BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG
-------o0o-------

TÌM HIỂU KỸ THUẬT TRUYỀN FILE MULTICAST

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

NGÀNH CÔNG NGHỆ THÔNG TIN

Sinh viên thực hiện : Nguyễn Thị Hằng
Người hướng dẫn : Ths. Đỗ Xuân Toàn
Mã số sinh viên: 121175

Hải Phòng - 2012

MỤC LỤC

LỜI CẢM ƠN

Em xin chân thành cảm ơn thầy Đỗ Xuân Toàn giảng viên trường Đại học dân lập Hải Phòng đã tận tình hướng dẫn và tạo mọi điều kiện thuận lợi để em hoàn thành bài báo cáo tốt nghiệp của mình.

Em xin chân thành cảm ơn tất cả các thầy, cô giáo khoa Công nghệ thông tin trường Đại Học Dân Lập Hải Phòng đã nhiệt tình giảng dạy và cung cấp những kiến thức quý báu để em có thể hoàn thành tốt đồ án tốt nghiệp này.

Em xin cảm ơn tất cả các bạn đã động viên, góp ý và trao đổi hỗ trợ cho em trong suốt thời gian vừa qua.

Và cuối cùng ,em kính chúc thầy cô sức khỏe, tiếp tục đạt được nhiều thành tích trong giảng dạy, cũng như trong nghiên cứu khoa học và trong sự nghiệp giáo dục.

Vì thời gian tìm hiểu đồ án có hạn, trình độ bản thân còn nhiều hạn chế. Cho nên trong đề tài khó tránh khỏi những thiếu sót, em rất mong nhận đuợc đuợc sự đóng góp ý kiến quý báu của các thầy cô giáo cũng như các bạn để đề tài của em được hoàn thiện hơn.

Em xin chân thành cảm ơn !

Hải Phòng, tháng 7 năm 2012

Sinh viên thực hiện

Nguyễn Thị Hằng

LỜI MỞ ĐẦU

Thế kỷ 21 được mệnh danh là thế kỷ của công nghệ thông tin, với sự bùng nổ mạnh mẽ về khoa học công nghệ. Đây là kỷ nguyên của nền văn minh dựa trên cơ sở công nghiệp trí tuệ. Ngày nay, tin học đã trở thành một môn khoa học quan trọng trên thế giới.

Sự phát triển mạnh mẽ như vậy thì công việc lập trình các ứng dụng nhằm phục vụ nhu cầu, lợi ích của con người trở nên cấp thiết. Máy tính đã trở thành công cụ đắc lực và không thể thiếu của con người.

Các tổ chức, công ty hay các cơ quan cần phải xây dựng luận văn – báo cáo – tiểu luận chuyên ngành xây dựng hệ thống mạng máy tính cho riêng mình để trao đổi dữ liệu giữa các bộ phận. Dữ liệu được truyền đi trên mạng phải đảm bảo : dữ liệu được chuyển tới đích nhanh chóng và chính xác. Hầu hết dữ liệu được truyền qua mạng là truyền dưới dạng file.

Trong thời gian gần đây Multicast ngày càng được chú ý nhiều hơn, chuẩn Multicast hỗ trợ hàng ngàn người có thể nhận thông tin đồng thời mà không ảnh hưởng đến băng thông chung. Nếu các thiết bị mạng hỗ trợ Multicast, chỉ cần gửi một gói tin duy nhất vào mạng cho nhiều người nhận.

Nhằm tìm hiểu thấu đáo một trong số các phương pháp truyền file em chọn đề tài "**Tìm hiểu kỹ thuật truyền file Multicast**.". Giải pháp truyền file Multicast –UFTP là giao thức truyền file dựa trên UDP được mã hóa theo cơ chế Multicast, được thiết kế an toàn, đáng tin cậy trong việc truyền file đến nhiều người nhận trong cùng một lúc. Với lập trình mã hóa Multicast dựa trên TLS với phần mở rộng cho phép nhiều người nhận có thể chia sẻ một khóa chung

Mục tiêu của đồ án là tìm hiểu về Multicast và giao thức UFTP, trên cơ sở đó xây dựng ứng dụng truyền file sử dụng mã nguồn mở UFTP với ngôn ngữ lập trình C#. Đồ án trình bày gồm các chương :

Chương 1: Tổng quan về Multicast và FTP.

Chương 2: Giải pháp truyền file Multicast.

Chương 3 : Xây dựng chương trình thực nghiệm.

DANH SÁCH CÁC TỪ VIẾT TẮT

FTP	File Transfer Protocol	Phương thức truyền file
MAC	Medium Access Control Address	Địa chỉ thiết bị mạng
UDP	User Datagram Protocol	Giao thức truyền vận
TCP	Transmission Control Protocol	Giao thức truyền vận
PIM	Protocol Independent Multicast	Giao thức định tuyến
CGMP	Cisco Group Membership Protocol	Giao thức chuẩn của Cisco
ARP	Address Resolution Protocol	Giao thức tìm địa chỉ
OSPF	Open Shortest Path First	Giao thức định tuyến chuẩn Internet. Giao thức định tuyến dạng link-state
NTP	Network Time Protocol	Giao thức đồng bộ thời gian
IGMP	Internet Group Management Protocol	Giao thức quản lý nhóm Internet
IGMP snooping	Internet Group Membership Protocol Snooping	Giao thức hoạt động trên switch để biết (học) về các cuộc truyền Multicast động.
TTL	Time To Live	Thời gian sống
RSA		Thuật toán mật mã hóa khóa công khai

CHƯƠNG 1: TỔNG QUAN VỀ MULTICAST VÀ FTP

1.1. Tổng quan về Multicast

1.1.1. Cơ chế truyền Multicast

Unicast: Các gói tin được gửi từ một địa chỉ nguồn đến một địa chỉ đích. Một Router hoặc một thiết bị lớp 3 sẽ chuyển các gói tin bằng cách tìm địa chỉ đích trong bảng định tuyến. Nếu một thiết bị là L2, nó chỉ cần dựa và địa chỉ MAC.

Broadcast: Các gói tin được gửi từ một máy nguồn đến một địa chỉ đích broadcast. Địa chỉ đích có thể là địa chỉ tất cả các host (255.255.255.255) hoặc là một phần của địa chỉ subnet. Một Router hoặc một L3 switch sẽ không cho phép chuyển các dữ liệu Broadcast này. Một thiết bị L2 sẽ cho phép phát tán lưu lượng Broadcast ra tất cả các cổng của nó.

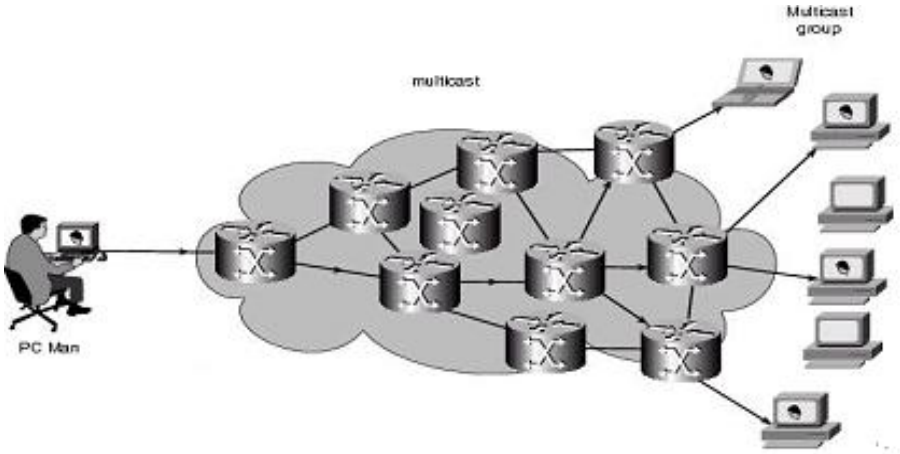
Multicast: Các gói được gửi từ một địa chỉ nguồn đến một nhóm các máy tính. Địa chỉ đích tượng trưng bằng các host muốn nhận traffic này. Mặc định, một Router hoặc một L3 switch sẽ không chuyển các gói tin này trừ khi phải cấu hình Multicast routing. Một thiết bị L2 switch không thể nhận biết được vị trí của địa chỉ Multicast đích. Tất cả các gói sẽ được phát tán ra tất cả các cổng ở chế độ mặc định.

Có hai thái cực được mô tả ở đây. Cơ chế dùng Unicast thì dữ liệu sẽ đi từ host đến host, broadcast thì traffic sẽ đi đến tất cả các host trên phân đoạn mạng đó. Cơ chế Multicast sẽ nằm giữa hai thái cực này, trong đó máy nguồn chỉ gửi những gói tin từ một host đến các người dùng muốn nhận loại lưu lượng đó. Nhóm này gọi là nhóm Multicast. Các máy nhận lưu lượng Multicast có thể nằm ở bất cứ nơi nào chứ không chỉ trên phân đoạn mạng cục bộ.

Các traffic dạng Multicast thường là một chiều. Do có nhiều host nhận cùng một dữ liệu, nên thông thường các gói tin không được phép gửi ngược về máy nguồn trên cơ chế Multicast. Một host đích sẽ trả traffic ngược về nguồn theo cơ chế Unicast. Cơ chế Multicast cũng sẽ được truyền theo kiểu phi kết nối. Multicast dùng UDP chứ không dùng TCP.

Các host muốn nhận dữ liệu từ một nguồn Multicast có thể tham gia hoặc rời khỏi một nhóm Multicast ở bất kỳ thời điểm nào. Hơn nữa, một host sẽ quyết định có trở thành

thành viên của một hay nhiều nhóm Multicast hay không. Nguyên tắc cần quan tâm là sẽ hoạch định làm thế nào để phân phối các lưu lượng Multicast đến các thành viên của nhóm mà không ảnh hưởng đến các thành viên ngoài nhóm.



Hình 1: Multicast Transmission Sends a Single Multicast Packet Addressed to All Intended Recipients

1.1.2. Điều kiện cần có để truyền Multicast

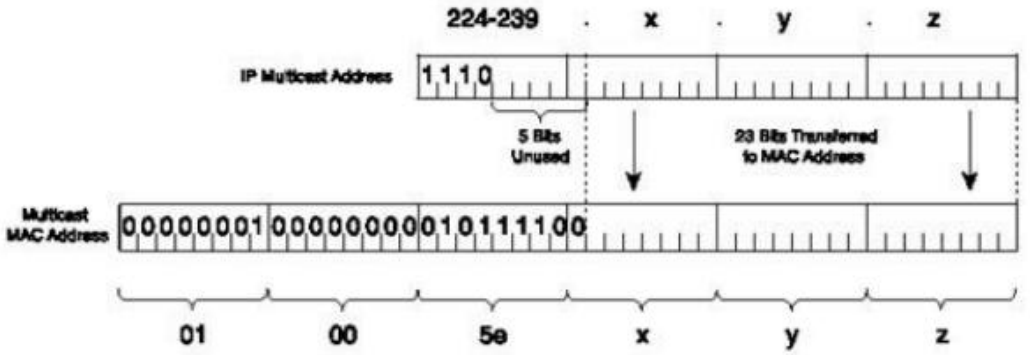
Có ba yêu cầu cơ bản để có thể triển khai Multicast trên một mạng:

- Phải có một tập hợp các địa chỉ dành cho các nhóm Multicast.
- Phải có một cơ chế trong đó các host có thể tham gia và rời khỏi nhóm.
- Phải có một giao thức định tuyến cho phép các Router phân phối các lưu lượng Multicast tới các thành viên của nhóm mà không làm quá tải tài nguyên mạng.

1.1.3. Địa chỉ Multicast

Các Router và Switch phải có phương thức để phân biệt traffic dạng Multicast với dạng Unicast hay Broadcast. Điều này thực hiện thông qua việc gán địa chỉ IP, bằng cách dùng địa chỉ lớp D từ 224.0.0.0 đến 239.255.255.255 chỉ cho Multicast. Các thiết bị mạng có thể nhanh chóng lọc ra các địa chỉ Multicast bằng cách đọc 4 bit bên trái của một địa chỉ. Bốn bit này của một địa chỉ Multicast luôn luôn bằng 1110. Không giống như dãy địa chỉ lớp A, B, và C, địa chỉ lớp D này không có quá trình Subnetting. Vì vậy có đến 2^{28} địa chỉ nhóm Multicast được trích dẫn ra từ lớp D này.

Làm thế nào mà một Router và Switch kết hợp một địa chỉ Multicast của IP với một địa chỉ MAC. Do không có cơ chế tương đương với cơ chế ARP, một dạng giá trị đặc biệt dành riêng cho địa chỉ MAC của Multicast sẽ được dùng. Các địa chỉ này bắt đầu bằng 0100.5e. Phần 28 bit sau của địa chỉ IP Multicast sẽ được ánh xạ vào 23 bit thấp của địa chỉ MAC bằng một giải thuật đơn giản.



Hình 2: Chuyển đổi IP sang MAC

Hình trên cho thấy cơ chế ánh xạ địa chỉ. Chỉ có 23 bit cuối của địa chỉ là được chép từ địa chỉ IP sang địa chỉ MAC.

Tuy nhiên chú ý rằng có 5 bit của địa chỉ IP không được chuyển sang địa chỉ MAC. Khả năng này làm cho nảy sinh một vấn đề là có thể có 32 địa chỉ Multicast khác nhau có thể ánh xạ vào cùng một địa chỉ MAC. Do sự nhập nhằng này, một host Multicast có một vấn đề nhỏ khi nó nhận một Ethernet frame của một địa chỉ Multicast. Một MAC có thể tương ứng với 32 địa chỉ Multicast khác nhau. Vì vậy, khi một host phải nhận và kiểm tra tất cả các frame có MAC mà nó quan tâm. Sau đó host này phải kiểm tra phần địa chỉ IP bên trong mỗi frame để nhận ra phần địa chỉ của từng nhóm Multicast.

Một vài không gian địa chỉ được dành riêng:

Toàn bộ không gian địa chỉ Multicast:224.0.0.0 đến 239.255.255.255

- Địa chỉ link-local: 224.0.0.0 - 224.0.0.255 được dùng bởi các giao thức định tuyến. Router sẽ không chuyển các gói tin có địa chỉ này.

- Các địa chỉ bao gồm địa chỉ tất cả các host all-hosts 224.0.0.1

- Tất cả các Router 224.0.0.2.

- Tất cả các OSPF Routers 224.0.0.5…224.0.1.1 dùng cho giao thức NTP. Đây là địa chỉ các nhóm cố định vì các địa chỉ này được định nghĩa trước.

- Địa chỉ GLOP trong tầm 233.0.0.0 - 233.255.255.255.

- Tầm địa chỉ dành cho quản trị (239.0.0.0 - 239.255.255.255) được dùng trong các vùng Multicast riêng, giống như dãy địa chỉ dành riêng trong RFC1918. Địa chỉ này không được Router giữa các domain nên nó có thể được dùng lại nhiều lần.

- Địa chỉ toàn cục (224.0.1.0-238.255.255.255) được dùng bởi bất cứ đối tượng nào. Các địa chỉ này có thể được định tuyến trên Internet, vì vậy địa chỉ này phải duy nhất.

1.1.4. IGMP

Làm thế nào một Router biết được các máy cần nghe lưu lượng Multicast? Để nhận lưu lượng Multicast từ một nguồn, cả nguồn và các máy nhận đầu tiên phải gia nhập vào một nhóm Multicast. Nhóm này được xác định thông qua địa chỉ Multicast.

Một host có thể tham gia vào một nhóm Multicast bằng cách gửi các yêu cầu đến Router gần nhất. Tác vụ này được thực hiện thông qua giao thức IGMP. IGMPv1 được định nghĩa trong RFC1112 và bản cải tiến của nó, IGMPv2 được định nghĩa trong RFC2236.

Khi có vài host muốn tham gia vào nhóm, giao thức PIM sẽ thông báo cho nhau giữa các Router và hình thành nên cây Multicast giữa các Routers. IGMP và ICMP có nhiều điểm tương đồng, cùng chia sẻ một vài chức năng tương tự. IGMP cũng đóng gói trong gói tin IP (protocol number 2), nhưng IGMP giới hạn chỉ trong một kết nối lớp 2.

Để đảm bảo Router không bao giờ tiếp tục forward gói tin, trường TTL của IGMP luôn có giá trị bằng 1.

a. IGMPv1

Để tham gia vào một nhóm Multicast, một host sẽ gửi một thông điệp đăng ký tham gia vào nhóm đến Router cục bộ của nó. Thông điệp này có tên là Membership

Report IGMP. Thông điệp này sẽ thông báo cho Router về địa chỉ nhóm Multicast mà host muốn tham gia vào. Địa chỉ Multicast 224.0.0.1 all-hosts được dùng như địa chỉ đích. Trong thông điệp này có chứa địa chỉ nhóm Multicast.

Cứ mỗi 60s, một Router trên mỗi phân đoạn mạng sẽ gửi truy vấn đến tất cả các host để kiểm tra xem các host này có còn quan tâm nhận lưu lượng Multicast nữa không? Router này gọi là IGMPv1 Querier và chức năng của nó là mời các host tham gia vào nhóm. Nếu một host muốn tham gia vào một nhóm, hoặc nó muốn tiếp tục nhận lưu lượng từ một nhóm mà nó đã tham gia, nó phải trả lời lại bằng thông điệp Membership-report.

Các host có thể tham gia vào các nhóm Multicast ở bất kỳ thời điểm nào. Tuy nhiên IGMPv1 không có cơ chế để cho phép một host rời khỏi một nhóm nếu host đó không còn quan tâm đến nội dung của nhóm Multicast đó. Thay vào đó, Router sẽ kết luận là một cổng giao tiếp của nó không còn thuộc về một nhóm Multicast nào nếu Router không nhận được Membership-report trong ba chu kỳ truy vấn liên tiếp. Điều này có nghĩa là, ở chế độ mặc định, các lưu lượng Multicast vẫn gửi vào một phân đoạn mạng trong ba chu kỳ truy vấn liên tiếp sau khi tất cả các thành viên của nhóm không còn lắng nghe lưu lượng Multicast nữa.

Ngoài ra, Router không có giữ một danh sách đầy đủ các host thành viên cho từng nhóm Multicast. Thay vào đó, nó cần phải lưu những nhóm Multicast nào là đang tồn tại trên những cổng nào của nó.

b. IGMPv2

IGMPv2 dùng ba loại thông điệp:

- Membership report.
- V1 membership report message
- Leave group message.

Membership report sẽ được gửi khi một host muốn tham gia một nhóm. Thỉnh thoảng, thông điệp loại này cũng được dùng để trả lời cho loại thông điệp truy vấn query từ Router. Khi một host muốn tham gia một nhóm, nó sẽ không chờ gói tin Query từ

Router. Thay vào đó, nó sẽ gửi Membership Report. Địa chỉ đích của Membership Report sẽ là địa chỉ đích của nhóm. Để đảm bảo rằng Router nhận được thông điệp này, host sẽ gửi vài message, cách nhau mỗi 10s.

Phiên bản IGMPv2 giới thiệu vài sự khác biệt so với phiên bản đầu tiên. Các gói tin truy vấn bây giờ được gọi là General Queries. Các gói này có thể gửi tới địa chỉ all-hosts hoặc tới từng nhóm cụ thể. Một cải tiến khác nữa là các host được phép rời khỏi nhóm.

Khi một host quyết định rời khỏi một nhóm nó đã tham gia, nó sẽ gửi thông điệp LeaveGroup đến địa chỉ all-Router 224.0.0.2. Tất cả các Router trên một phân đoạn mạng nội bộ sẽ lưu ý thông điệp này và Router truy vấn sẽ tiếp tục quá trình. Router sẽ trả lời thông điệp trên bằng thông điệp truy cập gửi theo nhóm. Thông điệp này sẽ hỏi rằng có còn host nào muốn nhận traffic cho nhóm đó nữa không? Bất cứ host nào cũng phải trả lời lại bằng thông điệp membership report. Nếu khác đi, Router sẽ kết luận một cách an toàn là không cần thiết chuyển traffic cho nhóm đó trên phân đoạn mạng đó. Khoảng thời gian này mặc định là 3 phút.

Nếu có nhiều Router trên cùng một kết nối, Router có IP address nhỏ nhất sẽ gửi ra gói tin query. Vì vậy, khi một Router nhận được một gói tin query từ một Router nào đó, nó sẽ kiểm tra địa chỉ nguồn của gói tin đó. Nếu địa chỉ nguồn của Router cục bộ nhỏ hơn địa chỉ nguồn trong gói tin vừa đến, Router sẽ vẫn tiếp tục gửi gói tin query vì nó biết rằng nó sẽ giữ vai trò query. Còn nếu địa chỉ nguồn của gói tin query là nhỏ hơn, Router sẽ từ bỏ vai trò truy vấn.

c. Switching Multicast Traffic

Các Router hoặc các MLS switch có thể xây dựng các cây Multicast và chuyển các gói tin đi một cách hiệu quả. Tuy nhiên ở lớp 2, một switch chỉ kiểm tra phần header của frame Ethernet để tìm địa chỉ nguồn và địa chỉ đích. Các switch này không thể hoạt động 'theo yêu cầu' giống như Router.

Thông tin tốt nhất mà một switch biết được là địa chỉ Multicast đích và khi đó frame đó cần phải được phát tán ra tất cả các cổng của VLAN. Có hai phương thức được

phát triển để cho phép các switch chuyển các gói tin Multicast một cách thông minh: dùng IGMP snooping và dùng CGMP. Một phương thức đòi hỏi phần cứng mạnh, phương thức kia thì học hỏi thông tin từ các Router láng giềng.

d. IGMP Snooping

Trong chế độ hoạt động bình thường, một host muốn tham gia vào một nhóm Multicast phải liên lạc với một Router gateway để Router đưa host đó vào nhóm Multicast. IGMP snooping cho phép một switch lắng nghe các thông điệp IGMP membership report này sao cho nó có thể tìm ra host nào đang yêu cầu nhóm nào. Để tham gia vào một nhóm, một host phải gửi các thông điệp Multicast membership report về chính địa chỉ Multicast của nhóm đó.

Một switch L2 phải lắng nghe đến tất cả các Multicast frame để tìm ra thông tin IGMP. Đây rõ ràng là một gánh nặng cho CPU của switch. Một thiết bị L3 switch thì có lợi thế rõ ràng hơn, nó có thể tách ra thông tin L3 trong một frame. Kiểu switch này phải lắng nghe mọi gói IGMP. Khi một thông điệp membership report được lắng nghe, switch sẽ thêm địa chỉ MAC của nhóm Multicast vào bảng CAM của nó cùng với port nguồn nơi mà một gói IGMP được nhận.

Tác vụ này sẽ liên kết địa chỉ nhóm với các host đã yêu cầu tham gia nhóm. Khi các host khác cũng yêu cầu tham gia nhóm, các switch port tương ứng sẽ được thêm vào bảng CAM. Khi có một frame cần đến một địa chỉ Multicast, nó có thể được nhân bản ra chính xác các cổng của các host nhận.

Với IGMP snooping, có hai trường hợp đặc biệt của thành viên nhóm trong bảng CAM:

- Tất cả các địa chỉ IGMP là nhận biết bởi switch (học động) cũng sẽ được lưu trong bảng CAM. Các frame Multicast phải được nhân bản về phía các Router để các traffic này có thể được Router nếu cần thiết.
- CPU của switch cũng là một thành viên của nhóm Multicast vì nó có thể xem các gói IGMP đến và đi. Chỉ có lưu lượng của IGMP là được xử lý. CPU sẽ không kiểm tra các frame Multicast khác. IGMP snooping được cho

phép trên tất cả các cổng của switch và các interface VLAN. Các switch 2950, 3550, 4500 và 6500 là có hỗ trợ IGMP Snooping.

e. CGMP

Khi một L2 switch không thể thực hiện tác vụ IGMP snooping, một Router Multicast láng giềng sẽ trợ giúp. Cisco phát triển ra một giao thức là Cisco Group Membership Protocol (CGMP) nhằm mục đích này.

Một Router hoặc một multilayer switch được cấu hình cho định tuyến Multicast có thể được cấu hình cho CGMP. Khi các host gửi các thông điệp membership report để tham gia và rời khỏi một nhóm, Router CGMP sẽ trung chuyển các thông điệp này đến các switch quan tâm. Các thông điệp CGMP dùng các địa chỉ Multicast nổi tiếng là *0100.0cdd.dddd*.

Theo định nghĩa, địa chỉ nhóm Multicast này sẽ được phát tán như là một trường hợp đặc biệt sao cho các thông điệp CGMP có thể được truyền trên các non-CGMP switch. Các thông điệp CGMP bao gồm địa chỉ MAC của host cùng với địa chỉ MAC của nhóm Multicast nó muốn tham gia hay rời bỏ.

Khi một layer 2 switch nhận thông điệp CGMP này, tác vụ đơn giản của nó là thêm vào địa chỉ nhóm Multicast và các host của nhóm đó vào bảng CAM. Router sẽ trở thành thiết bị "trợ thính" cho một switch trong việc lắng nghe các thông điệp CGMP.

Chỉ có Router Multicast phải được cấu hình cho CGMP. tất cả các IOS-based L2 switch đều có CGMP cho phép ở chế độ mặc định, vì vậy switch sẽ tự động xử lý CGMP từ Routers.

Mặc dù là cả Router và switch đều cấu hình để chạy CGMP, thật sự chỉ có Router là tạo ra các gói tin CGMP. Có hai loại gói tin CGMP

- Gói tin join được gửi ra bởi Router, ra lệnh cho switch thêm vào các member mới.
- Gói tin leave được dùng để báo cho switch xóa bớt thành viên hoặc xóa hẳn một nhóm.

1.1.5. Định tuyến Multicast

Chức năng của Unicast – routing là tìm đường đi ngắn nhất đến một địa chỉ đích nào đó. Tác vụ này được thực hiện bằng cách dùng các giao thức distance vector hay dùng các cơ sở dữ liệu liên kết (linkstate). Kết quả của tác vụ này là một hàng trong bảng định tuyến, chỉ ra cổng outbond hoặc chỉ ra Router kế tiếp. Interface đầu ra có thể hiểu như là Interface gần nhất để đi về mạng đích.

Trái lại, chức năng của Multicast routing là tìm ra upstream interface, tức là đường đi ngắn nhất về source. Bởi vì các Multicast routing protocol quan tâm tìm ra đường đi ngắn nhất về source hơn là đường đi ngắn nhất về đích nên quá trình forward gói tin Multicast được gọi là Reverse path forwarding.

Cách ngắn nhất để một giao thức định tuyến Multicast xác định đường đi ngắn nhất về source là tham khảo bảng định tuyến Unicast. Tuy nhiên, các gói Multicast sẽ được forward đi dựa trên thông tin của bảng Multicast Routing riêng biệt. Lý do cho việc này là Router không chỉ lưu lại cổng Upstream mà còn lưu lại cổng Downstream của cả nhóm Multicast.

Các traffic IP phải được định tuyến giống như bất cứ một gói tin L3 nào. Sự khác nhau là ở điểm cần phải biết để chuyển gói tin về đâu. Các gói tin L3 dạng Unicast chỉ có một cổng ra duy nhất trên Router(ngay cả khi có quá trình load-balancing xảy ra), trong khi lưu lượng Multicast có thể được chuyển mạch ra nhiều cổng, tùy thuộc vào các máy nhận nằm ở đâu.

Một vài giao thức định tuyến Multicast hiện có: PIM, DVMRP, MOSPF.

1.1.5.1. Cây Multicast

Các Router hoặc các multilayer switch trong một mạng phải xác định một tuyến đường để phân phối các gói tin Multicast từ máy nguồn đến từng máy nhận. Khi đó, toàn bộ mạng giống như một cấu trúc cây, trong đó gốc của cây là nguồn của luồng lưu lượng đó. Mỗi Router dọc theo đường đi sẽ là một nhánh rẽ của cây.

Nếu một Router biết tất cả các địa chỉ Multicast, Router cũng phải biết cần phải nhân bản luồng Multicast đó ra những nhánh nào của cây. Một vài Router không có các

máy nhận trong các phân đoạn mạng của nó thì các Router đó sẽ không chuyển lưu lượng đó nữa. Các Router khác sẽ có thể có các máy nhận lưu lượng Multicast.

Cấu trúc cây này tương tự như cấu trúc cây Spanning Tree vì nó có một root và các lá. Cấu trúc cây này cũng đảm bảo là không bị vòng lặp sao cho lưu lượng Multicast không bị chuyển ngược về cây.

1.1.5.2. Reverse Path Forwarding

Các Router thường phải thực hiện một phép kiểm tra trên tất cả các gói Multicast mà nó nhận. *Reverse Path Forwarding (RPF)* là một công cụ để đảm bảo rằng các gói tin không bị đưa ngược trở về cây Multicast ở một vị trí bất kỳ nào đó. Khi một gói tin Multicast được nhận trên một cổng của Router, ví dụ cổng E0 của Router, địa chỉ nguồn của gói sẽ được kiểm tra.

Sau đó Router sẽ so sánh địa chỉ nguồn này với một entry trong bảng định tuyến unicast. Nếu cột out-going interface của bảng định tuyến cũng đúng bằng cổng nhận gói Multicast (tức E0 trong ví dụ này), gói Multicast sẽ được xử lý và chuyển ra các nhánh của cây. Nếu cổng là không so trùng, điều này có nghĩa là có một ai đó đã đưa gói vào một vị trí không mong đợi, chuyển gói tin ngược về root. Gói tin lúc này sẽ bị loại bỏ.

Để thực hiện phép kiểm tra RPF này, Router chạy giao thức PIM phải tìm kiếm địa chỉ nguồn trong bảng định tuyến unicast.

1.1.5.3. Giao thức định tuyến PIM

Protocol Independent Multicast (PIM) là một giao thức định tuyến có thể được dùng để chuyển các lưu lượng Multicast. PIM hoạt động độc lập với các giao thức định tuyến Unicast IP vì vậy PIM sử dụng bảng định tuyến IP. Cần chú ý là bảng Unicast Routing cũng không phụ thuộc vào các giao thức định tuyến vì nhiều giao thức định tuyến có thể đóng góp vào cùng một bảng định tuyến. PIM có thể hoạt động ở hai chế độ:

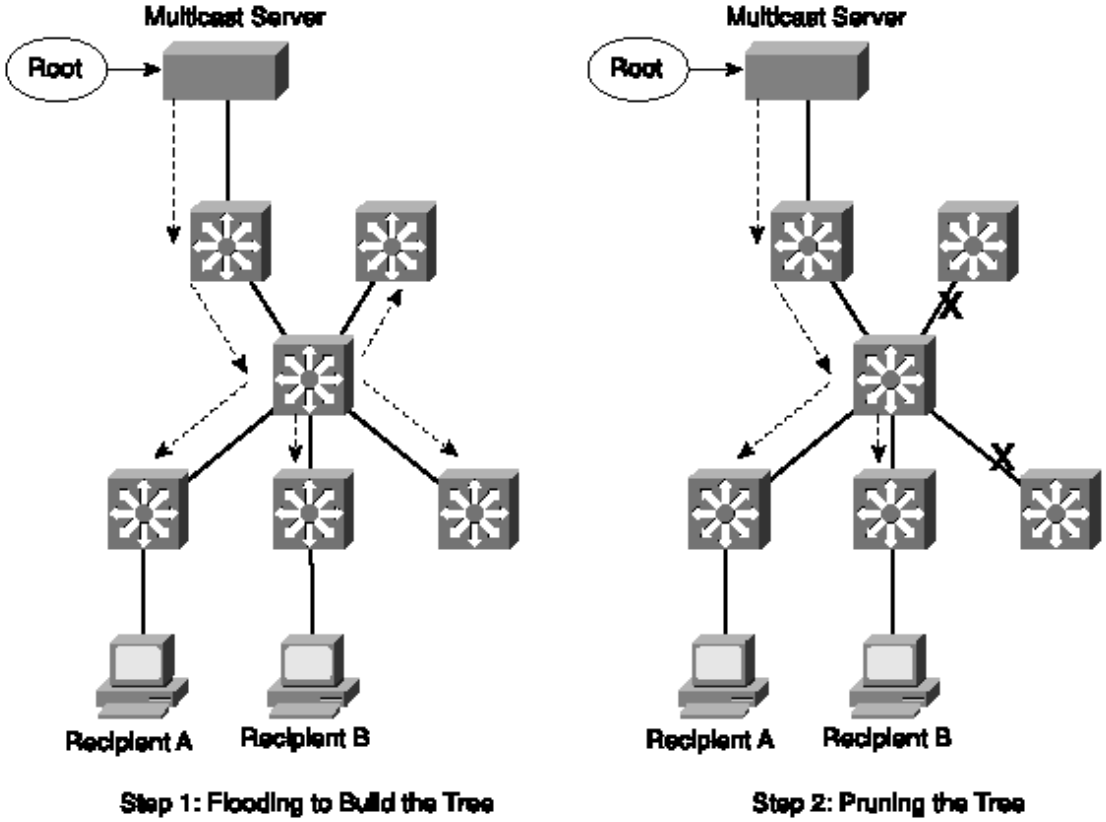
- PIM Dense Mode
- PIM Sparse Mode
- PIM Sparse Dense Mode (do Cisco đưa ra)

a. PIM Dense Mode

Các PIM Router có thể được cấu hình theo kiểu Dense Mode (còn gọi là PIM-DM) nếu các host tham gia vào Multicast group nằm ở khắp nơi trên tất cả các Subnet. Một sơ đồ mạng được xem là Dense nếu có rất nhiều nhóm Multicast so với số host tương đối.
Dense mode thường được dùng trong môi trường LAN và Campus.
Sparse mode thường được dùng trong môi trường WAN.

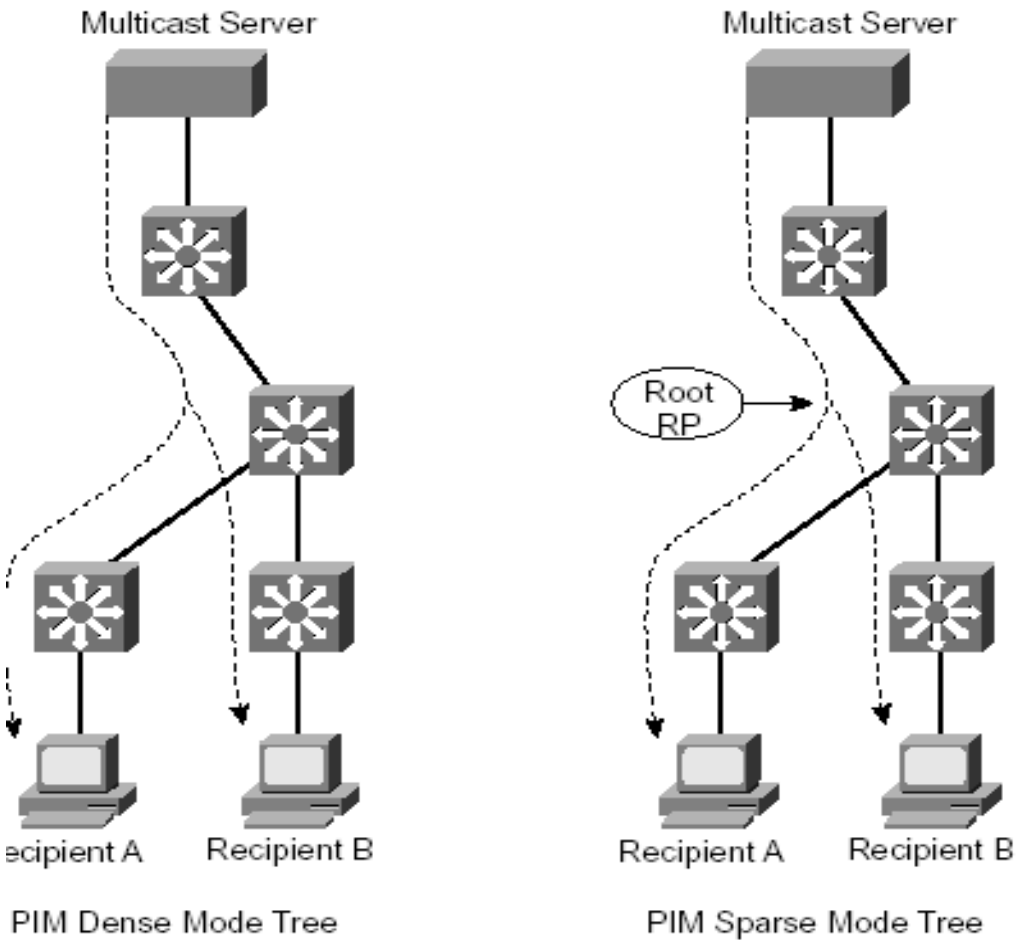
Địa chỉ Multicast nguồn trở thành gốc của cây và cây Multicast được xây dựng từ nguồn đến đích. Cơ chế này còn được gọi bằng ký hiệu (S,G) trong đó đường đi từ nguồn đến các thành viên trong nhóm là duy nhất và được xác định.

Cây Multicast được xây dựng bằng cách cho phép phát tán các traffic từ nguồn đến tất cả các Router trong mạng. Cây sẽ phát triển từ trên xuống dưới. Trong một thời gian ngắn, các lưu lượng không cần thiết sẽ được lưu chuyển giống như trong lưu lượng Broadcast. Tuy nhiên khi các Router nhận được traffic cho một nhóm, Router sẽ quyết định nó có các máy nhận muốn nhận dữ liệu hay không? Nếu là muốn, Router sẽ duy trì tình trạng im lặng và để dòng traffic tiếp tục. Nếu không có host nào đăng ký cho nhóm Multicast đó (thông qua IGMP), Router sẽ gửi thông điệp Prune đến các Router láng giềng của nó (theo hướng về gốc của cây. Nhánh của cây sau đó sẽ bị loại bỏ (prune) sao cho các traffic không cần thiết sẽ không được phát tán về hướng đó.



Hình 3 : Cây Multicast theo kiểu Dense-Mode

Cây Multicast sẽ được xây dựng theo một làn sóng của các yêu cầu tham gia vào nhóm. Sau đó các switch không có các host tham gia sẽ bị xóa ra khỏi cây. Cây kết quả sau cùng được hiển thị ở hình. kế tiếp.



Hình 4: Cây Multicast sau khi loại bỏ các switch không có host tham gia

PIM-DM sẽ nhận biết các thiết bị láng giềng bằng cách trao đổi các gói hello. Thông tin láng giềng này được dùng trước để xây dựng cây đến tất cả các láng giềng. Sau đó, các nhánh của cây sẽ lần lượt được loại bỏ. Nếu một dòng Multicast bắt đầu, cây sẽ được xây dựng, cây sẽ chỉ tồn tại khi các thành viên tích cực còn tồn tại. Nếu một host mới đăng ký tham gia nhóm, nhánh của phân đoạn mạng đó sẽ được đính thêm vào cây.

b. PIM Sparse Mode

Có vài điểm giống nhau giữa PIM-SM và PIM-DM:

- Cả hai đều dùng cơ chế hello để tìm ra láng giềng.
- Tính toán và kiểm tra RPF khi bảng định tuyến unicast routing thay đổi.

Bầu chọn designated Router trên môi trường Multiaccess.

- Sử dụng cơ chế prune trên môi trường Multiaccess.

Tuy nhiên, PIM-SM dùng cơ chế explict join (join tường minh).

PIM Sparse Mode (PIM-SM) dùng một giải pháp khác. Cây Multicast không mở rộng đến Router cho đến khi nào một host đã tham gia vào một nhóm. Cây Multicast được xây dựng bằng các thành viên ở các Node lá và mở rộng ngược về Root. Cây được xây dựng từ dưới lên. SM cũng hoạt động dựa trên ý tưởng cấu trúc shared-tree, trong đó gốc của cây không nhất thiết là nguồn của Multicast.

Thay vào đó, root là Router PIM-SM thường được đặt ở trung tâm của mạng. Router làm gốc này gọi là *Rendezvous Point (RP: điểm hẹn hò)*. Các Router có thể nhận biết được RP bằng 3 cách:

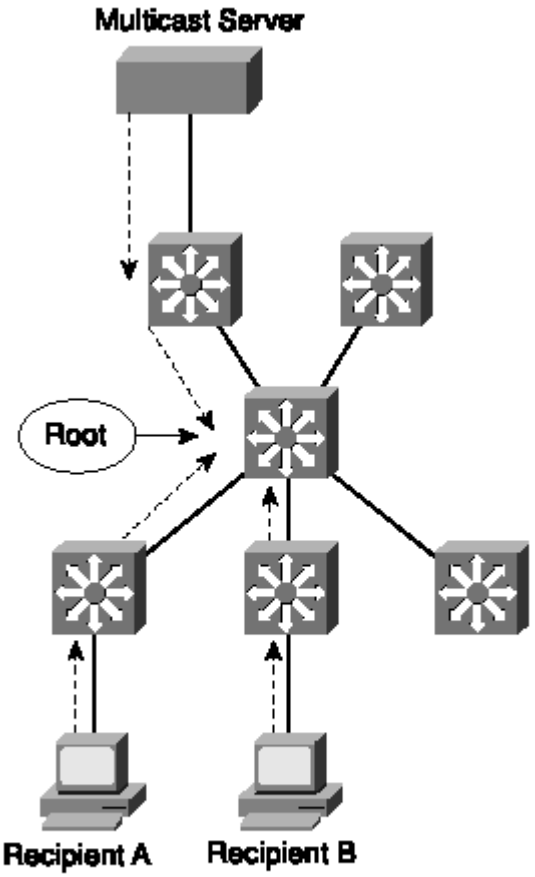
- Địa chỉ RP có thể được cấu hình tĩnh trên từng Router: kiểm soát được mạng nhưng chi phí quản trị cao.

- Bầu chọn RP có thể dùng giao thức Bootstrap.

- Dùng cơ chế Auto-RP của Cisco: phù hợp hệ thống mạng nhỏ.

Cây từ điểm RP đến các thành viên thật ra là một cây con của cây từ nguồn đến các thành viên. Nếu một Router ở bất kỳ đâu trong mạng có thể đăng ký với RP, cấu trúc cây này sẽ hoàn tất. Chế độ spare-mode còn được gọi là Shared tree. Các dòng Multicast được mô tả như (*,G) bởi vì cây luôn cho phép bất cứ nguồn nào gửi đến một nhóm. Ký hiệu (*,G) có ý nghĩa là bất kỳ source nào cũng có thể gửi về nhóm G.

Khi một host tham gia vào một nhóm Multicast dùng IGMP, Router cục bộ sẽ chuyển các thông điệp Membership report về gốc của cây Multicast. Mỗi Router dọc theo đường đi sẽ thêm nhánh đó vào cây dùng chung shared-tree. Quá trình loại bỏ nhanh chỉ thực hiện khi một thành viên của nhóm bị xóa ra khỏi một nhóm. Quá trình này được hiển thị ở hình dưới đây:



Hình 5: Cây Multicast khi có host tham gia

Chú ý là quá trình này chỉ bao gồm 1 bước. Các Router không tham gia vào nhóm sẽ không bị loại bỏ vì nó không bao giờ là một thành phần của cây.

Khi shared tree đã được thiết lập, định kỳ Router sẽ gửi các thông điệp join/prune đến các upstream Routers như cơ chế keepalive.

c. PIM Sparse-Dense Mode

PIM có khả năng hỗ trợ cả hai chế độ Dense và Sparse Mode bởi vì cả hai tồn tại trên những nhóm Multicast khác nhau trên một mạng. Cisco cho phép chế độ lai sparse-dense mode cho phép một PIM Router dùng chế độ dense hay chế độ Sparse tùy thuộc vào từng nhóm. Nếu một nhóm có RP được định nghĩa, Sparse-mode sẽ được dùng, nếu không có, Dense-mode sẽ được dùng

d. PIM Version 1

Ta có thể giới hạn phạm vi các nhóm Multicast được hỗ trợ bởi RP bằng cách dùng một access-list. Từ khóa Overrise cho phép RP được ưu tiên hơn bất cứ một RP nào được bầu chọn bởi quá trình tự động. RP phải được định nghĩa trên tất cả các vùng mạng chạy PIM, kể cả trên Router RP. Cisco cũng cung cấp một phương thức để tự động thông báo về PIM-SM cho một nhóm. Phương thức này gọi là Auto-RP. Quá trình này được thực hiện bằng cách chỉ ra một Router nằm ở trung tâm và các Router kết nối vào nó gọi là mapping agent. Các mapping agent sẽ học thông tin của tất cả các ứng cử viên cho RP. Các Router muốn làm ứng cử viên phải gửi ra một thông điệp Cisco-RP-Announce về địa chỉ Multicast 224.0.1.39.

Router mapping agent sẽ gửi các thông tin ánh xạ từ RP-nhóm đến tất cả các PIM Router khác dùng thông điệp Cisco-RP-Discovery về địa chỉ 224.0.1.40. Giá trị Time-to-live được thiết lập trong những thông điệp này hạn chế tầm vực của thông điệp này. Thông số này sẽ chỉ ra là thông tin sẽ còn hợp lệ trong bao nhiêu lâu nữa. Người quản trị phải định nghĩa tường minh các ứng cử viên cho RP. Khi một Router được xem như là RP, nó sẽ bắt đầu gửi các thông điệp đến Router mapping agent.

Các cổng giao tiếp sẽ quảng bá địa chỉ Router RP. Phạm vi của thông điệp quảng bá này sẽ bị giới hạn bởi thông số TTL. Router cũng sẽ quảng bá chính nó như là một candidate RP cho nhóm được định nghĩa trong access-list.

1.1.6. Ứng dụng của Multicast

Hiện nay, trong môi trường kinh doanh có một lượng rất lớn các thông tin cần phải chuyển tiếp đến nhiều nơi trong cùng một thời gian. Cùng thời điểm đó, các doanh nhân và các nhà nghiên cứu cần lấy một lượng lớn thông tin và thống kê trong cùng một ngày. Mạng hiện nay được sử dụng và phát triển để đáp ứng nhu cầu này, với sự phát triển đó thì các dịch vụ mới lại được thêm vào để phục vụ tốt hơn cho nhu cầu sử dụng.

Các ứng dụng hiện nay trên mạng Internet đều dựa vào giao thức truyền theo điểm, giống như là các ứng dụng trong mạng LAN truyền thống. IP Multicast giúp cho mạng tiết kiệm được băng thông. Multicast là một sự thay thế tốt cho quá trình truyền Unicast

khi mà các công ty cần chuyển thông tin đến nhiều khách hàng trong cùng một thời điểm. Sử dụng Multicast có thể giảm tải cho mạng.

Ví dụ: một ứng dụng về tài chính cần phải gửi thông tin báo cáo cho hàng trăm máy trong mạng của một công ty. Mỗi máy nếu sử dụng theo giao thức Unicast thì thông tin cần phải nhân bản lên hàng trăm lần và di chuyển trên cùng một liên kết mạng. Các gói tin chỉ được nhân bản khi gặp một nhánh rẽ trên cây chuyển tiếp.

Ứng dụng:

- Phân phối thông tin: Multicast làm cho việc phân phối thông tin trong các phòng ban trở nên dễ dàng. Ví dụ: một công ty có một số thay đổi về chính sách giá cả thì thông tin này sẽ được truyền tới toàn bộ các đại lý trong cùng một lúc, hoặc các công ty về IT sử dụng Multicast để đưa thông tin cập nhật về các phần mềm mới của mình tới khách hàng.

- Hội thảo truyền hình: Thông thường người ta hay sử dụng các đường ISDN đắt tiền phục vụ cho việc hội thảo truyền hình hoặc sử dụng dịch vụ do các công ty viễn thông cung cấp. Hội thảo tương tác qua Internet, Intranet hoặc extranet sử dụng Multicast thì có giá trị kinh tế hơn nhiều, và cho phép số người tham gia cùng một lúc không giới hạn.

- Phát hiện dịch vụ: Các giao thức phát hiện dịch vụ trên mạng sử dụng Multicast thay vì broadcast thì có hiệu quả cao hơn, điều đó làm cho dịch vụ không còn bị giới hạn trong phạm vi của một subnet mà có thể mở rộng trên toàn mạng TCP/IP mà không cần phải cung cấp các thông tin về cấu hình.

1.2. Giao thức truyền file FTP

1.2.1. Tổng quan FTP

FTP(File Transfer Protocol) là giao thức truyền tập tin. Thường được dùng để trao đổi tập tin qua mạng lưới truyền thông dùng giao thức TCP/IP. FTP sử dụng giao thức TCP có độ tin cậy cao, đảm bảo tính toàn vẹn dữ liệu truyền. Sử dụng cơ chế truyền Unicast. Và là một giao thức hướng kết nối.

Hoạt động của FTP cần có hai máy tính, một máy chủ và một máy khách. Máy chủ

FTP, dùng chạy phần mềm cung cấp dịch vụ FTP, gọi là trình chủ, lắng nghe yêu cầu về dịch vụ của các máy tính khác trên mạng lưới.

Máy khách chạy phần mềm FTP dành cho người sử dụng dịch vụ, gọi là trình khách, thì khởi đầu một liên kết với máy chủ.

Một khi hai máy đã liên kết với nhau, máy khách có thể xử lý một số thao tác về tập tin, như tải tập tin lên máy chủ, tải tập tin từ máy chủ xuống máy của mình, đổi tên của tập tin, hoặc xóa tập tin ở máy chủ...

1.2.2. Các phương thức truyền dữ liệu trong FTP

Khi kênh dữ liệu đã được thiết lập xong giữa máy chủ với máy khách, dữ liệu sẽ được truyền trực tiếp từ phía Client tới phía Server, hoặc ngược lại, dựa theo các lệnh được sử dụng. Do thông tin điều khiển được gửi đi trên kênh điều khiển, nên toàn bộ kênh dữ liệu có thể được sử dụng để truyền dữ liệu.

Tất nhiên, hai kênh logic này được kết hợp với nhau ở lớp dưới cùng với tất cả các kết nối TCP/UDP khác giữa hai thiết bị, do đó điều này không hẳn đã cải thiện tốc độ truyền dữ liệu so với khi truyền trên chỉ một kênh - nó chỉ làm cho hai việc truyền dữ liệu và điều khiển trở nên độc lập với nhau mà thôi.

FTP có ba phương thức truyền dữ liệu, nêu lên cách mà dữ liệu được truyền từ một thiết bị tới thiết bị khác trên một kênh dữ liệu đã được khởi tạo, đó là: stream mode, block mode, và compressed mode.

- Stream mode: Trong phương thức này, dữ liệu được truyền đi dưới dạng các byte không cấu trúc liên tiếp. Thiết bị gửi chỉ đơn thuần đẩy luồng dữ liệu qua kết nối TCP tới phía nhận. Không có một trường tiêu đề nhất định được sử dụng trong phương thức này làm cho nó khá khác so với nhiều giao thức gửi dữ liệu rời rạc khác. Phương thức này chủ yếu dựa vào ***tính tin cậy*** trong truyền dữ liệu của TCP. Do nó không có cấu trúc dạng header, nên việc báo hiệu kết thúc file sẽ đơn giản được thực hiện việc phía thiết bị gửi ngắt kênh kết nối dữ liệu khi đã truyền xong. Trong số ba phương thức, stream mode là phương thức được sử dụng nhiều nhất trong triển khai FTP thực tế. Có một số lý do giải thích điều đó.

- Thứ nhất, nó là phương thức mặc định và đơn giản nhất, do đó việc triển khai nó là dễ dàng nhất.
- Thứ hai, nó là phương pháp phổ biến nhất, vì nó xử lý với các file đều đơn thuần như là xử lý dòng byte, mà không để ý tới nội dung của các file.
- Thứ ba, nó là phương thức hiệu quả nhất vì nó không tốn một lượng byte "overload" để thông báo header.

- Block mode: Đây là phương thức truyền dữ liệu mang tính quy chuẩn hơn, với việc dữ liệu được chia thành nhiều khối nhỏ và được đóng gói thành các FTP blocks. Mỗi block này có một trường header 3 byte báo hiệu độ dài, và chứa thông tin về các khối dữ liệu đang được gửi. Một thuật toán đặc biệt được sử dụng để kiểm tra các dữ liệu đã được truyền đi và để phát hiện, khởi tạo lại đối với một phiên truyền dữ liệu đã bị ngắt.

- Compressed mode: Đây là phương thức truyền sử dụng một kỹ thuật nén khá đơn giản, là "run-length encoding" có tác dụng phát hiện và xử lý các đoạn lặp trong dữ liệu được truyền đi để giảm chiều dài của toàn bộ thông điệp. Thông tin khi đã được nén, sẽ được xử lý như trong block mode, với trường header. Trong thực tế, việc nén dữ liệu thường được sử dụng ở những chỗ khác, làm cho phương thức truyền kiểu Compressed mode trở nên không cần thiết nữa.

Ví dụ: nếu bạn đang truyền đi một file qua Internet với modem tương tự, modem của bạn thông thường sẽ thực hiện việc nén ở lớp 1, các file lớn trên FTP server cũng thường được nén sẵn với một số định dạng như Zip, làm cho việc nén tiếp tục khi truyền dữ liệu trở nên không cần thiết.

1.2.3. Ứng dụng của FTP

- Khuyến khích việc dùng chung tập tin(như chương trình ứng dụng vi tính hoặc dữ liệu).
- Khuyến khích việc sử dụng máy tính ở xa một cách gián tiếp/ âm thầm.
- Che đậy sự khác biệt về hệ thống lưu trữ tập tin giữa các máy chủ, hầu như cho người dùng không cần phải quan tâm đến những sự khác biệt riêng tư của chúng
- Truyền tải dữ liệu một cách đáng tin cậy và có hiệu quả cao.

1.2.4. Ưu điểm và nhược điểm của FTP.

Ưu điểm:

- FTP là cách nhanh và hiệu quả của việc vận chuyển số lượng lớn dữ liệu qua Internet.

- Nó có thể sao lưu tự động. Bất cứ khi nào bạn chỉnh sửa các tập tin của bạn trong hệ thống của bạn, bạn có thể cập nhật bằng cách sao chép nó vào hệ thống máy chủ lưu trữ trang web của bạn. Vì vậy, trong trường hợp trang web của bạn bị hỏng và tất cả các dữ liệu bị mất, bạn sẽ có một bản sao của nó trong hệ thống của bạn. Nó cũng hoạt động theo chiều khác.

- FTP cho phép bạn kiểm soát chuyển giao. Đó là, bạn có thể chọn chế độ mà trong đó dữ liệu được chuyển qua mạng. Các dữ liệu có thể được chuyển giao hoặc trong chế độ ASCII (đối với các tập tin văn bản) hoặc ở chế độ nhị phân (thực thi hoặc các tập tin nén).

- Bạn có thể làm việc với các thư mục trên hệ thống từ xa, xóa hoặc đổi tên các tập tin từ xa trong khi chuyển dữ liệu giữa 2 máy.

- Trong khi sử dụng FTP, công cụ giống như một lệnh riêng bằng ngôn ngữ lập trình cũng có thể được sử dụng để làm cho công việc của bạn hiệu quả và dễ dàng hơn.

Nhược điểm:

- Do sử dụng TCP nên chỉ có thể truyền theo cơ chế Unicast nên chỉ truyền 1-1

- Khi cần truyền 1- nhiều hoặc nhiều - nhiều sẽ cần tạo nhiều kết nối 1 - 1 dẫn đến tốn tài nguyên, thời gian truyền lâu.

- Vì vậy khi cần truyền file theo kiểu 1- nhiều hoặc nhiều – nhiều cần có một giải pháp khác.

CHƯƠNG 2 : UFTP - GIẢI PHÁP TRUYỀN FILE MULTICAST

2.1. Tổng quan UFTP

UFTP – Encrypted UDP based FTP with Multicast

UFTP là chương trình truyền tập tin được mã hóa theo cơ chế Multicast, được thiết kế an toàn, đáng tin cậy, và rất hiệu quả trong truyền dữ liệu đến nhiều người nhận cùng một lúc. Chương trình mã hóa Multicast dựa trên TLS với phần mở rộng để cho phép nhiều người nhận có thể chia sẻ một khóa chung

Ý tưởng chủ đạo là đưa các cơ chế kiểm soát dữ liệu của TCP vào UDP.

2.2. Mô tả giao thức UFTP

Một phiên truyền UFTP gồm 3 giai đoạn:

- *Giai đoạn 1: Thông báo/ Đăng ký*

- Giai đoạn này thiết lập phiên truyền tập tin Multicast và thương lượng tất cả các thông số mã hóa

- Các server sẽ gửi một thông báo trên một địa chỉ Multicast. Tất cả các thông tin từ máy chủ sau đó đi qua một hệ thống Multicast

- Client đăng ký gửi trả lời thông báo

- Các server sau đó sẽ gửi một xác nhận. Trong đó xác nhận sẽ chứa các khóa mã hóa nếu cơ chế mã hóa kích hoạt. Nếu Client nhận được các khóa mã hóa, nó sẽ gửi một gói xác nhận trở lại máy chủ.

- *Giai đoạn 2: Truyền file*

+ Bư ớc 1: Bắt đầu Server sẽ gửi một tin nhắn mô tả tập tin truyền bao gồm:

- Tên và kích thước của tập tin

- Cách phân chia tập tin thành các khối. Một tập tin được chia thành một số khối, và các khối được nhóm lại thành các phần. Khối A là một phần của tập tin được gửi đi trong một đơn gói tin. Phần A là một nhóm các khối có thể được gửi đính kèm trước khi Server yêu cầu thông tin phản hồi từ Clients.

- Tổng số số lượng các khối.

+ Bước 2: Truyền dữ liệu

- Các gói dữ liệu được gửi bởi các Server theo một tỷ lệ được quy định bởi người sử dụng.

- Bởi vì UDP không đảm bảo rằng các gói tin đến theo thứ tự để các Client có thể tổ hợp lại tập tin nên khi kết thúc một phần, Server gửi một thông điệp đến các Client yêu cầu trạng thái.

- Các Client gửi lại một thông báo trạng thái có chứa danh sách các NAKs cho các khối trong phần đã gửi.

- Một khi tất cả các phần đã được gửi đi, nếu Server nhận được một số khác 0 NAKs từ Client bất kỳ, nó sẽ bắt đầu gửi lại lần hai các dữ liệu, lần này nó chỉ gửi các gói tin bị sai.

- Server tiếp tục gửi lại dữ liệu cho đến khi tất cả các Client gửi đến thông báo hoàn thành hoặc hết thời gian chờ thông tin trạng thái từ Clients.

- Khi một Client đã nhận được toàn bộ tập tin, nó sẽ gửi một tin nhắn hoàn thành tới Server.

Bước 1 và 2 trong giai đoạn này được lặp đi lặp lại trong suốt phiên làm việc

- *Giai đoạn 3: Hoàn thành/xác nhận*

- Giai đoạn này kết thúc phiên làm việc giữa Server và Client. Nó bắt đầu với một thông điệp từ thông báo kết thúc phiên.

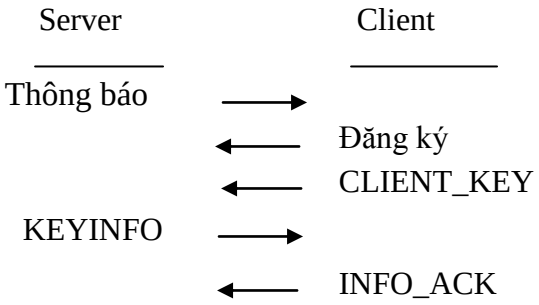
- Các Client trả lời với một thông điệp hoàn thành.

- Server gửi thông điệp xác nhận với mỗi thông điệp hoàn thành mà nó nhận.

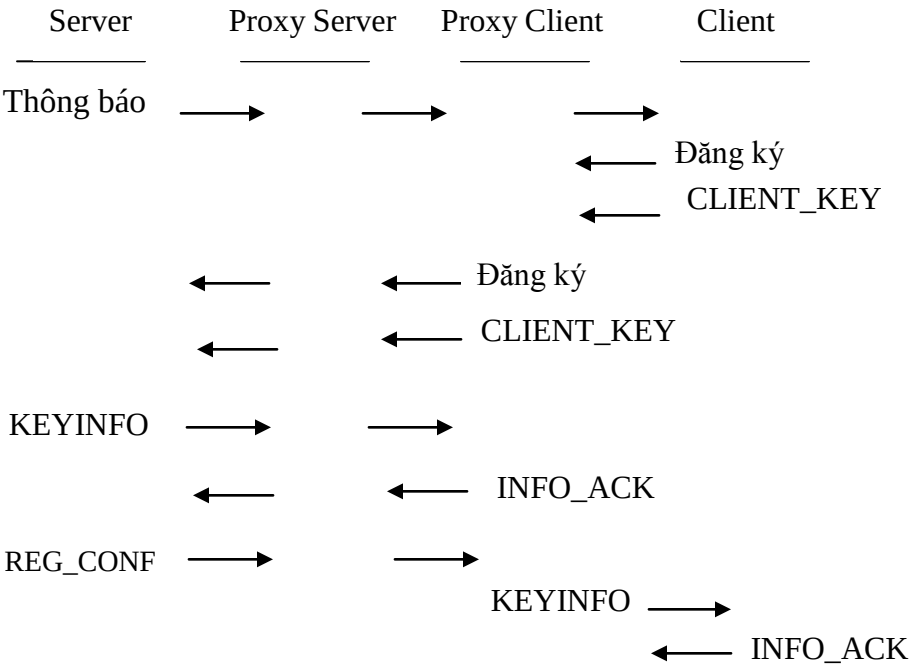
2.2.1. Cơ chế làm việc

1. Giai đoạn thông báo/ đăng ký có mã hóa

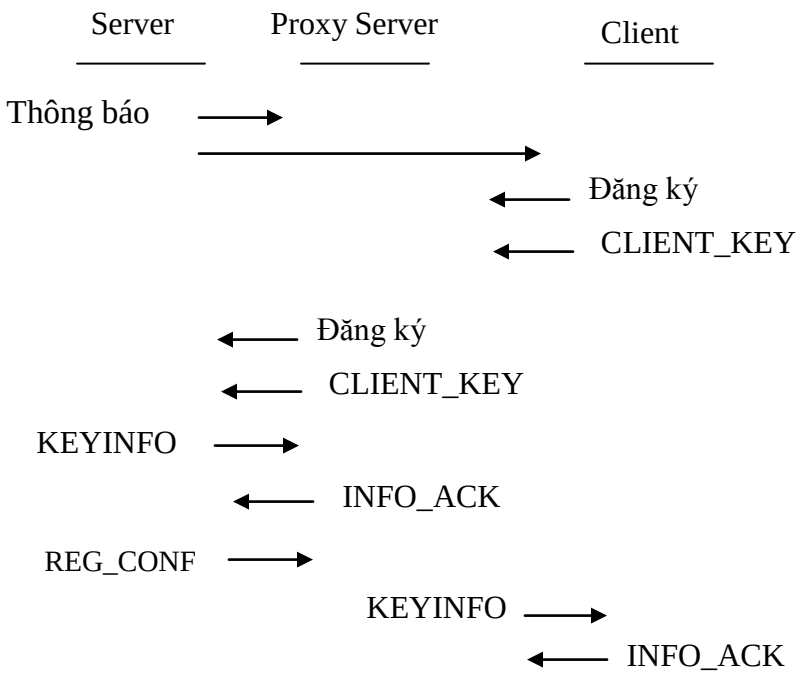
1.1. Trường hợp không có proxy.



1.2. Trường hợp có proxy server/client

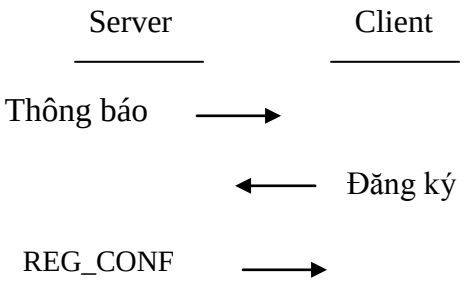


1.3. Trường hợp proxy đáp ứng.

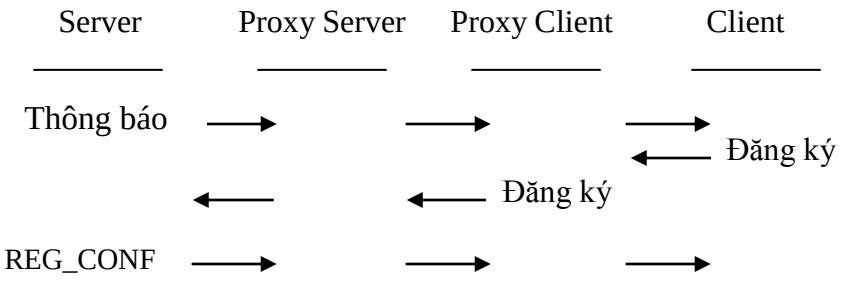


2. Giai đoạn thông báo/ đăng ký mà không cần mã hóa

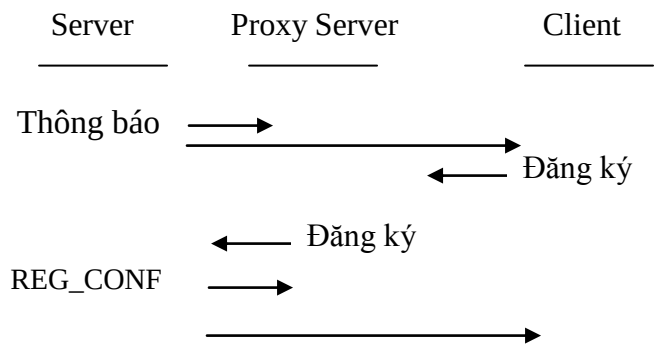
2.1. Trường hợp không có proxy.



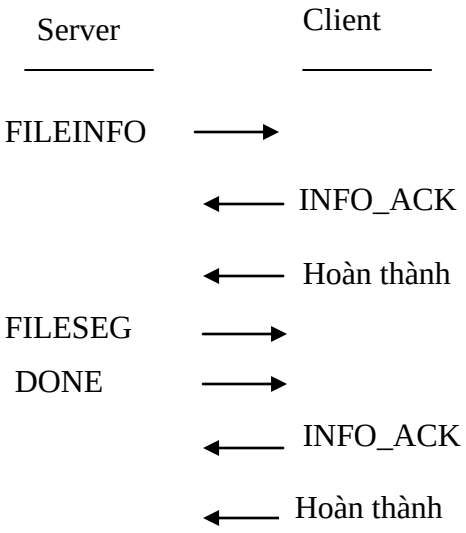
2.2. Trường hợp proxy Server/ Client.



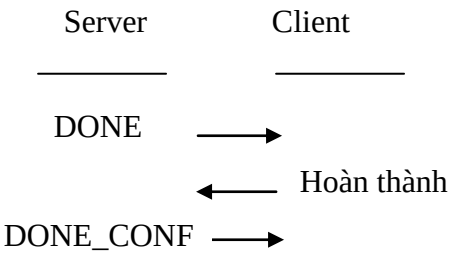
2.3. Trường hợp proxy chủ đáp ứng.



3. Giai đoạn truyền tập tin.



4. Giai đoạn hoàn thành/ xác nhận.



2.2.2. Thông điệp

2.2.2.1. HEADER UFTP.

<table>
<tr><td colspan="32">0 1 2 3</td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td></tr>
<tr><td colspan="8">UFTP ID</td><td colspan="8">Hàm</td><td colspan="16">Kích thước khối</td></tr>
<tr><td colspan="32">Nhóm ID</td></tr>
<tr><td colspan="32">Địa chỉ nguồn</td></tr>
<tr><td colspan="32">Địa chỉ đích</td></tr>
</table>

- UFTP ID: 8 bits.

Xác định số phiên bản của giao thức. Hiện nay là 0x30

- Hàm: 8 bits.

Số lượng thông điệp của thông điệp bao hàm. Nếu thông điệp được mã hóa, điều này luôn luôn xác định số lượng thông điệp cho mã hóa.

- Kích thước khối : 16 bits.

Kích thước của toàn bộ gói UFTP trong byte, ngoài ra không bao gồm tiêu đề này(phải là bội số của 4).

- Nhóm ID : 32 bit

Một ký hiệu nhận dạng duy nhất cho phiên hiện tại

- Địa chỉ nguồn : 32 bit

Địa chỉ IP của người gửi. Đối với server, đây là sự xác định địa chỉ Multicast đi ra. Đối với một client hoặc proxy, đây là một trong hai IP phù hợp với một danh sách địa chỉ IP được liệt kê trong một thông điệp hoặc địa chỉ IP không phải của vòng lặp đầu tiên.

- Địa chỉ đích : 32 bit

Địa chỉ IP đây là thông điệp dành cho bước cuối cùng.

2.2.2.2 .MÃ HÓA.

<table>
<tr><td colspan="32">0 1 2 3</td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td></tr>
<tr><td colspan="32">Nhãn thời gian giây</td></tr>
<tr><td colspan="32">Nhãn thời gian micro giây</td></tr>
<tr><td colspan="16">Chiều dài chữ ký</td><td colspan="16">Tải trọng chiều dài</td></tr>
<tr><td colspan="32">Chữ ký</td></tr>
<tr><td colspan="32">Tải trọng mã hóa</td></tr>
</table>

Chi tiết hóa một thông điệp được mã hóa. Có chứa chữ ký kèm theo tải trọng có ích thông điệp được mã hóa(đó là một trong các thông điệp dưới đây). Chữ ký là một trong HMAC sử dụng nhóm khóa hoặc chữ ký RSA sử dụng RSA của người gửi trên toàn bộ thông điệp UFTP, tùy thuộc vào chữ ký mà server đã chọn.

- Nhãn thời gian giây :32 bit

Thời gian hiện tại được biểu diễn theo giây tính từ giai đoạn UNIX 01/01/1970 UTC

- Nhãn thời gian micro giây: 32 bit

Một phần triệu giây của dòng thời gian hiện hành. Cái miền này và miền ở trên được sử dụng trong sự tính toán tính đối xứng mã hóa tải trọng khóa IV.

- Chiều dài chữ ký: 16 bits

Chiều dài của chữ ký trong nhóm bit (phải là bội số của 4).

- Tải trọng chiều dài : 16 bits

Chiều dài của tải trọng mã hóa trong nhóm bit (phải là bội số của 4).

- Chữ ký: biến thiên.

Chữ ký cho thông điệp này. Nó áp dụng đến toàn bộ gói tin UFTP, bao gồm cả tiêu đề UFTP. Có thể là hoặc một HMAC sử dụng các nhóm khóa HMAC hoặc chữ ký RSA sử dụng khóa riêng RSA của người gửi.

- Tải trọng mã hóa: biến thiên.

Thông điệp được mã hóa, mã hóa với nhóm khóa đối xứng.

2.2.2. 3 .THÔNG BÁO.

<table>
<tr><td colspan="32">0 1 2 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1</td></tr>
<tr><td colspan="8">Hàm</td><td colspan="8">Cờ</td><td colspan="16">Số lượng đích</td></tr>
<tr><td colspan="16">Khoảng thời gian thông báo</td><td colspan="16">Khoảng thời gian thông báo</td></tr>
<tr><td colspan="16">Khoảng thời gian đăng ký</td><td colspan="16">Khoảng thời gian thực hiện</td></tr>
<tr><td colspan="8">Thời gian thông báo</td><td colspan="8">Thời gian trạng thái</td><td colspan="16">MTU</td></tr>
<tr><td colspan="32">Địa chỉ Multicast riêng</td></tr>
<tr><td colspan="8">Quyền máy khách</td><td colspan="8">Loại chữ ký</td><td colspan="8">Loại Hash</td><td colspan="8">Loại khóa</td></tr>
<tr><td colspan="16">Chiều dài modun khóa công khai</td><td colspan="16">Dự trữ</td></tr>
<tr><td colspan="32">Số mũ khóa công khai</td></tr>
<tr><td colspan="32">Số server ngẫu nhiên</td></tr>
<tr><td colspan="32">Modun khóa công khai</td></tr>
<tr><td colspan="32">Địa chỉ IP máy khách</td></tr>
<tr><td colspan="32">...</td></tr>
</table>

Gửi bởi server để bắt đầu truyền tập tin. Chứa thông tin cơ bản cần thiết bởi client để bắt đầu một phiên. Đối với thành viên của nhóm, danh sách khách hàng được phép được quy định cụ thể. Nhiều thông điệp có thể được gửi để phù hợp danh sách đầy đủ của client. Thông điệp này đi qua các địa chỉ Multicast công khai. Tất cả các thông điệp server tiếp theo sau đó đi qua địa chỉ Multicast riêng. Nếu server cần phải gửi lại thông điệp này theo thành viên của nhóm, chỉ có client đã không đáp ứng được liệt kê.

- Hàm: 8 bits

Số lượng cho thông điệp này luôn là 1

- Cờ: 8 bits

0x01- RESTART.

Nếu thiết lập, điều này chỉ ra rằng phiên được biểu diễn bởi thông điệp này là một sự khởi động lại của phiên trước. Tất cả các bits nên thiết lập là 0.

- Số lượng đích: 16 bits

Số lượng địa chỉ IP của client được liệt kê trong thông điệp này. Với thành viên ngoài nhóm, cái này luôn là 0. Với thành viên của nhóm, cái này luôn khác 0.

- Khoảng thời gian thông điệp: 16 bits

Số một phần nghìn giây server sẽ chờ đợi trước khi truyền lại một (A DONE)cái đã thực hiện.

- Khoảng thời gian đăng ký: 16 bits

Số một phần nghìn giây, client nên chờ đợi trước khi truyền lại một ĐĂNG KÝ THÀNH VIÊN

- Khoảng thời gian thực hiện: 16 bits

Số một phần nghìn giây, client nên chờ đợi trước khi truyền lại một bổ sung (COMPLETE)khi đã đầy đủ một DONE_CONF.

- Thời gian thông điệp: 8 bits

Số giây server có thể chời đợi cho một ĐĂNG KÝ hoặc INFO_ACK

Giai đoạn thông báo/ đăng ký hoặc INFO_ACK trong giai đoạn thông tin tập tin trước khi tiếp tục giai đoạn tiếp theo.

- Thời gian trạng thái: 8 bits

Số giây server có thể đợi sau khi gửi một hoặc nhiều thông điệp thực hiện(DONE) trước khi cho bất cứ client nào đã không thể gửi một trạng thái hoặc hoàn thành(COMPLETE) và tiếp tục với giai đoạn truyền dữ liệu.

- MTU: 16 bits

Tổng kích thước của IP thông điệp để gửi như là qui định của server. Cái này bao gồm IP và UDP tiêu đề cũng như thông điệp UFTP. Chú ý tăng điều này không tính đến cho sự tồn tại của IP tiêu đề tùy chọn, do đó khi nó được dự kiến IP tiêu đề tùy chọn có thể được thêm vào, giá trị này nên được đặt thấp hơn đường dẫn MTU để tránh bị phân mảnh gói tin IP.

- Quyền hạn client: 8 bits

Định rõ liệu client có nên gửi một thông điệp CLIENT_KEY bổ sung đến đăng ký thành viên(REGISTER) khi đáp ứng. Giá trị hợp lý là 1 cho trường hợp đúng, 0 cho trường hợp sai.

- Loại chữ ký: 8 bit

Quy định cụ thể số loại chữ ký của chữ ký để sử dụng mã hóa thông điệp. Xem phần

Hằng số thông điệp để xem danh sách các giá trị hợp lý.

- Loại Hash (dữ liệu hỏng): 8 bits

Quy định số loại hash của các thuật toán băm để sử dụng cho chữ ký HMAC và khóa dẫn xuất. Xem phần Hằng số thông điệp cho danh sách các giá trị hợp lý.

- Loại khóa: 8 bits

Quy định số loại khóa của thuật toán mã hóa đối xứng sử dụng. Xem phần Hằng số thông điệp để xem danh sách các giá trị hợp lý.

- Chiều dài modun khóa công khai: 16 bit

Chiều dài trong bytes của modun khóa công khai RSA của server.

- Dự trữ: 8 bit

Dành riêng cho tương lai sử dụng và cần phải được thiết lập là 0.

- Số mũ khóa công khai: 32 bit

Số mũ khóa công khai của khóa công khai RSA của server.

- Số server ngẫu nhiên: 256 bit

32 byte số ngẫu nhiên được lựa chọn bởi server sử dụng để lấy được khóa bí mật chủ giữa server và mỗi client.

- Modun khóa công khai: biến thiên

Modun khóa công khai của khóa công khai RSA của server.

- Địa chỉ IP client: mỗi một 32 bit.

Địa chỉ của một hoặc nhiều client được phép tham gia vào thành viên của nhóm.

2.2.2.4. ĐĂNG KÝ.

<table>
<tr><td colspan="32">0 1 2 3</td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td></tr>
<tr><td colspan="8">Hàm</td><td colspan="24">Dự trữ</td></tr>
<tr><td colspan="16">Số lượng đích</td><td colspan="16">Chiều dài mã hóa chính</td></tr>
<tr><td colspan="16">Khoảng thời gian đăng ký</td><td colspan="16">Khoảng thời gian thực hiện</td></tr>
<tr><td colspan="32">Số client ngẫu nhiên</td></tr>
<tr><td colspan="32">Mã hóa bí mật chính</td></tr>
<tr><td colspan="32">Địa chỉ IP client</td></tr>
<tr><td colspan="32">...</td></tr>
</table>

Được gửi bởi client để xác nhận một thông điệp.Nếu mã hóa là yêu cầu, cũng có chứa một số ngẫu nhiên và khóa chính bí mật được mã hóa với khóa công khai RSA của server.

- Hàm: 8 bit

Số lượng thông điệp cho thông điệp này. Luôn là 2.

- Dự trữ: 24 bit

Dành riêng cho tương lai sử dụng và cần phải được thiết lập là 0.

- Số lượng đích: 16 bits

Số lượng địa chỉ IP của client được liệt kê trong thông điệp này. Với client, cái này luôn là 0.

- Chiều dài mã hóa chính: 16 bít

Chiều dài của khóa bí mật chính được mã hóa.

- Số client ngẫu nhiên: 256 bit

32 byte số ngẫu nhiên được lựa chọn bởi client sử dụng để lấy được khóa bí mật chủ giữa server và mỗi client.

- Mã hóa bí mật chính:biến thiên.

Khóa bí mật chính được chọn bởi client, mã hóa với khóa công khai RSA của server.

- Địa chỉ IP client: mỗi một 32 bit

Địa chỉ IP của một hoặc nhiều client rằng một proxy nhận được một biểu mẫu đăng ký thành viên và chuyển tiếp ngay đến server.

2.2.2.5 .CLIENT_KEY(Khóa client)

0								1								2								3							
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Hàm								Dự trữ																							
Chiều dài modun khóa công khai																Chiều dài xác minh dữ liệu đã ký															
Số mũ khóa công khai																															
Modun khóa công khai																															
Xác minh dữ liệu đã ký																															

Gửi bởi client nếu server yêu cầu client xác thực, hoặc nếu server yêu cầu chữ ký RSA thay vì HMAC.

- Hàm: 8 bit

  Số lượng thông điệp cho thông điệp. Luôn luôn là 3.

- Dự trữ: 24 bit

  Dành riêng cho tương lai sử dụng và cần phải được thiết lập là 0.

- Chiều dài modun khóa công khai. 16 bit

  Chiều dài trong nhóm bit của modun khóa công khai RSA của client.

- Chiều dài xác minh dữ liệu đã ký: 16 bit

  Chiều dài trong nhóm bit của trường xác minh dữ liệu đã ký.

- Số mũ khóa công khai: 32 bit

  Số mũ khóa công khai của khóa công khai RSA của client.

- Modun khóa công khai: biến thiên

  Modun khóa công khai của khóa công khai RSA của client.

- Xác minh dữ liệu đã ký: biến thiên

  Chữ ký từ khóa riêng RSA của client xác minh dữ liệu dựa trên các hash của các chuyển đổi thông số mã hóa đến thời điểm này.

2.2.2.6 .REG_CONF

<table>
<tr><td colspan="32">0 1 2 3</td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td></tr>
<tr><td colspan="8">Hàm</td><td colspan="8">Dự trữ</td><td colspan="16">Số lượng đích</td></tr>
<tr><td colspan="32">Địa chỉ IP máy khách</td></tr>
<tr><td colspan="32">...</td></tr>
</table>

Gửi bởi server để đáp ứng một đăng ký thành viên nếu không có mã hóa. Chứa một danh sách các client mà server nhận đã được cho đăng ký, và nhiều thông điệp có thể được gửi để cung cấp danh sách đầy đủ của client. Ngoài ra được gửi nếu đăng ký thành viên đến từ một proxy. Điều này cho phép các proxy có thể xác nhận đăng ký khi mã hóa được kích hoạt. Các server sẽ không gửi lại thông điệp này cho một client nhất định, trừ khi nó nhận được thêm một đăng ký từ client đó.

- Hàm: 8 bit

Số lượng thông điệp cho thông điệp này. Luôn là 4.

- Dự trữ: 8 bit

Dành riêng cho tương lai sử dụng và cần phải được thiết lập là 0.

- Số lượng đích: 16 bit

Số lượng địa chỉ IP của client được liệt kê trong thông điệp này.

- Địa chỉ IP client: mỗi một 32 bit

Địa chỉ IP của một hoặc nhiều client mà server nhận được từ đăng ký.

2.2.2.7.KEYINFO.

<table>
<tr><td colspan="32">0 1 2 3</td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td></tr>
<tr><td colspan="8">Hàm</td><td colspan="8">Dự trữ</td><td colspan="8">Số lượng đích</td><td colspan="8">Chiều dài nhóm mã hóa chính</td></tr>
<tr><td colspan="32">Nhãn thời gian giây</td></tr>
<tr><td colspan="32">Nhãn thời gian micro giây</td></tr>
<tr><td colspan="32">Địa chỉ IP client</td></tr>
<tr><td colspan="32">Nhóm mã hóa chủ</td></tr>
<tr><td colspan="32">Địa chỉ IP client</td></tr>
<tr><td colspan="32">...</td></tr>
</table>

Gửi bởi server trong việc đáp ứng đến một đăng ký nếu một mã hóa được kích hoạt. Nó có chứa danh sách các client mà server nhận được cho việc đăng ký, và có nhiều thông điệp có thể được gửi để thích hợp với danh sách đầy đủ của client. Nếu server cần phải gửi lại thông điệp này, chỉ có client không được đáp ứng được liệt kê. Ngoài ra, với mỗi client được liệt kê, nó chứa nhóm khóa chủ được mã hóa với khóa chủ bí mật thương lượng với client đó. Nếu một đăng ký nhận được từ proxy, KEYINFO được gửi trực tiếp đến proxy, không phải client nó xử lý.

- Hàm: 8 bit

Số lượng thông điệp cho thông điệp này luôn luôn là 6.

- Dự trữ: 8 bit

Dành riêng cho tương lai sử dụng và cần phải được thiết lập là 0.

- Số lượng đích: 16 bits

Số lượng khóa mã hóa thông điệp được liệt kê trong thông điệp này.

- Chiều dài nhóm mã hóa chính: 8 bit

Chiều dài của nhóm khóa mã hóa chính trong mỗi khóa mã hóa thông điệp luôn là 48.

- Nhãn thời gian giây: 32 bit

Thời gian hiện tại được biểu diễn theo giây kể từ giai đoạn UNIX 01/01/1970 UTC.

- Nhãn thời gian micro giây: 32 bit

Một phần triệu giây của dòng thời gian hiện hành. Cái nhãn thời gian này và nhãn ở trên được sử dụng trong sự tính toán tính đối xứng khóa IV cho nhóm khóa mã hóa chủ với mỗi client được liệt kê.

- Địa chỉ IP client: mỗi một 32 bit.

Địa chỉ IP của một hoặc nhiều client mà server nhận được cho việc đăng ký.

- Nhóm mã hóa chủ:mỗi một 384 bit.

Cuối cùng 47 byte của 48 byte nhóm chủ bí mật, mã hóa bằng cách sử dụng khóa đối xứng chính cho client được liệt kê trước.

2.2.2.8.FILEINFO(thông tin tập tin).

<table>
<tr><td colspan="32">0 1 2 3</td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td></tr>
<tr><td colspan="8">Hàm</td><td colspan="8">Loại tập tin</td><td colspan="16">ID tập tin</td></tr>
<tr><td colspan="32">Tổng khối</td></tr>
<tr><td colspan="16">Tổng số đoạn</td><td colspan="16">Số lượng đích</td></tr>
<tr><td colspan="32">Kích thước tập tin</td></tr>
<tr><td colspan="32">Tên tập tin</td></tr>
<tr><td colspan="32">Địa chỉ IP client</td></tr>
<tr><td colspan="32">...</td></tr>
</table>

Gửi bởi server để cung cấp thông tin trên một tập tin riêng được gửi. Nó chứa danh sách client đang hoạt động hiện tại, và có nhiều thông điệp có thể được gửi để thích hợp với danh sách đầy đủ của client. Nếu server cần phải gửi lại thông điệp này, chỉ có client không được đáp ứng được liệt kê. Nếu mã hóa được kích hoạt, thông điệp này mã hóa và nhúng vào bên trong thông điệp được mã hóa.

- Hàm: 8 bit

Số lượng thông điệp cho thông điệp luôn là 5.

- Loại tập tin: 8 bit

Quy định loại tập tin được gửi đi(tệp thông thường, thư mục, hoặc liên kết ký hiệu). Xem phần Hằng số thông điệp để xem danh sách các giá trị hợp lệ.

- ID của tập tin: 16 bit

Ký hiệu nhận dạng của tập tin hiện hành. Được liên tục chọn bởi server bắt đầu từ 1.

- Tổng khối: 32 bit

Số lượng của khối tập tin được chia nhỏ ra.

- Tổng số đoạn: 16 bit

Số lượng các đoạn của các khối được chia thành các nhóm.

- Số lượng đích: 16 bit

Số lượng địa chỉ IP của client được liệt kê trong thông điệp này.

- Kích thước tập tin: 64 bit.

Kích thước của tập tin được gửi trong byte.

- Tên tập tin: 300 byte.

Tên đường dẫn của tập tin để gửi. Dấu gạch chéo(/) được sử dụng như phân cách thư mục. Các tập tin được tạo ra với đường dẫn trong thư mục đích của client.

- Địa chỉ IP client: 32 bit

Địa chỉ IP của một hoặc nhiều client đang hoạt động.

2.2.2.9.INFO_ACK.

<table>
<tr><td colspan="32">0 1 2 3</td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td></tr>
<tr><td colspan="8">Hàm</td><td colspan="8">Cờ</td><td colspan="16">ID tập tin</td></tr>
<tr><td colspan="16">Số lượng đích</td><td colspan="16">Dự trữ</td></tr>
<tr><td colspan="32">Thẩm tra dữ liệu</td></tr>
<tr><td colspan="32">Địa chỉ IP client</td></tr>
<tr><td colspan="32">...</td></tr>
</table>

Gửi bởi client trong việc đáp ứng một KEYINFO hoặc một FILEINFO. Nếu được gửi trong đáp ứng đến KeyINFO, chứa dữ liệu xác nhận dựa trên thông số mật mã trao đổi đến lúc này. Nếu gửi trong đáp ứng đến FILEINFO, chứa các ID tập tin. Nếu mã hóa được kích hoạt, thông điệp này được mã hóa và nhúng trong thông điệp đã được mã hóa.

- Hàm : 8 bit

Số lượng thông điệp cho thông điệp này luôn là 7.

- Cờ : 8 bit

0x01: phần riêng

Nếu thiết lập, điều này chỉ ra rằng các client phần riêng nhận được tập tin chỉ thị trên một máy chạy trước. Giá trị chỉ trong đáp ứng đến FILEINFO. Tất cả các bit nên thiết lập là 0.

- ID tập tin: 16 bit

Nếu gửi trong đáp ứng đến FILEINFO, nhận dạng của tập tin hiện hành. Nếu gửi trong đáp ứng đến KEYINFO, 0.

- Số lượng đích: 16 bit

Số lượng địa chỉ IP của client được liệt kê trong thông điệp này.

- Dự trữ: 16 bit

Dành sử dụng cho tương lai và cần phải được thiết lập là 0.

- Địa chỉ IP client: mỗi một 32 bit

Địa chỉ IP của một hoặc nhiều client mà proxy nhận được từ INFO_ACK và chuyển tiếp ngay đến server.

2.2.2.10.FILESEG

<table>
<tr><td colspan="32">0 1 2 3</td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td></tr>
<tr><td colspan="8">Hàm</td><td colspan="8">Dự trữ</td><td colspan="16">ID tập tin</td></tr>
<tr><td colspan="8">Pass</td><td colspan="8">Dự trữ</td><td colspan="16">Phiên</td></tr>
<tr><td colspan="32">Số khối</td></tr>
</table>

Được gửi bởi server, và chứa một khối dữ liệu kế tiếp tiêu đề này. Nếu mã hóa được kích hoạt, thông điệp này được mã hóa và nhúng vào với thông điệp đã được mã hóa.

- Hàm : 8 bit

Số lượng thông điệp cho thông điệp này luôn là 8.

- ID tập tin: 16 bit

Nhận dạng của tập tin hiện hành.

- Pass: 8 bit

Số pass mà server hiện thời.

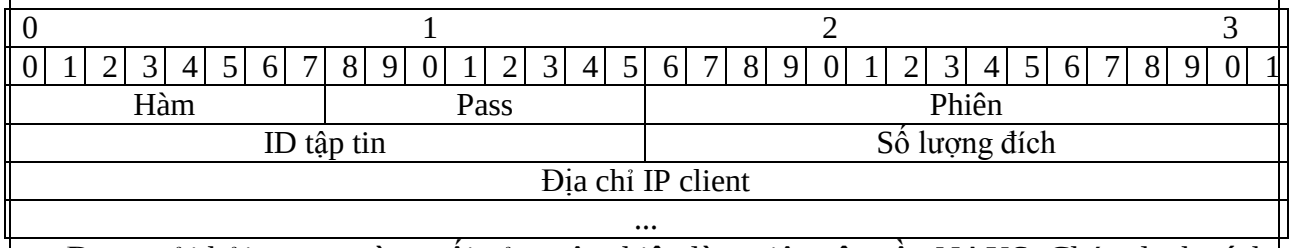
- Phiên : 16 bit

Số phiên cho khối.

- Số khối: 32 bit

Số lượng của khối này.

2.2.2.11.DONE(thực hiện)



Được gửi bởi server vào cuối của một phiên làm việc yêu cầu NAKS. Chứa danh sách client mà server cần cho trạng thái, và có nhiều thông điệp có thể được gửi để thích hợp với danh sách đầy đủ của client. Nếu server cần phải gửi lại thông điệp này, chỉ có client không được đáp ứng được liệt kê. Nếu mã hóa được kích hoạt, thông điệp này được mã hóa và nhúng và trong thông điệp đã được mã hóa.

- Hàm: 8 bit

Số lượng thông điệp cho thông điệp này luôn là 9.

- Pass: 8 bit

Số pass mà server hiện thời.

- Phiên : 16 bit

Số phiên cho khối này.

- ID tập tin: 16 bit

Nhận dạng của tập tin hiện hành. Nếu bằng không, chứng tỏ rằng bắt đầu của giai đoạn hoàn thành/ xác nhận đã đến kết thúc phiên.

- Điểm đích: 16 bit

Số lượng của địa chỉ IP client được liệt kê trong thông điệp này.

- Địa chỉ IP client: 32 bit

Địa chỉ IP của một hoặc nhiều client mà server cần trạng thái chờ.

2.2.2.12 .TRẠNG THÁI(STATUS)

<table>
<tr><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td></tr>
<tr><td colspan="8">Hàm</td><td colspan="8">Dự trữ</td><td colspan="16">ID tập tin</td></tr>
<tr><td colspan="12">Pass</td><td colspan="12">Dãy số</td><td colspan="8">Phiên</td></tr>
<tr><td colspan="32">Số lượng NAK</td></tr>
</table>

Gửi bởi client trong việc đáp ứng một thông điệp được thực hiện. Nếu mã hóa được kích hoạt, thông điệp này được mã hóa và nhúng vào trong một thông điệp đã được mã hóa.

- Hàm : 8 bit

Số thông điệp cho thông điệp này luôn là 10.

- Dự trữ: 8 bit

Dành sử dụng cho tương lai và cần phải được thiết lập là 0.

- ID tập tin: 16 bit

Nhận dạng tập tin hiện hành.

- Pass: 8 bit

Số pass mà server hiện thời.

- Dãy số: 8 bit

Một dãy số so khớp trạng thái này(STATUS) đến trạng thái gần nhất(PRSTATUS).

- Phiên: 16 bit

Số phiên cho khối.

- Số lượng NAK: 32 bit

Tổng số của NAK chứa trong thông điệp này. Nếu bằng không, thông điệp này chỉ chứa tiêu đề. Nếu khác không, chứa một mặt nạ bit của các NAK cho phiên xác định, và mặt nạ bit này là kích thước của một khối.

2.2.2.13 .PRSTATUS(trạng thái tiếp, trạng thái gần nhất).

0								1								2								3							
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Hàm								Dự trữ								ID tập tin															
Pass								Dãy số								Phiên															
Số lượng đích																Dự trữ															
Địa chỉ IP client																															
...																															

Được gửi bởi proxy client cùng với một thông điệp trạng thái. Chứa danh sách của client hầu hết áp dụng trạng thái gần đây nhất. Nếu mã hóa được kích hoạt, thông điệp này được mã hóa và nhúng vào trong thông điệp đã được mã hóa.

- Hàm : 8bit

Số thông điệp cho thông điệp này luôn là 11.

- Dự trữ: 8 bit

Dành sử dụng cho tương lai và cần phải được thiết lập là 0.

- ID tập tin: 16 bit

Nhận dạng của tập tin hiện hành.

- Pass: 8 bit

Số pass mà server hiện thời.

- Dãy số: 8 bit

Một dãy số so khớp trạng thái gần nhất này(PRSTATUS) đến một trạng thái (STATUS).

- Phiên : 16 bit

Số phiên cho khối này.

- Số lượng đích: 16 bit

Số lượng địa chỉ IP client được liệt kê trong thông điệp này.

- Dự trữ: 16 bit

Dành sử dụng cho tương lai và cần phải được thiết lập là 0.

- Địa chỉ IP client: mỗi 32 bit

Địa chỉ IP của một hoặc nhiều client mà proxy nhận được một trạng thái(STATUS) và chuyển tiếp ngay đến server.

2.2.2.14.HOÀN THÀNH(COMPLETE)

<table>
<tr><td colspan="32">0 1 2 3</td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td></tr>
<tr><td colspan="8">Hàm</td><td colspan="8">Trạng thái</td><td colspan="16">ID tập tin</td></tr>
<tr><td colspan="16">Số lượng đích</td><td colspan="16">Dự trữ</td></tr>
<tr><td colspan="32">Địa chỉ IP client</td></tr>
<tr><td colspan="32">...</td></tr>
</table>

Được gửi bởi client để đáp ứng đến thông điệp đã được thực hiện(DONE) khi client nhận được toàn bộ tập tin. Cũng có thể được gửi trong một đáp ứng đến FILEINFO nếu phiên là phiên khởi động lại và client nhận được toàn bộ tập tin trên sự thử nghiệm trước.

Nếu mã hóa được kích hoạt, thông điệp này được mã hóa và nhúng và trong một thông điệp đã được mã hóa. Nếu ID tập tin là 0, chứng tỏ kết thúc phiên làm việc, tất cả các tập tin và thư mục gửi trong suốt phiên này được chuyển từ thư mục tạm thời của client đến thư mục đích, nếu một thư mục tạm thời được thiết lập. Các tập tin trong thư mục được di chuyển như một phần chứa thư mục.

- Hàm : 8bit

Số thông điệp cho thông điệp này luôn là 12.

- Trạng thái: 8 bit

Qui định trạng thái của thông điệp hoàn thành. Khi ở chế độ đồng bộ hóa, một trạng thái của COMP_STAT_NORMAL xác định rằng tập tin là một bản sao chép mới trên, trạng thái COMP_STAT_SKIPPED qui định rằng tập tin đã được bỏ qua bởi vì tập tin đến là cũ, và một trạng thái của COMP_STAT_OVERWRITE qui định rằng tập tin ghi đè lên một tập tin đã tồn tại. Khi không ở chế độ đồng bộ hóa, trạng thái được thiết lập COMP_STAT_NORMAL nếu tập tin đã được gửi đi thành công. Nếu khách hàng từ chối các tập tin do một vấn đề có thể đường dẫn hoặc tên tập tin, mặc dù đang ở chế độ đồng bộ, trạng thái được thiết lập là COMP_STAT_REJECTED.

- ID tập tin: 16 bit

Nhận dạng của tập tin hiện hành.

- Số lượng đích: 16 bit

Số lượng địa chỉ IP client được liệt kê trong thông điệp này.

- Dự trữ: 16 bit

Dành sử dụng cho tương lai và cần phải được thiết lập là 0.

- Địa chỉ IP client: 32 bit

Địa chỉ IP của một hoặc nhiều client mà proxy nhận được từ hoàn thành và chuyển tiếp ngay đến server.

2.2.2.15 .DONE_CONF

<table>
<tr><td colspan="32">0 1 2 3</td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td></tr>
<tr><td colspan="8">Hàm</td><td colspan="8">Dự trữ</td><td colspan="16">ID tập tin</td></tr>
<tr><td colspan="16">Số lượng đích</td><td colspan="16">Dự trữ</td></tr>
<tr><td colspan="32">Địa chỉ IP client</td></tr>
<tr><td colspan="32">...</td></tr>
</table>

Gửi bởi server đáp ứng với một thông điệp hoàn thành(COMPLETE) vào cuối phiên. Chứa danh sách client đã hoàn thành. Và có nhiều thông điệp có thể được gửi để thích hợp với danh sách đầy đủ của client. Các server sẽ không gửi lại thông điệp này cho một client nhất định trừ khi nó nhận được thêm một thông điệp hoàn thành từ client đó. Nếu mã hóa được kích hoạt, thông điệp này được mã hóa và nhúng vào trong thông điệp đã được mã hóa.

- Hàm : 8 bit

Số lượng cho thông điệp này luôn là 13

- Dự trữ: 8 bit

Dành sử dụng cho tương lai và cần phải được thiết lập là 0.

- ID tập tin: 16 bit

Nhận dạng của tập tin hiện hành. TODO: loại bỏ trường này, không cần thiết.

- Số lượng đích: 16 bit

Số lượng địa chỉ IP client được liệt kê trong thông điệp này.

- Dự trữ: 16 bit

Dành sử dụng cho tương lai và cần phải được thiết lập là 0.

- Địa chỉ IP client: 32 bit

Địa chỉ IP của một hoặc nhiều client mà server nhận được cho việc hoàn thành.

2.2.2.16. HỦY BỎ.

0								1								2														3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Hàm								Cờ								Dự trữ															
Máy chủ(Host)																															
Thông điệp																															

Gửi bởi một trong client hoặc server khi có lỗi điều kiện xảy ra. Thông điệp này có thể hoặc không thể mã hóa, tùy thuộc vào dù có hay không nhóm khóa chủ đã được đưa ra.

- Hàm : 8 bit

Số lượng cho thông điệp này luôn là 99.

- Cờ : 8 bit

0x01: tập tin hiện hành.

Chỉ áp dụng nếu gửi cho client, và chỉ khi nếu trường Host bằng 0. Nếu thiết lập, qui định cho tất cả client không hoạt động liên tục trên tập tin hiện hành phải hủy bỏ. Client hoàn thành trên tập tin hiện hành thì không hủy bỏ và có thể nhận tập tin tiếp theo trong phiên.

Tất cả các bit khác nên thiết lập là 0.

- Dự trữ: 16 bit

Dành sử dụng cho tương lai và cần phải được thiết lập là 0.

- Host: 32 bit

Nếu được gửi bởi server, qui định client là server muốn hủy bỏ, hoặc bằng 0 để định rõ rằng tất cả client phải hủy bỏ. Nếu được gửi bởi client, cái này được thiết lập bằng 0. Nếu được gửi bởi một proxy đại diện cho một client, nó được thiết lập cho IP của client mà hủy bỏ. Nếu gửi bởi proxy đại diện riêng của nó, nó được thiết lập bằng 0.

- Thông điệp: 300 bytes

Văn bản mô tả nêu rõ nguyên nhân việc hủy bỏ.

2.2.2.17.HB_REQ

0								1								2								3							
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Hàm								Dự trữ																							
Chiều dài modun khóa công khai																Chiều dài ký														
Nonce																															
Số mũ khóa công khai																															
Modun khóa công khai																															
Giá trị(nonce) ký																															

Gửi bởi một proxy(thường là proxy client) đến proxy upstream cho mục đích mở ra một lỗ hổng trong một bức tường lửa mà proxy upstream có thể gửi thông qua, và cho thấy IP NAT của proxy đến proxy upstream biết nơi gửi các yêu cầu khác.

- Hàm: 8 bit

Số lượng thông điệp cho thông điệp này luôn là 14.

- Dự trữ: 24 bit

Dành sử dụng cho tương lai và cần phải được thiết lập là 0.

- Chiều dài modun khóa công khai: 16 bit

Chiều dài trong byte của modun khóa công khai RSA của proxy.

- Chiều dài ký: 16 bit

Chiều dài trong bytes của trường ký lúc này.

- Nonce: một số duy nhất sinh ra ngẫu nhiên: 32 bit

Giá trị nhận được từ một HB_RESP trước đó sẽ được ký.

- Số mũ khóa công khai: 32 bit

Số mũ khóa công khai của khóa công khai RSA của proxy.

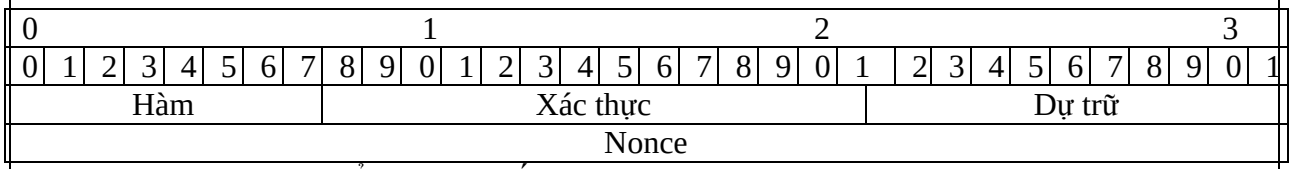
- Modun khóa công khai: biến thiên

Modun khóa công khai của khóa công khai RSA của proxy.

- Giá trị(nonce) ký: biến thiên

Chữ ký từ khóa riêng RSA của proxy của số(giá trị: nonce) qui định.

2.2.2.18. HB_RESP



Gửi bởi một proxy để đáp ứng đến thông điệp HB_REQ.

- Hàm : 8 bit

Số lượng thông điệp cho thông điệp này luôn là 15

- Xác thực: 8 bit

Định rõ trạng thái của HB_REQ thông điệp này đã được đáp ứng. Giá trị của HB_AUTH_OK có nghĩa đã xác thực thành công hoặc là không được yêu cầu. Giá trị của HB_AUTH_CHALLENGE có nghĩa xác thực thông tin dự tính không được định rõ. Giá trị của HB_AUTH_FAILED có nghĩa thông tin xác thực đưa ra là không hợp lệ.

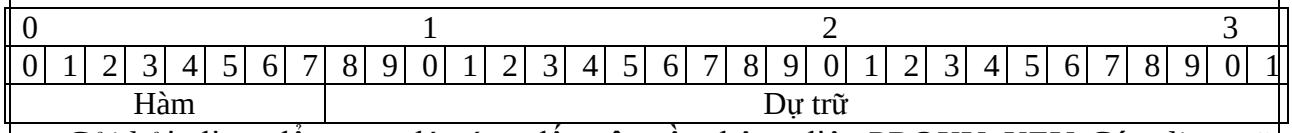
- Dự trữ: 16 bit.

Dành sử dụng cho tương lai và cần phải được thiết lập là 0.

- Nonce(số ngẫu nhiên): 32 bit

Khi xác thực= HB_AUTH_CHALLENGE, giá trị nonce được dự tính để ký trong xác thực HB_REQ.

2.2.2.19.KEY_REQ



Gửi bởi client để proxy đáp ứng đến yêu cầu thông điệp PROXY_KEY. Các client sẽ gửi thông điệp này 5 giây 1 lần cho đến khi nó nhận được đáp ứng một cách hợp lệ.

- Hàm: 8 bit

Số lượng thông điệp cho thông điệp này luôn là 16

- Dự trữ: 24 bit

Dành sử dụng cho tương lai và cần phải được thiết lập là 0.

2.2.2.20.PROXY_KEY

0								1								2								3							
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Hàm								Dự trữ																							
Chiều dài modun khóa công khai																Chiều dài ký														
Nonce																															
Số mũ khóa công khai																															
Modun khóa công khai																															
Giá trị(nonce) ký																															

Gửi bởi proxy đáp ứng cho client với mục đích cung cấp khóa công khai RSA của nó. Khi một client nhận được một thông điệp trực tiếp từ server, nó có chứa khóa công khai RSA của server.

Các client sau đó có thể sử dụng khóa proxy để mã hóa bí mật chủ chính trong đăng ký thay vì khóa của server. Thông điệp này được gửi về địa chỉ Multicast công khai được qui định đầu tiên, do đó tất cả client upstream đều có thể đọc nó.

Để tránh tấn công thâm nhập dịch vụ, proxy sẽ không gửi thông điệp nhiều hơn 5s một lần.

- Hàm : 8 bit

Số lượng thông điệp cho thông điệp này luôn là 17.

- Dự trữ: 24 bit

Dành sử dụng cho tương lai và cần phải được thiết lập là 0.

- Chiều dài modun khóa công khai: 16 bit

Chiều dài trong byte của modun khóa công khai RSA của proxy.

- Chiều dài ký: 16 bit

Chiều dài trong bytes của trường ký lúc này.

- Nonce: 32 bit

Một giá trị được lựa chọn ngẫu nhiễn sẽ được ký bởi khóa công khai RSA của proxy.

- Số mũ khóa công khai: 32 bit

Số mũ khóa công khai của khóa công khai RSA của proxy.

- Modun khóa công khai: biến thiên

Modun khóa công khai của khóa công khai RSA của proxy.

- Giá trị(nonce) ký: biến thiên

Chữ ký từ khóa riêng RSA của proxy của số(giá trị: nonce) qui định.

BẢNG CÁC HẰNG SỐ THÔNG ĐIỆP

1. Số loại thông điệp.

Thông điệp	1
Đăng ký	2
CLIENT_KEY	3
REG_CONF	4
FILEINFO	5
KeyInfo	6
INFO_ACK	7
FILESEG	8
DONE(thực hiện)	9
STATUS(trạng thái)	10
PRSTATUS	11
COMPLETE(hoàn thành)	12
DONE_CONF	13
HB_REQ	14
HB_RESP	15
KEY_REQ	16
PROXY_KEY	17
ENCRYTED(Mã hóa)	80
ABORT (Hủy bỏ)	99

2. Số loại khóa

Nonce(loại ngẫu nhiên)	0
DES	1
Triple DES	2
AES 128	3
AES 256	4

3. Các kiểu Hash.

Nonce	0
MD5	1
SHA-1	2
SHA-256	3

4. Các loại chữ ký số.

Nonce	0
HMAC	1
RSA	2

5. Mã xác thực hearbeat.

HB_AUTH_FAILED	0
HB_AUTH_OK	1
HB_AUTH_CHALLENGE	2

6. Kiểu tập tin.

Tệp thông thường	0
Thư mục	1
Liên kết ký hiệu	2

7. Trạng thái hoàn thành.

COMP_STAT_NORMAL	0
COMP_STAT_SKIPPED	1
COMP_STAT_OVERWRITE(ghi đè)	2
COMP_STAT_REJECTED(không được chấp nhận)	3

CHƯƠNG 3: XÂY DỰNG CHƯƠNG TRÌNH THỰC NGHIỆM

3.1. Mô tả chương trình

Chương trình được xây dựng theo mô hình Server-Client sử dụng mã nguồn mở UFTP, bao gồm 2 thành phần :

+ Phần Server :

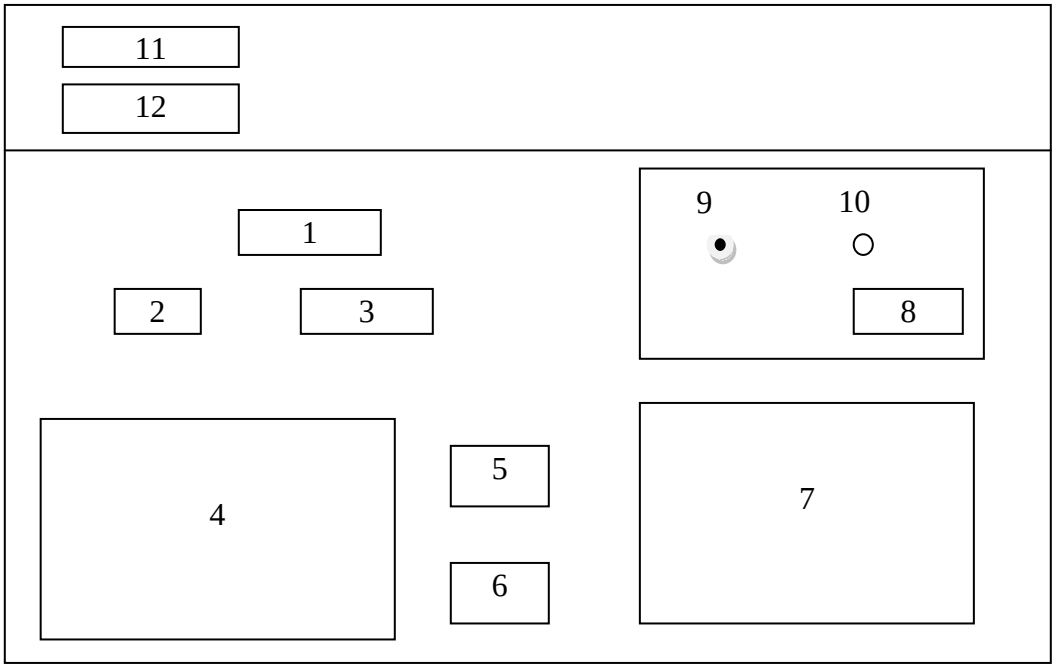
- Truyền file cho các Client tham gia nhóm Multicast yêu cầu nhận file.

+ Phần Client :

- Thực hiện kết nối đến Server.
- Gửi lệnh tham gia vào nhóm Multicast
- Thực hiện nhận file

3.2.Thiết kế chương trình

3.2.1. Server



(1)Textbox : Nhập địa chỉ IP nhóm Multicast

(2)Textbox : Nhập cổng nhóm Multicast

(3)Combobox : Load ổ đĩa của server

(4)Listview : Hiển thị tất cả các thư mục của server.

(5)Button: Lấy file để gửi đi

(6)Button : Xóa file gửi đi

(7)Listview : Hiển thị file mà Server muốn truyền.

(8) Textbox : Nhập thời gian Server muốn phát lại việc truyền file

(9)Checkbox : Chọn thời gian dừng phát lại

(10)Checkbox : Chọn thời gian phát liên tục sau khi ấn định thời gian phát

(11) Button : Start Server

(12) Button : Stop Server

** Quá trình hoạt động*

- Server thực hiện load ổ đĩa, tất cả các thư mục hiện lên trong listview (4)

- Chọn file cần truyền nhấn nút Get. File cần chuyển sẽ được chuyển sang listview (7).

-Nếu muốn hủy không lấy file đó truyền, nhấn nút Clear.

- Sau khi chọn được file cần truyền nhấn nút Start Server. Quá trình truyền file sẽ bắt đầu.

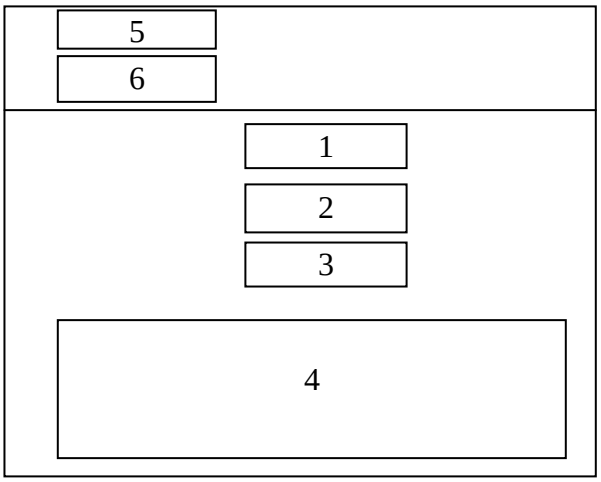
+ Nếu có Client tham gia nhóm Multicast yêu cầu nhận file, Server sẽ truyền file cho Client

+ Nếu không có Client tham gia nhóm Multicast , Server sẽ tự động ngắt.

- Khi quá trình truyền file xảy ra, Server muốn thực hiện phát lại việc truyền file kích chọn Time Loop. Dừng việc phát lại kích chọn Time Off.

- Nhấn vào Stop Server nếu muốn dừng truyền.

3.2.2. Client



(1)Textbox : Nhập địa chỉ IP nhóm Multicast

(2)Textbox : Nhập cổng nhóm Multicast

(3)Combobox : Load ổ đĩa của Client để lưu file nhận được

(4) Textbox : Hiển thị thông tin khi Client kết nối đến Server, và nhận file từ Server.

(5) Button : Kết nối

(6) Button : Ngắt kết nối

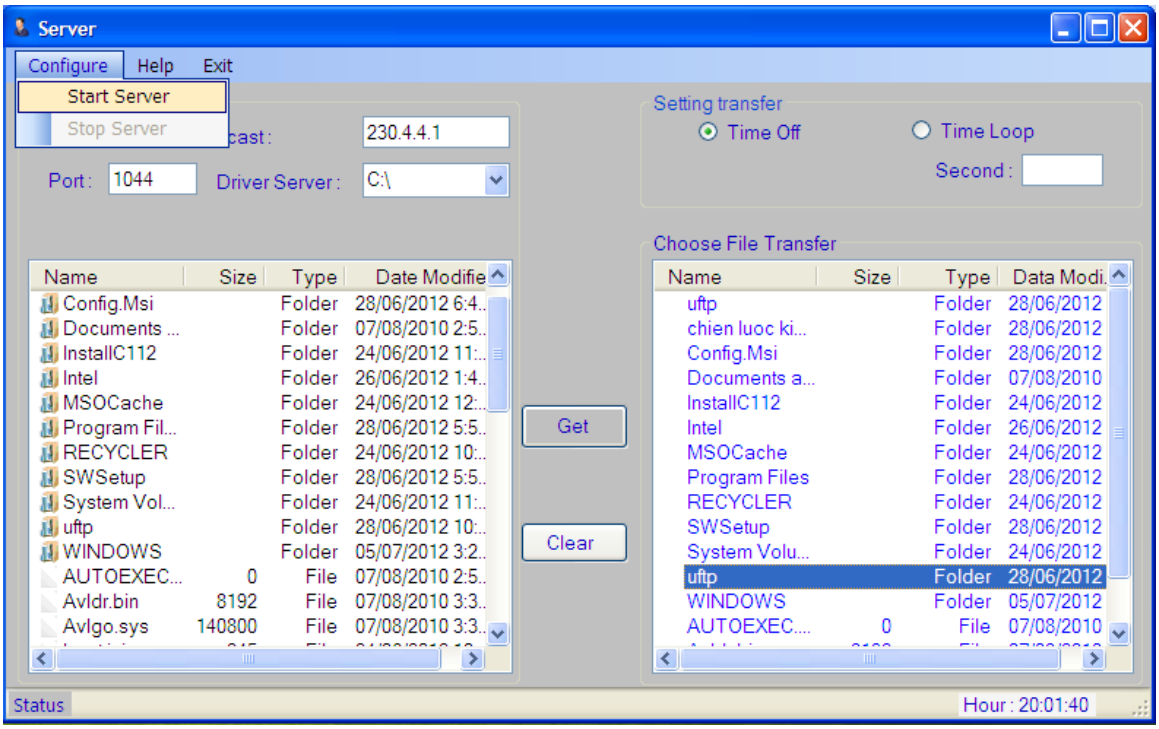
** Quá trình hoạt động*

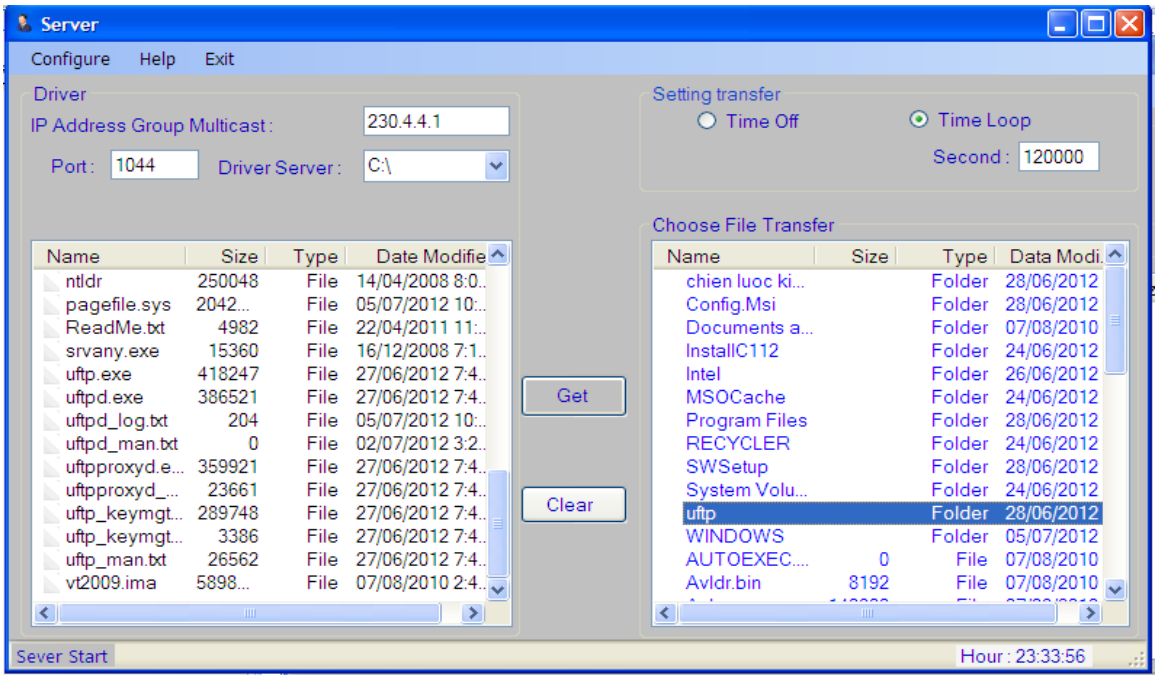
- Client thực hiện chọn ổ đĩa cần lưu file nhận được.
- Nhập địa chỉ IP nhóm Multicast muốn tham gia
- Nhập số cổng của nhóm Multicast
- Thực hiện nhấn nút *Connect* để thực hiện tham gia nhóm Multicast và nhận file
- Nếu muốn dừng việc tham gia và nhận file nhấn nút *Disconnect.*
- Thông tin về việc tham gia nhóm Multicast và việc nhận file từ Server được hiển thị tại hộp *Information.*

3.3. Kết quả đạt được.

Một số hình ảnh quá trình chạy chương trình :

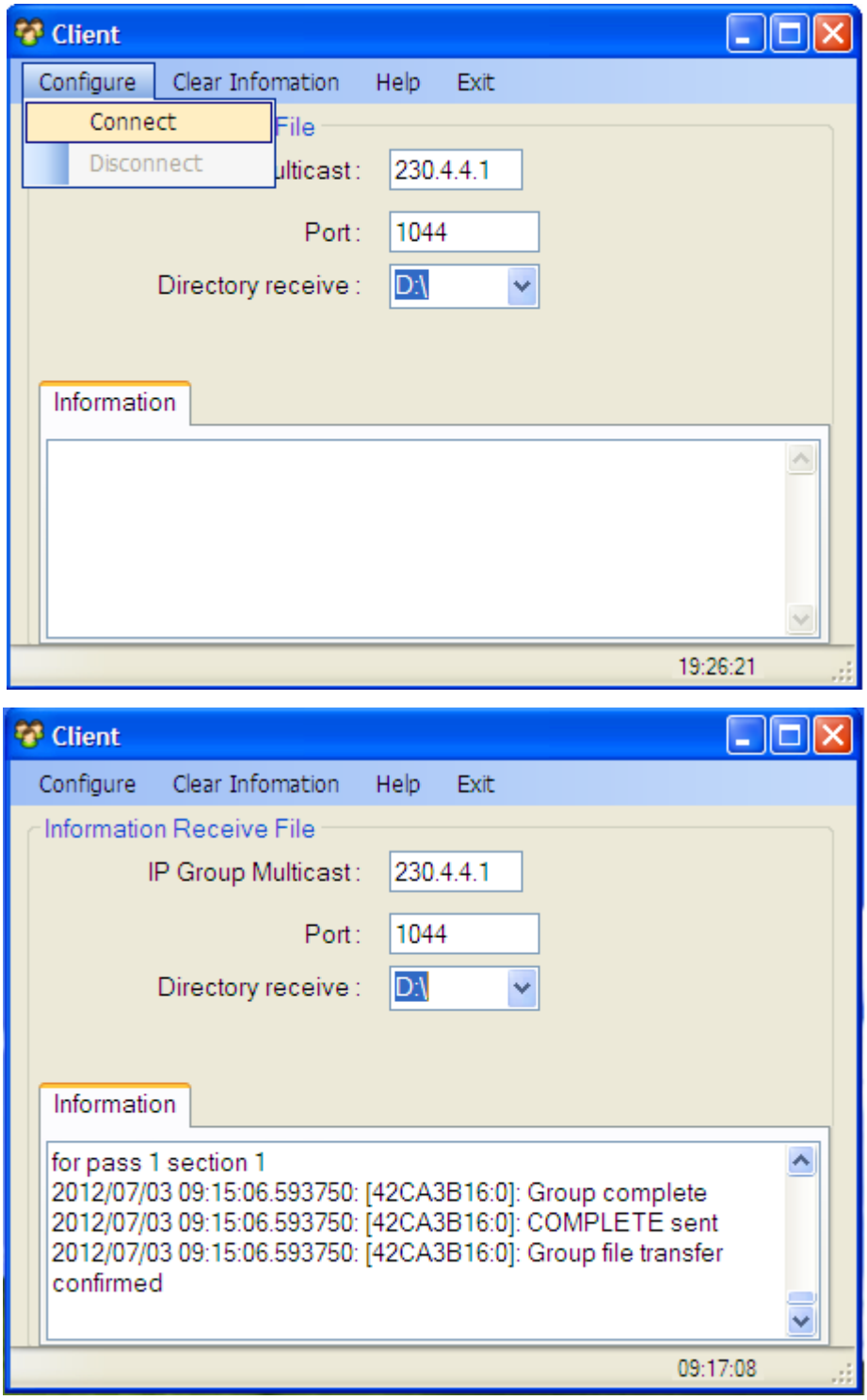
Server:





Hình 6: Server thực hiện gửi file.

Client :



Hình 7 : Client thực hiện tham gia vào nhóm Multicast và nhận file từ Server

KẾT LUẬN

Trong đồ án này, em đã nghiên cứu được một số vấn đề:

- Công nghệ Multicast
- Giao thức UFTP

Qua đồ án em đã củng cố lại kiến thức đã được học. Và em cũng hiểu hơn về Multicast, xây dựng được chương trình sử dụng mã nguồn mở UFTP trong việc truyền file. Nhưng vẫn còn nhiều có vấn đề tồn tại như : cần tìm hiểu thêm về C# để xây dựng thêm tiện ích cho chương trình. Giao diện còn khá đơn giản, tính chuyên nghiệp chưa cao.

Chương trình của em còn nhiều hạn chế mong thầy cô và các bạn có những nhận xét đóng góp ý kiến để nhóm có thể hoàn thiện và phát triển chương trình hơn nữa nhằm làm cho chương trình có thể được ứng dụng dễ dàng nhưng mang lại hiệu quả khi truyền file.

Hải Phòng, tháng 7 năm 2012

Sinh viên

Nguyễn Thị Hằng

TÀI LIỆU THAM KHẢO

Các sách tham khảo :

[1]. Richard Blum _ *C# Network Programming* _ ISBN:0782141765_2003

[2]. Jesse Liberty & O‟Reilly, "*Programming C#*".

Các trang web :

[1]. http://www.codeproject.com

[2]. http://www.tcnj.edu/~bush/uftp.html

[3]. http://www.google.com.