# BÀI THỰC HÀNH

## Bài 1. Cài đặt các máy tính kết nối với nhau trong mạng

Sinh viên tiến hành cài đặt 3 máy tính kết nối với nhau trong một mạng.

Sinh viên có thể sử dụng máy ảo (cài VirtualBox) hoặc máy thật. Hệ điều hành sử dụng là Ubuntu. Gọi 3 máy đó là A, B, và C.

## Bài 2. Cấu hình mạng cho các máy

(gợi ý: sử dụng lệnh ifconfig)

Câu hỏi 1: Trình bày các bước (các lệnh) để thực hiện quá trình cấu hình mạng sao cho các máy nằm trong cùng một mạng đó. Em thực hiện lệnh nào để biết các máy đã được kết nối trong cùng một mạng?

Bước 1: Cài đặt 2 máy ảo Ubuntu thông qua VirtualBox.

A screenshot of a computer

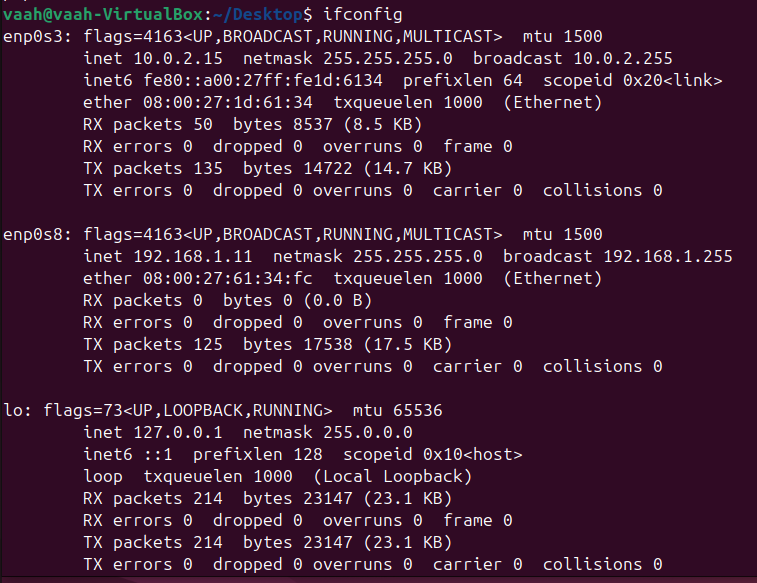
Description automatically generated

Bước 2: Sử dụng VirtualBox thiết lập 1 mạng cho 2 máy.

A screenshot of a computer

Description automatically generated

Bước 3: Sử dụng câu lệnh ifconfig ở 2 máy cho kết quả:



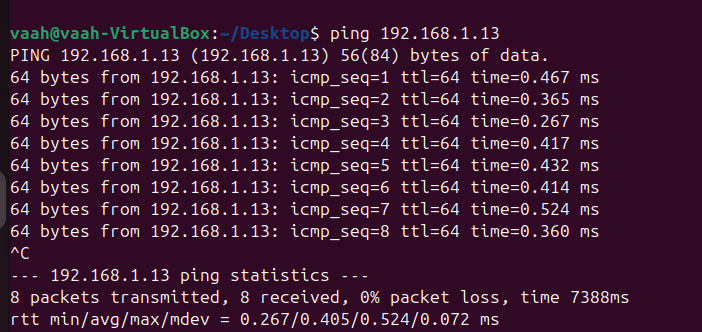
A screenshot of a computer

Description automatically generated

Câu hỏi 2: Em thực hiện lệnh nào để biết các máy đã được kết nối trong cùng một mạng?

Hai máy có địa chỉ IP **192.168.1.11** và 192.168.1.13 có cùng phần dải mạng 192.168.1, cho thấy chúng nằm trong cùng một mạng con (subnet). Với subnet mask 255.255.255.0, ba octet đầu 192.168.1 xác định mạng, và octet cuối (11 và Hai máy có địa chỉ IP **192.168.1.11** và **192.168.1.13** có cùng phần dải mạng **192.168.1**, cho thấy chúng nằm trong cùng một mạng con (subnet). Với subnet mask **255.255.255.0**, ba octet đầu **192.168.1** xác định mạng, và octet cuối (11 và 13) là địa chỉ của từng máy. Điều này có nghĩa là hai máy có thể giao tiếp trực tiếp với nhau trong cùng mạng.

Ngoài ra, khi sử dụng lệnh **ping** từ máy này đến máy kia, bạn sẽ nhận được phản hồi từ máy đích, xác nhận rằng kết nối giữa hai máy hoạt động bình thường.



**Bài 3**. Cài đặt wireshark cho máy A. Thử kết nối giữa các máy. Quan sát màn hình wireshark của máy A khi được B và C thực hiện lệnh *ping*.

Thực hiện bắt các gói tin với wireshark:

* Ấn vào nút Capture
* Lựa chọn giao diện mạng (interface) phù hợp (chú ý: phải chọn đúng giao diện đang có kết nối giữa các máy)

Câu hỏi 3: Thực hiện lệnh ping giữa các máy. Những dòng thông tin nào trên cửa sổ wireshark cho thấy thông tin của lệnh *ping* đó?

A screenshot of a computer program

Description automatically generated

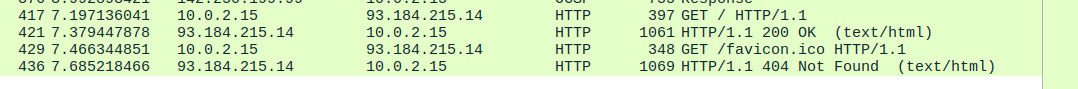
A screenshot of a computer

Description automatically generated

Câu hỏi 4: Dùng trình duyệt của máy đang chạy wireshark truy cập vào các trang web khác nhau. Những dòng thông tin nào trên cửa sổ wireshark cho thấy thông tin của quá trình duyệt web đó (các gói tin liên quan HTTP/HTTPS traffic).

Khi truy cập vào trang web example.com, wrireshark hiện ra các thông tin:

* HTTP yêu cầu lấy tài nguyên



* TLS dữ liệu được mã hoá, bắt tay

A screenshot of a computer

Description automatically generated

* DNS query thành công

A screenshot of a computer

Description automatically generated

Thực hiện phân tích các luồng dữ liệu TCP và UDP với wireshark:

* **Đối với UDP:**
  + Ở máy server, sử dụng công cụ netcat để chạy lệnh: nc -u -l 9999
  + Ở máy client: echo "Test UDP traffic" | nc -u 127.0.0.1 9999
  + Sau khi chạy được một lúc, dừng wireshark lại và nhập “udp” vào thanh filter để lọc ra các thông điệp UDP

Ở bảng giữa sẽ có các thông tin chi tiết về cấu trúc gói:

* Frame: Thông tin chung về gói (kích thước, thời gian, v.v.).
* Ethernet: Địa chỉ MAC nguồn và đích.
* IP: Địa chỉ IP nguồn và đích.
* UDP: Cổng nguồn và đích, độ dài và checksum.

Thực hiện câu lệnh:



A computer screen with blue text

Description automatically generated

A screenshot of a computer

Description automatically generated

Câu hỏi 5: Quan sát UDP packet trên wireshark, phân tích về tính đơn giản của UDP. Gợi ý: không có kết nối, do đó không có cờ (flags) để thiết lập hoặc hủy kết nối.

**UDP** là giao thức **không kết nối**, không cần thiết lập hay hủy kết nối, do đó không có cờ (flags) như TCP. Nó cũng không có điều khiển luồng hoặc tắc nghẽn, giúp **nhanh** và **đơn giản**. Tiêu đề UDP rất ngắn, chỉ chứa các trường cơ bản như **cổng nguồn**, **cổng đích**, **độ dài**, và **checksum**. Do không có các cơ chế phức tạp, UDP thường được sử dụng cho các ứng dụng thời gian thực như **truyền thông thoại/video**, nơi **tốc độ** được ưu tiên hơn **độ tin cậy**

* **Đối với TCP:**
  + Ở máy server, sử dụng công cụ netcat để chạy lệnh: nc -l 8888
  + Ở máy client: echo "Test TCP traffic" | nc 127.0.0.1 8888
  + Sau khi chạy được một lúc, dừng wireshark lại và nhập “tcp” vào thanh filter để lọc ra các thông điệp TCP
* Quan sát thông tin 1 TCP packet, chúng ta sẽ thấy các thông tin sau: Frame: Chi tiết chung về gói tin.
* Ethernet: Địa chỉ MAC.
* IP: Thông tin địa chỉ IP.

A screenshot of a computer screen

Description automatically generated



A screenshot of a computer

Description automatically generated

Câu hỏi 6: Ấn vào trường thông tin TCP, quan sát sẽ thấy nhiều trường hơn so với UDP. Đó là những trường nào? Ý nghĩa của từng trường là gì?

Dưới đây là các trường cơ bản trong **TCP** và ý nghĩa:

1. **Source Port (Cổng nguồn)**:
   * Cổng của ứng dụng gửi dữ liệu.
2. **Destination Port (Cổng đích)**:
   * Cổng của ứng dụng nhận dữ liệu.
3. **Sequence Number (Số thứ tự)**:
   * Xác định thứ tự của byte dữ liệu để đảm bảo dữ liệu được sắp xếp đúng khi đến nơi.
4. **Acknowledgment Number (Số xác nhận)**:
   * Xác nhận rằng đã nhận được dữ liệu từ phía bên kia.
5. **Flags (Cờ điều khiển)**:
   * **SYN**: Bắt đầu kết nối.
   * **ACK**: Xác nhận đã nhận dữ liệu.
   * **FIN**: Kết thúc kết nối.
6. **Window Size (Kích thước cửa sổ)**:
   * Xác định lượng dữ liệu có thể gửi đi mà không cần xác nhận.
7. **Checksum (Kiểm tra lỗi)**:
   * Dùng để kiểm tra lỗi trong quá trình truyền dữ liệu.

**Tổng kết:**

TCP phức tạp hơn UDP vì cần quản lý kết nối, đảm bảo dữ liệu đến đúng thứ tự và không bị mất. Những trường cơ bản trên giúp TCP thực hiện các chức năng này.

Liên quan đến Bắt tay ba bước:

Gói SYN từ máy khách. Gói SYN-ACK từ máy chủ.

Gói ACK từ máy khách để thiết lập kết nối.

Câu hỏi 7: Giải thích ý nghĩa của quy trình thiết lập kết nối bắt tay 3 bước này đối với TCP.

**Quy trình bắt tay 3 bước của TCP** gồm:

1. **SYN**: Máy gửi (client) gửi gói SYN đến máy nhận (server) để yêu cầu kết nối.
2. **SYN-ACK**: Máy nhận phản hồi bằng gói SYN-ACK để xác nhận đã nhận được yêu cầu và đồng ý kết nối.
3. **ACK**: Máy gửi gửi gói ACK để xác nhận đã nhận phản hồi từ máy nhận, hoàn tất việc thiết lập kết nối.

**Ý nghĩa:**

Quy trình này đảm bảo rằng cả hai bên (máy gửi và máy nhận) đều sẵn sàng truyền và nhận dữ liệu một cách tin cậy.

**Bài 4**. **Cài đặt Webserver apache2**

Giao thức http: Cài đặt webserver apache2 cho máy A. Thử truy cập vào trang web của A từ 2 máy B và C (cổng 80). Quan sát màn hình wireshark của máy A.

Câu hỏi 8: Những dòng thông tin nào trên cửa sổ wireshark cho thấy thông tin của việc truy cập web đó?

(gợi ý: có thể tham khảo đường link sau để cài đặt apache2: https://[www.digitalocean.com/community/tutorials/how-to-install-the-apache-web-server-on-ubuntu-16-](http://www.digitalocean.com/community/tutorials/how-to-install-the-apache-web-server-on-ubuntu-16-) 04 )