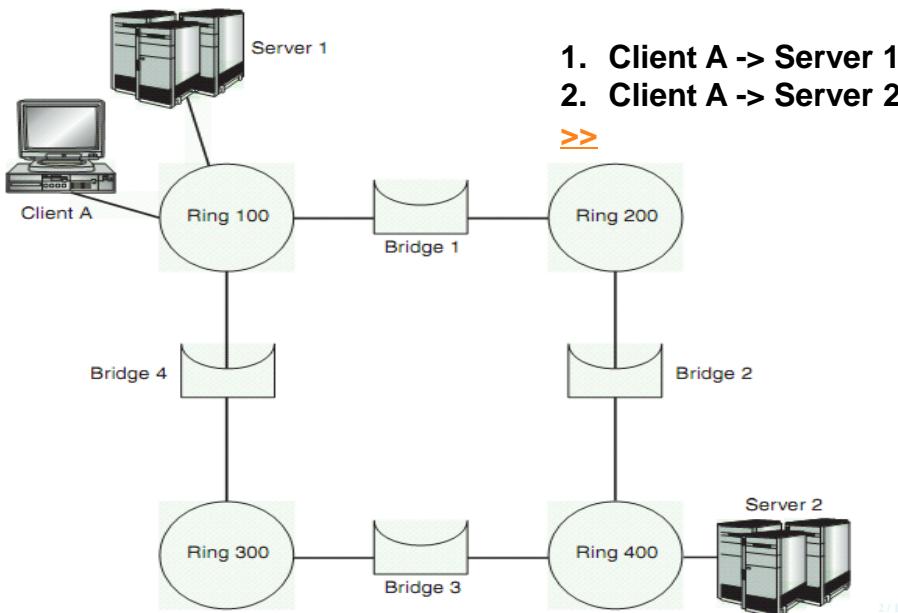
Java Simplified / Session 22 / 16 of 15

Concept

- ▶ routing information field (RIF): A field used in source route bridging that maintains the correct path that frame used to traverse a series of Token Ring networks. Token Ring bridges or switches inhabit this field
- ▶ routing information indicator (RII): A field used in source route bridging that identifies the packet as a local frame if set to 0, or a source route frame if set to 1.
- ▶ all routes explorer (ARE): A frame used to discover the best path to a client in source route bridging networks

Java Simplified / Session 22 / 17 of 45

b. Source route bridging, ex



2/18 18

b. Source route bridging, ex

- ▶ Client A needs to communicate with Server 1.
 - 1. Client A sends out a local explorer frame to determine whether Server 1 is on the same ring as itself. A special bit in the frame called the routing information indicator (RII) is set to 0, indicating to all bridges that the frame should be ignored.
 - 2. Server 1 sees a frame with its address and responds to the local explorer frame. At this point all communications take place locally without involving any of the bridges or other rings.

Java Simplified / Session 22 / 19 of 20 19

Source route bridging, ex

- ▶ Client A needs to communicate with Server 2. 
- 1. Client A sends out a local explorer frame to determine whether Server 2 is on the same ring as itself.
- 2. No response is received to the local explorer frame, so Client A sends out an all routes explorer (ARE) frame, then the RII field is set to 1, indicating that bridges should pass the frame and fill in the routing information field (RIF).
- 3. Bridge 1 receives the ARE from Client A and fills in the RIF. The first entry indicates the source ring and the local bridge ID, like this: *ring100-bridge1*
- 4. Bridge 1 retransmits the frame as a local explorer frame attempting to find Server 2 on ring 200.

Java Simplified / Session 22 / 20 20

b. Source route bridging

5. No response to the local explorer frame is received, so Bridge 1 retransmits the frame with the RII set to 1 and the discovery process continues at Bridge 2. \geq
6. Bridge 2 receives the frame and notes that the RII is set to 1. It fills in the RIF field with the next set of entries, as follows:
ring100-bridge1-ring200-bridge2
7. Bridge 2 sends the frame out as a local explorer frame attempting to find Server 2 on ring 400.
8. Server 2 responds to the local explorer frame and retransmits the RIF field, sending the frame back to Client A along the same path.

Java Simplified / Session 22 / 21 21

c. Translation bridging

- ▶ Using in some networks contain a mixture of Ethernet and Token Ring clients
- ▶ A bridge that communicates both Ethernet and Token Ring can translate mismatched fields and allow networks to communicate properly.
- ▶ Unfortunately, few standards exist for translational bridging and implementations vary between vendors.
- ▶ Solution: a router or other Layer 3 device should be used instead of a Layer 2 bridge or switch.

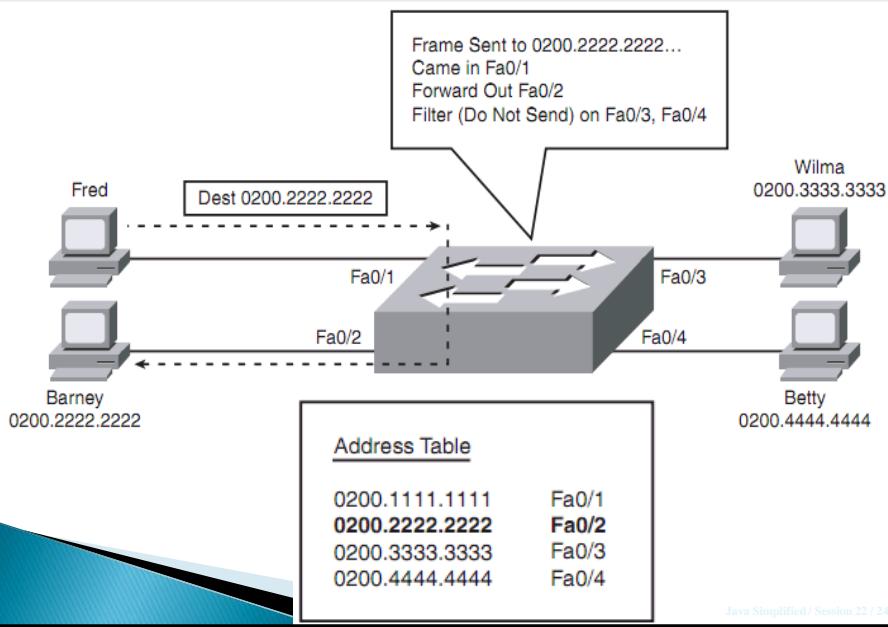
Java Simplified / Session 22 / 22 22

II. Switch & Switching

- The concept of switching is identical to bridging: They both forward, flood, and filter traffic using the same rules.
- Both bridges and switches create multiple collision domains as well (one collision domain per port).

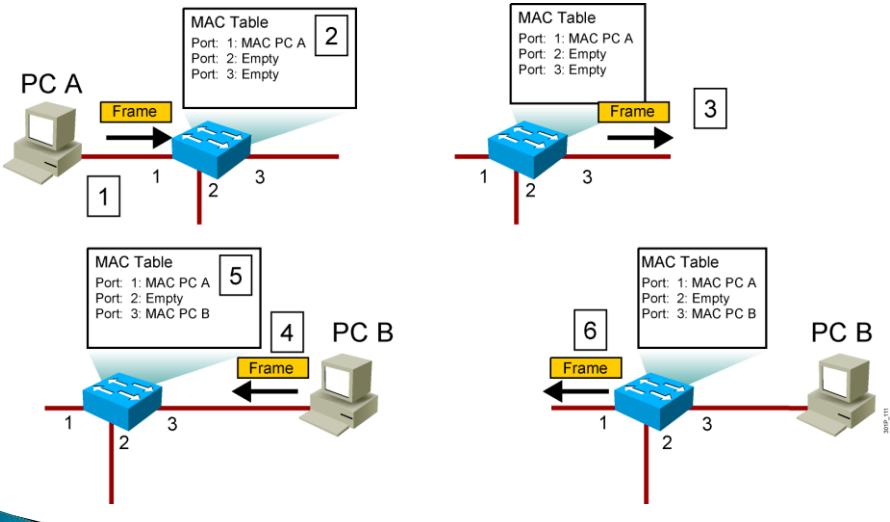
Java Simplified / Session 22 / 23 23

Switch forwarding and filtering Decision



Java Simplified / Session 22 / 24 24

Switching Frames



Java Simplified / Session 22 / 25 25

II. Switch & Switching

- ▶ However, switching offers the following improvements over bridging:
 - Higher port density
 - Faster packet-processing capabilities
 - Quality-of-service (QoS) capabilities
 - Use of virtual LAN (VLAN) technology
 - Avoid loops using Spanning Tree Protocol

Java Simplified / Session 22 / 26 26

II. Switch & Switching

- ▶ Faster packet-processing capabilities in a number of different ways:
 - Application-specific integrated circuits (ASICs)
 - today's networking applications and services require much faster processing capabilities. This is achieved by imbedding the decision routines into hardware ASICs installed on each port on a switch.
 - Better central processing units (CPUs)
 - switches use the latest and most powerful CPUs on the market that help process all of the port-based ASIC decisions as well as other switching functions such as QoS and VLAN configuration

Java Simplified / Session 22 / 27 / 28

27

II. Switch & Switching

- ▶ Faster packet-processing capabilities in a number of different ways:
 - Cut-through switching
 - allows the switch to forward immediately after receiving the destination MAC address
 - The major limit: It is that malformed (not good) frames from collisions or a misbehaving NIC are also forwarded
 - Fragment-free switching
 - The frame is forwarded after receipt of the 64th byte frame
 - A successful transmission to the 64th byte ensures that a high probability frame is good (not a collision). Because Ethernet requires that collisions are detected prior to this byte being transmitted

Java Simplified / Session 22 / 28

28

III. Redundant and Broadcast storm

- ▶ Introduction
 - Device layer 2,3
 - Loop in network
- ▶ STP
 - Concept
 - Active

Java Simplified / Session 22 / 29 29

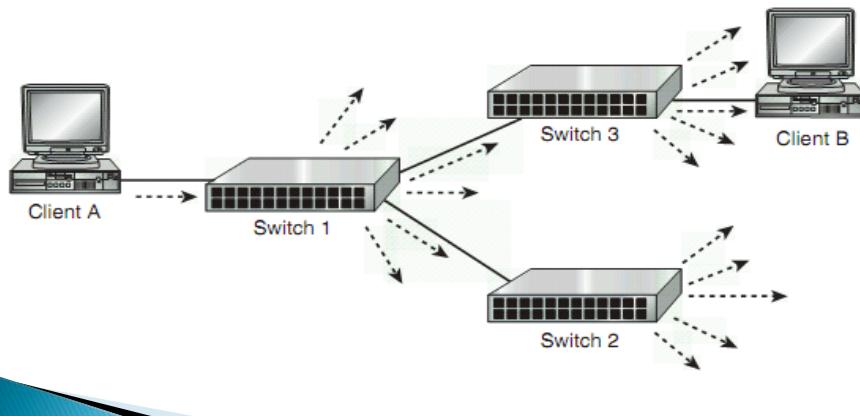
Device layer 3,2

- ▶ Layer 3: router
 - Routing, Multiple paths to destination
 - Backup path
 - Limit risk in the network
- ▶ Layer 2: Switch, Bridge
 - Not routing, not backup
 - Redundant link: use looped topology, result:
 - Prevent network failure in the even one link stop working
 - But, create errored loop, because of:
 - Broadcast storm,
 - forwarding table corruption

Java Simplified / Session 22 / 30 30

Normal broadcast flooding

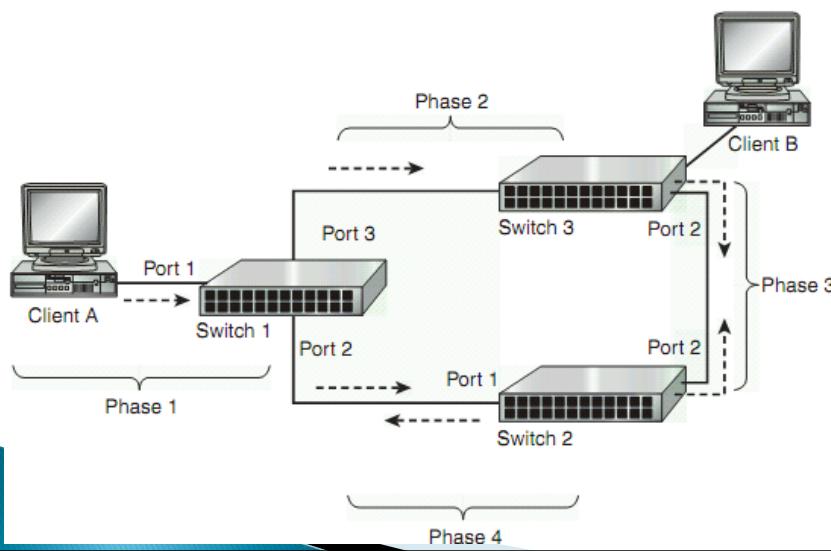
- ▶ Broadcast from Client A is flooded out of all ports of all switches.



Java Simplified / Session 22 / 31 | 31

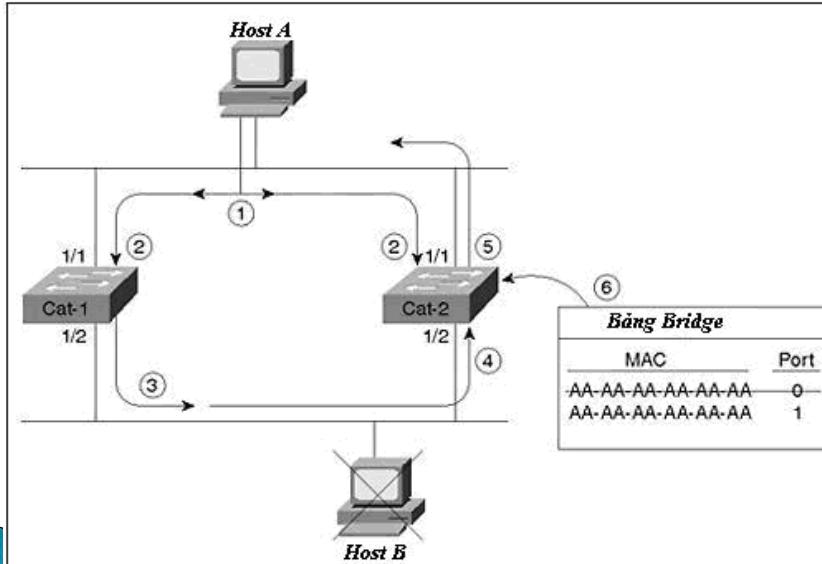
Broadcast storm

- ▶ Because the switches are plugged together in a looped



ion 22 / 32 | 32

Forwarding table corruption



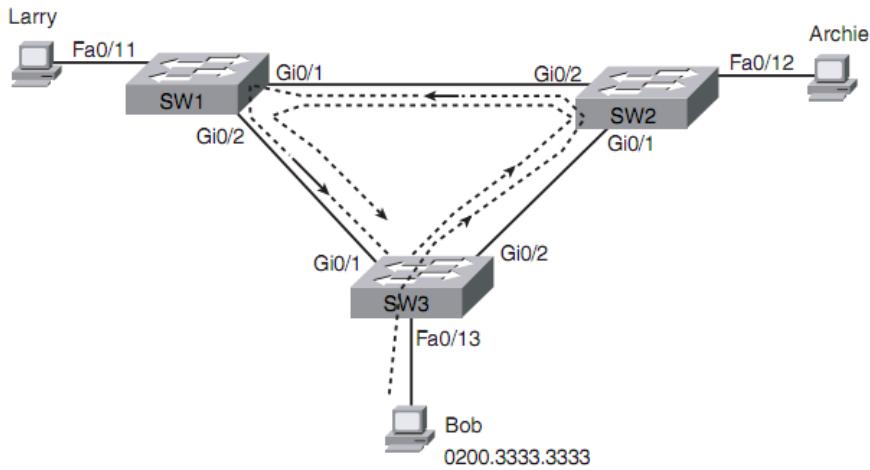
Java Simplified / Session 22 / 33 33

III. Solution STP

- Without STP, a LAN with redundant links would cause Ethernet frame to loop for an indefinite period of time.
- With STP enabled, some switches block ports so that these ports do not forward frames. STP chooses which ports block so that only one active path exists between any pair of LAN segment.
- As a result, frames can be delivered to each device, without causing the problems created when frames loop through the network
- => STP's main task is to stop network loops from occurring on layer 2 network (Switch or Bridge)

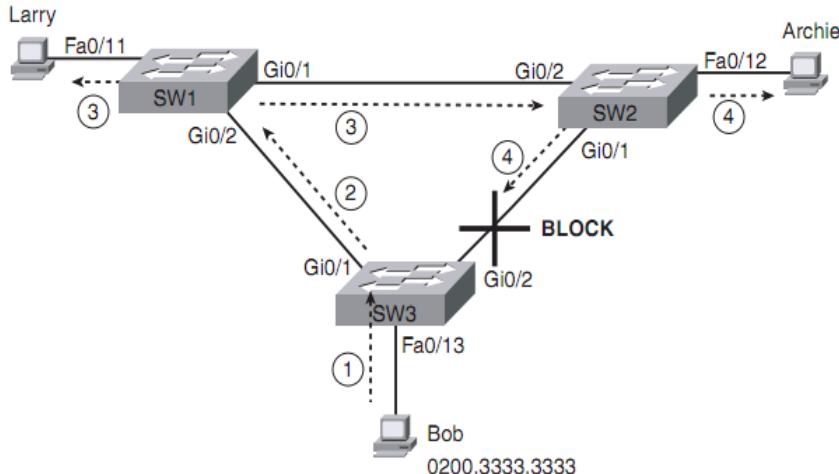
Java Simplified / Session 22 / 34 34

Ex without STP



Java Simplified / Session 22 / 35 35

With STP



Java Simplified / Session 22 / 36 36

Concepts in STP

- ▶ BID (Bridge ID)
- ▶ Root Bridge
- ▶ Port cost
- ▶ Root port
- ▶ Designated port

Java Simplified / Session 22 / 37 / 45 37

BID (Bridge ID)

- ▶ BID (Bridge ID): is 8 bytes long and consists of:
 - a bridge priority: is a user-configurable 2-byte field offering a range of values from 0 to 65535 (2^{16}).
 - a MAC address is a 6-byte field and varies from vendor to vendor but is typically the lowest configured MAC address on the bridge. It is important to remember that the lowest bridge ID wins the root bridge election

Java Simplified / Session 22 / 38 / 45 38

Concepts

- ▶ Root Bridge: is the bridge with the best bridge ID (lowest)
- ▶ Port cost: determines the best path when multiple links are used between 2 switches and none of the links is root port (by bandwidth of a link)
- ▶ Root port: is the shortest path to the root bridge (lowest-cost port).
- ▶ Designated port: is one that has been determined as having the best (lowest) cost.

Java Simplified / Session 22 / 39 39

BPDU - bridge protocol data unit

- ▶ BPDU - bridge protocol data unit:

Some of the fields in the BPDU are:

- Root bridge ID: This is the bridge ID of the current root bridge.
- Root path cost: This is the cost of the path to the root bridge
- Sender bridge ID: This is the ID of the bridge that sent the BPDU
- Port ID: This is the ID of port on the bridge that sent the BPDU

Java Simplified / Session 22 / 40 40

STP Cost (port cost)

802.1D defines:

$\text{Cost} = 1000 \text{ Mbps/Bandwidth}$

Bandwidth	STP Cost
4 Mbps	250
10 Mbps	100
16 Mbps	62
45 Mbps	39
100 Mbps	19
155 Mbps	14
622 Mbps	6
1 Gbps	4
10 Gbps	2

Java Simplified / Session 22 / 41 / 45

41

Decision of STP

- Root BID: lowest
- Path Cost to Root Bridge: lowest
- Sender BID: lowest
- PortID: lowest

Java Simplified / Session 22 / 42 / 45

42

STP process

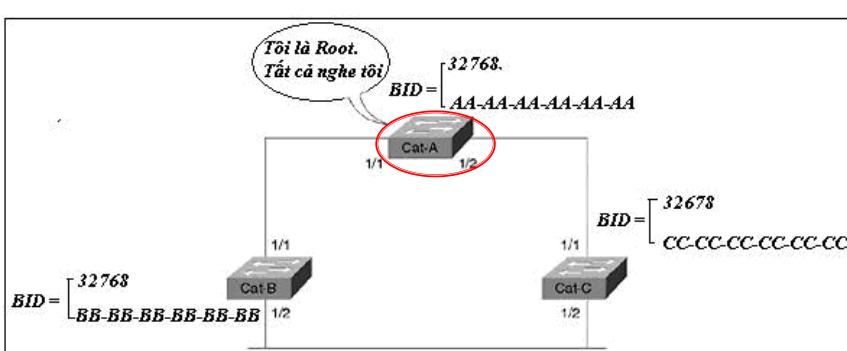
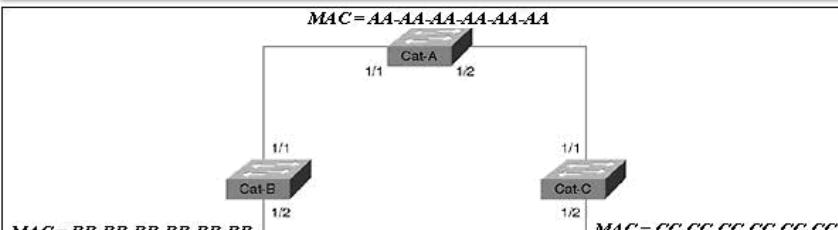
- The STP process is accomplished by:

(Initial STP Convergence)

- Election of a **root bridge**:
Root BID: lowest (by MAC)
- Identification of a **root port**
the shortest path to the root bridge
- Identification of a **designated port**

Java Simplified / Session 22 / 43 43

Election of a root bridge



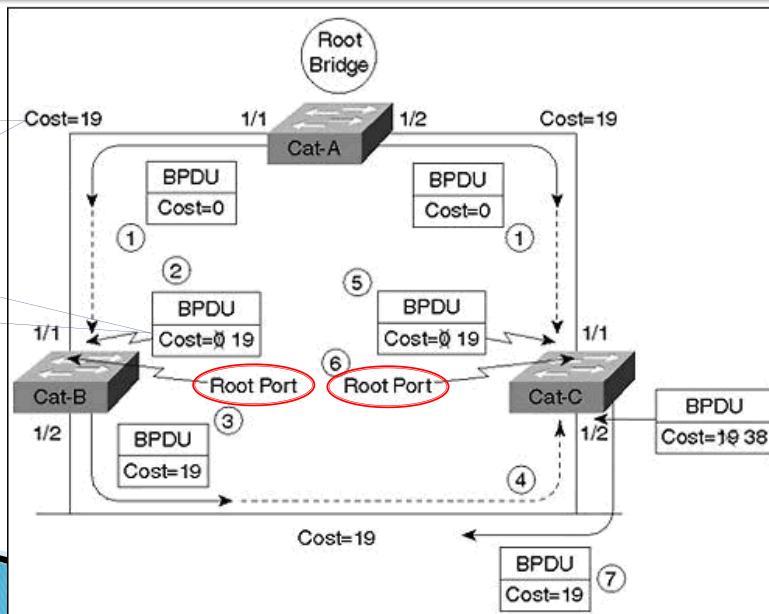
27/43 44

Identification of a root port

- Every switches except root bridge chooses root port.
- Root port (lowest-cost port): the shortest path to the root bridge.
(root path cost: Accumulation cost of all links connected to Root bridge)

Java Simplified / Session 22 / 45 45

Identification of a root port

path
costroot
path
cost

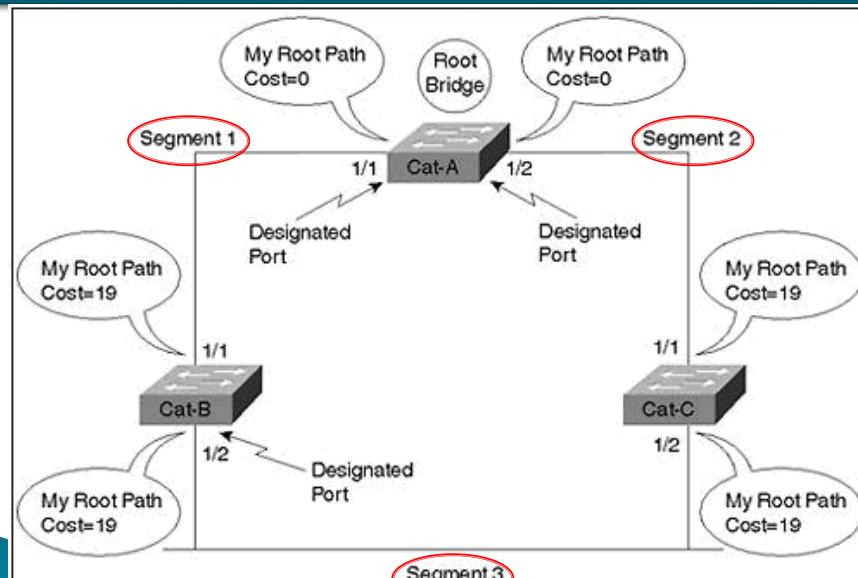
6

Identification of a designated port

- ▶ Note:
 - all port on Root Bridge: Designated port
 - (every segment):
 - Root BID: lowest **Between 2 switches**
 - Path Cost to Root Bridge: lowest
 - Sender BID: lowest
 - PortID: lowest
- ▶ The switch with the lowest bridge ID forwards traffic while the other blocks it

Java Simplified / Session 22 / 47/194

Identification of a designated port



48

Loop-free

- Now, we have a loop-free switch environment, and are protected from broadcast storms and forwarding table corruption.
 - Root Port and Designated Port: Forwarding
 - None Designated Port: Blocking

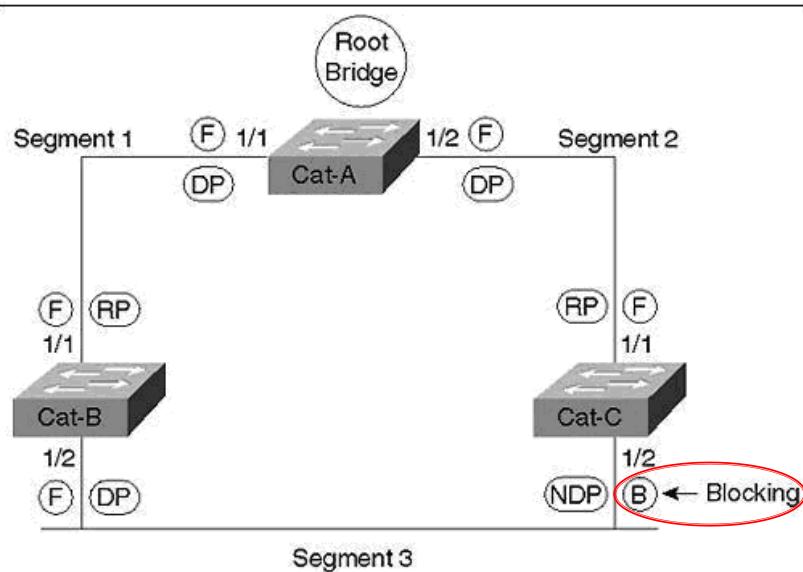
Java Simplified / Session 22 / 49 of 50 49

STP State

State	Purpose
Forwarding	Sending and receiving data
Learning	Create forwarding Table
Listening	Initial STP network
Blocking	Block traffic
Disable	Port stops working

Java Simplified / Session 22 / 50 50

Result



Java Simplified / Session 22 / 51 | 51

Symbol

State/ Port	symbol
Blocking	B
Forwarding	F
Designated Port	DP
Root Port	RP
None Designated Port	NDP

Java Simplified / Session 22 / 52 | 52

Example

STP	Number
Root Bridge	1
Root port	14
Designated port	146
<i>15 switches và 146 segments</i>	

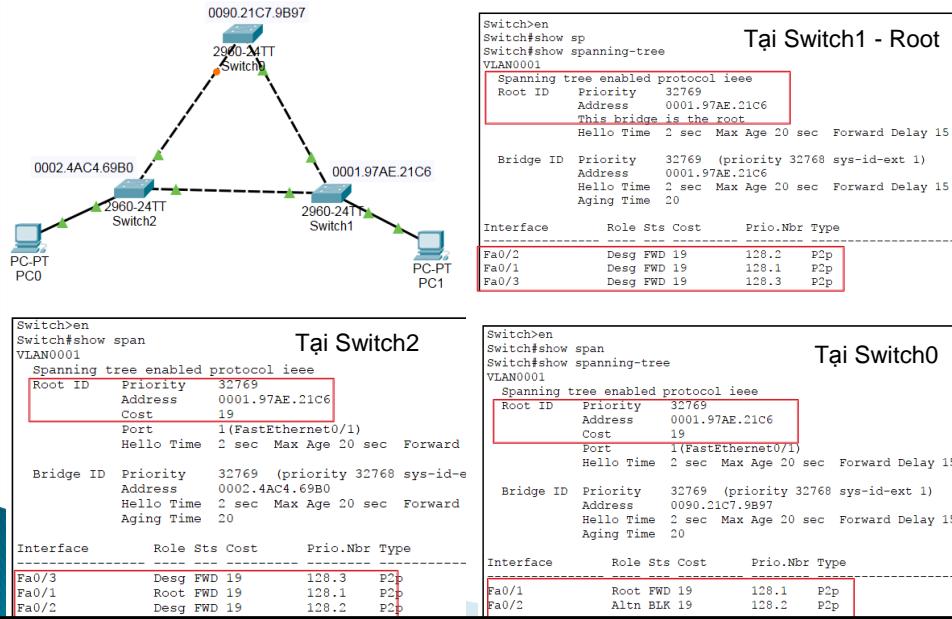
Java Simplified / Session 22 / 53 53

Cấu hình STP

- ▶ Cấu hình Spanning Tree mặc định
 - Spanning-tree được hoạt động mặc định trên VLAN 1 và trên tất cả các VLAN được tạo sau đó
 - show spanning-tree vlan 1 (show span)
- ▶ Disable Spanning Tree
 - no show spanning-tree vlan id
 - Để cho phép spanning-tree hoạt động trở lại trên một VLAN nào đó: spanning-tree vlan vlan-id (chế độ global configuration)

Java Simplified / Session 22 / 54 54

Ví dụ: Xem Cấu hình STP (default)



Cấu hình STP

- ▶ Cấu hình một Switch trở thành Root Switch.
 - Cần sửa đổi switch priority từ giá trị mặc định (32768) thành một giá trị khác thấp hơn. Ex:

```
SW(config)# spanning-tree vlan 2 root primary
SW(config)# end
SW# show spanning-tree detail
SW# show spanning-tree vlan 2
SW# copy running-config startup-config
(Copy nội dung của running-config trong RAM vào NVRAM)
```
- ▶ Cấu hình một Secondary Root Switch.
 - điều chỉnh lại priority từ giá trị mặc định thành giá trị 28672. Switch này sẽ trở thành root switch trong một vlan khi mà root switch không hoạt động hoặc bị lỗi. Ex:

```
SW(config)# spanning-tree vlan 2 root secondary
SW(config)# end
SW# show spanning-tree detail
SW# show spanning-tree vlan 2
```

Ví dụ: Cấu hình Switch trở thành Root Switch

Ví dụ, cấu hình Switch0 thành Root Bridge

Tại Switch1

```

Switch#show spanning-tree vlan 1
VLAN0001
  Spanning tree enabled protocol ieee
    Root ID Priority 24577
      Address 0090.21C7.9B97
      Cost 19
      Port 1(FastEthernet0/1)
        Hello Time 2 sec Max Age 20 sec Forward Delay 15
    Bridge ID Priority 32769 (priority 32768 sys-id-ext 1)
      Address 0001.97AE.21C6
      Hello Time 2 sec Max Age 20 sec Forward Delay 15
      Aging Time 20

```

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/1	Desg	FWD	19	128.1	P2p
Fa0/2	Desg	FWD	19	128.2	P2p
Fa0/3	Desg	FWD	19	128.3	P2p

Tại Switch2

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/1	Root	FWD	19	128.1	P2p
Fa0/2	Desg	FWD	19	128.2	P2p
Fa0/3	Desg	FWD	19	128.3	P2p

Cấu hình STP

- ▶ Cấu hình độ ưu tiên của port (Port Priority)
 - int có giá trị priority thấp nhất về trạng thái forwarding và blocks những int còn lại

```

SW(config)# interface fa0/10
SW(config-if)# spanning-tree port-priority 112
SW(config-if)# spanning-tree vlan 2 port-priority 112
SW(config-if)# end
SW# show spanning-tree interface fa0/10
SW# show spanning-tree vlan 2

```
- ▶ Cấu hình chi phí cho các đường đi (Path Cost)
 - int có giá trị cost thấp nhất trở về trạng thái forwarding và blocks những int còn lại

```

SW(config)# interface fa0/11
SW(config-if)# spanning-tree cost 10
SW(config-if)# spanning-tree vlan 2 cost 10
SW(config-if)# end
SW# show spanning-tree interface fa0/1

```
- ▶ Cấu hình độ ưu tiên của switch trong một Vlan
 - cho switch đó được lựa chọn làm root switch

```

SW(config-if)# spanning-tree vlan 2 priority 4098
SW(config-if)# end
SW# show spanning-tree vlan 2

```

IV. VLAN – Virtual LAN

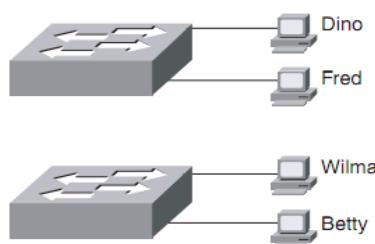
- ▶ 1 LAN bao gồm các thiết bị mạng nằm trong 1 broadcast domain (tất cả các thiết bị có thể nhận được frame -> giảm băng thông).
- ▶ Cần hạn chế broadcast traffic -> Router, VLAN
- ▶ VLAN là một kỹ thuật cho phép tạo lập các mạng LAN độc lập một cách logic trên cùng một kiến trúc hạ tầng vật lý (gom port).
- ▶ Thiết bị dùng để chia VLAN là switch
- ▶ 1 LAN -> nhiều VLAN: giúp giảm thiểu broadcast domain cũng như tạo thuận lợi cho việc quản lý một LAN rộng lớn. VLAN tương đương như mạng con (*subnet*)

VLAN = Broadcast Domain = Logical Network (Subnet)

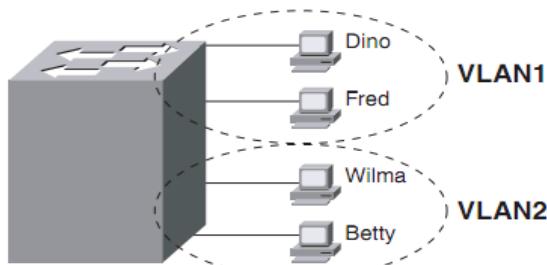
Java Simplified / Session 22 / 59 59

Sample Network

Sample Network with Two Broadcast Domains and No VLANs



Sample Network with Two VLANs Using One Switch



Java Simplified / Session 22 / 60 60

Ưu điểm và nhược điểm

- ▶ Tối ưu hóa băng thông và không làm hệ thống quá tải:
 - Giảm broadcast size
 - Tăng broadcast number
- ▶ Tăng khả năng bảo mật:

Các VLAN khác nhau không truy cập được vào nhau (trừ khi có khai báo định tuyến).
- ▶ Dễ dàng quản lý giám sát:

Có thể cấu hình VLAN khác nhau cho từng cổng của Switch, do đó dễ dàng kết nối thêm/bớt các máy tính với các VLAN.
- ▶ Mạng có tính linh động cao

Java Simplified / Session 22 / 61/63

Phân loại VLAN

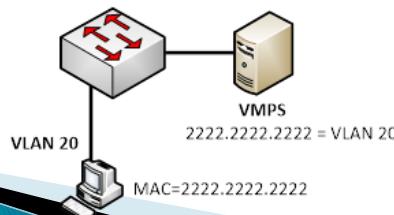
▶ Static VLAN

các cổng của Switch được cấu hình thuộc về một VLAN nào đó, các thiết bị gắn vào cổng đó sẽ thuộc về VLAN đã định trước. Đây là loại VLAN dùng phổ biến



▶ Dynamic VLAN

sử dụng một server lưu trữ địa chỉ MAC của các thiết bị và qui định VLAN mà thiết bị đó thuộc về, khi một thiết bị gắn vào Switch, Switch sẽ lấy địa chỉ MAC của thiết bị và gửi cho server kiểm tra và cho vào VLAN định trước

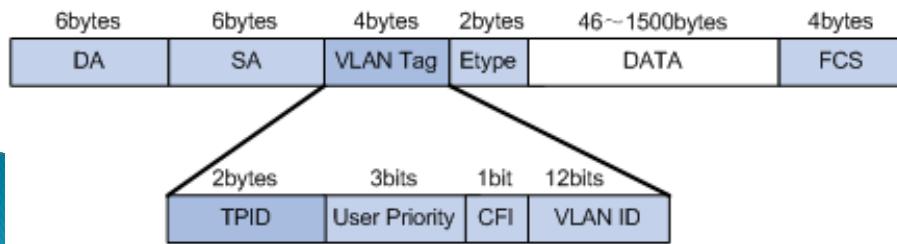


Java Simplified / Session 22 / 62

62

VLAN Tag Header

- Chuẩn IEEE 802.1Q là chuẩn về dán nhãn (*tagging*) trong VLAN, là giao thức thông dụng được sử dụng trong việc cấu hình các VLAN
- Cấu trúc khung Ethernet được gán thêm một ID (VLAN ID).
- TPIP – Tag Protocol Identifier (=0x8100): used to identify the tagged frame.
- TCI – Tag Control Information:
 - Priority: ranges 0-7. Classify the voice, data , video traffic.
 - CFI - Canonical Format Indicato (=0)
 used to identify the MAC address format
- VLANID: determines the protocol ID.



Các dạng VLAN

- Default VLAN (VLAN1)
 - Tất cả các giao diện Ethernet trên switch mặc định thuộc VLAN 1
 - Mặc định, native VLAN, management VLAN và user VLAN sẽ là thành viên của VLAN 1
- User VLAN:
 - được tạo ra nhằm tạo ra một nhóm người sử dụng mà không phụ thuộc vào vị trí địa lý hay logic và tách biệt với phần còn lại của mạng ban đầu.
 - lệnh switchport access vlan được dùng để chỉ định các giao diện vào các VLAN khác nhau
- Native VLAN
 - là một VLAN có các cổng được cấu hình trunk.
- Management VLAN
 - Khi truy cập Switch từ xa thì chúng ta nên đặt vào trong một VLAN, được gọi là Management VLAN.
 - VLAN này độc lập với các VLAN khác như user VLAN, native VLAN. Do đó khi mạng có vấn đề như: hội tụ với STP, broadcast storms, thì một Management VLAN cho phép nhà quản trị vẫn có thể truy cập được vào các thiết bị và giải quyết các vấn đề đó.

Management VLAN độc lập với user VLAN là việc tách các thiết bị đáng tin cậy với các thiết bị không tin cậy.

Configuration VLAN

Bước 1. Tạo VLAN

```
Switch(config)#vlan <vlan-id>
Switch(config-vlan)#name <vlan-name>
```

Bước 2. Gán các cổng cho VLAN

Case 1: Gán 1 cổng vào LAN

```
Switch(config)#interface <interface>
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan <vlan-id>
```

Case 2: Gán 1 dãy các cổng liên tiếp

```
Switch(config)#interface range <start>-<end-intf>
Switch(config-if-range)#switchport mode access
Switch(config-if-range)#switchport access vlan <vlan-id>
```

Case 3: Gán nhiều cổng không liên tiếp

```
Sw(config)#int range <int1, int2,...>
Sw(config-if-range)#switchport mode access
Sw(config-if-range)#switchport access vlan <vlan-id>
```

Ví dụ:

```
Switch(config)#vlan 10
Switch(config-if)#name P.KyThuat
```

Ví dụ:

```
Switch(config)#interface fa0/5
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 10
```

Ví dụ:

```
Switch(config)#interface fa0/10 - 20
Switch(config-if-range)#switchport mode access
Switch(config-if-range)#switchport access vlan 10
```

65

Configuration VLAN

- ❖ **Xóa VLAN:** Xóa một VLAN trên switch bằng cách sử dụng lệnh “no” trước câu lệnh tạo VLAN.
- ❖ **Lệnh kiểm tra cấu hình VLAN**

`Switch#show vlan [brief]`

- Lệnh này cho phép hiển thị các VLAN-ID (số hiệu VLAN), tên VLAN, trạng thái VLAN và các cổng được gán cho VLAN trên switch.

Java Simplified / Session 22 / 66 66

Ex: Cấu hình VLAN trên Switch

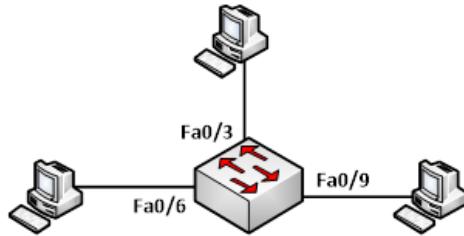
- Tạo 3 VLAN:
 - VLAN 10: Fa0/1 –Fa0/6,
 - VLAN 20: Fa0/7 – Fa0/9,
 - VLAN 30: Fa0/10 – Fa0/12

1. Tạo vlan:

```
Switch(config)#vlan 10
Switch(config)#vlan 20
Switch(config)#vlan 30
```

2. Gán các cổng vào VLAN

```
Switch(config)#interface range f0/1 - 6
Switch(config-if-range)#switchport mode access
Switch(config-if-range)#switchport access vlan 10
Switch(config)#interface range f0/7 - 9
Switch(config-if-range)#switchport mode access
Switch(config-if-range)#switchport access vlan 20
Switch(config)#interface range f0/10 - 12
Switch(config-if-range)#switchport mode access
Switch(config-if-range)#switchport access vlan 30
```



3. Kiểm tra cấu hình:

Thực hiện các câu lệnh sau để kiểm tra cấu hình

```
Switch#show run
Switch#show vlan [brief]
```

Gắn PC vào các cổng như trên sơ đồ, đặt IP cho các PC và dùng lệnh “ping” để kiểm tra kết nối.

Các PC gắn SW cùng vlan sẽ thấy (ping) nhau, khác vlan sẽ ko thấy nhau

Java Simplified/ Session 22 / 67/94

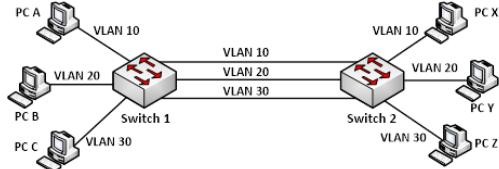
67

VLAN trên nhiều Switch

Hai cách kết nối:

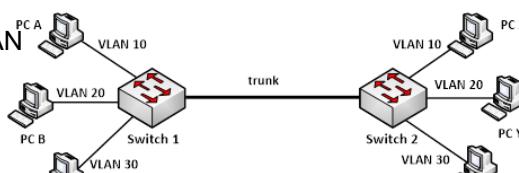
- Kết nối riêng: Mỗi VLAN ở trên các switch sẽ được kết nối bằng một đường kết nối riêng.

=> Lãng phí



- Trunking: dữ liệu của các VLAN có thể cùng lưu thông

=> tiết kiệm



Các thành viên

- trong cùng VLAN ở các Switch khác nhau vẫn có thể giao tiếp với nhau
- Khác VLAN (cùng Switch hay khác Switch) không thể giao tiếp với nhau

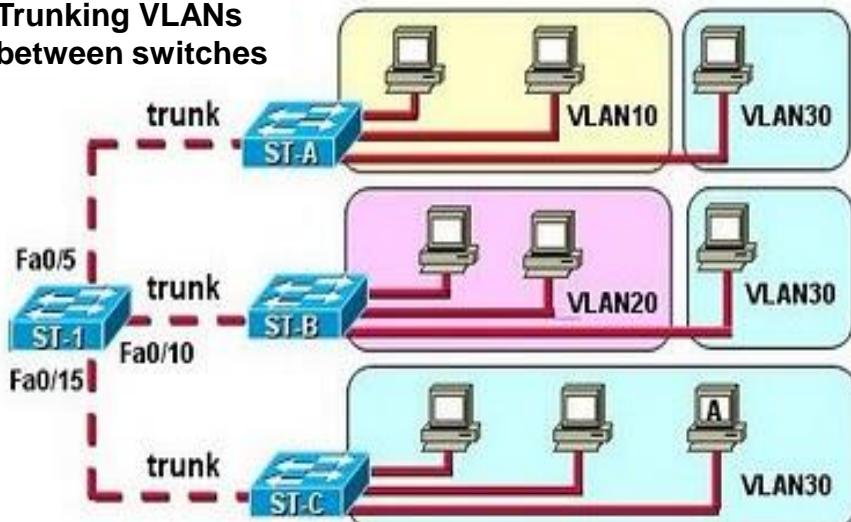
=> cần định tuyến

Java Simplified/ Session 22 / 68/94

68

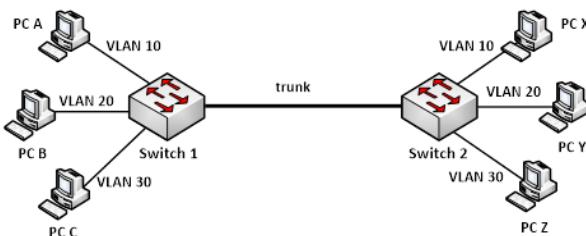
Trunking VLANs between switches

Trunking VLANs between switches



Java Simplified / Session 22 / 69

Cấu hình VLAN trunking trên Switch



- Kết nối 2 SW qua cable **cross-over**. Xác định cổng trunk: ex, f0/15 trên 2 SW

- Cấu hình 2 cổng “trunk” ở 2 SW như sau:

```

switch(config)#interface <interface>
switch(config-if)#switchport mode trunk encapsulation dot1q
or   switch(config-if)#switchport mode trunk
      switch(config-if)#switchport trunk native vlan <id>
  
```

Update configuration:

Switch#**show run**

Lệnh xem trunking:

Switch#**show interfaces trunk**

Java Simplified / Session 22 / 70 70

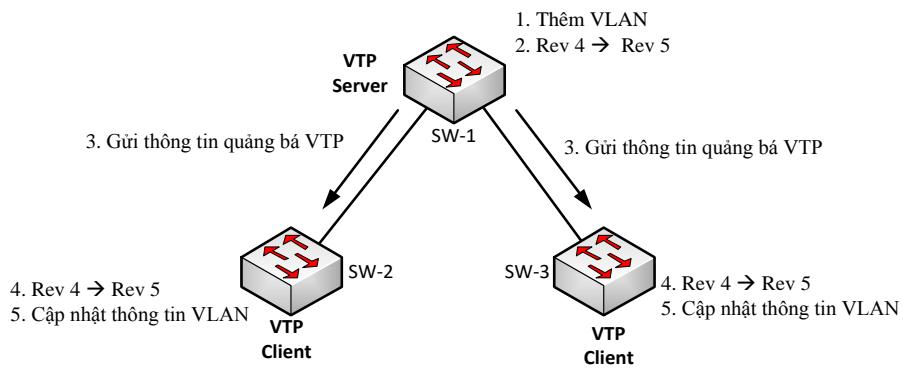
VLAN Trunking Protocol – VTP

- ▶ Giao thức mạch nối các VLAN (VTP):
 - Quản lý việc lưu chuyển các Frame từ các VLAN khác nhau trên một đường truyền vật lý.
 - Thiết lập các thoả thuận cho việc sắp xếp các Frame vào các cổng được liên kết với nhau ở hai đầu đường trunk.
 - Có 2 kỹ thuật Trunking là Frame Filtering và Frame Tagging.
 - Các tag được thêm vào trên đường gói tin đi ra vào đường trunk và được bỏ đi khi ra khỏi đường trunk.
 - Các gói tin có gắn tag không phải là gói tin Broadcast.
- Để lưu trữ, mỗi Frame được gắn tag để nhận dạng trước khi gửi đi, Frame của VLAN nào thì đi về VLAN đó.

Java Simplified / Session 22 / 71 of 45

VLAN Trunking Protocol – VTP

- ▶ Hoạt động của VTP



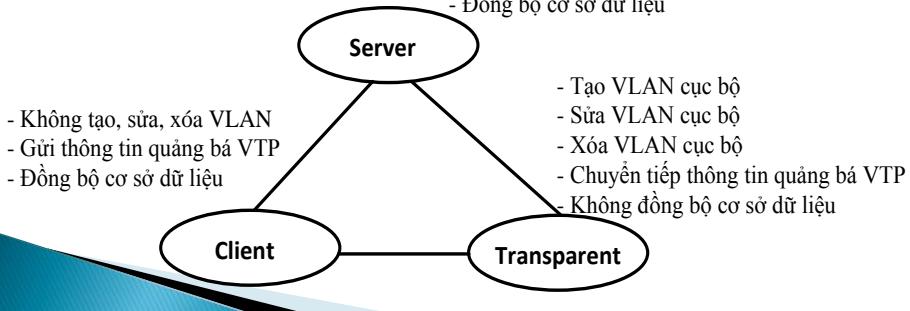
Java Simplified / Session 22 / 72 of 45

VLAN Trunking Protocol – VTP

▶ VTP hoạt động ở một trong ba chế độ

- Server
- Client
- Transparent

- Tạo VLAN
- Sửa VLAN
- Xóa VLAN
- Gửi thông tin quảng bá VTP
- Đồng bộ cơ sở dữ liệu



Java Simplified / Session 22 / 73

73

Lợi ích của VTP

▶ Các kết nối của VLAN sẽ bị chồng chéo khi các VLAN bị đặt trùng tên. Các cấu hình không đúng này có thể bị cắt kết nối khi chúng được ánh xạ từ một kiểu LAN tới một kiểu LAN khác.

▶ VTP cung cấp các lợi ích sau:

- Cấu hình đúng các VLAN qua mạng
- Hệ thống ánh xạ cho phép 1 VLAN được trunk qua các môi trường truyền hỗn hợp.

Giống như ánh xạ các VLAN Ethernet tới đường cáp trực tốc độ cao như ATM hoặc FDDI.

- Theo dõi chính xác và kiểm tra VLAN
- Báo cáo động về việc thêm vào các VLAN
- Dễ dàng cấu hình khi thêm mới VLAN

Java Simplified / Session 22 / 74

74

Cấu hình VTP

Cấu hình VTP domain

```
Switch(config) #vtp domain <domain_name>
```

Cấu hình VTP mode

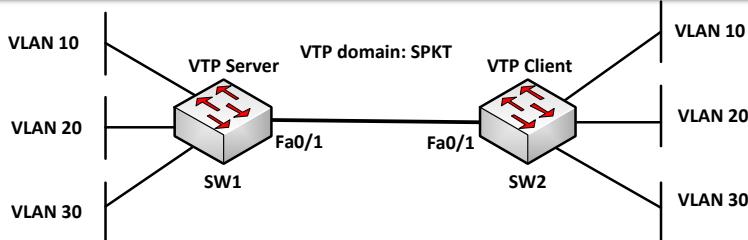
```
Switch(config) #vtp [client| transparent| server]
```

Lệnh xem cấu hình VTP

```
Switch#show vtp status
```

Java Simplified / Session 22 / 75 75

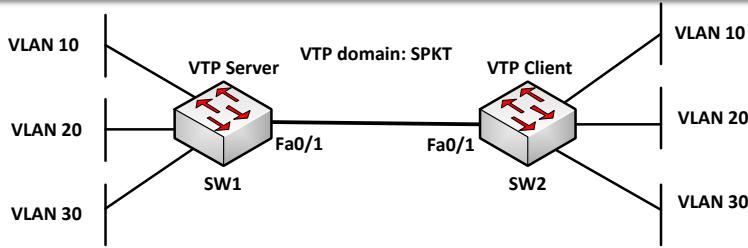
Cấu hình VTP



- ▶ Hai switch kết nối với nhau qua đường “trunk”.
- ▶ Tạo 3 vlan: VLAN 10, VLAN 20, VLAN 30 trên SW1
- ▶ Cấu hình VTP để các thông tin các VLAN trên SW1 cập nhật cho SW2
- ▶ Trên SW1:
 - VLAN 10 (Fa0/2 – Fa0/4), VLAN 20 (Fa0/5 – Fa0/7), VLAN 30 (Fa0/8 – Fa0/10)
- ▶ Trên SW2:
 - VLAN 10 (Fa0/4 – Fa0/6), VLAN 20 (Fa0/7 – Fa0/9), VLAN 30 (Fa0/10 – Fa0/12)

Java Simplified / Session 22 / 76 76

Cấu hình VTP



- ▶ Hai switch kết nối với nhau qua đường “trunk”.
- ▶ Tạo 3 vlan: VLAN 10, VLAN 20, VLAN 30 trên SW1
- ▶ Cấu hình VTP để các thông tin các VLAN trên SW1 cập nhật cho SW2
 - Trên SW1: (Thiết lập VTP domain: SPKT, VTP mode Server, và tạo các VLAN)
 - VLAN 10 (Fa0/2 – Fa0/4), VLAN 20 (Fa0/5 – Fa0/7), VLAN 30 (Fa0/8 – Fa0/10)
 - Trên SW2: (Cấu hình vtp domain: SPKT, vtp mode: client)
 - VLAN 10 (Fa0/4 – Fa0/6), VLAN 20 (Fa0/7 – Fa0/9), VLAN 30 (Fa0/10 – Fa0/12)

Java Simplified / Session 22 / 77/78 77

Cấu hình port security on Switch

- ▶ Configure port security on the Switch
- ▶ Ex Configure ports fa0/6, so that it allows only a single host and learn the MAC address of the host dynamically.

```
SW(config)#interface fa0/6
SW(config-if)#switchport port-security
SW(config-if)#switchport port-security maximum 1
SW(config-if)#switchport port-security mac-address sticky
```

Java Simplified / Session 22 / 78 78

Định tuyến giữa các VLAN

- ▶ Mỗi VLAN là một miền quảng bá.

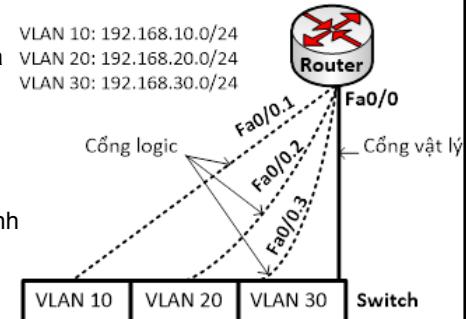
 - Một máy tính trong một VLAN muốn liên lạc với một máy tính thuộc một VLAN khác thì nó phải thông qua thiết bị định tuyến như là router, Switch layer 3.

- ▶ Việc sử dụng 1 cổng trên router kết nối với một cổng trên switch và cấu hình đường trunk (trunk layer 3) để định tuyến cho các VLAN.
 - Đường kết nối cho phép mang lưu lượng của nhiều VLAN
 - Nó không phải là của riêng VLAN nào.
 - Trunking layer 3 đòi hỏi cổng trên VLAN phải có thể hoạt động ở tốc độ FastEthernet trở lên.

Java Simplified / Session 22 / 79

Cổng vật lý và cổng logic trên Router

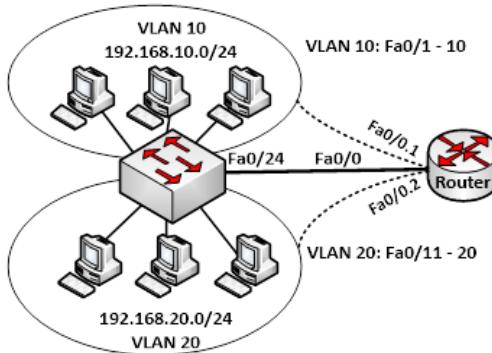
- ▶ Đường “trunk” có ưu điểm:
 - giảm số lượng cổng cần sử dụng của router và switch.
 - tiết kiệm chi phí
 - giúp cho cấu hình bớt phức tạp.
- ▶ Cổng trên Router:
 - Một cổng vật lý có thể được chia thành nhiều cổng luận lý.
 - Mỗi cổng luận lý tương ứng với một VLAN và được đặt một địa chỉ IP của VLAN đó.
 - Mỗi VLAN là một mạng riêng, do đó cổng luận lý thuộc VLAN nào thì có địa chỉ IP thuộc mạng của VLAN đó.



Cấu hình một giao diện của router ở mode trunk link sử dụng subinterface (logic port):
encapsulation dot1q vlan id

Java Simplified / Session 22 / 80

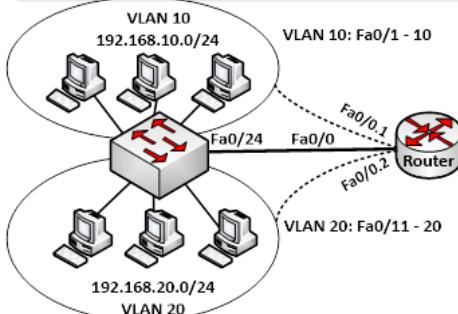
Cấu hình định tuyến VLAN trên router



- ▶ Trên Switch:
 - Tạo 2 VLAN: VLAN 10 và VLAN 20
 - Gán port: 10: fa0/1 – 10, 20: fa0/11 - 20
 - Tạo đường trunk: **switchport mode trunk**
- ▶ Trên router:
 - cấu hình định tuyến cho VLAN10 và VLAN20 giao tiếp với nhau

Java Simplified / Session 22 / 81/84

Cấu hình định tuyến VLAN trên router



- ▶ Trên Switch:
 - Tạo 2 VLAN: VLAN 10 và VLAN 20
 - Gán port: 10: fa0/1 – 10, 20: fa0/11 - 20
 - Tạo đường trunk: **switchport mode trunk**
- ▶ Trên router:
 - cấu hình định tuyến cho VLAN10 và VLAN20 giao tiếp với nhau như sau:

- ▶ Chọn cổng fa0/0 để cấu hình trunk


```
router(config)#interface fa0/0
router(config-if)#no shutdown
```
- ▶ Kích hoạt trunk trên subinterface fa0/0.1 và đóng gói bằng dot1q


```
router(config)#int fa0/0.1
router(config-if)#encapsulation dot1q 10
```
- ▶ Cấu hình thông tin lớp 3 cho sub-interface **fa0/0.1**

```
router(config-subif)#ip address 192.168.10.1 255.255.255.0
```
- ▶ Kích hoạt "trunk" trên sub-interface **fa0/0.2** và đóng gói bằng **dot1q**

```
router(config)#int fa0/0.2
router(config-subif)#encapsulation dot1q 20
```
- ▶ Cấu hình thông tin lớp 3 cho sub-interface **fa0/0.2**

```
router(config-subif)#ip address 192.168.20.1 255.255.255.0
```

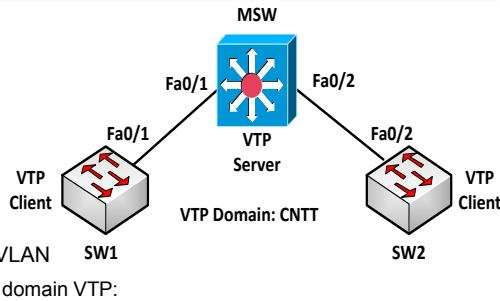
Định tuyến cho các VLAN dùng switch layer 3

▶ Tạo:

VLAN10: 192.168.10.0/24
 VLAN20: 192.168.20.0/24
 VLAN30: 192.168.30.0/24
 VLAN40: 192.168.40.0/24

▶ Yêu cầu:

- Cấu hình đường Trunk
- Cấu hình VTP
- Cấu hình MSW để định tuyến cho 4 VLAN



▶ Trên SW1,2, cấu hình trunk và VTP client, domain VTP:

```

SW1(config)#interface fa0/1
SW1(config-if)#switchport mode trunk
SW1(config-if)#switchport trunk encapsulation dot1q
SW1(config)#vtp domain CNTT
SW1(config)#vtp mode client

MSW(config)#interface fa0/1
MSW(config-if)#switchport mode trunk
MSW(config-if)#switchport trunk encapsulation dot1q
MSW(config)#interface fa0/2
MSW(config-if)#switchport mode trunk
MSW(config-if)#switchport trunk encapsulation dot1q
    
```

Java Simplified / Session 22 / 83/83 83

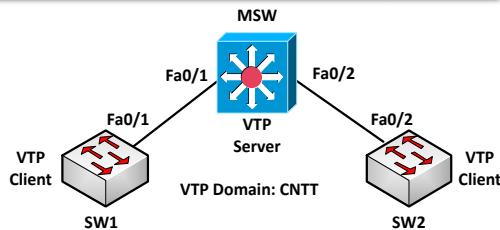
Định tuyến cho các VLAN dùng switch layer 3

▶ Trên Switch MSW: cấu hình trunk

Cấu hình VTP, VLAN

```

MSW(config)#vtp domain CNTT
MSW(config)#vtp mode server
MSW(config)#vlan 10
MSW(config)#vlan 20
MSW(config)#vlan 30
MSW(config)#vlan 40
    
```



▶ Cấu hình Switch MSW để routing giữa 4 VLAN

```

MSW(config)#ip routing
MSW(config)#interface vlan 10
MSW(config-if)#ip address 192.168.10.1 255.255.255.0
MSW(config)#interface vlan 20
MSW(config-if)#ip address 192.168.20.1 255.255.255.0
MSW(config)#interface vlan 30
MSW(config-if)#ip address 192.168.30.1 255.255.255.0
MSW(config)#interface vlan 40
MSW(config-if)#ip address 192.168.40.1 255.255.255.0
    
```

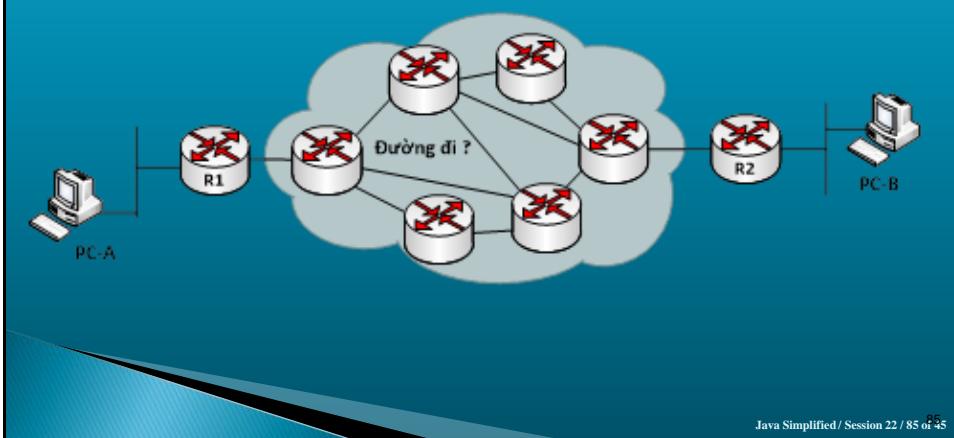
▶ Kiểm tra cấu hình

```

show interface trunk
show vtp status
show vlan brief
show ip route
    
```

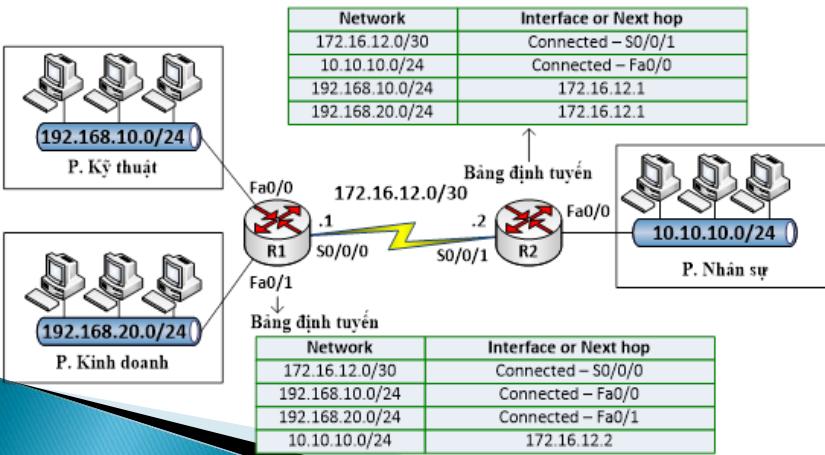
Java Simplified / Session 22 / 84/84 84

Routing



Router

- Router dựa vào địa chỉ IP đích (destination IP) trong các gói tin và sử dụng bảng định tuyến (routing table) để xác định đường đi cho chúng



Phân loại định tuyến

▶ Định tuyến tĩnh

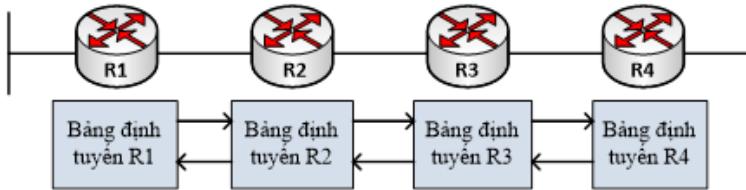
- là loại định tuyến mà trong đó router sử dụng các tuyến đường đi tĩnh để vận chuyển dữ liệu đi.
- Các tuyến đường đi tĩnh này có được do người quản trị cấu hình thủ công vào các router.

▶ Định tuyến động

- là loại định tuyến mà trong đó router sử dụng các tuyến đường đi động để vận chuyển dữ liệu đi.
- Các tuyến đường đi động này có được do các router sử dụng các giao thức định tuyến động trao đổi thông tin định tuyến với nhau tạo ra.
- Ex: RIP, OSPF, BGP,...
- 2 loại: *distance-vector* và *link-state*

Java Simplified / Session 22 / 87/94

distance-vector (RIP)



▶ Các router định tuyến loại “distance vector”

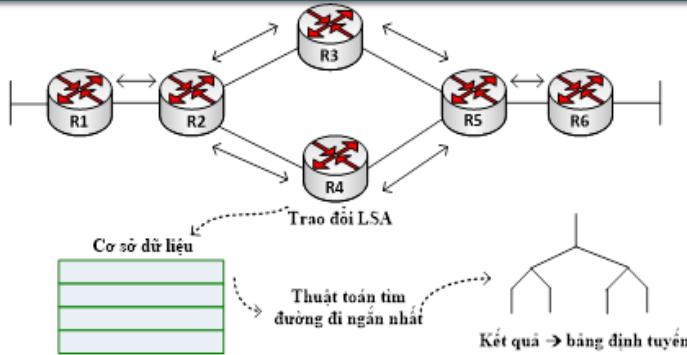
- thực hiện gửi định kỳ toàn bộ bảng định tuyến của mình và chỉ gửi cho các router láng giềng kết nối trực tiếp với mình. => nên tồn tại nhiều bảng thông tin đường truyền
- không biết được đường đi đến đích một cách cụ thể, không biết về các router trung gian trên đường đi và cấu trúc kết nối giữa chúng.

▶ Bảng định tuyến:

- là nơi lưu kết quả chọn đường tốt nhất của mỗi router.
- khi chúng trao đổi bảng định tuyến với nhau, các router chọn đường dựa trên kết quả đã chọn của router láng giềng.
- Mỗi router nhìn hệ thống mạng theo sự chi phối của các router láng giềng.

Java Simplified / Session 22 / 88

Link – state (OSPF, IS-IS)



▶ Các router định tuyến dạng link-state:

- sẽ trao đổi các LSA (*link state advertisement*) với nhau để xây dựng và duy trì cơ sở dữ liệu về trạng thái các đường liên kết
- dùng thuật toán tìm đường đi ngắn nhất (SPF - Shortest Path First) để tính toán chọn đường đi tốt nhất đến từng mạng đích.
- chỉ cập nhật khi nào có sự thay đổi xảy ra. Do đó, thời gian hội tụ nhanh và ít tốn băng thông.

Java Simplified / Session 22 / 89/94 89

Metric và AD

▶ Hai tham số dùng trong định tuyến

◦ Metric:

Là tham số được sử dụng để chọn đường tốt nhất cho việc định tuyến. Đây là giá trị mà bất kỳ giao thức định tuyến nào cũng phải dùng để tính toán đường đi đến mạng đích.

Trong trường hợp có nhiều đường đi đến một mạng đích thì đường đi nào có metric thấp nhất sẽ được lựa chọn để đưa vào bảng định tuyến. Mỗi giao thức định tuyến có một kiểu metric khác nhau.

◦ AD - Administrative Distance

là giá trị quy ước dùng để chỉ độ tin cậy của các giao thức định tuyến, giao thức nào có AD nhỏ hơn sẽ được xem là đáng tin cậy hơn.

Trong trường hợp router học được một mạng đích thông qua nhiều giao thức định tuyến khác nhau, thì tuyến của giao thức định tuyến nào có AD nhỏ nhất thì sẽ được lựa chọn và đưa vào bảng định tuyến.

Java Simplified / Session 22 / 90/94 90

Metric và AD

- ▶ Giá trị AD mặc định

Route Type	Administrative Distance
Connected	0
Static	1
BGP (external routes)	20
EIGRP (internal routes)	90
IGRP	100
OSPF	110
IS-IS	115
RIP	120
EIGRP (external routes)	170
BGP (internal routes)	200
Unusable	255

- ▶ Giá trị AD có thể chỉnh sửa trong câu lệnh cho router học route

```
Router(config)#ip route 10.10.10.0 255.255.255.0 fa0/0 ?
<1-255>  Distance metric for this route
<cr>
```

91

cấu hình định tuyến tĩnh

- ▶ Cú pháp sau:

```
Router(config)#ip route <Destination> <Subnet-mask>
{Next-hop |Outgoing -interface}
```

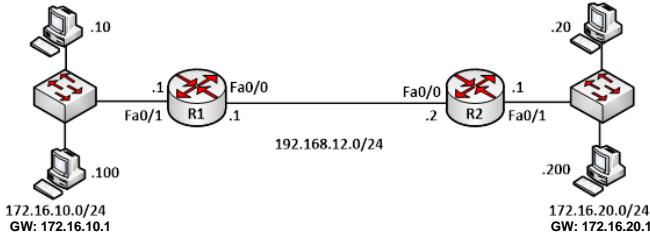
- ▶ Trong đó:

- *Destination-network*: là địa chỉ mạng đích cần đi tới
- *Subnet-mask*: Subnet mask của *destination-network*
- *Next-hop-address*: địa chỉ IP của cổng trên router kế tiếp có kết nối trực tiếp với router đang xét.
- *Out-bound-interface*: cổng của router sẽ gửi dữ liệu ra
- *Distance*: thay đổi giá trị AD cho tuyến này. Mặc định các tuyến tĩnh có AD=1.

Java Simplified / Session 22 / 92

92

cấu hình định tuyến tĩnh



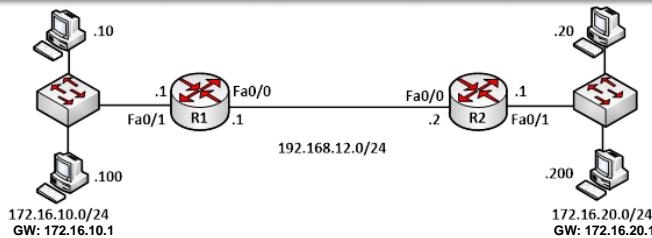
➤ Thực hiện:

- PCs: gán IP và GW theo sơ đồ
- Router R1:


```
Router>enable
Router#configure terminal
Router(config)#hostname R01
R01(config)#
R01(config)#interface fa0/1
R01(config-if)#ip address 172.16.10.1 255.255.255.0
R01(config-if)#no shutdown
R01(config)#interface fa0/0
R01(config-if)#ip address 192.168.12.1 255.255.255.0
R01(config-if)#no shutdown
```
- Router R2: tương tự, IP: 172.16.20.1/24, 192.168.12.2/24
(dùng ping để check kết nối từng đoạn)

93

cấu hình định tuyến tĩnh



➤ Router R1: cấu hình định tuyến tĩnh như sau:

```
R1(config)#ip route 172.16.20.0 255.255.255.0 fa0/0
  hoặc R1(config)#ip route 172.16.20.0 255.255.255.0 192.168.12.2
```

➤ Router R2: tương tự router R1,

```
R2(config)#ip route 172.16.10.0 255.255.255.0 fa0/0
  hoặc R2(config)#ip route 172.16.10.0 255.255.255.0 192.168.12.1
```

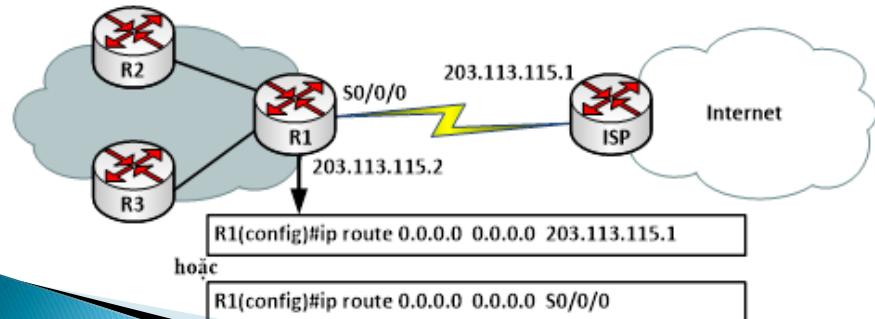
➤ Kiểm tra: Router#show ip route

➤ Test: Tại PC1, ping đến PC2, hoặc tracert 172.16.20.200

1	0 ms	11 ms	4 ms	172.16.10.1
2	12 ms	11 ms	2 ms	192.168.12.1
3	12 ms	13 ms	14 ms	172.16.20.200

Cấu hình Default route

- ▶ Default route nằm ở cuối bảng định tuyến và được sử dụng để gửi các gói tin đi trong trường hợp mạng đích không tìm thấy trong bảng định tuyến.
- ▶ Nó rất hữu dụng trong các mạng dạng “stub network” như kết nối từ mạng nội bộ ra ngoài Internet.



Java Simplified / Session 22 / 95

Cấu hình định tuyến động - RIP

- ▶ RIP: RIP có hai phiên bản là RIPv1 và RIPv2.
 - là một giao thức định tuyến theo kiểu “*distance-vector*”.
 - “*Hop count*” được sử dụng làm *metric* cho việc chọn đường
 - Giá trị *AD* mặc định của RIP là 120.

Cấu hình

- Khởi tạo tiến trình định tuyến RIP

```
Router(config)#router rip
```

- Bật chế độ RIPv2

```
Router(config-router)#version 2 //sử dụng cho RIPv2
```

- Chọn cổng tham gia vào quá trình trao đổi thông tin định tuyến

```
Router(config-router)#network <major-classful-network>
```

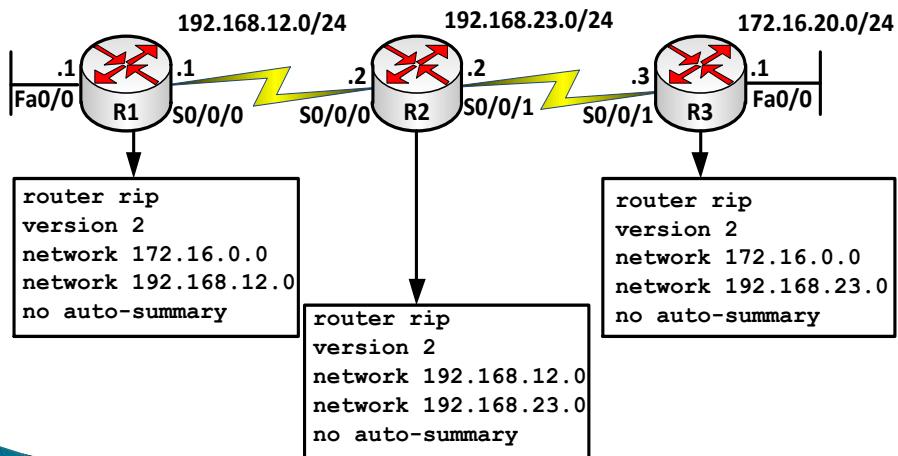
- Tắt tính năng tự động tóm tắt các tuyến

```
Router(config-router)#no auto-summary //sử dụng cho RIPv2
```

Java Simplified / Session 22 / 96

Cấu hình định tuyến động - RIP

Ex:



Java Simplified / Session 22 / 97/98

Cấu hình chứng thực trong định tuyến RIPv2



Dạng plain text

```

R1(config)#key chain aaaa
R1(config-keychain)#key 1
R1(config-keychain-key)#key-string ccna
R1(config)#interface S0/0/0
R1(config-if)#ip rip authentication key-chain aaaa
R2(config)#key chain bbbb
R2(config-keychain)#key 1
R2(config-keychain-key)#key-string ccna
R2(config)#interface S0/0/0
R2(config-if)#ip rip authentication key-chain bbbb
    
```

Dạng MD5: thêm dòng:

```

R1(config-if)#ip rip authentication mode md5
    
```

Java Simplified / Session 22 / 98

98

Cấu hình định tuyến động – OSPF

❖ OSPF (Open Shortest Path First):

- là một giao thức định tuyến dạng *link-state*, sử dụng thuật toán Dijkstra để xây dựng bảng định tuyến.
- có đặc điểm của giao thức *link-state*.
- Nó có ưu điểm là hội tụ nhanh, hỗ trợ được mạng có kích thước lớn và không xảy ra “*routing loop*”.
- OSPF hỗ trợ chứng thực dạng *Plain-Text* và dạng *MD5*.
- OSPF sử dụng *metric* là *cost*. (cộng dồn *cost* dọc theo tuyến đường đi của packet)

❖ Các loại môi trường OSPF

- Multiple access (ethernet)
- Point-to-Point
- NBMA (Non-Broadcast Multiple Access)

Java Simplified / Session 22 / 99 of 99

Cấu hình định tuyến động – OSPF

❖ Cấu hình OSPF

- Khởi tạo tiến trình định tuyến OSPF

```
Router(config)#router ospf <process-id>
```

- Chọn cổng tham gia vào quá trình trao đổi thông tin định tuyến

```
Router(config-router)#network <address> <wildcard> area <area-id>
```

- Trong đó:

- *Process-id*: chỉ số tiến trình của OSPF, mang tính chất cục bộ, có giá trị 1 đến 65535.
- *Address*: địa chỉ cổng tham gia định tuyến
- *Wildcard*: điều kiện kiểm tra giữa địa chỉ cấu hình trong address và địa chỉ các cổng trên router, tương ứng bit 0 – phải so khớp, bit 1 – không cần kiểm tra.
- *Area-id*: vùng mà cổng tương ứng thuộc về trong kiến trúc OSPF.

❖ Các câu lệnh kiểm tra cấu hình OSPF

```
Router#show ip protocol
```

```
Router#show ip route
```

```
Router#show ip ospf interface
```

```
Router#show ip ospf neighbor
```

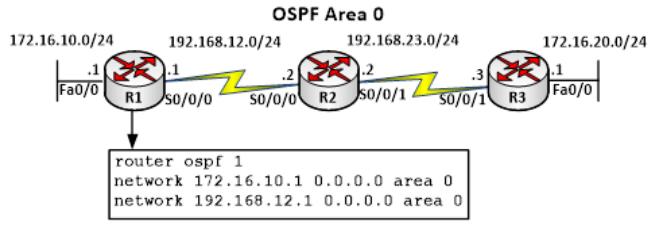
```
Router#debug ip ospf events
```

```
Router#debug ip ospf packet
```

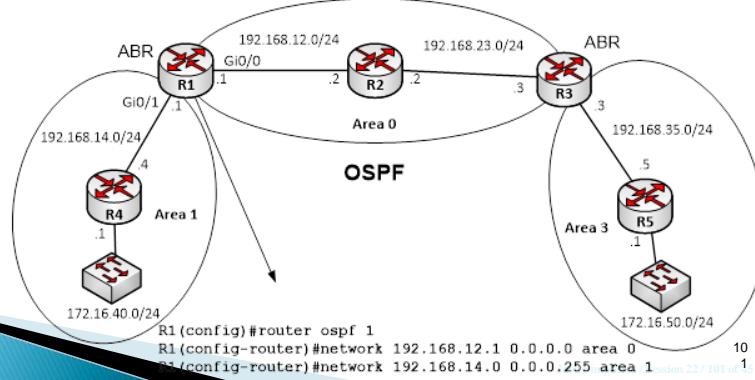
10
Java Simplified / Session 22 / 100 of 0

Cấu hình định tuyến động – OSPF

- ▶ Single area



- ▶ Multi-area



Cấu hình chứng thực trong định tuyến động

- Chứng thực bằng “Plain Text”

```

R(config)#interface <interface>
R(config-if)#ip ospf authentication
R(config-if)#ip ospf authentication-key <password>

```

- Chứng thực bằng MD5

```

R(config)#interface <interface>
R(config-if)#ip ospf authentication message-digest
R(config-if)#ip ospf messages-digest-key 1 md5 <pass>

```

EIGRP Enhanced Interior Gateway Routing Protocol

- EIGRP là giao thức định tuyến do Cisco tạo ra,
- EIGRP là một giao thức định tuyến lai: "distance vector" và "link-state".
- Cách tính metric của EIGRP

$$\text{metric}_{\text{EIGRP}} = \left[K1 * \text{BW} + \frac{K2 * \text{BW}}{(256 - \text{load})} + K3 * \text{Delay} \right] * \frac{K5}{(\text{reliability} + K4)}$$

- Với K1, K2, K3, K4, K5 là hằng số
- Mặc định: K1=1, K2=0, K3=1, K4=0, K5=0

- Do đó, ta có: metric = bandwidth + delay

Cấu hình EIGRP

- **Bước 1.** Kích hoạt giao thức định tuyến EIGRP

```
Router(config) #router eigrp <autonomous-system>
```

Trong đó: autonomous-system: có giá trị từ 1 đến 65535, giá trị này phải giống nhau ở tất cả các router trong hệ thống chạy EIGRP

- **Bước 2.** Chọn cổng tham gia vào quá trình trao đổi thông tin định tuyến

```
Router(config-router) #network <network-number>
```

Trong đó: network-number là địa chỉ cổng theo đúng lớp mạng của nó.

Để quảng bá các mạng con và hỗ trợ mạng không liên tục, chúng ta phải sử dụng lệnh sau:

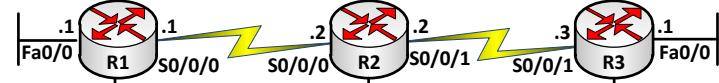
```
Router(config-router) #no auto-summary
```

10

Java Simplified / Session 22 / 103 of 3

EIGRP Enhanced Interior Gateway Routing Protocol

- Ex:



```
router eigrp 100
network 172.16.0.0
network 192.168.12.0
no auto-summary
```

```
router eigrp 100
network 192.168.12.0
network 192.168.23.0
no auto-summary
```

- Cấu hình chứng thực:

```
R1(config)#key chain my_keychain1
R1(config-keychain)#key 1
R1(config-keychain-key)#key-string cisco

R1(config)#interface fa0/0
R1(config-if)#ip authentication mode eigrp 100 md5
R1(config-if)#ip authentication key-chain eigrp 100 my_keychain1
```

Java Simplified / Session 22 / 104 of 104

Summary

- Bridges & Bridging
 - Transparent Bridging
 - Source route bridging
 - Translational Bridging
- Redundancy & Broadcast Storm
- STP
- VLAN
- VTP
- Routing: RIP, OSPF, EIGRP

10

Java Simplified / Session 22 / 105 of 5

LAB

- ▶ Configuration
 - VLAN
 - VTP
 - Routing: static, RIP, OSPF

10

Java Simplified / Session 22 / 106 of 6