TÌM HIỂU CÔNG CỤ BẮT GÓI TIN WIRESHARK

I. Mục tiêu :

Bài hướng dẫn giúp sinh viên có thể:

- Sử dụng công cụ bắt gói tin Wireshark.
- Biết cấu trúc gói tin một số giao thức cơ bản trong mạng máy tính

II. Wireshark

II.1.Giới thiệu

Wireshark là một phần mềm mã nguồn mở dùng để bắt và phân tích các gói tin lưu thông qua card mạng của máy tính. Phần mềm này có thể sử dụng trên nhiều nền tảng khác nhau như Linux, windows, Mac OS X, Solaris …

Tên nguyên bản của phần mềm Wireshark là Ethereal, vào tháng 5 năm 2006 dự án được chuyển tên thành Wireshark.

Phần mềm Wireshark giúp :

- Người quản trị hệ thống phân tích và sửa chữa hệ thống.
- Người phát triển chương trình xây dựng các ứng dụng.
- Sinh viên tìm hiểu hoạt động của các giao thức mạng.

Các tính năng chính của Wireshark gồm :

- Bắt các gói tin đi qua một card mạng.
- Liệt kê một cách chi tiết các gói tin bắt được.
- Lưu trữ và mở lại các thông tin bắt được dưới dạng file.
- Tiến hành lọc các gói tin bắt được dưới nhiều tiêu chuẩn khác nhau.
- Tạo ra các biểu đồ thống kê các gói tin qua card mạng.
- Và nhiều các tính năng khác

II.2.Cách cài đặt

1. Gói cài đặt có thể được download tại http://www.wireshark.org.
2. Cài đặt từ file vừa download về.

   Trên windows quá trình này diễn ra tự động và gồm bước

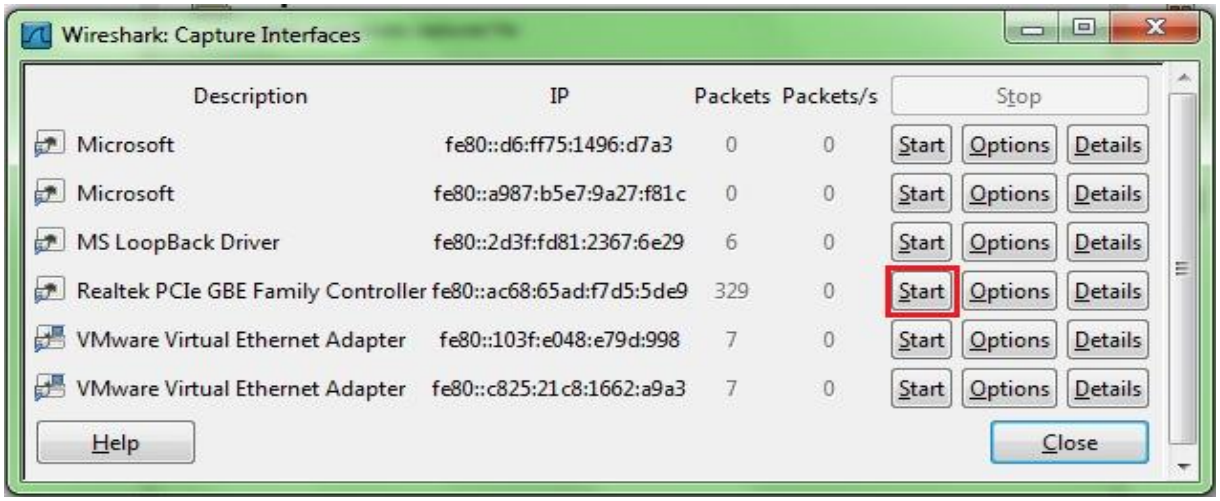
   a. Cài đặt bộ thư viện WinPcap là một bộ thư viện trên windows cung cấp chức năng bắt các gói tin trên card mạng.

   b. Cài đặt phần mềm wireshark sẽ hoạt động dựa trên bộ thư viện này.

II.3.Cách bắt gói tin thông qua một card mạng

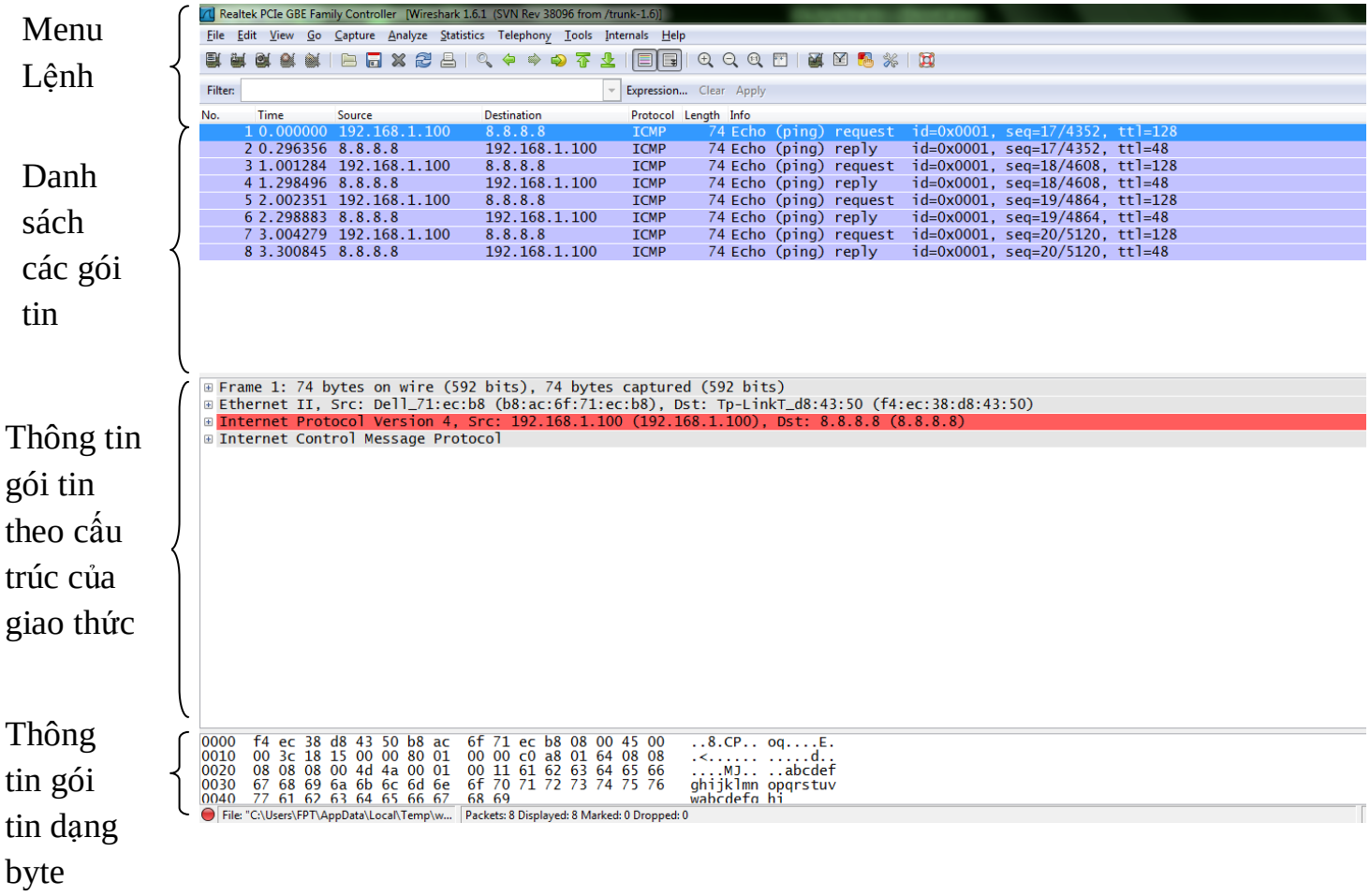
Khởi động chương trình Wireshark.

Lưu ý rằng wireshark không bắt hết các gói tin của máy mà chỉ bắt các gói tin thông qua một card mạng được chọn, nên đầu tiên là ta phải chọn card mạng muốn lắng nghe. Chọn Menu **Capture ➔ Interface** hay phím tắt là **Ctr+I** :



Ở đây liệt kê tất cả các card mạng mà máy tính có, ta chọn một card mạng muốn lắng nghe và khởi động quá trình Capture .

Thử **ping 8.8.8.8** và ta nhận được kết quả bắt gói tin như sau :



Sau khi thu thập đủ các dữ liệu cần, ta sẽ dừng quá trình lắng nghe tại một card mạng bằng cách vào menu **Capture ➔ Stop**

II.4.Lọc các gói tin sau khi Capture

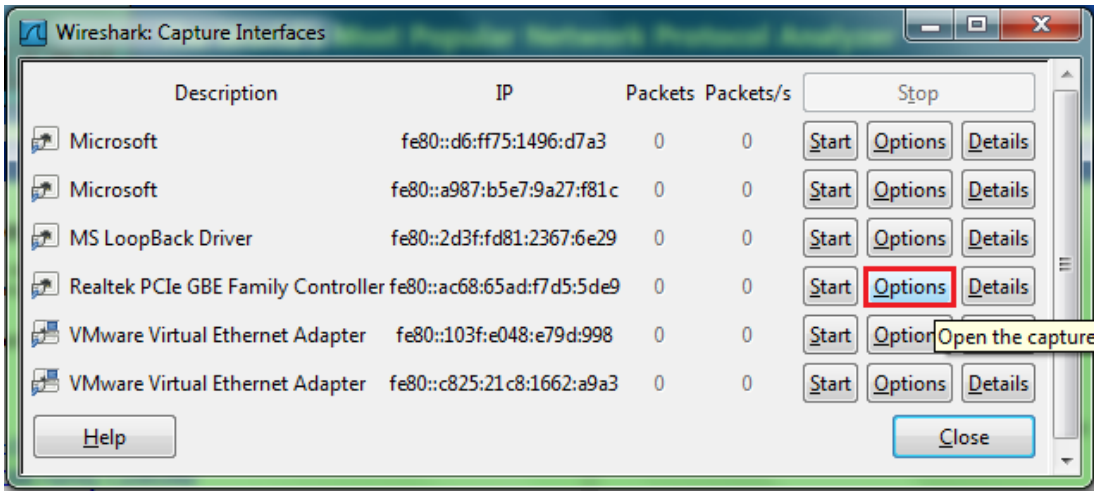
Trong quá trình làm việc thực tế thường có rất nhiều các loại gói tin khác nhau thông qua card mạng mà ta khó có thể kiểm soát hết được. Trong khi đó ta thường chỉ muốn tiến hành thu thập dữ liệu và phân tích một số loại gói tin nhất định. Chính vì thế Wireshark cung cấp cho người dùng khả năng lọc các gói tin theo các tiêu chí cụ thể.

Wireshark cung cấp cho người dùng 2 phương pháp để lọc gói tin vào 2 thời điểm khác nhau của quá trình bắt gói tin. Tuy nhiên, do 2 thời điểm lọc gói tin là khác

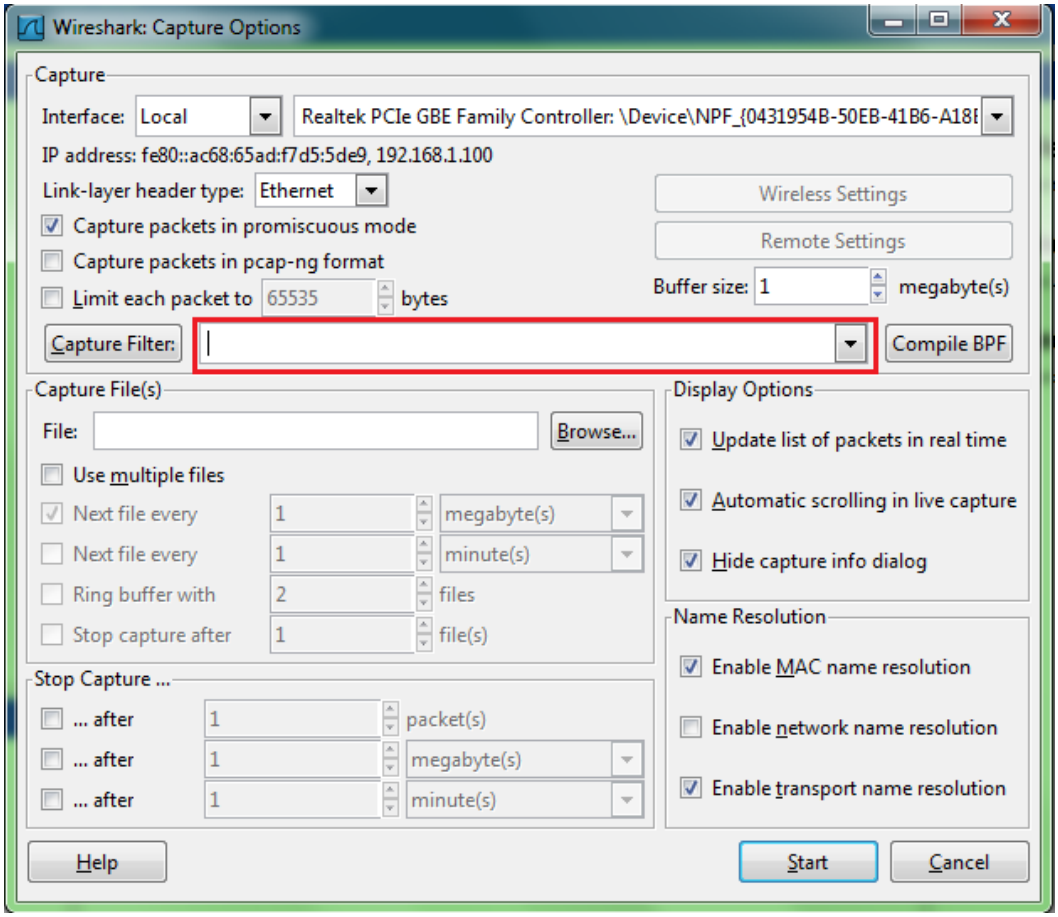
nhau và do 2 thành phần khác nhau đứng ra lọc gói tin là WinPCap và chương trình Wireshark nên ta sẽ thấy được sự khác nhau trong ngôn ngữ mô tả của 2 chức năng này. Sau đây ta sẽ đi tìm hiểu cả 2 phương pháp.

II.4.1. Lọc gói tin ngay khi bắt:

Khi mở hộp thoại chọn card mạng, thay vì bấm **Start** để bắt đầu, ta tiên nhấn nút **Options** để tùy khởi động việc tùy chọn cho việc bắt gói tin.



Hộp thoại **Capture Options** sẽ hiện ra :



Hộp thoại này cho phép ta tùy chỉnh rất nhiều các tính năng trong quá trình bắt gói tin như chức năng lọc các gói tin, chức năng hiển thị các gói, chức năng lưu trữ các gói tin và chức năng hẹn giờ tắt chương trình.

Ở đây chúng ta quan tâm đến chức năng lọc các gói tin bắt được. Việc lọc các gói tin bắt được sẽ được thực hiện theo mô tả mà người dùng đánh vào ở mục capture Filter. Các gói tin sẽ được lọc theo tiêu chí được mô tả và chỉ những gói tin thỏa các tiêu chí này mới được lưu lại để xem xét.

Phương pháp mô tả các gói tin :

Vì việc bắt các gói tin ở phần này được thực hiện dưới sự hỗi trợ bộ thư viện WinPcap, nên ngôn ngữ mô tả ở đây được sử dụng là ngôn ngữ mô tả của WinPcap.

Bạn có thể tìm thấy nhiều ví dụ ở http://wiki.wireshark.org/CaptureFilters . Sau đây sẽ trình bày một cách khái quát phương pháp mô tả này.

Câu lệnh mô tả là sự kết hợp của nhiều câu lênh mô tả con và được nối với nhau bằng **[and|or]**, ta có thể phủ định câu lênh mô tả con bằng cách đặt chữ not trước nó.

```
[not] Mô Tả [and|or] [not] Mô Tả ...
```

Ví dụ :

+Lọc các gói tin Telnet (port 23) từ máy chủ 10.0.0.5

```
tcp port 23 and host 10.0.0.5
```

Các mô tả thành phần là một trong những mô tả sau :

[src|dst] host <host>

Là một thành phần cho phép bạn lọc các gói tin theo địa chỉ IP hay theo tên của nguồn hay đích. Bạn có thể chỉ rõ địa chỉ nguồn hay đích bằng cách đặt các tham số phụ ở đầu là **src|dst** . Nếu trường này không được chỉ ra, về mặc định các gói tin có địa chỉ nguồn hay đích phù hợp điều kiện sẽ được nhận.

ether [src|dst] host <ehost>

Thành phần này cho phép bạn filter trên địa chỉ Ethernet của nguồn hay đích. Tương tự như thành phần ở trên bạn có thể chỉ rõ loại địa chỉ mà bạn quan tâm bằng tham số phụ là [src|dst].

[src|dst] net <net> [{mask <mask>}|{len <len>}]

Thành phần này cho phép bạn tiến hành lọc các gói tin theo địa chỉ network của một gói tin. Bạn có thể thêm các thành phần phụ như src|dst vào để nhấn mạnh rằng bạn quan tâm đến địa chỉ nguồn hay đích. Nếu không thêm trường này vào thì các gói tin có địa chỉ nguồn hoặc đích thỏa yêu cầu sẽ được lưu lại.

[tcp|udp] [src|dst] port <port>

Cho phép bạn lọc các gói tin theo TCP và UDP port. Bạn có thể thêm các tham số **src|dst** và **tcp|udp** cho phép bạn nhấn mạnh rằng quan tâm đến địa chỉ port nguồn hay đích, UDP hay TCP. Chú ý rằng từ tcp|udp phải xuất hiện trước **src|dst**.

Nếu các tham số đó không được sử dụng, gói tin sẽ được lựa chọn trên cả 2 giao thức là TCP và UDP khi mà địa chỉ và port của gói tin thỏa mãn điều kiện đề ra.

less|greater <length>

Thành phần này cho phép bạn lọc các gói tin có chiều dài nhỏ hơn, hay bằng hoặc lớn hơn một độ dài cho trước.

ip|ether proto <protocol>

Thành phần này cho phép bạn lọc các gói tin ở một số giao thức nhất định ở cả tầng Ethernet hay tầng IP.

ether|ip broadcast|multicast

Cho phép bạn tiến hành lọc các gói tin ở cả tầng Ethernet hay IP với broadcasts or multicasts.

<expr> relop <expr>

Cho phép bạn tạo ra một điều kiện lọc gói tin phức tạp bằng cách nhấn mạnh bằng cách chỉ ra một byte hay một khoảng bytes của gói tin. Tham khảo chi tiết tại http://www.tcpdump.org/tcpdump_man.html.

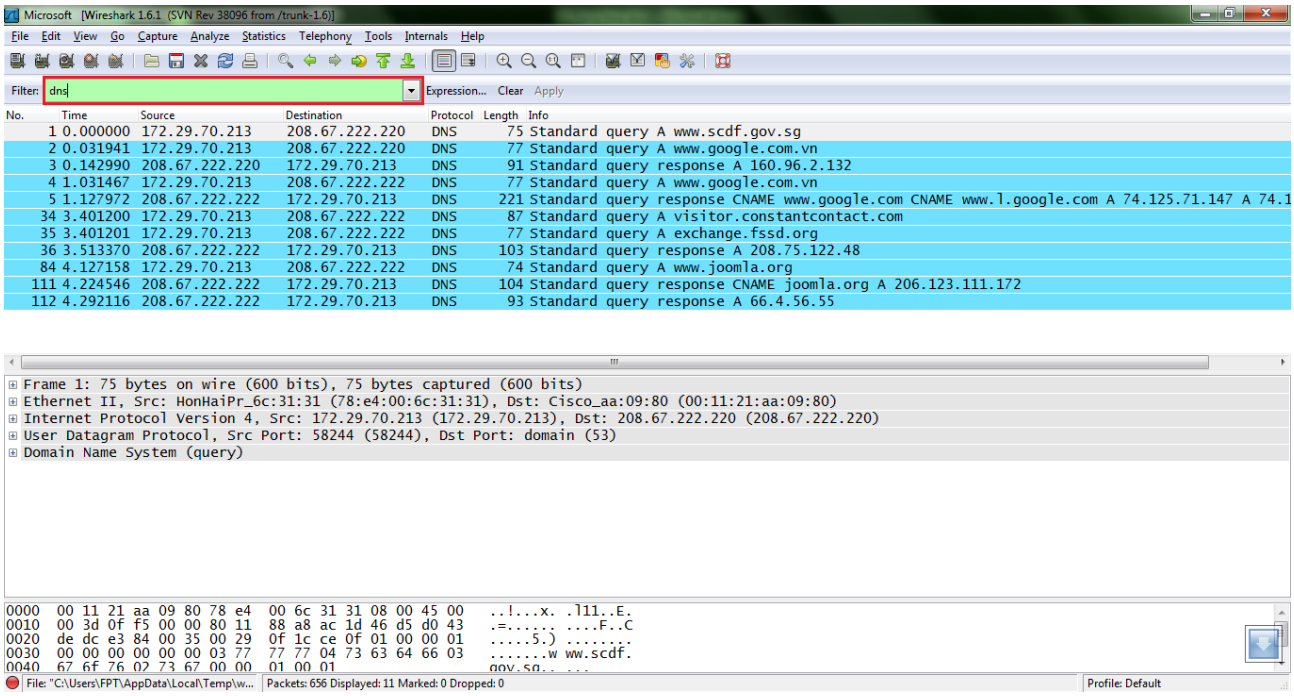
II.4.2. Lọc các gói tin sau khi bắt:

Wireshark cung cấp một cách lọc các gói tin khác sau khi bắt và lưu trữ nó một cách khá hiệu quả và đơn giản hơn. Ngôn ngữ mô tả ở đây được Wireshark xây dựng một cách đơn giản hơn vì thế cho phép bạn có thể tạo ra những điều kiện lọc gói tin chính xác và hiệu quả hơn. Bạn có thể so sánh giá trị của các trường của một gói tin thông qua các biểu thức một cách trực quan. Bạn có thể tiến hành lọc các gói tin theo :

- Loại giao thức.
- Sự xuất hiện của một trường
- Giá trị của một trường
- Và nhiều các giá trị khác.

VD :

Ta tiến hành lọc các gói tin DNS từ các gói tin bắt được bằng cách nhập chữ DNS vào trường Filter của cửa sổ hiển thị :



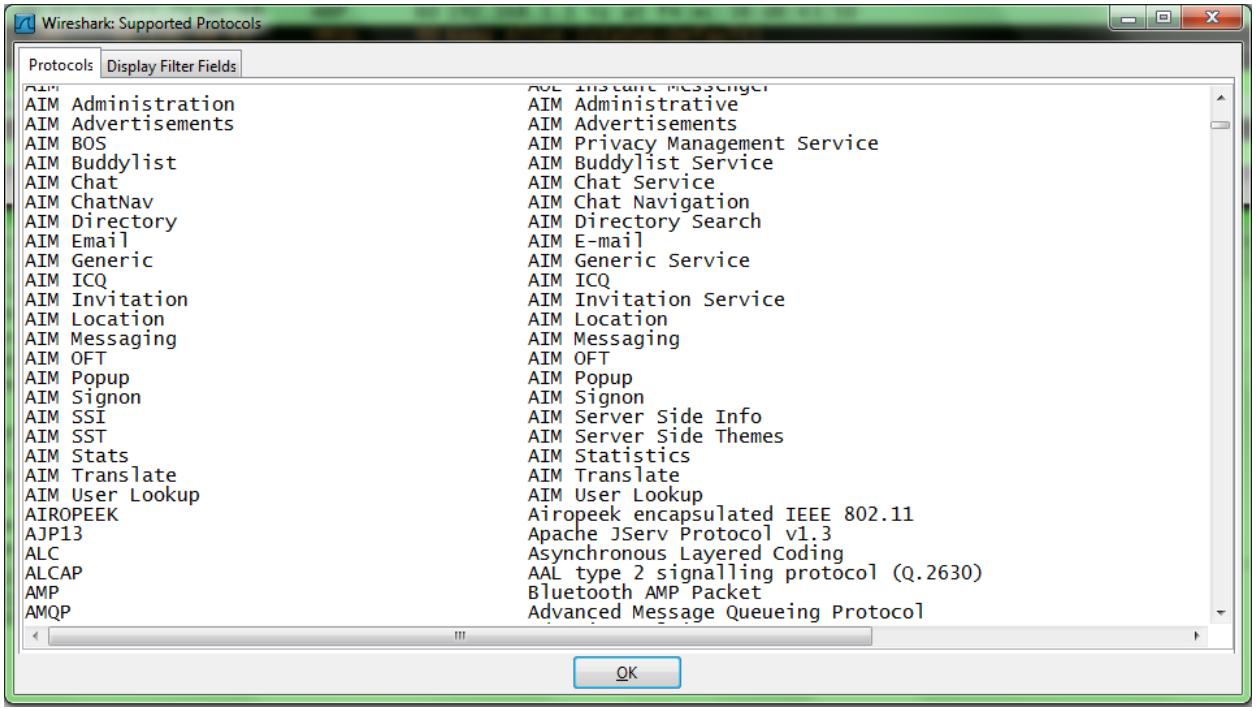
Để xây dựng tốt các miêu tả lọc gói tin bạn nên tham khảo chi tiết tại http://wiki.wireshark.org/DisplayFilters . Sau đây sẽ trình bày một cách sơ lược cách xây dựng biểu thức lọc gói tin.

Phương pháp mô tả các gói tin :

Mọi trường trong khung thông tin của Packet mà Wireshark thể hiện đều có thể sử dụng ở trong ô Filter.

Ví dụ : nếu Filter là tcp thì Wireshark sẽ tiến hành lọc các gói tin có trường này.

Một bảng danh sách đầy đủ các trường có thể tiến hành lọc được thể hiện ở Menu **Internals➔ Supported Protocals**



Tiến hành so sánh các trường :

Ta có thể tiến hành so sánh các trường của một gói tin theo các giá trị cụ thể. Bạn có thể sử dụng từ viết tắt cho tiếng anh hay sử dụng các phép so sánh của ngôn ngữ C để thể hiện việc so sánh. Bảng các phép so sánh có giá trị được liệt kê bên dưới:

English	C	Định nghĩa và ví dụ
Eq	==	**Bằng** `ip.src==10.0.0.5`
Ne	!=	**Khác** `ip.src!=10.0.0.5`
Gt	>	**Lớn hơn** `frame.len > 10`
Lt	<	**Bé hơn** `frame.len < 128`
Ge	>=	**Lớn hơn hay bằng** `frame.len ge 0x100`
Le	<=	**Bé hơn hay bằng** `frame.len <= 0x20`

Bảng sau thể hiện các trường mà bạn có thể tiến hành so sánh cũng như cách sử dụng chúng :

Type	Example

Giá trị số không dấu (8-bit, 16-bit, 24-bit, 32-bit)	`Ta có thể tiến hành so sánh các giá trị số với với trên hệ 10 hay hệ 16`
	
	`ip.len le 1500`
	`ip.len le 02734`
	`ip.len le 0x436`

Boolean	Nhấn mạnh một trường nào đó của gói tin có tồn tại hay không. Nếu trường đó tồn tại, giá trị trả ra là True và gói tin thỏa điều kiện lọc. VD : Lọc các gói tin có cờ SYN của giao thức TCP **tcp.flags.syn**

Địa chỉ Ethernet (6 bytes)	Dấu ngăn cách sử dụng ở đây có thể là dấu hai chấm (:), dấu chấm (.), dấu gạch ngang (-).
	
	`eth.dst == ff:ff:ff:ff:ff:ff`
	`eth.dst == ff-ff-ff-ff-ff-ff`
	`eth.dst == ffff.ffff.ffff`

Type	Example
IPv4	ip.addr == 192.168.0.1 Tiến hành lọc IP từ một miền xác định : ip.addr == 129.111.0.0/16
IPv6	ipv6.addr == ::1
IPX	ipx.addr == 00000000.ffffffffffff
Chuỗi	http.request.uri == "http://www.wireshark.org/"

Các phép liên kết giữa các biểu thức :

English	C-	Định nghĩa và ví dụ
and	&&	`ip.src==10.0.0.5 and tcp.flags.fin`
or	||	`ip.scr==10.0.0.5 or ip.src==192.1.1.1`
xor	^^	`tr.dst[0:3] == 0.6.29 xor tr.src[0:3] == 0.6.29`
not	!	`not llc`

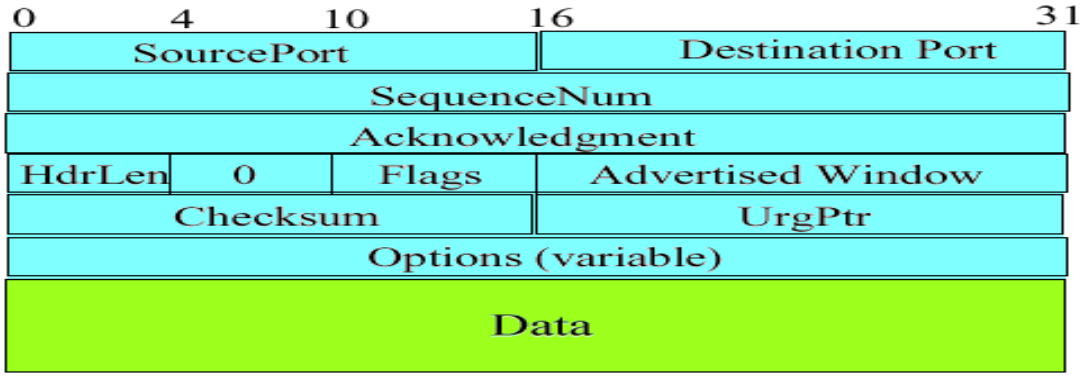
[...]		**Phân đoạn**
		
		Wireshark cho phép bạn chia các tham số thành các đoạn để so sánh với một cách khá phức tạp. Sau trường so sánh, bạn có thể đặt dấu [] và chỉ ra khoảng mà bạn muốn sử dụng để so sánh.
		
		VD:
		
		[n:m] giá trị so sánh lấy từ vị trí n và lấy m giá trị
		
		`eth.src[0:3] == 00:00:83]`
		
		[n-m] Lấy từ vị trí thứ n đến vị trí thứ m
		
		`eth.src[1-2] == 00:83`
		
		[:m] lấy các giá trị từ vị trí bắt đầu cho đến vị trí thứ m. điều này tương đương với [0:m]
		
		`eth.src[:4] == 00:00:83:00`

English	C-	Định nghĩa và ví dụ

		[n:] Lấy các giá trị từ điểm n trở về sau.
		
		`eth.src[4:] == 20:20`
		
		[n] lấy chính xác giá trị tại vị trí thứ n. Tương đương với [n:1]
		
		`eth.src[2] == 83`
		
		Wireshark cho phép bạn nối các giá trị này lại với nhau bằng dấu phẩy ngăn cách giữa chúng.
		
		`eth.src[0:3,1-2,:4,4:,2] == 00:00:83:00:83:00:00:83:00:20:20:83`

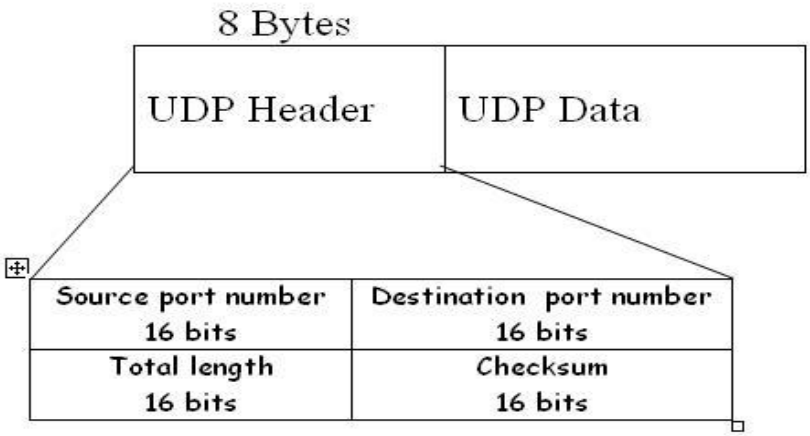
III. Cấu trúc các gói tin thông dụng

III.1.Gói tin TCP:



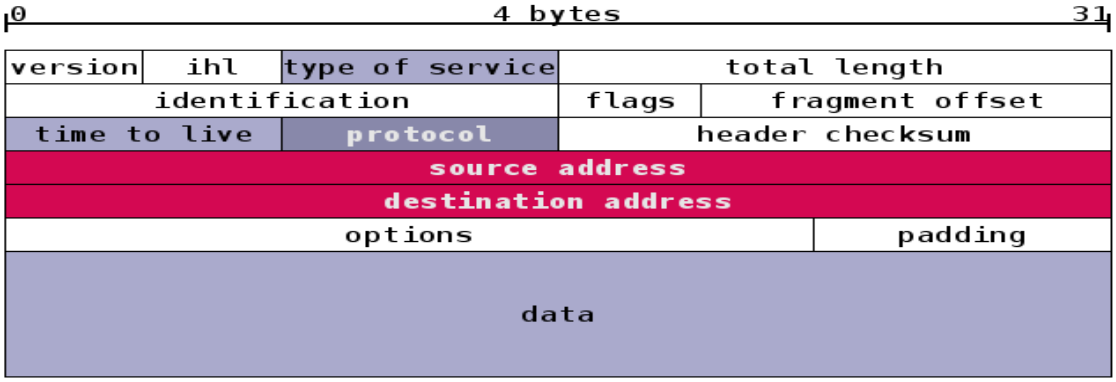
Chi tiết tham khảo tại
http://en.wikipedia.org/wiki/Transmission_Control_Protocol

III.2.Gói tin UDP



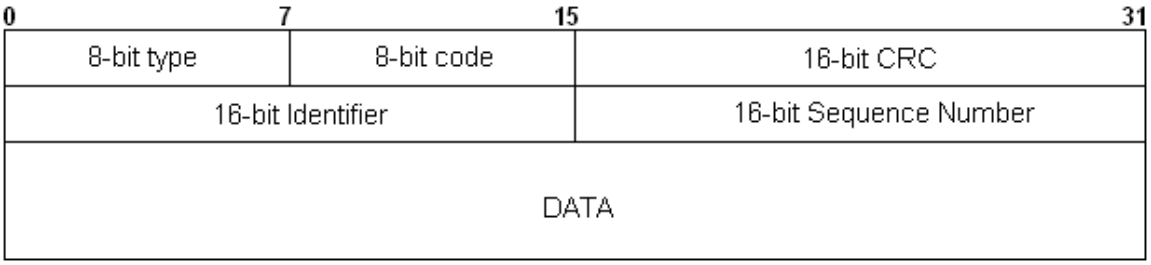
Chi tiết tham khảo tại http://en.wikipedia.org/wiki/User_Datagram_Protocol

III.3.Gói tin IP



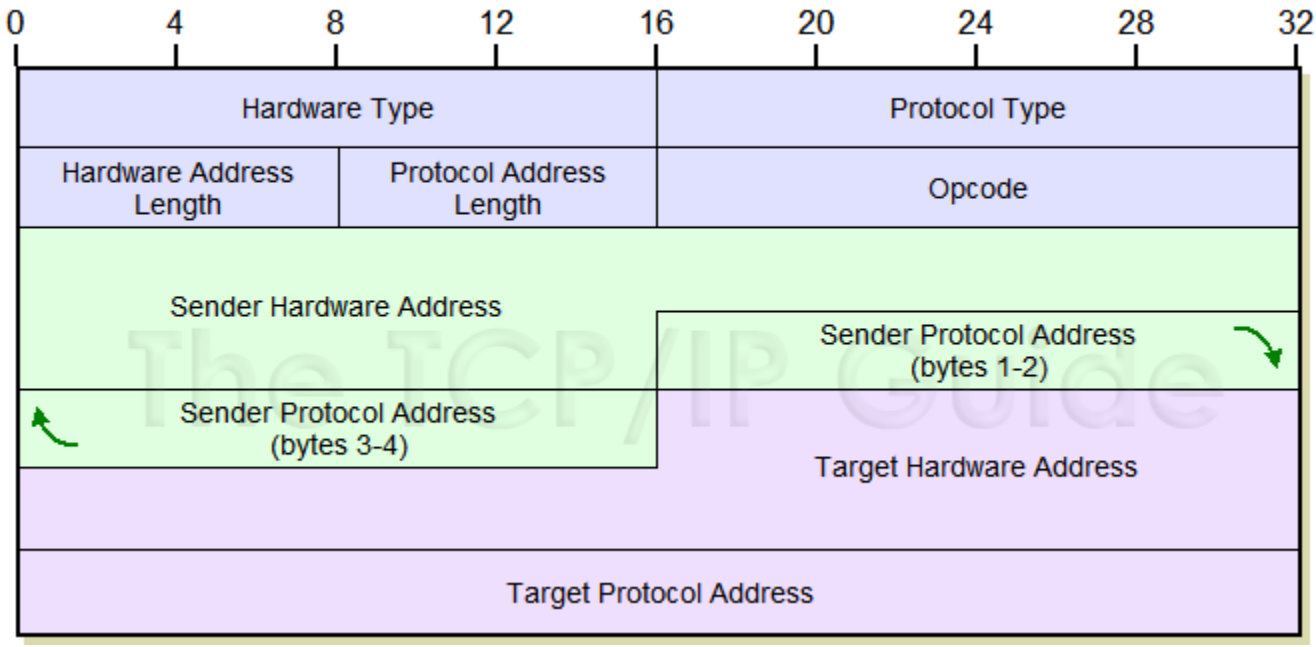
Chi tiết tham khảo tại http://en.wikipedia.org/wiki/Internet_Protocol

III.4.Gói tin ICMP



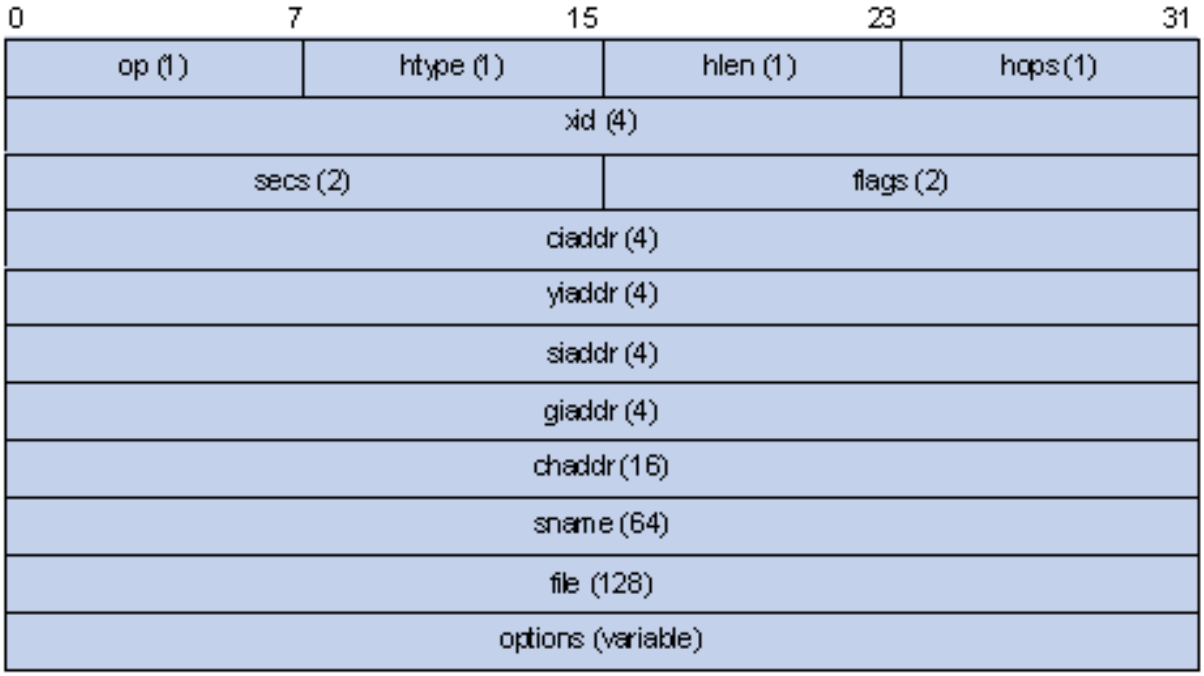
Chi tiết tham khảo tại
http://en.wikipedia.org/wiki/Internet_Control_Message_Protocol

III.5.ARP Packet:



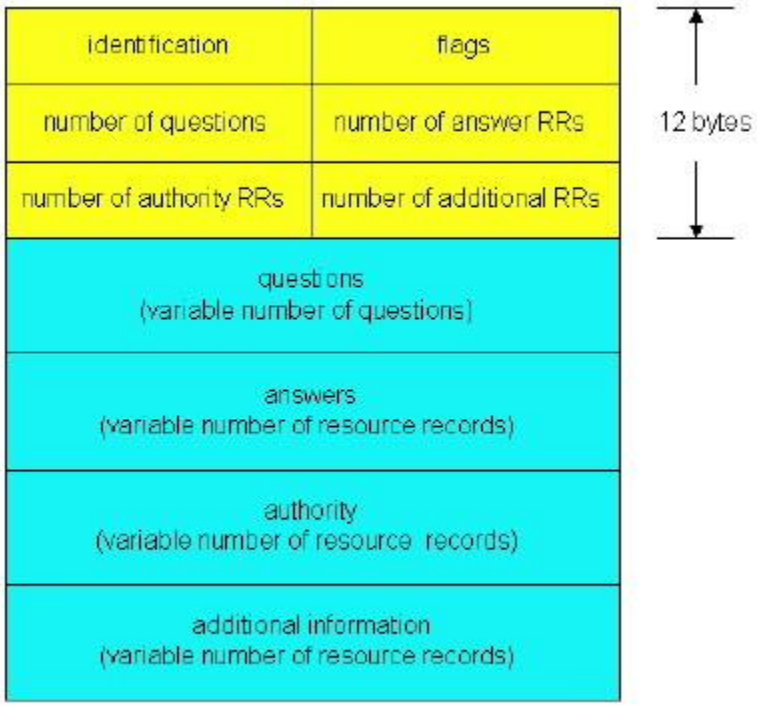
Chi tiết tham khảo tại http://en.wikipedia.org/wiki/Address_Resolution_Protocol

III.6. Gói tin DHCP:



Chi tiết tham khảo tại
http://en.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol

III.7. Gói tin DNS:



Chi tiết tham khảo tại http://en.wikipedia.org/wiki/Domain_Name_System