

# BUỔI THỰC HÀNH 1

## Mục đích:

- Làm quen một số công cụ quản trị mạng trên nền tảng Linux
- Làm quen với công cụ mô phỏng mạng Netkit: cài đặt và sử dụng lệnh
- Xây dựng mạng và cấu hình tĩnh cơ bản với Netkit

## I. CÔNG CỤ QUẢN TRỊ MẠNG TRÊN LINUX

**ping**: công cụ cho phép gửi 1 gói tin đến từ địa chỉ IP máy nguồn đến địa chỉ IP máy đích. Nếu như địa chỉ IP máy đích là tồn tại, ping trên máy đích sẽ tự động hồi đáp bằng 1 gói tin ngược lại máy nguồn. Cả 2 gói tin ping này đều chứa thông điệp ICMP - Internet Control Message Protocol.

**ifconfig**: công cụ cho phép cấu hình giao diện mạng (network interface) của máy, ví dụ: đặt địa chỉ IP và netmask, tắt hoặc mở giao diện mạng.

**tcpdump**: công cụ cho phép bắt các gói tin luân chuyển qua một hoặc nhiều giao diện mạng. Công cụ này cung cấp 2 chức năng lớn, là packet sniffing và packet analyze với thư viện lệnh phong phú.

**route**: công cụ cho phép xem bảng dẫn đường hiện tại của host.

**traceroute**: công cụ cho phép lần vết của dữ liệu luân chuyển qua host.

## II. BỘ CÔNG CỤ MÔ PHỎNG NETKIT

Netkit là một bộ công cụ mã nguồn mở trên nền tảng Linux cho phép người dùng giả lập (emulate) các hệ thống mạng phục vụ cho nhiều mục đích sử dụng khác nhau như từ đơn giản cho đến phức tạp. Netkit hỗ trợ mô phỏng các thiết bị mạng như Router, Switch, PC...

**Netkit có 1 ưu điểm rất lớn là việc xây dựng máy ảo trên Linux kernel cho phép người dùng có thể hack và modify lại thành 1 máy ảo phù hợp với mô hình mạng cần xây dựng.** Ngoài ra trong các đánh giá về hiệu năng hoạt động của các Network Emulator, Netkit đạt được đánh giá tốt theo nhiều tiêu chí khác nhau. Trong thực hành Mạng máy tính, ta sử dụng Netkit để mô phỏng mô hình mạng nhằm minh họa kiến thức lý thuyết đã được giảng dạy.

**Bên cạnh Netkit, còn có Netkit-NG**, có thể xem là 1 phiên bản anh em của Netkit, tuy nhiên ảnh (image) của máy ảo được cập nhật với phiên bản Linux mới hơn.

**Hiện nay Netkit đã ngưng phát hành phiên bản mới mà chuyển sang phát triển trên nền tảng mới với nhiều tính năng hấp dẫn hơn đó là Kathará.** Tương tự như Netkit, Kathará cũng là phần mềm mô phỏng mạng ảo với chi phí thấp nhưng hiệu suất của Kathará (về Tiêu hao bộ nhớ, CPU sử dụng...) vượt trội hơn so với Netkit.

**Đội ngũ phát triển Netkit cũng giới thiệu một công cụ đồ họa hữu ích, Netkit Lab Generator.** Công cụ này cho phép xây dựng cấu trúc thư mục mạng ảo Netkit thông qua giao diện đồ họa thân thiện, dễ sử dụng. Địa chỉ để sử dụng công cụ Netkit Lab Generator: <http://www.kathara.org/tools/nlg/>

### 1. Các đặc điểm chính của Netkit

- Netkit cho phép tạo ra nhiều máy ảo trên một máy thực. Máy thực (Host Machine) là một hệ điều hành Linux, trong môi trường thực hành Mạng máy tính là Ubuntu 16.04. Các máy ảo (Virtual Machine) Netkit hoạt động dưới dạng User Mode Linux (chỉ có chế độ dòng lệnh, không có đồ họa).
- Mỗi máy ảo Netkit được cấp phát tài nguyên (cố định hoặc thay đổi) để hoạt động.

- Các máy ảo Netkit được tạo ra từ 1 ảnh (image) có sẵn trong mã nguồn cài đặt. Ảnh này có thể thay đổi, chẳng hạn, cài đặt thêm môi trường java, python...tùy theo nhu cầu sử dụng máy ảo.
- Các máy ảo được cấu hình và cài đặt thêm các phần mềm, thư viện mới (nếu cần) để mô phỏng cho các thiết bị trong một mạng máy tính, chẳng hạn: Router, Switch, DNS Server...
- Một mạng ảo Netkit được tạo ra bằng cách xây dựng các máy ảo Netkit trong cùng một môi trường phát triển. Các mạng ảo Netkit có thể liên thông với nhau trong cùng một máy tính (môi trường Ubuntu 16.04) hoặc giữa các máy tính khác nhau (các máy tính Ubuntu 16.04).
- Một mạng các máy ảo Netkit kết nối với nhau theo hình thái trục (bus topology) mô phỏng công nghệ Ethernet II.

## 2. Hệ thống tập lệnh của Netkit

Netkit cung cấp 2 tập lệnh là: **v-commands** và **l-commands**. 2 tập lệnh này được sử dụng trên màn hình điều khiển (Terminal) của máy thực (Host Machine)

- **v-commands** sử dụng để tương tác với một máy ảo đơn lẻ.
- **l-commands** sử dụng để tương tác với một mô hình mạng gồm nhiều máy ảo tham gia.

Một số thao tác (commands) có thể được thực hiện bằng lệnh v hoặc lệnh l

	<b>v-commands</b>	<b>l-commands</b>
<b>start</b>	Khởi động 1 máy ảo. Cách dùng: <i>vstart pc1 --eth0=A</i>	Khởi động 1 nhiều máy ảo được chỉ định trong 1 mạng ảo. Cách dùng: <i>lstart</i> hoặc <i>lstart pc1 pc2</i>
<b>crash</b>	Hủy 1 máy ảo. Cách dùng: <i>vcrash pc1</i>	Hủy 1 hoặc nhiều máy ảo được chỉ định trong 1 mạng ảo. Cách dùng: <i>lcrash</i> hoặc <i>lcrash pc1 pc2</i>
<b>halt</b>	Tạm dừng hoạt động 1 máy ảo. Các thay đổi trên máy ảo sẽ được giữ nguyên khi máy ảo được kích hoạt lại. Cách dùng: <i>vhalt pc1</i>	Tạm dừng hoạt động 1 hoặc nhiều máy ảo được chỉ định trong 1 mạng ảo. Các thay đổi trên máy ảo sẽ được giữ nguyên khi máy ảo được kích hoạt lại. Cách dùng: <i>lhalt</i> hoặc <i>lhalt pc1 pc2</i>

## 3. CÀI ĐẶT NETKIT TRÊN LINUX

❖ **Bước 1:** Truy cập vào trang chủ của Netkit, vào phần Download, tiến hành download 3 file nén dưới đây: *Netkit core version 2.8 and documentation, Netkit file system version 5.2, Netkit kernel version 2.8.*

❖ **Bước 2:** Tạo 1 folder mới có tên là **THMMT** trong **Home**, di chuyển 3 file nén đã download được vào trong thư mục này. Chuyển đến thư mục **THMMT**, thực hiện giải nén 3 file này bằng cách viết 1 script là **giainen.sh** với nội dung như sau:

```
tar -xjSf netkit-2.8.tar.bz2
tar -xjSf netkit-filesystem-i386-F5.2.tar.bz2
tar -xjSf netkit-kernel-i386-K2.8.tar.bz2
```

Chú ý rằng **netkit-2.8.tar.bz2**, **netkit-filesystem-i386-F5.2.tar.bz2** và **netkit-kernel-i386-K2.8.tar.bz2** là tên của 3 file nén được download về.

❖ **Bước 3:** Trên máy thực (Ubuntu 16.04) mở **Terminal** và thực thi script **giainen.sh** bằng lệnh **./giainen.sh** (có thể cấp phát quyền bằng lệnh **chmod 777 giainen.sh** trước nếu cần). Sau khi giải nén xong, ta có được một thư mục có tên là **netkit**.

❖ **Bước 4:** Trên **Terminal**, di chuyển vào thư mục netkit bằng lệnh **cd <Tên thư mục>**.

Kiểm tra việc cấu hình biến môi trường thực thi bằng lệnh `./check_configuration.sh`, nhận được thông báo lỗi vì biến môi trường chưa được đặt. Thông báo lỗi này sẽ hướng dẫn cách thức đặt biến môi trường cho netkit để chương trình có thể hoạt động được.

❖ **Bước 5:** Tiến hành cấu hình biến môi trường `MANPATH`, `PATH` để Netkit có thể hoạt động và thực thi như sau:

```
$ export NETKIT_HOME=~/.THMMT/netkit
$ export MANPATH=~/.NETKIT_HOME/man
$ export PATH=~/.NETKIT_HOME/bin:$PATH
```

Để biến môi trường không cần tạo lại mỗi lần khởi động Netkit. Ta thiết lập biến môi trường trong file `~/.bashrc` bằng cách copy các lệnh export ở trên (bỏ đi dấu \$) và khởi động lại terminal.

❖ **Bước 6:** Kiểm tra cấu hình đã cài đặt bằng lệnh `./check_configuration.sh` (thực hiện lệnh này trong thư mục cài đặt `netkit`). Nếu cài đặt thành công thì sẽ nhận được thông báo chúc mừng.

```
passed.
> Checking for availability of terminal emulator applications:
      xterm      : found
      konsole    : not found
      gnome-terminal : found
passed.
> Checking filesystem type... passed.
> Checking whether 32-bit executables can run... passed.

[ READY ] Congratulations! Your Netkit setup is now complete!
          Enjoy Netkit!
```

## 4. MẠNG ẢO NETKIT

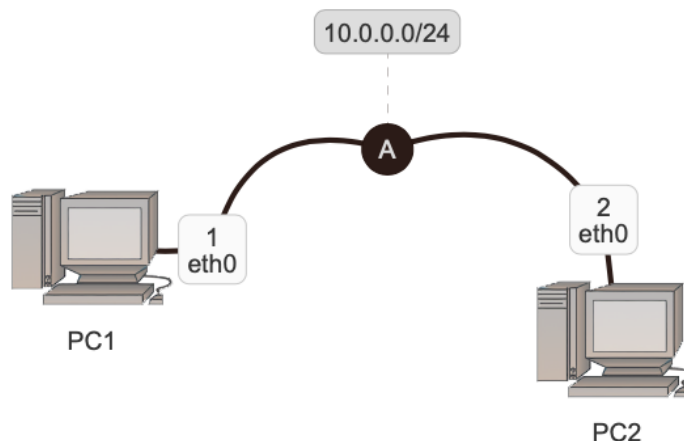
Mỗi sinh viên tạo ra một không gian làm việc (workspace) của riêng mình dưới thư mục `/home/student/`. Ví dụ: `/home/student/vanlong-workspace`. Workspace của 1 sinh viên sẽ là nơi chứa tất cả các bài tập thực hành về Netkit. Workspace cho phép các sinh viên quản lý tách biệt các bài tập thực hành của mình với các sinh viên khác.

Một mô hình mạng ảo Netkit có thể được xây dựng bằng 2 cách:

- **Cách 1:** Các máy ảo được tạo riêng lẻ mà không nằm trong cùng một không gian mô phỏng (một thư mục chứa các tệp tin mô phỏng theo quy chuẩn của Netkit) bằng tập lệnh `v-commands`. Các máy ảo sau đó sẽ được liên kết lại với nhau tạo thành một mạng ảo. Hình thức này không phù hợp trong việc xây dựng một mạng ảo lớn, phức tạp đòi hỏi cần phải quản trị mạng tốt → **Chỉ sử dụng trong Bài tập 1 của phần thực hành Mạng máy tính để sinh viên hiểu hoạt động của máy ảo.**
- **Cách 2:** Một mạng ảo được xây dựng dựa trên cấu trúc thư mục chứa các tệp tin mô phỏng theo quy chuẩn của Netkit.
  - o Các máy ảo được khai báo dưới dạng là các thư mục con
  - o Các cấu hình cho máy ảo được định nghĩa qua file `<tenmayao>.startup`
  - o Hình thái mạng ảo (topology) được định nghĩa qua file `lab.conf`
  - o Khi khởi động mạng ảo, tất cả các máy ảo được kích hoạt cùng 1 lúc. Hình thức này phù hợp với việc xây dựng một mạng ảo gồm nhiều thiết bị mạng với các chứng năng khác nhau. Ngoài ra, việc các máy ảo hoạt động trong cùng 1 không gian mô phỏng sẽ hỗ trợ quản trị mạng tốt hơn → **Được sử dụng trong phần thực hành Mạng máy tính.**

## III. BÀI TẬP THỰC HÀNH

**BÀI TẬP 1:** Xây dựng một mạng LAN đơn giản theo cách 1



◆ **Bước 1:** Quan sát mô hình mạng cần xây dựng và nhận diện các thiết bị, giao diện với các địa chỉ IP được gán

◆ **Bước 2:** Tạo thư mục **BaiTap1** (nằm dưới *workspace /home/student/your\_workspace*). Trên chế độ dòng lệnh Terminal của máy thực Ubuntu 16.04, di chuyển đến thư mục BaiTap1 bằng lệnh **`cd /home/student/your_workspace/BaiTap1`**

◆ **Bước 3:** Khởi tạo máy ảo PC1 bằng lệnh: **`vstart pc1 --eth0=A`** hoặc **`vstart --eth0=A pc1`**  
Ý nghĩa: tạo ra máy ảo tên là PC1 với một giao diện mạng tên là eth0. Giao diện eth0 kết nối PC1 vào mạng LAN A.

Thực hiện tương tự với máy ảo PC2.

◆ **Bước 4:** Trên giao diện của máy ảo PC1 hoặc PC2, dùng lệnh **`ifconfig`** để kiểm tra cấu hình mạng. Tại sao chỉ hiển thị thông tin của giao diện *lo* mà không có giao diện *eth0* đã khai báo tại bước 2  
Cách hiển thị ra (kích hoạt) giao diện eth0 trên PC1 hoặc PC2: **`ifconfig eth0 up`**

◆ **Bước 5:** Đặt địa chỉ IP cho giao diện eth0 của PC1: **`ifconfig eth0 10.0.0.1 netmask 255.255.255.0 broadcast 10.0.0.255`**

Gợi ý: Lệnh `ifconfig eth0 up` ở bước 3 và bước 4 có thể được thay thế bằng một lệnh duy nhất đơn giản hơn: **`ifconfig eth0 10.0.0.1/24 up`**

Trong đó *netmask* và *broadcast* sẽ tự động được tính bởi máy ảo Netkit.

◆ **Bước 6:** Thực hiện đặt địa chỉ IP cho giao diện eth0 của PC2 tương tự như đã làm với PC1. Kiểm tra lại cấu hình mạng của 2 máy xem đã có giao diện eth0 và đã gán địa chỉ IP hay chưa bằng lệnh **`ifconfig`**.

◆ **Bước 7:** Trên PC1, dùng lệnh **`ping`** đến pc2 (10.0.0.2). Kết quả lệnh này là gì?

Sử dụng lệnh **`traceroute`** để kiểm tra thông tin đường đi của gói tin từ PC1 đến PC2.

Sử dụng lệnh **`route`** để hiển thị thông tin bảng vạch đường của PC1 hoặc PC2 trong mạng LAN A.

◆ **Bước 8:** Trên máy thực Ubuntu 16.04, sử dụng lần lượt lệnh **`vcrash pc1`** và **`vcrash pc2`** để hủy 2 máy ảo vừa tạo

◆ **Bước 9:** Trong thư mục */home/student/your\_workspace/BaiTap1*, xóa đi 2 file ***PC1.disk*** và ***PC2.disk*** bằng dòng lệnh (**`rm`**) hoặc bằng đồ họa. Kết thúc Bài tập 1.

## **BÀI TẬP 2:** Xây dựng một mạng LAN đơn giản theo cách 2

◆ **Bước 1:** Quan sát mô hình mạng cần xây dựng và nhận diện các thiết bị, giao diện với các địa chỉ IP được gán

◆ **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 **BaiTap2**. Trên chế độ dòng lệnh Terminal của máy thực Ubuntu 16.04, di chuyển đến thư mục **BaiTap2** bằng lệnh *cd /home/student/your\_workspace/BaiTap2*.

Cấu trúc thư mục được tổ chức như sau:

```
----lab1.2      (folder)
----pc1         (folder)
----pc2         (folder)
----pc1.startup (file)
----pc2.startup (file)
----lab.conf    (file)
```

Trong đó, (**folder**) thể hiện cho các thư mục và (**file**) thể hiện cho tập tin.

- File **lab.conf** là nơi chứa miêu tả về hình thái (topology) của một mạng ảo
- Thư mục PC1 và PC2 đại diện cho 2 máy ảo của mạng muốn đưa vào hoạt động. Lưu ý: Tên thư mục sẽ được gán cho tên máy ảo khi mạng ảo khởi động.
- File **PC1.startup** và **PC2.startup** (hay gọi chung là các file *.startup*) là nơi chứa các cấu hình muốn áp dụng cho một máy ảo (PC1 hoặc PC2) ngay khi máy ảo được khởi động cùng mạng ảo. Lưu ý: Tên file *.startup* phải được đặt trùng với tên thư mục đại diện cho máy ảo, chẳng hạn: PC1
- Để thực hiện tạo thư mục, file cũng như soạn thảo nội dung cho file, ngoài chế độ đồ họa GUI trên Ubuntu 16.04, người dùng có thể lựa chọn chế độ dòng lệnh để thực hiện
  - o Đối với việc tạo mới thư mục: lệnh *mkdir <new\_folder>*
  - o Đối với việc tạo mới file: *touch <new\_file>*
  - o Công cụ soạn thảo file trên Terminal: *nano <file\_needed\_to\_edit>*

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

```
pc1[0]=A
pc2[0]=A
```

◆ **Bước 4:** Trên file **PC1.startup**, cấu hình của eth0 được miêu tả như sau

```
ifconfig eth0 10.0.0.1/24 up
```

Trên file **PC2.startup**, cấu hình của eth0 được miêu tả như sau

```
ifconfig eth0 10.0.0.2/24 up
```

◆ **Bước 5:** Trên màn hình dòng lệnh **Terminal** của máy thực Ubuntu 16.04, tại thư mục **BaiTap2** sử dụng lệnh *Istart* để khởi động mạng ảo **BaiTap2** đã tạo

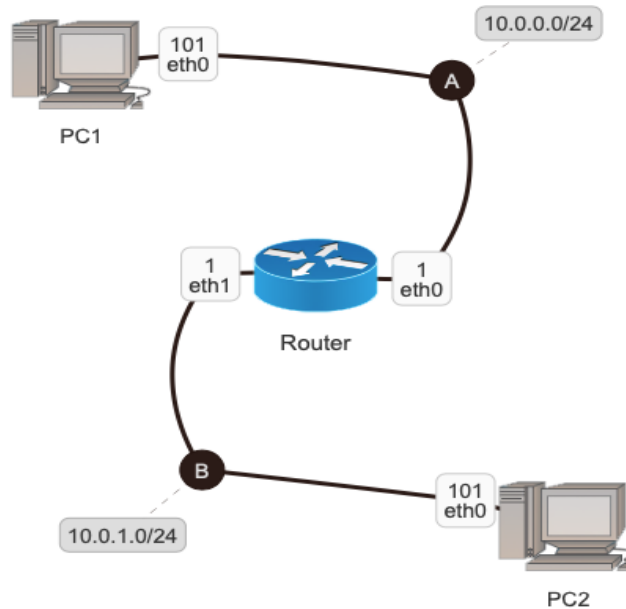
Lưu ý: có thể sử dụng *Istart* để khởi động từng máy ảo riêng lẻ trong trường hợp muốn kiểm tra từng máy ảo một. Ví dụ: *Istart PC1* hoặc *Istart PC2*

◆ **Bước 7:** Trên PC1, lần lượt dùng các lệnh *ping*, *traceroute* và *route* để kiểm tra tính liên thông trong LAN A giống như Bước 7 **BaiTap1**

◆ **Bước 8:** Trên máy thực Ubuntu 16.04, sử dụng lệnh *Icrash* để hủy 2 máy ảo vừa tạo. Các file PC1.disk và PC2.disk sẽ được tự động thu hồi. Kết thúc Bài tập 2.

**BÀI TẬP 3:** Xây dựng 2 mạng LAN kết nối bởi 1 Router

◆ **Bước 1:** Quan sát mô hình mạng cần xây dựng và nhận diện các thiết bị, giao diện với các địa chỉ IP được gán



♦ **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 **BaiTap3**. Trên chế độ dòng lệnh Terminal của máy thực Ubuntu 16.04, di chuyển đến thư mục **BaiTap3** bằng lệnh *cd /home/student/your\_workspace/BaiTap3*.

```

$NETKIT_HOME
----Lab1.3          (folder)
----pc1            (folder)
----pc2            (folder)
----router          (folder)
----pc1.startup    (file)
----pc2.startup    (file)
----router.startup (file)
----lab.conf       (file)
  
```

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

```

PC1[0]=A
PC2[0]=B
Router[0]=A
Router[1]=B
  
```

♦ **Bước 4:** Trên file *PC1.startup*, cấu hình của eth0 được miêu tả như sau

```

ifconfig eth0 10.0.0.101/24 up
route add default gw 10.0.0.1
  
```

**Ý nghĩa:** Lệnh *route add default gw* cho phép thêm thông tin vạch đường mặc nhiên (default) vào bảng vạch đường của một thiết bị. Tham số **gw** đại diện cho Gateway là hướng mà thiết bị sẽ gửi gói tin đến để các gói tin có thể đi ra bên ngoài mạng.

♦ **Bước 5:** Trên file *PC2.startup*, cấu hình của eth0 được miêu tả như sau

```
ifconfig eth0 10.0.1.101/24 up  
route add default gw 10.0.1.1
```

◆ **Bước 6:** Trên file **Router.startup**, cấu hình của eth0 và eth1 được miêu tả như sau

```
ifconfig eth0 10.0.0.1/24 up  
ifconfig eth1 10.0.1.1/24 up
```

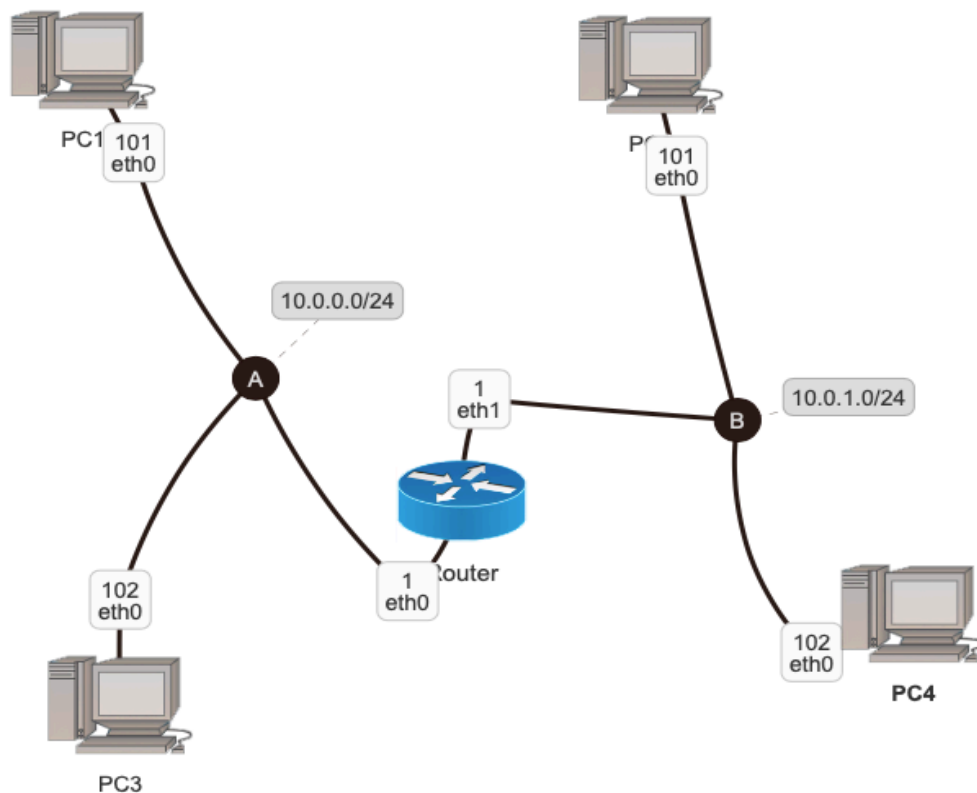
◆ **Bước 7:** Trên màn hình dòng lệnh **Terminal** của máy thực Ubuntu 16.04, tại thư mục **BaiTap3** sử dụng lệnh **Istart** để khởi động mạng ảo **BaiTap3** đã tạo

◆ **Bước 8:** Trên PC1 lần lượt dùng các lệnh **ping**, **traceroute** và **route** để kiểm tra tính liên thông tới Router và PC2.

◆ **Bước 9:** Trên máy thực Ubuntu 16.04, sử dụng lệnh **Icrash** để hủy mạng ảo vừa tạo. Kết thúc Bài tập 3.

**BÀI TẬP 4:** Xây dựng 2 mạng LAN có nhiều máy tính kết nối bởi 1 Router

◆ **Bước 1:** Quan sát mô hình mạng cần xây dựng và nhận diện các thiết bị, giao diện với các địa chỉ IP được gán



◆ **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 **BaiTap4**. Trên chế độ dòng lệnh Terminal của máy thực Ubuntu 16.04, di chuyển đến thư mục **BaiTap4** bằng lệnh **cd /home/student/your\_workspace/BaiTap4**.

Lưu ý: Có thể sao chép nội dung thư mục **BaiTap3** sang cho **BaiTap4** và bổ sung thêm phần khai báo, cấu hình cho máy PC3 và PC4.

◆ **Các bước tiếp theo:** Thực hiện giống hướng dẫn trong Bài Tập 3 đã thực hiện.