

## BUỔI THỰC HÀNH 3

### Mục đích:

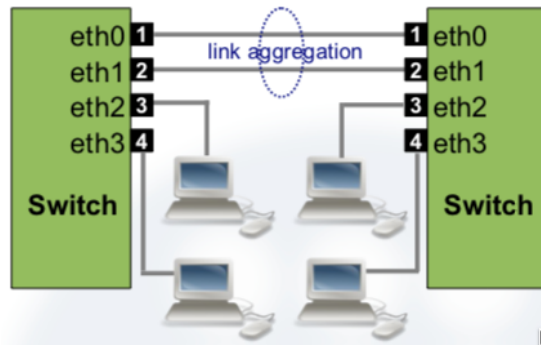
- Khảo sát hoạt động của Switch.
- Mô phỏng liên mạng với Switch và Router.

### I. GIỚI THIỆU VỀ LINUX BRIDGE

Linux Bridge là một *soft switch*, một công nghệ cung cấp *Switch ảo* trong hệ thống Linux giúp các máy ảo Netkit trong mạng ảo giao tiếp được với nhau và giao tiếp với mạng bên ngoài (mạng thực) trong môi trường máy thực (Ubuntu 16.04).

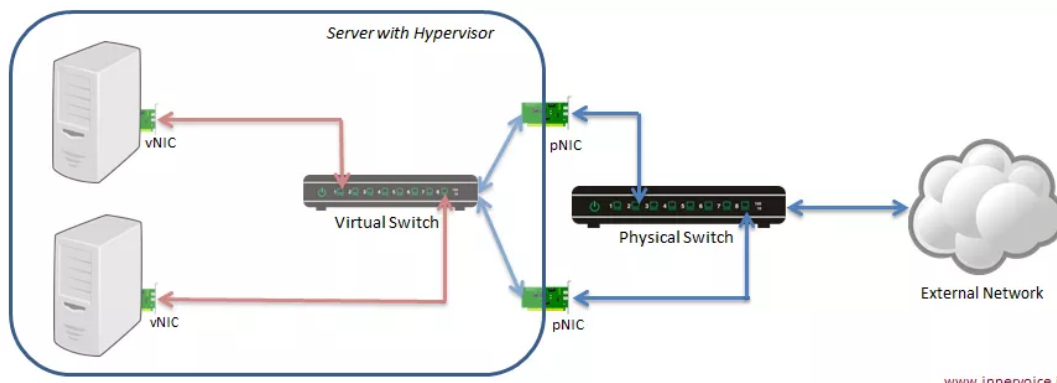
Các Switch ảo mà Linux Bridge tạo ra sẽ *hoạt động ở tầng 2 (Data Link Layer) trong mô hình OSI*. Một Switch ảo còn cho phép tách một mạng (Net hoặc Subnet) làm hai hoặc nhiều nhánh mạng (segments). Các máy ảo trong cùng 1 nhánh mạng dễ dàng truyền tải dữ liệu cho nhau theo hình thức quảng bá (broadcasting). Tuy nhiên đối với trường hợp truyền tải dữ liệu giữa 2 máy ảo khác nhánh mạng (dù thuộc chung Net/Subnet), thì *Switch ảo có nhiệm vụ kiểm tra đích đến (Destination) của từng khung dữ liệu (Data frame) và quyết định xem là khung sẽ được chuyển tiếp hoặc không đến nhánh mạng nào đó*. Việc kiểm tra của Switch được thực hiện nhờ vào phương pháp MAC lookup table (Bảng lưu trữ địa chỉ vật lý các máy tính đã từng gửi dữ liệu lưu thông qua Switch).

Như vậy, *Switch cũng là một thiết bị liên mạng (cho phép kết nối các mạng) giống như Router nhưng hoạt động nhanh hơn và ít dụng độ hơn*.



Hầu hết các công cụ mô phỏng mạng trên nền tảng Linux, chẳng hạn: KVM, QEMU, Netkit, Kathara...đều được cài đặt dưới dạng là một module trong kernel để có thể sử dụng Linux Bridge.

Cấu trúc hệ thống sử dụng Linux Bridge được minh họa như sau:



Trong đó các khái niệm được định nghĩa:

- *Host with Hypervisor*: Máy thực cài đặt công cụ mô phỏng mạng Netkit, ở đây là Ubuntu 16.04
- *Virtual NIC (vNIC)*: các giao diện mạng của các máy ảo Netkit được tạo trong máy thực
- *Physical switch*: là thiết bị Switch thực cho phép nhiều máy tính thực kết nối lại với nhau qua các cổng vật lý (Physical port)
- *Virtual switch*: là thiết bị Switch ảo (phần mềm) chứa các cổng ảo (Virtual port) kết nối các máy tính ảo Netkit qua cáp ảo (Virtual cable)

Các tính năng mà một Linux Bridge cung cấp bao gồm:

- STP: Spanning Tree Protocol – một giao thức chống việc các gói tin bị gửi lặp lại (loop) trong một mạng.
- VLAN: cho phép trên một Switch ảo quản lý được các mạng LAN ảo (Virtual LAN). Các VLANs có đặc tính cô lập về dữ liệu lưu thông dù được kết nối vào cùng một thiết bị liên mạng là Switch.
- FDB: hỗ trợ chuyển tiếp các gói tin theo dạng cơ sở dữ liệu (Forward Database) nhằm nâng cao hiệu năng của Switch.

Các lệnh của Linux Bridge được thao tác bởi **brctl** và có thể tham khảo theo link dưới đây: <http://man7.org/linux/man-pages/man8/bridge.8.html>

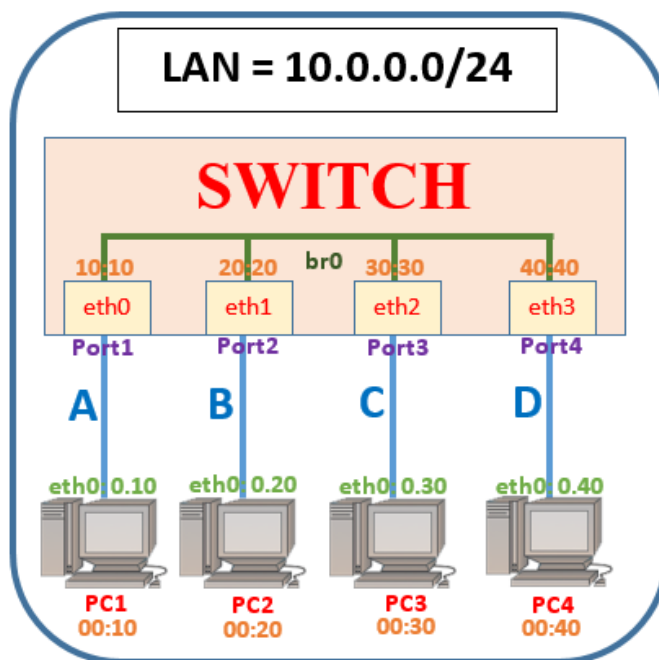
## II. BÀI TẬP THỰC HÀNH

**BÀI TẬP 8:** Mạng LAN phân nhánh với 1 Switch ảo và giao thức ARP

♦ **Bước 1:** Quan sát mô hình mạng cần xây dựng. Nhận diện các thiết bị (PC, Switch...), giao diện (eth0, eth1...) với các địa chỉ IP được gán.

**Vùng dụng độ (Collision domain) A, B và C là các nhánh mạng (segment) thuộc mạng LAN 10.0.0.0/24**  
Các địa chỉ được highlight màu vàng là phần sau của một địa chỉ vật lý (MAC). **Địa chỉ MAC hiển thị đầy đủ như sau:** 00:00:00:00:10:00

**Trong bài tập này chúng ta đặt địa chỉ MAC trên giao diện của các máy ảo để dễ ghi nhớ và theo dõi khi quan sát việc cập nhật MAC Lookup Table trên Switch ảo.**



♦ **Bước 2:** Xây dựng cấu trúc thư mục mạng ảo (nằm dưới *workspace /home/student/your\_workspace*) với đầy đủ các thư mục con và các file cấu hình (*.startup*, *lab.conf*). Thư mục mạng ảo đặt tên là **BaiTap8**. Lưu ý: **Switch ảo cũng là các máy ảo Netkit**. Sinh viên có thể tham khảo cấu trúc thư mục được Giảng viên hướng dẫn.

♦ **Bước 3:** Trên file *lab.conf*, soạn thảo nội dung mô tả hình thái mạng theo thiết kế. Nội dung tham khảo:

```
PC1[0]=A
PC2[0]=B
PC3[0]=C
PC4[0]=D
Switch[0]=A
Switch[1]=B
Switch[2]=C
Switch[3]=D
```

♦ **Bước 4:** Lần lượt trên các file *PC1.startup*, *PC2.startup*, *PC3.startup* và *PC4.startup* soạn thảo nội dung cấu hình cho giao diện mạng eth0 của các máy tính PC1, PC2, PC3 và PC4. Nội dung *PC1.startup* tham khảo:

```
ifconfig eth0 10.0.0.10/24 up
ifconfig eth0 hw ether 00:00:00:00:00:10
```

♦ **Bước 5:** Trên file *Switch.startup* soạn thảo nội dung cấu hình cho giao diện mạng eth0. Nội dung *Switch.startup* tham khảo:

```
ifconfig eth0 up
ifconfig eth0 hw ether 00:00:00:00:10:10
ifconfig eth1 up
ifconfig eth1 hw ether 00:00:00:00:20:20
ifconfig eth2 up
ifconfig eth2 hw ether 00:00:00:00:30:30
ifconfig eth3 up
ifconfig eth3 hw ether 00:00:00:00:40:40
```

Mỗi giao diện (eth0, eth1...) sẽ là một port ảo trên Switch cho phép các máy ảo kết nối vào.

Có thể bỏ qua việc đặt IP trên các giao diện mạng của Switch ảo vì Switch hoạt động trên tầng 2 của OSI.

♦ **Bước 6:** Trên file *Switch.startup* bổ sung vào nội dung như dưới đây:

```
brctl addbr br0
brctl addif br0 eth0
brctl addif br0 eth1
brctl addif br0 eth2
brctl addif br0 eth3
brctl stp br0 on
ifconfig br0 up
```

**Ý nghĩa:** Tạo ra cầu nối (Bridge) **br0**. Cầu nối **br0** giúp cho các máy tính ở những phân nhánh vật lý khác nhau (A, B...) vẫn có thể truyền dữ liệu cho nhau được nhờ vào MAC Lookup Table trên Switch ảo. Giải thích cụ thể các lệnh:

- *addbr*: Tạo một cầu nối mới (br0, br1...) để nối 2 nhánh A, B, C và D với nhau

- *addif*: Đăng ký các giao diện (cổng) của Switch ảo vào cầu nối. Các cổng (giao diện) thuộc chung 1 cầu nối, chẳng hạn: br0, thì truyền dữ liệu cho nhau được.
- *stp <name\_of\_bridge> on*: Kích hoạt giải thuật STP trên một cầu nối (br0, br1...) của Switch ảo
- *ifconfig <name\_of\_bridge> on*: Kích hoạt cầu nối (br0, br1...)

◆ **Bước 7:** Khởi động mạng ảo **Bài Tập 8**. Trên máy ảo **Switch** kiểm tra nội dung của Mac Lookup Table bằng lệnh: **brctl showmacs br0**. Kết quả hiển thị là MAC Lookup Table mà **Switch** lưu trữ cho br0

- Bảng này có bao nhiêu địa chỉ vật lý?
- Các địa chỉ vật lý này là của các giao diện của máy ảo nào trong mạng LAN?

◆ **Bước 8:** Trên máy ảo **Switch**, **PC1** và **PC3** thực hiện lệnh:

```
tcpdump -e -q -w /hosthome/BT8_Switch.pcap
tcpdump -e -q -w /hosthome/BT8_PC1.pcap
tcpdump -e -q -w /hosthome/BT8_PC3.pcap
```

◆ **Bước 9:** Trên PC2, thực hiện lệnh **ping** đến PC3 (10.0.0.30) và chờ khoảng 10 giây. Sau đó dùng lệnh **ping** trên PC2 lại. Dùng lệnh **tcpdump** trên **Switch**, **PC1** và **PC3** lại.

◆ **Bước 10:** Trên **Switch** kiểm tra lại nội dung Mac Lookup Table bằng lệnh **brctl showmac br0**. Kết quả trên Switch thay đổi như thế nào. Ví dụ: Switch biết thêm địa chỉ vật lý của máy tính nào, ở port bao nhiêu?

◆ **Bước 11:** Trên máy thực Ubuntu 16.04

- Dùng Wireshark mở file **BT8\_Switch.pcap**. Chọn khung dữ liệu có giao thức ARP đến từ địa chỉ **MAC 00:00:00:00:00:20**
  - Tại sao **Switch** nhận được khung dữ liệu này?
  - Khung dữ liệu này có ý nghĩa như thế nào với **Switch**?
- Chọn khung dữ liệu có giao thức ARP đến từ địa chỉ **MAC 00:00:00:00:00:30**.
  - Tại sao **Switch** nhận được khung dữ liệu này?
  - Khung dữ liệu này có ý nghĩa như thế nào với **Switch**?
- Dùng Wireshark mở file **BT8\_PC1.pcap**. Chọn khung dữ liệu có giao thức ARP đến từ địa chỉ **MAC 00:00:00:00:00:20**
  - Tại sao **PC1** nhận được khung dữ liệu này?
  - Khung dữ liệu này có ý nghĩa như thế nào với **PC1**? **PC1** có hồi đáp cho khung này hay không?
- Dùng Wireshark mở file **BT8\_PC3.pcap**. Chọn khung dữ liệu có giao thức ARP đến từ địa chỉ **MAC 00:00:00:00:00:20**
  - Tại sao **PC3** nhận được khung dữ liệu này?
  - Khung dữ liệu này có ý nghĩa như thế nào với **PC3**? **PC3** có hồi đáp cho khung này hay không?

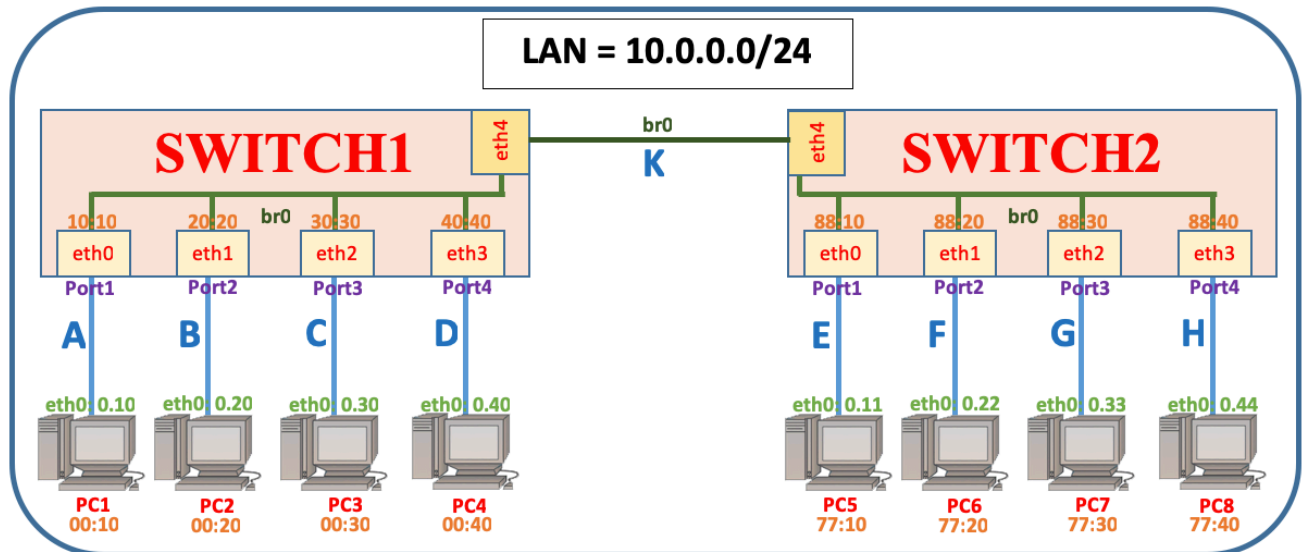
◆ **Bước 13:** Kết luận về hoạt động “**Học**” địa chỉ vật lý của các máy tính trong một mạng LAN của một Switch

◆ **Bước 14:** Hủy mạng ảo bằng lệnh **lcrash** sau khi đã thực hiện xong **Bài tập 8**

**Bài tập 9:** Mạng LAN phân nhánh với 2 Switch ảo

◆ **Bước 1:** Quan sát mô hình mạng cần xây dựng. Nhận diện các thiết bị (PC, Switch...), giao diện (eth0, eth1...) với các địa chỉ IP được gán.

**Nhận xét:** Đây là mô hình mạng mở rộng của Bài tập 8, vì vậy sinh viên có thể sử dụng lại kết quả của Bài tập 8 và phát triển thêm.



♦ **Bước 2:** Xây dựng cấu trúc thư mục mạng ảo (nằm dưới *workspace /home/student/your\_workspace*) với đầy đủ các thư mục con và các file cấu hình (*.startup*, *lab.conf*). Thư mục mạng ảo đặt tên là **BaiTap9**. Sinh viên có thể tham khảo cấu trúc thư mục được Giảng viên hướng dẫn.

♦ **Bước 3:** Trên file *lab.conf*, soạn thảo nội dung mô tả hình thái mạng theo thiết kế

♦ **Bước 4:** Lần lượt trên các file *.startup* của các **PC** soạn thảo nội dung cấu hình cho giao diện mạng eth0 của chúng.

♦ **Bước 5:** Lần lượt trên các file *Switch1.startup*, và *Switch2.startup* soạn thảo nội dung cấu hình cho giao diện mạng eth0 đồng thời thêm vào nội dung tạo ra cầu nối ảo (*br0*).

♦ **Bước 6:** Khởi động mạng ảo **BaiTap9**. Trên máy ảo *Switch1* và *Switch2* lần lượt kiểm tra nội dung của Mac Lookup Table bằng lệnh: *brctl showmacs br0*. Kết quả hiển thị là MAC Lookup Table mà *Switch1* và *Switch2* lưu trữ cho *br0*

#### A – Gửi khung dữ liệu (frame) giữa 2 máy tính cùng nối kết vào 1 Switch

♦ **Bước 7A:** Lần lượt trên máy ảo *Switch1* và *Switch2* thực hiện lệnh:

```
tcpdump -e -q -w /hosthome/BT9_Switch1_A.pcap trên máy ảo Switch1
tcpdump -e -q -w /hosthome/BT9_Switch2_A.pcap trên máy ảo Switch2
```

♦ **Bước 8A:** Trên PC1, thực hiện lệnh *ping* đến PC4 (10.0.0.40) và chờ khoảng 10 giây. Sau đó dừng lệnh *ping* trên PC1 lại. Dừng các lệnh *tcpdump* trên *Switch1* và *Switch2* lại.

♦ **Bước 9A:** Trên *Switch1* và *Switch2* lần lượt kiểm tra lại nội dung Mac Lookup Table bằng lệnh *brctl showmac br0*.

- *Switch1* có học được địa chỉ của cả 2 máy PC1 và PC4 hay không?
- *Switch2* có học được địa chỉ của cả 2 máy PC1 và PC4 hay không? Nếu không thì tại sao?
- Các máy tính từ *PC5* đến *PC8* có nhận được gói tin ICMP đến từ PC1 hay không? Nếu không thì các máy tính này chỉ nhận được gói tin có giao thức gì từ PC1?

#### B– Gửi khung dữ liệu (frame) giữa 2 máy tính nối kết trên 2 Switch khác nhau

♦ **Bước 7B:** Lần lượt trên máy ảo *Switch1* và *Switch2* thực hiện lệnh:

```
tcpdump -e -q -w /hosthome/BT9_Switch1_B.pcap trên máy ảo Switch1
tcpdump -e -q -w /hosthome/BT9_Switch2_B.pcap trên máy ảo Switch2
```

♦ **Bước 8B:** Trên **PC2**, thực hiện lệnh **ping** đến PC7 (10.0.0.33) và chờ khoảng 10 giây. Sau đó dừng lệnh **ping** trên **PC2** lại. Dừng các lệnh **tcpdump** trên **Switch1** và **Switch2** lại.

♦ **Bước 9B:** Trên **Switch1** và **Switch2** lần lượt kiểm tra lại nội dung Mac Lookup Table bằng lệnh **brctl showmac br0**.

- **Switch1** có học được địa chỉ của cả 2 máy PC2 và PC7 hay không?
- **Switch2** có học được địa chỉ của cả 2 máy PC2 và PC7 hay không? Nếu có thì tại sao?
- Các máy tính từ **PC5**, **PC6** và **PC8** có nhận được gói tin ICMP đến từ PC2 hay không? Hãy chứng minh điều đó.

♦ **Bước 13:** Kết luận về hoạt động “**Học**” địa chỉ vật lý của các máy tính trong một mạng (Net/Subnet) được nối kết bởi 2 Switch ảo.

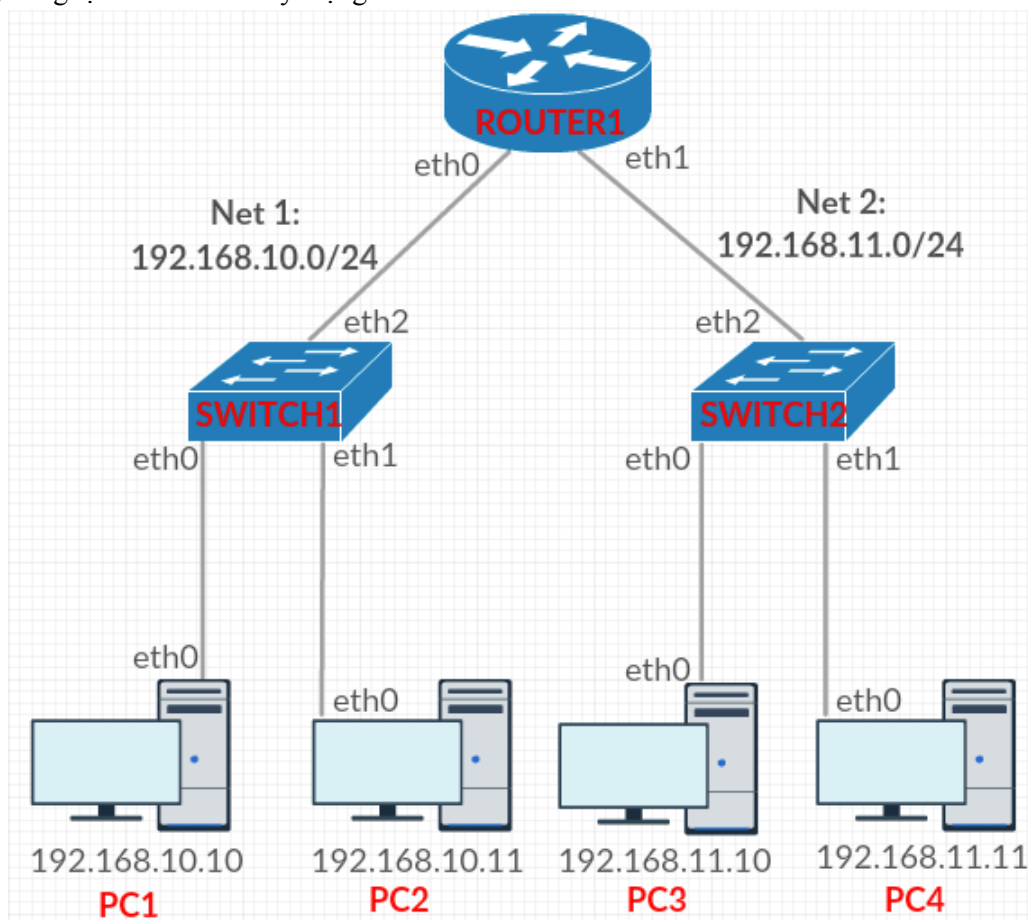
♦ **Bước 14:** Hủy mạng ảo bằng lệnh **lcrash** sau khi đã thực hiện xong **Bài tập 9**

**Bài tập 10:** Liên mạng với Router và Switch ảo

♦ **Bước 1:** Quan sát mô hình mạng cần xây dựng. Nhận diện các thiết bị (PC, Switch...), giao diện (eth0, eth1...) với các địa chỉ IP. **Lưu ý: Sinh viên tự xác định các vùng độ có trong sơ đồ thiết kế mạng.**

♦ **Bước 2:** Thực hiện xây dựng mạng ảo và lưu ở đường dẫn **/home/student/your\_workspace/BaiTap10**. Hoàn thành bài tập khi các thiết bị đều truyền tải dữ liệu được với nhau.

♦ **Bước 3:** Hoàn thành bài tập khi các thiết bị trong mô hình đều truyền tải dữ liệu được cho nhau. Sau khi kết thúc, dùng lệnh **lcrash** để hủy mạng.



**Bài tập thêm (Nâng cao):** Liên mạng với Router và Switch ảo

♦ **Bước 1:** Quan sát mô hình mạng cần xây dựng. Nhận diện các thiết bị (PC, Switch...), giao diện (eth0, eth1...) với các địa chỉ IP. **Lưu ý:** Sinh viên tự xác định các vùng mạng có trong sơ đồ thiết kế mạng.

♦ **Bước 2:** Thực hiện xây dựng mạng ảo và lưu ở đường dẫn `/home/student/your_workspace/BaiTapThem`. Hoàn thành bài tập khi các thiết bị đều truyền tải dữ liệu được với nhau.

♦ **Bước 3:** Hoàn thành bài tập khi các thiết bị trong mô hình đều truyền tải dữ liệu được cho nhau. Sau khi kết thúc, dùng lệnh `lcrash` để hủy mạng.

