**Bài tập thực hành**

**Môn: TH Lập Trình Mạng**

**Chương 1: Cơ bản về mạng máy tính**

1. **MỤC TIÊU**

Bài thực hành đưa ra với mục tiêu cho sinh viên ôn tập lại các kiến thức mạng máy tính cơ bản, đồng thời làm quen sử dụng các công cụ mạng cơ bản, sẽ trợ giúp cho các bài tập ở các chương sau.

1. **YÊU CẦU**

* Kiến thức cơ bản về mạng máy tính và các câu lệnh cơ bản Linux
* Máy tính cài đặt Hệ điều hành Linux (khuyến khích distro Ubuntu)

1. **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?

* *Các bước (các lệnh) để thực hiện quá trình cấu hình mạng sao cho các máy nằm trong cùng một mạng:*
* *Bước 1: Thiết lập network cho 2 máy ảo:*

*A screenshot of a computer

Description automatically generated*

* *Bước 2: Xác định giao diện mạng:*

A computer screen shot of a computer program

Description automatically generated

* *Bước 3: Cấu hình địa chỉ IP tĩnh cho từng máy*

*Máy A: 192.168.0.103*



*Máy B: 192.168.0.104*



* *Bước 4: Tắt tường lửa*

A computer screen with text

Description automatically generated with medium confidence

* *Lệnh để biết các máy đã được kết nối trong cùng một mạng*
* *Máy A đến máy B:*

*A screenshot of a computer

Description automatically generated*

* *Máy B đến máy A:*

*A screenshot of a computer program

Description automatically generated*

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

*Lệnh để biết các máy đã được kết nối trong cùng một mạng*

* *Máy A đến máy B:*

A screenshot of a computer

Description automatically generated

* *Máy B đến máy A:*

A screenshot of a computer program

Description automatically generated

**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*.

***Cài đặt wireshark cho máy A***





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?

*Trên cửa sổ wireshark:*

* *Những trường thông tin hiển thị thông tin của lệnh* ***ping*** *đã thực hiện là: Source, Destination, Protocol, Info.*
* *Trong trường hợp này, cứ 2 dòng sẽ hiển thị thông tin của 1 lệnh* ***ping****.*

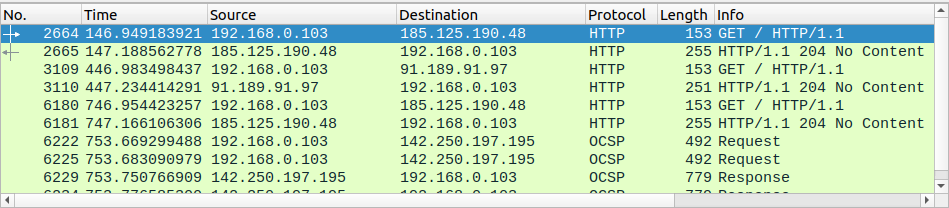
A screenshot of a computer

Description automatically generated

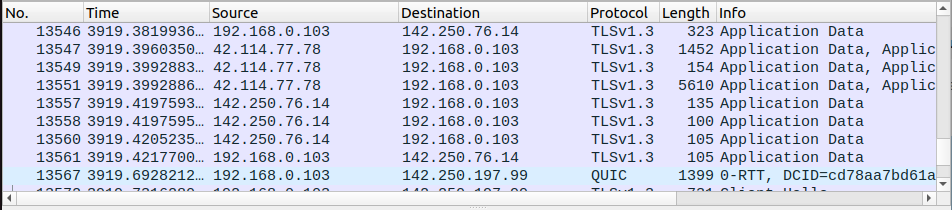
Câu hỏi 4: Dùng trình duyệt của máy đang chạy wireshark truy cập vào các trang web khác nhau. Những dòng thông tin nào trên cửa sổ wireshark cho thấy thông tin của quá trình duyệt web đó (các gói tin liên quan HTTP/HTTPS traffic).

*Những dòng thông tin trên cửa sổ wireshark cho thấy thông tin của quá trình duyệt web đó là:*

* *Dòng có giao thức HTTP để yêu cầu (GET, POST), phản hồi (HTTP/1.1 204 No Content)*

****

* *Dòng có giao thức TLSv1.2 hoặc TLSv1.3 (quá trình bắt tay)*

****

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

A screenshot of a computer

Description automatically generated

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.
* *UDP là giao thức không có kết nối, nghĩa là khi gửi 1 gói tin, nó không cần thiết lập kết nối trước 2 bên như TCP (bắt tay 3 bước).*

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 không có flags, vì thế không kiểm tra kết nối giữa 2 bên.*
* *UDP không có cơ chế xác nhận gói tin đã được nhận nên không thể kiểm tra và gửi lại khi mất gói tin. Bên cạnh đấy, các gói tin có thể bị tràn hoặc bỏ qua khi mạng quá tải.*

*Do đó, UDP sẽ được ứng dụng khi cần truyền dữ liệu thời gian thực (như: trò chơi trực tuyến, video streaming,…)*

* 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

A screenshot of a computer

Description automatically generated

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.

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?

*Ý nghĩa các trường thông tin của TCP (nhiều hơn so với UDP):*

* *Source Port: Là cổng của ứng dụng trên máy gửi, giúp xác định ứng dụng nào đang gửi dữ liệu.*
* *Destination Port: Là cổng của ứng dụng trên máy nhận, giúp chuyển dữ liệu đến đúng ứng dụng trên máy nhận.*
* *Sequence Number: Chứa số thứ tự của byte đầu trong dữ liệu được gửi, giúp theo dõi và đảm bảo dữ liệu được nhận theo đúng thứ tự.*
* *Acknowledgment Number: Chứa số thứ tự byte tiếp theo mà bên gửi mong muốn nhận được, giúp xác nhận đã nhận được dữ liệu.*
* *Data Offset: Xác định độ dài tiêu đề TCP, giúp xác định dữ liệu bắt đầu từ byte nào trong gói tin.*
* *Flags: Hiển thị trạng thái kết nối*
* *SYN: Thiết lập kết nối trong quá trình bắt tay 3 bước.*
* *ACK: Xác nhận đã nhận được dữ liệu.*
* *FIN: Kết thúc kết nối.*
* *RST: Đặt lại kết nối nếu có lỗi.*
* *PSH: Yêu cầu gửi dự liệu ngay lập tức.*
* *URG: Dữ liệu khẩn cấp cần được xử lý.*
* *Window Size: Cho biết lượng dữ liệu bên nhận có thể tiếp tục nhận tiếp mà không cần xác nhận lại.*
* *Checksum: Chứa giá trị kiểm tra lỗi của gói tin.*

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.

*Ý nghĩa của quy trình thiết lập kết nối bắt tay 3 bước đối với TCP là: Đảm bảo kết nối được thiết lập đáng tin cậy, phía nguồn chỉ gửi dữ liệu khi nào phía đích đã sẵn sàng, giúp tránh mất mát và nhầm lẫn trong quá trình trao đổi dữ liệu.*

**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.

***Cài đặt webserver apache2:***





***Truy cập trang web của A từ máy B:***

****

*Những dòng thông tin trên cửa sổ wireshark cho thấy thông tin việc truy cập web là:*

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

* *Dòng có giao thức HTTP: Quá trình yêu cầu (GET) từ máy B và phản hồi (200 OK) từ máy A.*
* *Dòng có giao thức TCP: Quá trình bắt tay 3 bước giữa 2 máy A và B.*

A screenshot of a computer

Description automatically generated

(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 )