1/Một gói tin TCP bao gồm 2 phần

* header
* dữ liệu

+ Chức năng của TCP header:

**Header**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **+** | **Bít 0 - 3** | **4 – 9** | **10 - 15** | **16 - 31** | | **0** | Source Port | | | Destination Port | | **32** | Sequence Number | | | | | **64** | Acknowledgement Number | | | | | **96** | Data Offset | Reserved | Flags | Window | | **128** | Checksum | | | Urgent Pointer | | **160** | Options (optional) | | | | | **160/192+** | Data | | | | |

**Source port**

Số hiệu của cổng tại máy tính gửi.

**Destination port**

Số hiệu của cổng tại máy tính nhận.

**Sequence number**

Trường này có 2 nhiệm vụ. Nếu cờ SYN bật thì nó là số thứ tự gói ban đầu và byte đầu tiên được gửi có số thứ tự này cộng thêm 1. Nếu không có cờ SYN thì đây là số thứ tự của byte đầu tiên.

**Acknowledgement number**

Nếu cờ ACK bật thì giá trị của trường chính là số thứ tự gói tin tiếp theo mà bên nhận cần.

**Data offset**

Trường có độ dài 4 bít quy định độ dài của phần header (tính theo đơn vị từ 32 bít). Phần header có độ dài tối thiểu là 5 từ (160 bit) và tối đa là 15 từ (480 bít).

**Reserved**

Dành cho tương lai và có giá trị là 0.

**Flags (hay Control bits)**

Bao gồm 6 cờ:

**URG**

Cờ cho trường Urgent pointer

**ACK**

Cờ cho trường Acknowledgement

**PSH**

Hàm Push

**RST**

Thiết lập lại đường truyền

**SYN**

Đồng bộ lại số thứ tự

**FIN**

Không gửi thêm số liệu

**Window**

Số byte có thể nhận bắt đầu từ giá trị của trường báo nhận (ACK)

**Checksum**

16 bít kiểm tra cho cả phần header và dữ liệu. Phương pháp sử dụng được mô tả trong [RFC 793](http://tools.ietf.org/html/793):

*16 bít của trường kiểm tra là bổ sung của tổng tất cả các từ 16 bít trong gói tin. Trong trường hợp số octet (khối 8 bít) của header và dữ liệu là lẻ thì octet cuối được bổ sung với các bít 0. Các bít này không được truyền. Khi tính tổng, giá trị của trường kiểm tra được thay thế bằng 0,*

Nói một cách khác, tất cả các từ 16 bít được cộng với nhau. Kết quả thu được sau khi đảo giá trị từng bít được điền vào trường kiểm tra. Về mặt thuật toán, quá trình này giống với [IPv4](https://vi.wikipedia.org/wiki/IPv4).

Điểm khác nhau chỉ ở chỗ dữ liệu dùng để tính tổng kiểm tra. Dưới đây là một header của IP:

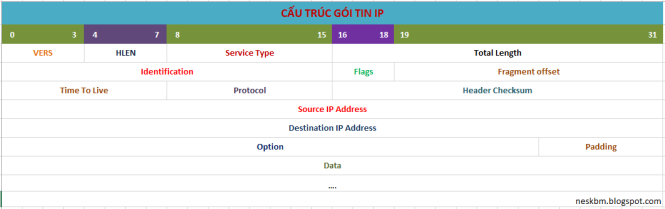
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **+** | **Bít 0 - 3** | **4 - 7** | **8 - 9** | **10 - 15** | **16 - 31** | | **0** | Source address | | | | | | **32** | Destination address | | | | | | **64** | Zeros | | Protocol | | TCP length | | **96** | Source Port | | | | Destination Port | | **128** | Sequence Number | | | | | | **160** | Acknowledgement Number | | | | | | **192** | Data Offset | Reserved | | Flags | Window | | **225** | Checksum | | | | Urgent Pointer | | **257** | Options (optional) | | | | | | **257/289+** | Data | | | | | |

Các địa chỉ nguồn và đích là các địa chỉ [IPv4](https://vi.wikipedia.org/wiki/IPv4). Giá trị của trường protocol là 6 (giá trị dành cho TCP, xem thêm: [Danh sách số hiệu giao thức IPv4](https://vi.wikipedia.org/w/index.php?title=Danh_s%C3%A1ch_s%E1%BB%91_hi%E1%BB%87u_giao_th%E1%BB%A9c_IPv4&action=edit&redlink=1)). Giá trị của trường TCP length field là độ dài của toàn bộ phần header và dữ liệu của gói TCP.

**Urgent pointer**

Nếu cờ URG bật thì giá trị trường này chính là số từ 16 bít mà số thứ tự gói tin (*sequence number*) cần dịch trái.

Câu 2:



**VERS**(4 bit): chỉ ra phiên bản hiện hành của IP đang được dùng, có 4 bit. Nếu trường này khác với phiên bản IP của thiết bị nhận, thiết bị nhận sẽ từ chối và loại bỏ các gói tin này.

**HLEN**(**4** bit):  chỉ độ dài phần tiêu đề (Internet Header Length) của datagram, tính theo đơn vị word (**32**bits). Nếu không có trường này thì độ dài mặc định của phần tiêu đề là 5 từ.

**Service Type** (**8**bits): cho biết các thông tin về loại dịch vụ và mức ưu tiên của gói IP, có dạng cụ thể như sau:

[https://kysudien.files.wordpress.com/2014/12/6c50f-untitled.png?w=400&h=32](https://kysudien.files.wordpress.com/2014/12/6c50f-untitled.png)

**Precedence**(**3**bits): chỉ thị về quyền ưu tiên gửi datagram, cụ thể là:

* 111 Network Control (cao nhất)
* 011- flash
* 110 Internetwork Control
* 010 Immediate
* 101 CRITIC/ECP
* 001 Priority
* 100 Flas Override
* 000 Routine (thấp nhất)

**D**(delay)  (1 bit) : chỉ độ trễ yêu cầu

* D = 0 độ trễ bình thường,
* D = 1 độ trễ thấp

**T**(Throughput) (1 bit) : chỉ số thông lượng yêu cầu

* T = 1 thông l ượng bình thường
* T = 1 thông l ượng cao

**R**(Reliability) (1 bit): chỉ độ tin cậy yêu cầu

* R = 0 độ tin cậy bình thường
* R = 1 độ tin cậy cao

– **Total Length** (**16**bits): chỉ độ dài toàn bộ datagram, kể cả phần header (tính theo đơn vị bytes), vùng dữ liệu của datagram có thể dài tới 65535 bytes. Để biết chiều dài của dữ liệu chỉ cần lấy tổng chiều dài này trừ đi HLEN.

– **Identification**(**16**bit):  cùng với các tham số khác như (Source Address và Destination Address) tham số này dùng để định danh duy nhất cho một datagram trong khoảng thời gian nó vẫn còn trên liên mạng. Đây là 1 số nguyên.

– **Flags**(**3**bits): Liên quan đến sự phân đoạn (Fragment) của datagram. cụ thể:

**0        DF         MF**

**Bit 0**: reserved chưa sử dụng luôn lấy giá trị 0

**Bit 1**:

* DF = 1: Gói tin bị phân đoạn, có nhiều hơn 1 đoạn
* DF = 0: Gói tin ko bị phân đoạn.

**Bit 2**:

* MF = 0: Đây là đoạn cuối cùng
* MF = 1: Đây chưa phải là đoạn cuối cùng, còn đoạn khác phía sau nữa

**Fragment Offset**(**13**bits):  chỉ vị trí của đoạn (fragment) ở trong datagram, tính theo đơn vị 64 bits, có nghĩa là mỗi đoạn (trừ đoạn cuối cùng) phải chứa một vùng dữ liệu có độ dài là bội của 64 bits. Nó được dùng để ghép các mảnh Datagram lai với nhau

**Time To Live** (**TTL**– **8**bit): giá trị này được đặt lúc bắt đầu gửi gói tin. Và nó sẽ giảm dần khi đi qua 1 router. gói tin sẽ bị hủy nếu giá trị này = 0 khi chưa đến đích. Việc làm này nhằm giải quyết vấn đề gói tin bị lặp vô hạn trên mạng.

**Protocol** (**8**bits): Chỉ ra giao thức lớp trên, chẳng hạn như TCP hay UDP

**Header Checksum**:  mã kiểm soát lỗi sử dụng phương pháp CRC (Cyclic Redundancy Check) dùng để đảm bảo thông tin về gói dữ liệu được truyền đi một cách chính xác (mặc dù dữ liệu có thể bị lỗi). Nếu như việc kiểm tra này thất bại, gói dữ liệu sẽ bị huỷ bỏ tại nơi xác định được lỗi. Cần chú ý là IP không cung cấp một phương tiện truyền tin cậy bởi nó không cung cấp cho ta một cơ chế để xác nhận dữ liệu truyền tại điểm nhận hoặc tại những điểm trung gian. Giao thức IP không có cơ chế Error Control cho dữ liệu truyền đi, không có cơ chế  kiểm soát luồng dữ liệu (flow control).

**Source Address** (**32**bits): địa chỉ của trạm nguồn.

**Destination Address** (**32**bits): địa chỉ của trạm đích.

**Option**(**có độ dài thay đổi**) sử dụng trong một số trường hợp, nhưng thực tế chúng rất ít dùng. Option bao gồm bảo mật, chức năng định tuyến đặc biệt

**Padding**(**độ dài thay đổi**): Các số 0 được bổ sung vào field này để đảm bảo IP Header luôn la bội số của 32 bit.

**Data**(**độ dài thay đổi**): vùng dữ liệu có độ dài là bội của 8 bits, tối đa là 65535 bytes.

Câu 3:

Giao thức ARP (Address Resolution Protocol) đã được xây dựng để chuyển đổi từ địa chỉ IP sang địa chỉ vật lý khi cần thiết

Câu 4:

Giao thức ICMP

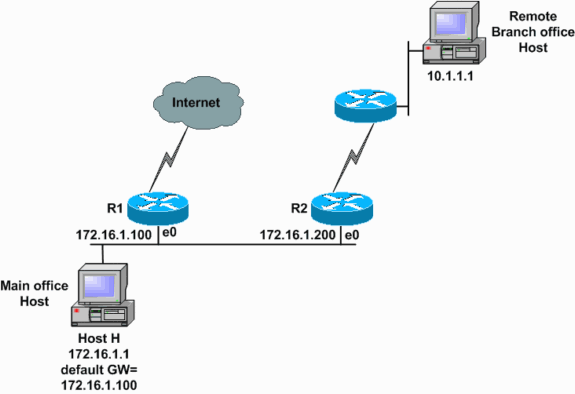
ICMP ((Internet Control Message Protocol) là một giao thức điều khiển của mức IP, được dùng để trao đổi các thông tin điều khiển dòng số liệu, thông báo lỗi và các thông tin trạng thái khác của bộ giao thức TCP/IP. Ví dụ:

Điều khiển lưu lượng dữ liệu (Flow control).

Thông báo lỗi : ví dụ  "Destination Unreachable".

Định hướng lại các tuyến đường: gói tin redirect

Kiểm tra các trạm ở xa: gói tin echo



Ví dụ khuôn dạng của thông điệp ICMP redirect như sau:

|  |  |  |
| --- | --- | --- |
| 0              7    8       15  16     31 | | |
| type (5) | Code(0-3) | Checksum |
| Địa chỉ IP của Router mặc định | | |
| IP header (gồm option) và 8 bytes đầu của gói dữ liệu IP nguồn | | |

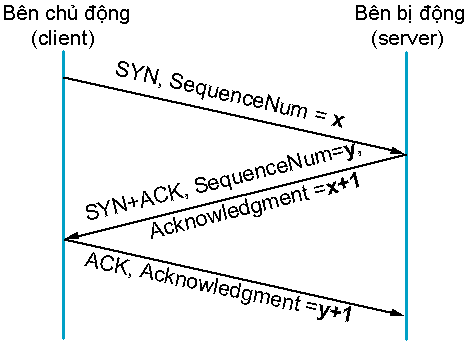
Câu 5:

SMTP là một giao thức dùng nền văn bản và tương đối đơn giản. Trước khi một thông điệp được gửi, người ta có thể định vị một hoặc nhiều địa chỉ nhận cho thông điệp - những địa chỉ này thường được kiểm tra về sự tồn tại trung thực của chúng). Việc kiểm thử một trình chủ SMTP là một việc tương đối dễ dàng, dùng chương trình ứng dụng "[telnet](https://vi.wikipedia.org/wiki/Telnet)" (xem dưới đây).

SMTP dùng [cổng](https://vi.wikipedia.org/w/index.php?title=Danh_s%C3%A1ch_c%C3%A1c_c%E1%BB%95ng_c%E1%BB%A7a_TCP_v%C3%A0_UDP&action=edit&redlink=1) 25 của giao thức [TCP](https://vi.wikipedia.org/wiki/TCP). Để xác định trình chủ SMTP của một tên miền nào đấy (*domain name*), người ta dùng một [mẫu tin MX](https://vi.wikipedia.org/w/index.php?title=M%E1%BA%ABu_tin_MX&action=edit&redlink=1) (*Mail eXchange - Trao đổi thư*) của [DNS](https://vi.wikipedia.org/wiki/DNS) (*Domain Name System - Hệ thống tên miền*).

SMTP định nghĩa tất cả những gì đã làm với email. Nó xác định cấu trúc của các địa chỉ, yêu cầu tên miền và bất cứ điều gì liên quan đến email. SMTP cũng xác định các yêu cầu cho Post Office Protocol (POP) và truy cập Internet Message Protocol (IMAP) máy chủ, do đó email được gửi đúng cách.

Câu 6:



Ở lớp Transport có 2 giao thức quan trọng là[UDP và TCP](http://kenhdaihoc.com/links.php?url=http://kenhdaihoc.com/links.php?url=http://kenhdaihoc.com/forum/showthread.php?t=2953).  
  
- [TCP](http://kenhdaihoc.com/links.php?url=http://kenhdaihoc.com/links.php?url=http://kenhdaihoc.com/forum/showthread.php?t=2859) là giao thức thuộc dạng connection-oriented (hướng kết nối). Có nghĩa là nó thiết lập kênh kết nối trước khi truyển data đi.  
  
- UDP là giao thức thuộc dạng connectionless (nghĩa là không hướng kết nối). Nó không cần thiết lập kênh truyền trước khi truyền dữ liệu đi.  
  
[TCP](http://kenhdaihoc.com/links.php?url=http://kenhdaihoc.com/links.php?url=http://kenhdaihoc.com/forum/showthread.php?t=2859)thiết lập kết nối bằng 3 bước bắt tay (3-way handshake)  
  
sender \_\_\_\_\_\_\_\_\_\_\_ receiver  
  
SYN seq=X ----------> SYN received (step 1)  
  
SYN received <--------send ACK X+1 and SYN Y (step 2)​

Send ACK Y+1 --------> (step 3)  
​

​

**Tìm hiểu:   
  
​**

Có lẽ kiến thức về TCP/IP là những gì bạn cần phải tìm hiểu đầu tiên trên con đường trở thành Network Administrator hay Security Advisor. Kiến thức liên quan đến bộ giao thức này có rất nhiều, bạn nên dành thời gian để tìm hiểu thật kỹ về nó. Trong bài này chúng ta sẽ đề cập đến vấn đề 3 bước bắt tay tạo kết nối trong TCP.  
Một gói dữ liệu TCP chứa các bit cờ (flag bits) để mô tả nội dung và mục đích của gói dữ liệu.  
Ví dụ:  
  
\* Gói dữ liệu với cờ SYN (synchronization <=> Sự đồng bộ) dùng để bắt đầu một connection.  
\* ACK (acknowledgement <=> Xác nhận).  
\* FIN (finish <=> hoàn thành) dùng để ngắt một connection.  
\* ...  
  
Bây giờ chúng ta sẽ tìm hiểu kỹ hơn một phiên (session) làm việc được bắt đầu như thế nào:  
  
  
  
1. SYN: các chương trình máy con (ví dụ yêu cầu từ browser, ftp client) bắt đầu connection với máy chủ bằng cách gửi một packet với cờ "SYN" đến máy chủ.  
SYN packet này thường được gửi từ các cổng cao (1024 - 65535) của máy con đến những cổng trong vùng thấp (1 - 1023) của máy chủ. Chương trình trên máy con sẽ hỏi hệ điều hành cung cấp cho một cổng để mở connection với máy chủ. Những cổng trong vùng này được gọi là "cổng máy con" (client port range). Tương tự như vậy, máy chủ sẽ hỏi HĐH để nhận được quyền chờ tín hiệu trong máy chủ, vùng cổng 1 - 1023. Vùng cổng này được gọi là "vùng cổng dịch vụ" (service port).  
Ví dụ (mặc định):  
  
- Web Server sẽ luôn chờ tín hiệu ở cổng 80 và Web browser sẽ connect vào cổng 80 của máy chủ.  
- FTP Server sẽ lắng ở port 21.  
  
Ngoài ra trong gói dữ liệu còn có thêm địa chỉ IP của cả máy con và máy chủ.  
  
2. SYN/ACK: khi yêu cầu mở connection được máy chủ nhận được tại cổng đang mở, server sẽ gửi lại packet chấp nhận với 2 bit cờ là SYN và ACK.  
SYN/ACK packet được gửi ngược lại bằng cách đổi hai IP của server và client, client IP sẽ thành IP đích và server IP sẽ thành IP bắt đầu. Tương tự như vậy, cổng cũng sẽ thay đổi, server nhận được packet ở cổng nào thì cũng sẽ dùng cổng đó để gửi lại packet vào cổng mà client đã gửi.  
Server gửi lại packet này để thông báo là server đã nhận được tín hiệu và chấp nhận connection, trong trường hợp server không chấp nhận connection, thay vì SYN/ACK bits được bật, server sẽ bật bit RST/ACK (Reset Acknowledgement) và gởi ngược lại RST/ACK packet.  
Server bắt buộc phải gửi thông báo lại bởi vì TCP là chuẩn tin cậy nên nếu client không nhận được thông báo thì sẽ nghĩ rằng packet đã bị lạc và gửi lại thông báo mới.  
  
3. ACK: khi client nhận được SYN/ACK packet thì sẽ trả lời bằng ACK packet. Packet này được gởi với mục đích duy báo cho máy chủ biết rằng client đã nhận được SYN/ACK packet và lúc này connection đã được thiết lập và dữ liệu sẽ bắt đầu lưu thông tự do.  
Đây là tiến trình bắt buộc phải thực hiện khi client muốn trao đổi dữ liệu với server thông qua giao thức TCP. Một số thủ thuật dựa vào đặc điểm này của TCP để tấn công máy chủ (ví dụ DOS).

Câu 7:

Cả hai giao thức này đều được dùng để gửi dữ liệu qua Internet hoặc qua mạng cục bộ . Khi cấu hình một số phần cứng hoặc phần mềm mạng , bạn có thể cần biết sự khác nhau giữa hai giao thức này .

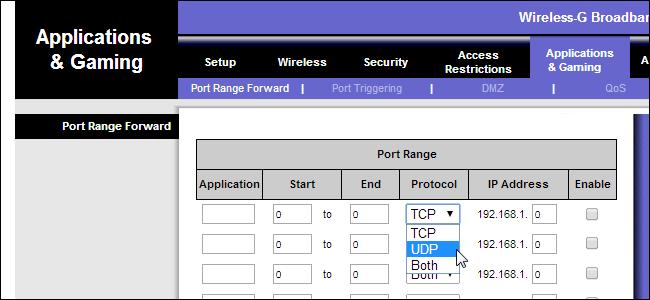
**Những điểm chung**

TCP và UDP là những giao thức dùng để gửi các Bit dữ liệu , hoặc là những gói ( Packet ) dữ liệu , qua Internet . Cả hai đều được tạo trên đầu của giao thức Internet .

Bạn đang truyền một gói dữ liệu qua TCP hoặc UDP , đó là gói dữ liệu được gửi tới một địa chỉ IP nào đó . Những gói dữ liệu này được xử lí tương tự nhau và chúng chuyển đi từ máy tính của bạn tới những Router trung gian và tới những đích cụ thể .

TCP và UDP không  chỉ là những giao thức làm việc phía trên cùng của IP mà chúng còn được sử dụng một cách rộng rãi .

Khái niệm hay được dùng nhất là TCP/IP có nghĩa là TCP trên IP . UDP cũng trên IP nên cũng được dùng gọi tắt là UDP/IP nhưng khai niệm này ít được sử dụng .



**TCP làm việc như thế nào ?**

TCP được viết tắt từ “Transmission Control Protocol” . Nó là giao thức được dùng thông dụng nhất trên Internet .

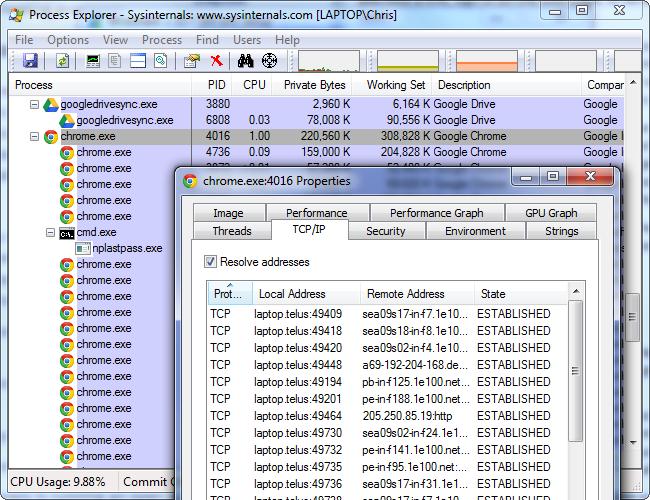
Khi bạn tải một trang web , máy tính của bạn gửi những gói TCP tới địa chỉ của máy chủ web , yêu cầu nó gửi trang web cho bạn . Máy chủ web trả lời lại bằng cách gửi một luồng dữ liệu TCP để trình duyệt  web xử lí để tái tạo lại nội dung trang web và hiển thị lại trên màn hình máy tính của bạn .

Khi bạn bấm vào một đường Link , đăng nhập , đăng bài bình luận hoặc làm bất kì điều gì , trình duyệt  web của bạn gửi những gió dữ liệu TCP tới máy chủ và máy chủ lại gửi những gói TCP phản hồi .

TCP bảo đảm người nhận sẽ nhận những gói dữ liệu theo thứ tự được đánh số . Người nhận gửi tin nhắn lại cho người gửi nói rằng đã nhận được tin . Nếu người gửi không  nhận được thông tin phản hồi chính xác , nó sẽ gửi lại những gói dữ liệu để bảo đảm người nhận đã nhận được chúng .

Những gói dữ liệu cũng được kiểm tra tìm lỗi vì thế TCP là giao thức có độ tin cậy cao . Những gói dữ liệu được gửi bằng TCP được theo dõi để không dữ liệu nào bị mất hoặc hỏng trong quá trình truyền . Điều đó có thể giải thích tại sao những file tải về không bị lỗi nếu như mạng bị ngắt giữa chừng .

[Process Explorer](http://www.howtogeek.com/school/sysinternals-pro/lesson2/) và những tiện ích hệ thống  khác có thể hiển thị kiểu kết nối trong quá trình xử lí . Chúng ta có thể thấy trình duyệt  Chrome sẽ mở những cổng kết nối TCP tới những máy chủ web khác nhau .



**UDP làm việc như thế nào ?**

UDP được viết tắt từ User Datagram Protocol , Datagram cũng tương tự như khái niệm gói thông tin . Giao thức UDP làm việc tương tự như TCP nhưng loại bỏ tất cả phần kiểm tra lỗi .

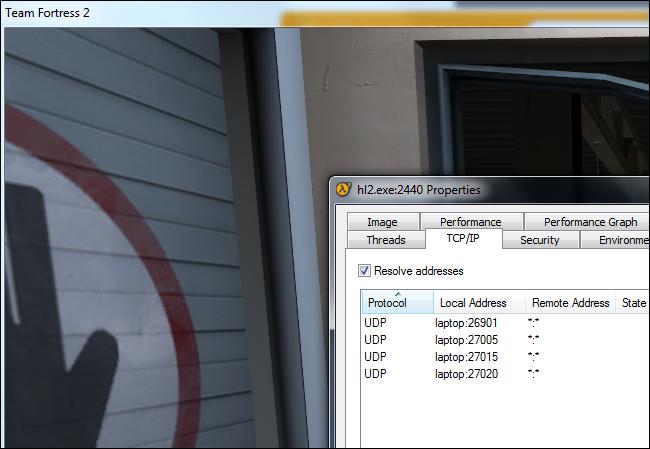
Khi dùng UDP , những gói dữ liệu chỉ gửi tới người nhận . Người gửi không chờ để bảo đảm người nhận đã nhận đủ dữ liệu và nó sẽ tiếp tục gửi những gói dữ liệu tiếp theo . Nếu là bên nhận và bạn thiếu mất vài gói dữ liệu bạn không  thể yêu cầu truyền lại những gói dữ liệu bị mất . UDP không bảo đảm bạn sẽ nhận được tất cả những gói dữ liệu và không có cách nào để yêu cầu gửi lại nếu như bạn không nhận được hoặc nhận bị lỗi . Tuy nhiên UDP có lợi thế về tốc độ truyền thông cực nhanh .

UDP được dùng  khi cần ưu tiến tốc độ và không thực sự cần sửa lỗi . Ví dụ UDP thường được dùng trong truyền tin trực tiếp và Game trực tuyến .

Ví dụ bạn đang xem nội dung video trực tiếp và việc truyền thông thường dùng UDP thay vì TCP . Máy chủ chỉ gửi nội dung những gói dữ liệu UDP tới những máy tính đang xem . Nếu bạn mất kết nối của mình trong vài giây , video sẽ bị dừng lại một lúc sau đó lại nhảy tới nội dung hiện thời đang truyền trực tiếp bỏ qua những gì bạn không nhận được .

Việc làm này cũng tương tự như game trực tuyến , nếu bạn mất một số gói UDP , những nhân vật trong game có thể xuất hiện dịch chuyển qua bản đồ bạn nhận được trong những gói UDP mới hơn . Sẽ không cần yêu cầu truyền lại gói dữ liệu cũ bị mất vì game vẫn tiếp tục chạy khi không có bạn .

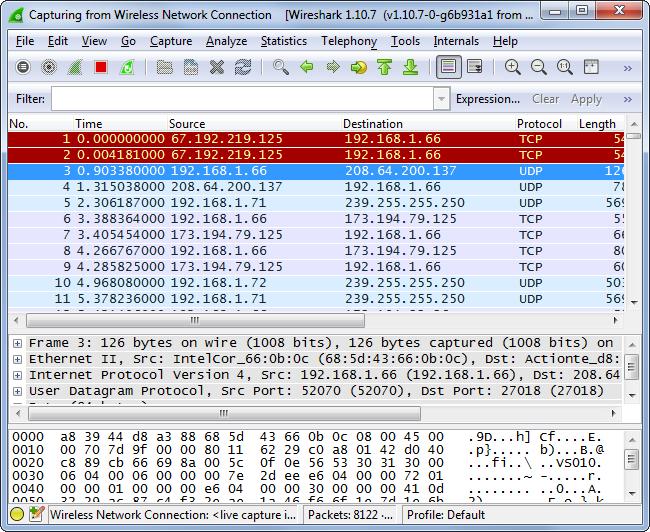
Vì bỏ qua tất cả những việc kiểm tra lỗi như của TCP nên cho phép tốc độ truyền dữ liệu nhanh hơn và giảm thời gian trễ .



Như vậy tùy theo ứng dụng mà bạn cần sử dụng theo giao thức truyền dữ liệu nào . Hầu hết những ứng dụng muốn sửa lỗi thì phải cần giao thức TCP , nhưng một số ứng dụng cần tốc độ và giảm thời gian trễ thì cần dùng UDP .

Dùng công cụ phân tích mạng như [Wireshark](http://www.wireshark.org/download.html) và bạn sẽ thấy những kiểu khác nhau của những gói dữ liệu đang truyền thông trong hệ thống  .

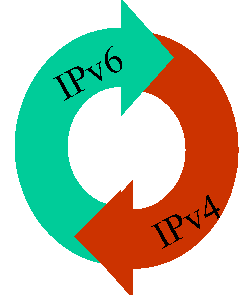
Nếu như đang cấu hình cho Router hoặc phần mềm Firewall bạn không rõ ứng dụng nào dùng TCP hay UDP , bạn có thể lựa chọn “Both” để chặn cả hai lưu lượng truyền thông theo giao thức TCP và UDP .



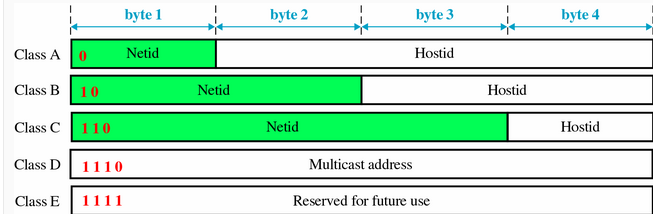
ICMP (Internet Control Message Protocol) là giao thức thông thường khác được dùng trên Internet . Nó thường được dùng để truy vấn và những thông báo lỗi giữa những Router và các thiết bị mạng khác . Những ứng dụng thông thường không dùng ICMP , mặc dù lệnh **ping** và **traceroute** dùng ICMP để cho những chẩn đoán của chúng .

Câu 8:

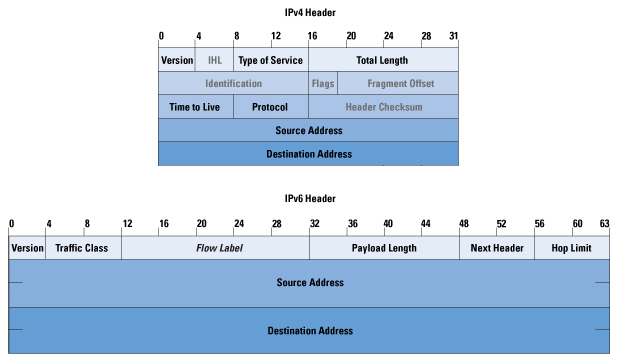
Mỗi máy kết nối với Internet thông qua một địa chỉ duy nhất và đó được gọi là địa chỉ IP.Địa chỉ IP chia ra thành 5 lớp A, B, C, D, E, hiện tại đã dùng hết lớp A,B và gần hết lớp C, còn lớp D và E Tổ chức internet đang để dành cho mục đích khác không phân, nên chúng ta chỉ nghiên cứu 3 lớp đầu. Bit nhận dạng là những bit đầu tiên của lớp A là 0, của lớp B là 10, của lớp C là 110. Lớp D có 4 bit đầu tiên để nhận dạng là 1110, còn lớp E có 4 bít đầu tiên để nhận dạng là 1111. Do đó địa chỉ ví dụ ở trên bắt đầu bằng 11000100 nên thuộc lớp C.



Một địa chỉ IP được phân biệt bởi hai phần, phần đầu gọi là Network ID (địa chỉ mạng) và phần sau là Host ID. Ví dụ đối với lớp A (có địa chỉ từ 0.0.0.0 đến 127.0.0.0 ), bit thứ nhất là bit nhận dạng lớp A = 0, 7 bit còn lại trong Octet thứ nhất dành cho địa chỉ mạng, 3 Octet còn lại có 24 bit dành cho địa chỉ của máy chủ. Do vậy, trên lớp A, có thể phân cho 126 mạng khác nhau và mỗi mạng có thể có tối đa 16777214 máy host.



Sự khác nhau đáng kể nhất giữa IPv4 và IPv6 là chiều dài của địa chỉ nguồn và địa chỉ của chúng. Việc chuyển sang sử dụng IPv6 là do ngày càng thiếu về số địa chỉ IP. Giao thức IPv6 có một không gian địa chỉ lớn hơn so với giao thức IPv4. Sau đây là sự khác biệt cụ thể:



**- Đối với giao thức IPv4**

Kể từ khi Internet tra đời và phát triển vào đầu những năm 80 thì đã được sử dụng IPv4với 32 bit và chia thành 4 Octet ( mỗi Octet có 8 bit, tương đương 1 byte ) cách đếm đều từ trái qua phải bit 1 cho đến bít 32, các Octet tách biệt nhau bằng dấu chấm (.). Ví dụ như 1 địa chỉ IP như sau: 196.84.156.67, địa chỉ IP được chia thành 4 số giới hạn từ 0 - 255 (vì 255 tương đương 11111111 (ở hệ nhị phân) là số lớn nhất có 8 bit).



**- Đối với giao thức IPv6**

Tương phản với IPv4, địa chỉ IPv6 có chiều dài là 128bit. Điều đó cho phép có thể biểu diễn đến 3.4x1038 (340.000.000.000.000.000.000.000.000.000.000.000.0 00) địa chỉ. Có một vài sự khác nhau trong cách biểu diễn địa chỉ của IPv6, một địa chỉ IPv6 thường được viết thành 8 nhóm, mỗi nhóm gồm có 4 số hex và mỗi nhóm được tách biệt với nhau bằng dấu (:) Ví dụ như sau thể hiện điều này 2001:0f68:0000:0000:0000:0000:1986:69af.



Bạn xem địa chỉ mẫu ở trên và có thể nghĩ rằng việc đánh một địa chỉ IPv6 phải mất rất nhiều thời gian và công sức. Nhưng thực chất, địa chỉ IPv6 có thể được viết vắn tắt bằng việc giảm thiểu các số 0. Có hai nguyên tắc phải tuân theo ở đây khi biểu diễn một địa chỉ IP, đầu tiên, một dãy bốn số 0 liên tục có thể được thay thế bằng hai dấu (::) Bằng cách đó địa chỉ IPv6 ở trên có thể được viết tắt như sau: 2001:0f68::0000:0000:0000:1986:69af.

Trong ví dụ ở trên, chúng ta chỉ có thể ước lượng một khối các chữ số 0 bởi vì nguyên tắc này phát biểu rằng chỉ có một cặp (::) trong một địa chỉ. Địa chỉ đang ví dụ ở trên vẫn còn rất nhiều chữ số cần phải đánh. Tuy nhiên, nguyên tắc thứ hai sẽ cho phép bạn thực hiện địa chỉ này ngắn hơn, các số 0 trong một nhóm có thể được bỏ qua. Nếu một khối 4 số bắt đầu của nó là số 0 thì số 0 này có thể được lược bỏ bớt để lại là 3 số 0 trong khối.

Nếu khối ba số đó cũng lại bắt đầu với một số 0 đứng đầu thì ta có thể tiếp tục loại bỏ, cứ như vậy đến khi gặp số khác 0 trong nhóm thì dừng. Trường hợp nếu 4 số trong nhóm đều là 0 thì số được giữ lại cuối cùng là một số 0. Nếu cứ nói mãi mà không biểu diễn trong ví dụ cụ thể để các bạn dễ theo dõi thì đó là một thiếu sót. Dưới đây là những gì mà bạn có thể áp dụng cả hai nguyên tắc đó cho địa chỉ:

*2001:0f68:0000:0000:0000:0000:1986:69af*

*2001:f68:000:000:000:000:1986:69af*

*2001:f68:00:00:00:00:1986:69af*

*2001:f68:0:0:0:0:1986:69af*

*2001:f68::1986:69af*

**Câu 9:**

Một **bộ định tuyến**[[một]](https://en.wikipedia.org/wiki/Router_(computing)#cite_note-1) là một thiết bị kết nối mạng cho phép gửi các [gói dữ liệu](https://en.wikipedia.org/wiki/Data_packet) giữa [các mạng máy](https://en.wikipedia.org/wiki/Computer_network)tính. Router thực hiện "chỉ đạo giao thông" các chức năng trên [Internet.](https://en.wikipedia.org/wiki/Internet)Một gói dữ liệu thường được chuyển tiếp từ một router khác thông qua mạng mà tạo thành liên mạng cho đến khi nó đạt đến node đích của nó

**Switch** ([tiếng Anh](https://vi.wikipedia.org/wiki/Ti%E1%BA%BFng_Anh)), hay còn gọi là **thiết bị chuyển mạch**, là một thiết bị dùng để kết nối các [đoạn mạng](https://vi.wikipedia.org/w/index.php?title=%C4%90o%E1%BA%A1n_m%E1%BA%A1ng&action=edit&redlink=1) với nhau theo mô hình [mạng](https://vi.wikipedia.org/wiki/M%E1%BA%A1ng_m%C3%A1y_t%C3%ADnh) hình sao (*star*). Theo mô hình này, switch đóng vai trò là thiết bị trung tâm, tất cả các máy tính đều được nối về đây. switch nhận tín hiệu vật lý, chuyển đổi thành dữ liệu, từ một cổng, kiểm tra địa chỉ đích rồi gửi tới một cổng tương ứng.