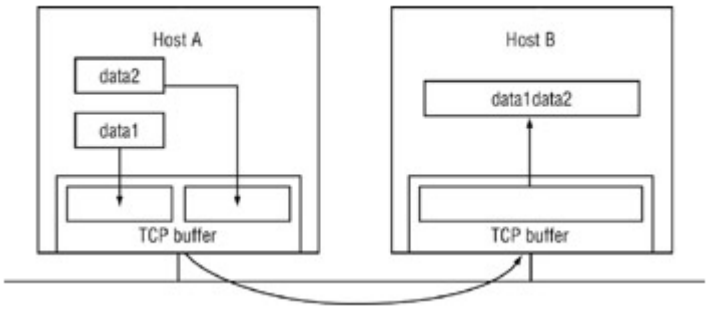
Mục lục

Chương 1: Giới thiệu giao thức TCP và UDP

1.1. Đặc điểm chương trình TCP

Trước hết, TCP là giao thức kết nối hướng đối tượng. Một dòng dữ liệu đã được thiết lập để đảm bảo dữ liệu được di chuyển chính xác từ thiết bị này đến thiết bị khác. Các ứng dụng dùng TCP không lo việc mất mát dữ liệu do có bộ đệm (buffers). TCP phải bảo đảm toàn vẹn dữ liệu. Nó lưu trữ toàn bộ dữ liệu đã gửi trong bộ đệm cho đến khi có sự xác nhận của bên nhận. Tương tự như khi nhận dữ liệu từ mạng, TCP phải giữ bộ đệm để đảm bảo đã nhận đủ tất cả các thành phần trước khi đưa dữ liệu vào chương trình. Bởi vì các buffer riêng lẻ, dữ liệu chuyển đổi giữa chương trình này với chương trình khác trên máy tính ở xa đã được điều khiển thay đổi điều mà mình không mong muốn.



Hình 1: Chuyển đổi dữ liệu TCP

Các chương trình dùng TCP trên hệ điều hành Windows có khả năng đồng bộ dữ liệu giữa chương trình trên máy tính với dữ liệu đầu vào từ thiết bị khác. Thay vì gửi ngay lên mạng, dữ liệu sẽ nằm trong bộ đệm trong một khoảng thời gian. Nghĩa là chương trình nên gửi thêm dữ liệu tới host, mọi sự thay đổi sẽ được thêm vào bộ đệm. Khi chương tình gửi dữ liệu đến thiết bị ở xa, nó sẽ gửi tất cả nội dung trong bộ đệm, không chia nhỏ dữ liệu mà cả bộ đệm data1 và data2 được gộp thành 1 gói để gửi. Dữ liệu đưa vào để xác định là 2 phần riêng biệt hay 1 gói dữ liệu lớn. Khi bạn gửi dữ liệu dưới dạng thông điệp đến 1 thiết bị ở xa, thiết bị đó sẽ không cần thiết nhận đúng số đơn vị thông điệp. Chương trình sẽ đặt tất cả các thông điệp riêng biệt vào bộ đệm TCP. Tuỳ thuộc vào tốc độ gửi và nhận dữ liệu, các thông điệp này được đẩy vào dòng dữ liệu (data

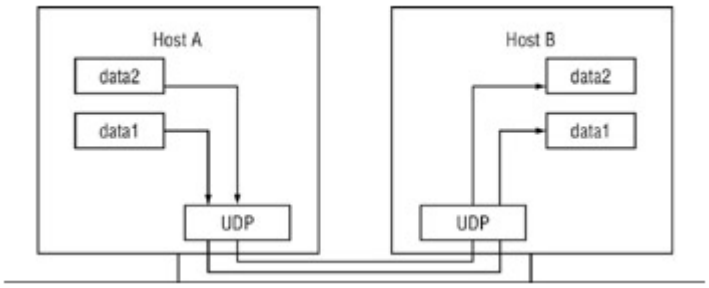
stream). Đặc điểm này của giao thức TCP gây bất ngờ cho nhiều lập trình viên mạng mới vào nghề. Có 2 cách thực hiện là:

- Tạo giao thức quy định đồng bộ dần các dữ liệu gửi.
- Thiết kế hệ thống đánh dấu để phân biệt dữ liệu bên ngoài và dòng dữ liệu

Cả 2 cách trên đều phải cân nhắc trước khi làm. Hầu hết các giao thức Internet dùng TCP. FTP và các giao thức tương tự cung cấp cơ chế dòng lệnh để client gửi thông điệp dạng dòng lệnh.

1.2. Đặc điểm chương trình dùng UDP

UDP được tạo ra để giải quyết vấn đề "thông điệp vùng biên" (message boundary) trong TCP. UDP lưu trữ dữ liệu bên ngoài của tất cả các thông điệp đã gửi từ chương trình. Bởi vì, UDP được thiết kế riêng biệt không quan tâm đến xác thực dữ liệu. Nó không dùng bộ đệm dữ liệu để lưu trữ dữ liệu đã gửi hoặc nhận. UDP lưu trữ thông điệp bên ngoài vào gói tin mạng như minh hoạ sau:



Hình 2: Chuyển đổi dữ liệu theo UDP

Khả năng này của UDP làm nảy sinh vấn đề khác. Bởi vì UDP không đảm bảo việc phân phối dữ liệu. Thiết bị gửi 1 gói tin UDP không có nghĩa là thiết bị nhận nhận được gói tin đó. Chương trình có thể hỏng do mất dữ liệu. Các bước gửi dữ liệu thông qua UDP:

- Gửi dữ liệu đến thiết bị khác
- Khởi động timer, đặt khoảng thời gian định trước.
- Đợi đồng nhất dữ liệu với thiết bị nhận. Khi đã nhận, dừng timer và tiếp tục chương trình.

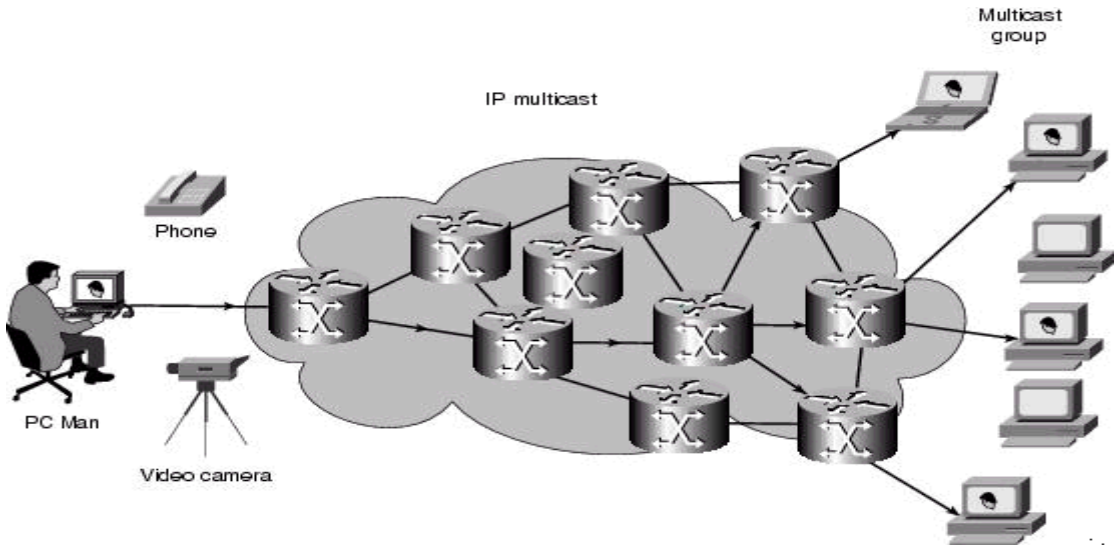
Nếu timer hết hiệu lực trước khi bạn nhận được đồng nhất dữ liệu, thì quay lại và lặp lại bước 1. Sau khi đặt lại khoảng thời gian mà không có phản hồi thì không thể kết nối với thiết bị ở xa.

Gửi dữ liệu dùng UDP nhanh và dễ dàng nhưng phức tạp hơn TCP bởi vì ta cần tự kiểm tra việc mất gói tin.

Chương 2: Giới thiệu về IP Multicast

2.1 Tìm hiểu về IP Multicast

- IP Multicast là công nghệ truyền thông dựa trên nền tảng IP. Nó khai thác hiệu quả môi trường mạng bằng cách gửi các gói tin. Một gói tin
- có thể chia thành nhiều gói tin gửi đến nhiều người nhận. Các nút mạng có trách nhiệm tái tạo và chuyển tiếp hướng tới người nhận. Giao thức phổ biến được dùng ở mức thấp là UDP. Nhưng UDP không có khả năng xác thực _ gây lỗi hoặc gói tin bị mất mát. Giao thức Realiable Multicast còn gọi là PGM được phát triển thêm vào đó cơ chế tự sửa lỗi và phát lại.
- IP Multicast là công nghệ băng thông rộng nhằm làm giảm lưu lượng trong việc phân phối dòng dữ liệu cho nhiều người. Các ứng dụng phổ biến là hội nghị truyền hình, học từ xa, truyền thông…
- Các gói tin Multicast được chuyển tiếp, phát lại trên mạng bởi các Router có chức năng PIM (Protocol Independent Multicast) hoặc các giao thức hỗ trợ multicast khác.



Hình 3: Multicast Transmission Sends a Single Multicast Packet Addressed to All Intended Recipients

- Trong các ứng dụng cần băng thông rộng như MPEG Video thì chỉ có cách duy nhất để gửi cho nhiều người một lúc là IP Multicast.

2.2 Broadcast và Multicast

- IP broadcasting được dùng bởi các thiết bị mạng để gửi 1 gói tin cho mọi thiết bị trên mạng. Bởi vì giao thức TCP yêu cầu 2 thiết bị phải có kết nối tin cậy. Vì vậy không thể gửi một gói tin broadcast trong môi trường TCP. Thay vào đó, UDP được dùng bởi vì giao thức này có khả năng gửi gói tin mà không cần khởi tạo một kết nối đặc biệt.
- Broadcasting là một cách để gửi tin tới tất cả các thiết bị trong cùng subnet nhưng nó có hạn chế là bị giới hạn trong local subnet. IP Multicast kế thừa cho phép một ứng dụng gửi gói tin tới một thiết bị trong cả local subnet và mạng khác. Tính năng này cho phép một chương trình kết nối tới nhóm multicast (multicast group) để thực hiện các hội nghị trên diện rộng (wide area conference).
- IP multicast dùng những địa chỉ IP đặc biệt. Các dải địa chỉ IP tạo ra các nhóm multicast khác nhau. Mỗi nhóm multicast bao gồm một nhóm thiết bị đang lắng nghe cùng một địa chỉ IP. Vì một gói tin gửi tới đích là địa chỉ nhóm nên mỗi thiết bị phải "lắng nghe địa chỉ" đó để nhận tin.

Dải địa chỉ 224.0.0.1 đến 239.255.255.255 được gọi là địa chỉ nhóm Multicast.

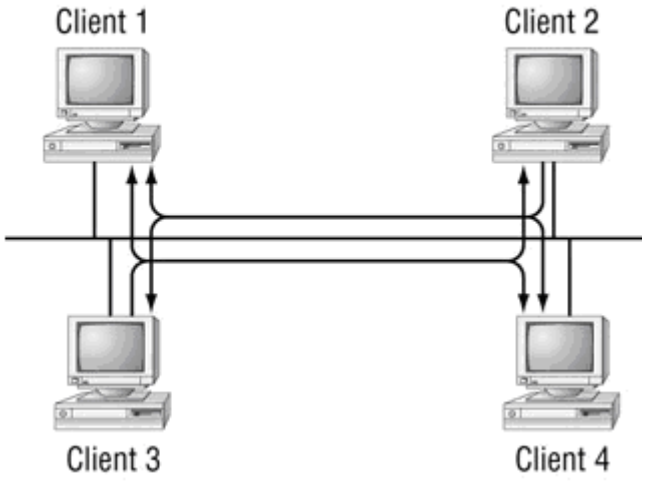
2.2.1 Các công nghệ Multicast

Có 2 công nghệ được dùng:

- Công nghệ peer – to- peer: tất cả các client có thể gửi thông điệp đến tất cả client khác trong nhóm.
- Central sever: gửi thông điệp tới nhóm client.

a. Công nghệ Peer – to- Peer

Tất cả các client trong nhóm multicast đều có quyền ngang nhau. Bất kỳ client đều có khả năng trao đổi thông điệp với client khác trong nhóm.

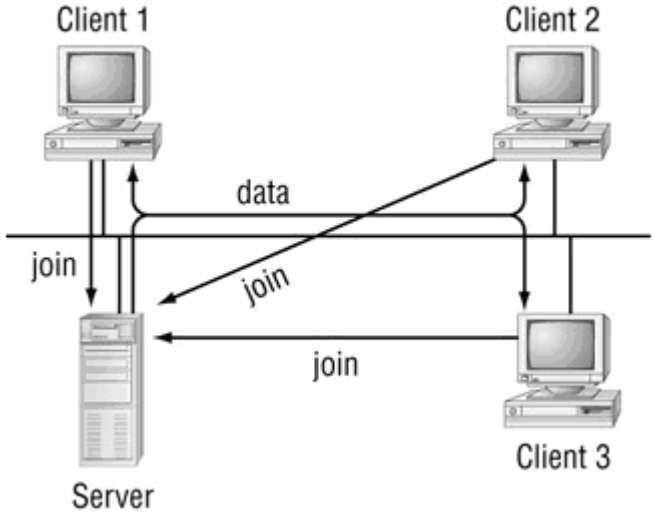


Hình 4: Mọi client có thể trao đổi thông điệp với client khác trong nhóm

Hệ thống IP hỗ trợ nhóm multicast theo peer – to – peer cho phép mọi thiết bị trên mạng gửi và nhận gói tin có đích là địa chỉ nhóm multicast. Một số yếu tố mã hoá để ngăn chặn client nặc danh nhận dữ liệu trong nhóm nhưng vẫn không có cách để nhận xác thực từ client về dữ liệu.

b.Central Server

Hệ thống dùng một thiết bị trên mạng để điều khiển toàn bộ hoạt động của nhóm multicast, gọi là central Server. Một client muốn kết nối vào nhóm phải xin quyền từ central server. Nếu central server từ chối cho client truy cập nhóm thì không một gói tin nào được chuyển tiếp tới nó. Như hình sau:

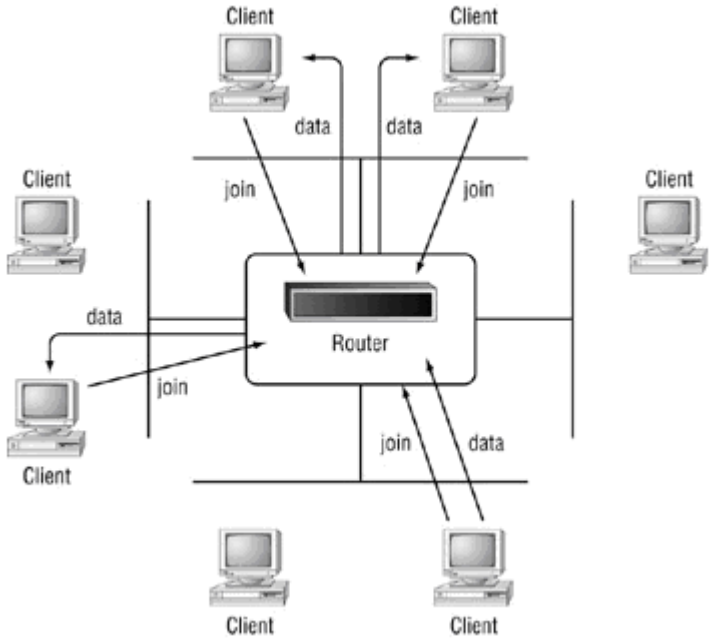


Hình 5: Central Server điều khiển nhóm Multicast

- Lưu ý: hệ thống nhóm multicast theo công nghệ central server không được hỗ trợ bởi IP. Hiện nay chỉ có mạng ATM có hỗ trợ nhóm multicast.

2.3. Gửi gói tin Multicast thông qua Routers

Mặc dù các gói tin multicast có thể gửi sang mạng khác nhưng theo mặc định hầu hết các router không chuyển tiếp gói tin sang subnet khác. Nếu router cho mọi gói tin chuyển tiếp qua thì có thể gây tràn gói tin. Chính vì vậy IGMP (Internet Group Management Protocol) được phát triển để giúp router tránh tắc nghẽn khi gửi gói tin sang subnet khác. Khi thiết bị mạng muốn kết nối vào nhóm multicast, nó sẽ gửi một gói tin IGMP tới router cục bộ trong subnet đó. Gói tin IGMP xác nhận thiết bị và địa chỉ nhóm của thiết bị nhận. Nó định tuyến chuyển tiếp gói tin từ nhóm đến subnet của thiết bị nhận. Hình sau biểu diễn quá trình xác nhận:



Hình 6:Quá trình xác thực thành viên trong router

Tương tự, khi một host rời khỏi nhóm multicast, một gói tin IGMP khác được gửi tới router để thông báo không chuyển tiếp gói tin.

2.4 Nhóm Multicast

Multicast hoạt động dựa trên cơ chế nhóm. Nhóm này không có đường biên giới tự nhiên hay địa lý nào. Các host nằm ở bất kỳ đâu trên mạng Internet. Host muốn nhận dữ liệu trong nhóm phải "join" vào nhóm. Giao thức được bên nhận sử dụng là IGMP (Internet Group Management Protocol). Host phải là thành viên của nhóm mới nhận được dữ liệu.

2.5 Địa chỉ nhóm – IP Multicast group address.

- IP Multicast group address được bên gửi và bên nhận dùng để gửi và nhận dữ liệu.

+ Bên gửi dùng địa chỉ nhóm như địa chỉ đích cho các gói tin.

+ Bên nhận dùng để báo cho mạng chúng đã nhận được các gói tin từ nhóm.

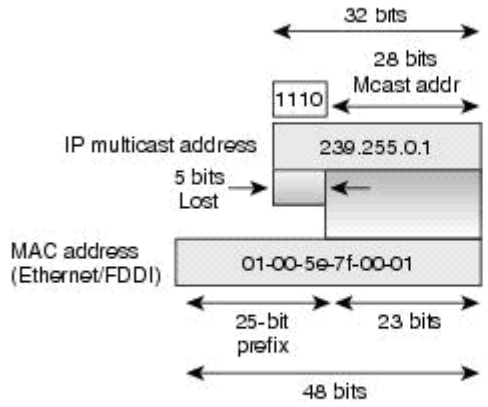
Ví dụ: địa chỉ nhóm là 239.1.1.1. Bên gửi sẽ gửi dữ liệu đến địa chỉ đích là 239.1.1.1. Bên nhận thông báo lên mạng đã nhận dữ liệu được gửi từ địa chỉ 239.1.1.1. Bên nhận phải "join" vào địa chỉ 239.1.1.1. Giao thức được bên nhận dùng để join là Internet Group Managenment Protocol.

- IANA xác định dải địa chỉ lớp D làm địa chỉ Multicast. Độ dài của địa chỉ nhóm là 254.0.0.0 đến 239.255.255.255.Dải địa chỉ này chỉ được dùng làm địa chỉ nhóm hoặc địa chỉ đích trong IP Multicast. Địa chỉ nguồn của các gói tin multicast luôn là địa chỉ unicast.
- 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 lũy thừa 28 địa chỉ nhóm multicast được trích dẫn ra từ lớp D này. Các địa chỉ multicast là tượng trưng một nhóm, không tượng trưng cho host.
- Trong đó, dải địa chỉ từ 224.0.0.0 đến 224.0.0.255 được dùng cho các giao thức trên mạng. Các gói tin mang địa chỉ này không được chuyển tiếp bởi các router. Chúng được để trong các phân đoạn mạng LAN cục bộ và có Time To Live (TTL) là 1.

2.6 Ánh xạ địa chỉ IP multicast sang địa chỉ MAC

- Làm thế nào mà một router và switch kết hợp một địa chỉ multicast của IP với một địa chỉ MAC. Việc gán địa chỉ multicast vào một nhóm L3 sang một nhóm multicast thường sẽ tự động tạo ra địa chỉ multicast lớp 2. 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 multicast IP sẽ được ánh xạ vào 23bit thấp của địa chỉ

MAC bằng một giải thuật đơn giản. Địa chỉ MAC được hình thành bằng cách dùng dạng OUI 01005E, sau đó là giá trị 0 và sau cùng là 23 bits địa chỉ của L3 multicast.



Hình 7: 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.

2.7 Tiến trình chuyển đổi địa chỉ Multicast:

- Bước 1: Chuyển đổi địa chỉ IP sang dạng nhị phân. Lưu ý 4bit đầu tiên luôn luôn là địa chỉ 1110 cho bất kỳ địa chỉ multicast nào.
- Bước 2: Thay thế bốn bit đầu tiên 1110 của địa chỉ IP với 6 ký tự (24bits) 01-00-5E như là địa chỉ bắt đầu trong tổng số 12 ký tự dạng thập lục phân (48bits) của địa chỉ multicast MAC.

- Bước 3: Thay thế 5bit kế tiếp của dạng địa chỉ IP vớI một bit 0 trong không gian địa chỉ MAC.
- Bước 4: Chép 23 bit cuối của địa chỉ IP dạng nhị phân vào 23 bit cuối của địa chỉ multicast.
- Bước 5: Chuyển đổi 24bit cuối của địa chỉ multicast từ dạng nhị phân sang dạng 6 số thập lục phân.
- Bước 6: Kết hợp sáu chữ số hexa đầu tiên 01-00-5E với sáu chữ số hexa vừa tính ở bước 5 để hình thành địa chỉ multicast đầy đủ.
- Theo cách thức nêu trên, địa chỉ 238.10.24.5 sẽ sinh ra địa chỉ MAC là 0x01-00-5E-0A-18-05 cũng giống như kết quả do địa chỉ 228.10.24.5. IETF đã chỉ ra rằng khả năng hai ứng dụng multicast trên cùng một LAN có thể tạo ra cùng những địa chỉ MAC là thấp. Nếu tình cờ điều này xảy ra, một gói tin từ một ứng dụng multicasat khác có thể sẽ được phân biệt bằng địa chỉ lớp 3. Người quản trị nên cẩn thận khi chọn lựa địa chỉ multicast, tránh việc tạo ra những địa chỉ MAC tương tự nhau.

2.7.1 Một vài không gian địa chỉ được dành riêng

- Toàn bộ không gian địa chỉ multicast:224.0.0.0239.255.255.255
- 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 232.0.0.0-232.255.255.255.
- 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 route 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.

2.7.2 Địa chỉ multicast cho những nhóm thường trực

- IANA dành ra hai dãy địa chỉ dành riêng cho multicast. Sự khác nhau giữa hai dãy địa chỉ này là dãy thứ nhất được dùng cho những gói tin không nên được truyền bởi router và nhóm thứ hai được dùng khi các gói tin phải được truyền bởi router.

1. Dãy địa chỉ được dùng cho cục bộ là 224.0.0.0 đến 224.0.0.255. Các địa chỉ này tương tự như các địa chỉ dùng bởi các giao thức định tuyến. Ví dụ như 224.0.0.5 và 224.0.0.6 được dùng bởi OSPF. Các ví dụ khác bao gồm địa chỉ multicast 224.0.0.1 chỉ ra tất cả các host có thể xử lý multicast và 224.0.0.2 chỉ ra tất cả các router có khả năng xử lý multicast. Dãy các địa chỉ nhóm được dùng khi các gói tin phải được định tuyến là 224.0.1.0 đến 224.0.1.255. Dãy địa chỉ này bao gồm 24.0.1.39 và 224.0.1.40 là hai địa chỉ được dùng bởi Auto-RP.

2. Địa chỉ multicast cho các ứng dụng multicast SSM.

- IANA đã cấp phát dãy địa chỉ 232.0.0.0 đến 232.255.255.255 cho các ứng dụng SSM. Mục đích của ứng dụng này là cho phép một host chọn ra một nguồn cho các nhóm multicast. SSM giúp cho việc định tuyến multicast trở nên hiệu quả hơn, cho phép một host chọn lựa một nguồn có chất lượng tốt hơn và giúp các nhà quản trị mạng giảm thiểu kiểu tấn công multicast DoS. Chỉ có các host chạy IGMPv3 có khả năng dùng tính năng SSM. IGMPv3 là một giao thức mới.

- Địa chỉ multicast cho các ứng dụng GLOP

IANA dành ra dãy địa chỉ 233.0.0.0 đến 233.255.255.255.255 gọi là địa chỉ GLOP. Địa chỉ này có thể được dùng bởi bất kỳ ai đang có một AS hợp lệ (registered autonomous system number-ASN) để tạo ra 256 địa chỉ multicast

toàn cục. IANA dành riêng các địa chỉ này để đảm bảo tính duy nhất toàn cục của địa chỉ. Bằng cách dùng giá trị 233 cho octet đầu tiên và bằng cách dùng ASN cho octet thứ hai và thứ ba, một AS có thể tạo ra một địa chỉ multicast toàn cục. Ví dụ nếu AS dùng số hiệu mạng ASN 5663, giá trị này có thể chuyển sang dạng nhị phân là 0001011000011111. 8 bit đầu tiên, 00010110, bằng với 22 trong dạng thập phân và 8 bit cuối, 00011111, bằng với 31 trong dạng thập phân. Ánh xạ 8bit đầu tiên vào octet thứ hai và 8bit cuối vào octet thứ ba trong dãy địa chỉ 233, công ty nào có mạng AS là 5663 sẽ được tự động cấp dãy địa chỉ 233.22.31.0 đến 233.22.31.255.

- Địa chỉ multicast cho những domain riêng.Dãy địa chỉ dành riêng cuối cùng là dãy địa chỉ dành cho quản trị. IANA gán dãy địa chỉ 239.0.0.0 đến 239.255.255.255 (RFC 2365) để dùng trong những miền multicast. IANA sẽ không gán các tầm địa chỉ này tới bất kỳ một giao thức nào hoặc một ứng dụng nào. Các nhà quản trị mạng có thể tự do sử dụng các địa chỉ trong dãy này, tuy nhiên họ phải cấu hình các router multicast để đảm bảo multicast traffic trong dãy địa chỉ này không vượt quá ranh giới của miền multicast.

- Địa chỉ multicast tạm thời cho các nhóm. Khi một doanh nghiệp muốn dùng một địa chỉ multicast toàn cục, doanh nghiệp cần một khối địa chỉ từ ISP hoặc từ IANA. Tuy nhiên, khi một doanh nghiệp muốn dùng một địa chỉ multicast mà không phải là một phần của các không gian địa chỉ multicast được mô tả trong các phần trước, các phần địa chỉ còn lại này được gọi là các địa chỉ multicast transient. Điều này có nghĩa là toàn bộ Internet phải chia sẽ địa chỉ này. Các địa chỉ này sẽ được cấp phát động khi cần thiết và phải được giải phóng khi không còn được dùng. Bởi vì các địa chỉ này không được gán vào bất cứ ứng dụng nào nên nó được gọi là tạm thời. Bất kỳ một doanh nghiệp có thể dùng các địa chỉ multicast này mà không cần sự cho phép từ IANA nhưng các doanh nghiệp cần giải phóng sau khi dùng xong.

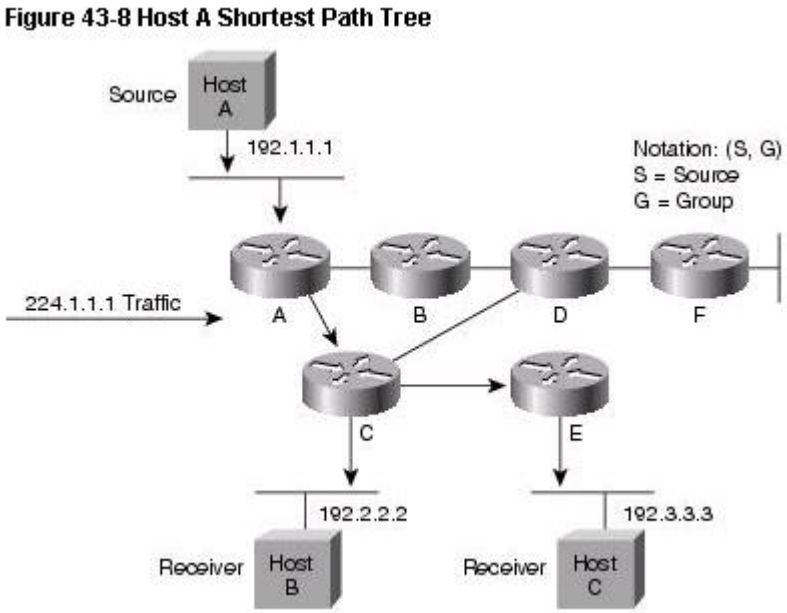
2.8 Cây phân phối Multicast (Multicast Distribution Trees)

Các router có khả năng Multicast tạo cây phân phối nhằm điều khiển đường đi của các gói tin trên mạng. Có 2 loại cơ bản:

2.8.1 Soure tree:

Là dạng đơn giản nhất của cây phân phối là cây một nguồn với gốc của nó là nguồn của cây. Nhánh tạo thành cây bao trùm thông qua những bên nhận. Vì cây này dùng thuật toán đường đi ngắn nhất trên mạng nên nó còn được gọi là SPT (shortest path tree).

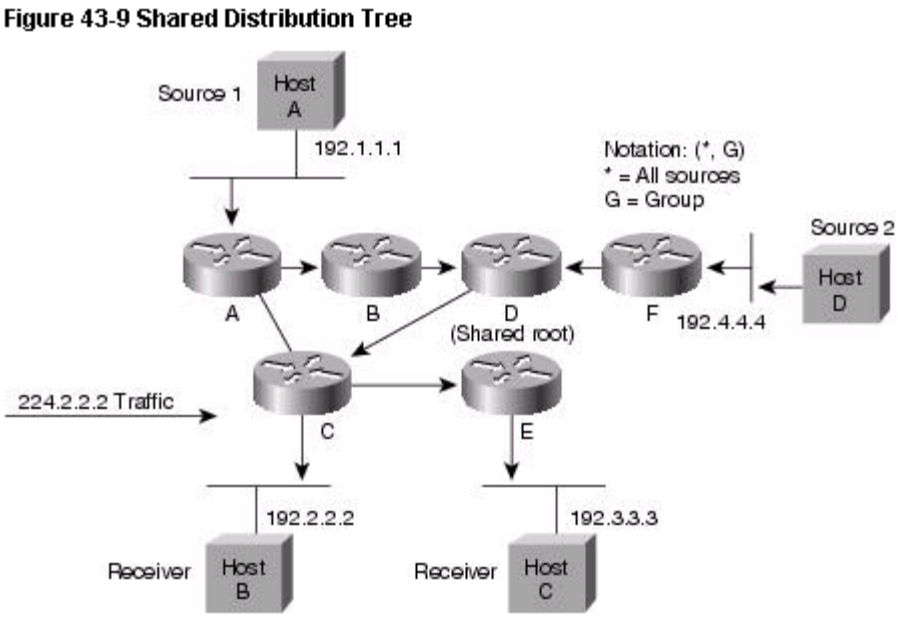
Hình sau là ví dụ cho SPT với nhóm 224.1.1.1 gốc đặt ở nguồn. Host A đang kết nối đến 2 bên nhận là host B và C.



Hình 8: Cây SPT

2.8.2 Share tree:

Không giống như Soure tree, Share tree có gốc được đặt tại một điểm được lựa chọn trên mạng. Nó còn được gọi là rendezvous point (RP).



Hình 9: Share tree

- Hình trên là ví dụ cho Share tree với địa chỉ nhóm là 224.2.2.2, gốc đặt tại router D. Khi sử dụng Share tree, các bên gửi đều phải qua gốc sau đó được chuyển tiếp lại cây để hướng tới các bên nhận.
- Trong ví dụ này, multicast traffic từ các nguồn là host A và D đi qua gốc là router D và quay lại cây phân phối tới 2 bên nhận là host B và C.Bởi vì nhiều bên gửi chỉ dùng 1 cây phân phối nên ký hiệu (*,G) với * đại diện cho tất cả các nguồn và G là nhóm multicast hiện thời.
- Cả SPT và share tree đều là loop tree. Các thông điệp được phát lại trên các nhánh của cây.Các thành viên nhóm multicast có thể "join" hoặc "leave" bất kỳ lúc nào. Do đó cây phân phối phải được cập nhật động. Khi tất cả các bên nhận trên các nhánh riêng biệt dừng truy vấn đến traffic của một nhóm multicast. Các router sẽ "chặt" nhánh trên cây phân phối và dừng chuyển tiếp trên các nhánh đó. Nếu một bên nhận trên nhánh hoạt động trở lại và truy vấn đến traffic, router tự động bổ sung vào cây phân phối và chuyển tiếp lại.
- SPT có lợi điểm là tạo đường đi tối ưu từ nguồn tới đích, tạo ra lượng nhỏ nhất các chuyển tiếp trên mạng. Nhưng nó gặp phải vấn đề là : Các router

phải nhớ thông tin đường dẫn của mỗi nguồn. Trên mạng có hàng nghìn nguồn và hàng nghìn nhóm multicast, chúng có thể nhanh chóng trở thành nguồn trong router. Bộ nhớ dành cho bảng định tuyến trong router trở thành một vấn đề cho những người thiết kế mạng.

- Share tree cơ ưu điểm là: tuỳ thuộc vào dung lượng nhỏ nhất của trạng thái hiện tại trên router. Nó giảm yêu cầu về bộ nhớ. Điểm bất lợi của nó là trong một số trường hợp, đường đi từ bên gửi đến bên nhận không phải là đường đi tối ưu. Nhà thiết kế mạng cần phải cân nhắc khi chọn RP.

2.9 Multicast Forwarding

- Trong định tuyến unicast, traffic chuyển tiếp trên mạng thông qua một đường từ bên gửi đến bên nhận. Một router unicast không cần biết địa chỉ bên gửi mà chỉ cần biết địa chỉ đích và cách chuyển tiếp đến đích. Router dùng bảng định tuyến để chuyển gói tin unicast.

- Trong định tuyến multicast, nguồn gửi traffic tới một nhóm bất kỳ nằm trong nhóm multicast. Router multicast cần phải xác định rõ đường upstream và đường downstream. Nếu có nhiều đường downstream, router cần tái tạo gói tin và chuyển tiếp lại các gói tin thích hợp- không cần thiết là tất cả.. Đó là cách chuyển tiếp theo kiểu multicast từ một nguồn đến nhiều người nhận. và được gọi là chuyển tiếp ngược (reverse path forwarding)

Chương 3 : Giao thức RTP (Real Time Transport Protocol)

3.1 RTP_Real Time Transport Protocol

- Phương thức thông thường để truyền tải dữ liệu dạng audio hay video cùng các dữ liệu đính kèm và khung truyền là RTP. RTP nhằm mục tiêu cung cấp các dịch vụ truyền tải hình ảnh thời gian thực như audio/video thông qua mạng (IP network).

- Các dịch vụ đó bao gồm khôi phục dữ liệu sau khoảng thời gian xác định(timing recovery),tìm và sửa lỗi(loss detection and correction), định dạng khung truyền và nguồn (payload and source identification), tiếp nhận phản hồi về chất lượng dịch vụ(reception quality feedback), đồng bộ dữ liệu (media synchronization)và quản lý thành viên(membership management). RTP được thiết kế để dùng trong trao đổi quảng bá (multicast conferences) dùng cơ chế lightweight sessions.

Nó tỏ ra hữu ích trong hàng loạt các ứng dụng: hội nghị truyền hình dùng cơ chế H323, webcasting, truyền hình,hệ thống thoại có dây và không dây. Bằng việc dùng giao thức này mỗi phiên truyền tải được gửi đến hàng nghìn người

- RTP được coi là chuẩn đóng gói để truyền tải dữ liệu dạng audio và video qua mạng Internet. Nó được phát triển bởi Audio-Video Transport Working Group và được công bố lần đầu năm 1996 là RFC 1889. Sau này là RFC 3550 năm 2003.

- RTP thường được dùng trong hệ thống truyền thông cùng với RTSP để thực hiện hội nghị truyền hình và hệ thống thoại. Nó chứa dòng dữ liệu được điều khiển bởi H323, MGCP và SIP. Đó là nền tảng công nghệ cho kiến trúc Voice IP.

- RTP thường được dùng kết hợp với RTCP. Trong khi RTP truyền dữ liệu or out-of-band signal(DTMF). RTCP được dùng để điều khiển trạng thái truyền và chất lượng dịch vụ QoS. Khi được dùng kết hợp RTP thường được sắp xếp nhận trên cổng chẵn, RTCP trên cổng lẻ.

- RTP ban đầu được tạo ra là giao thức multicast nhưng nó vẫn chấp nhận trong nhiều ứng dụng unicast. Trong truyền thông dạng host-to – host, RTP và RTCP thường dùng UDP qua các giao thức ở lớp vận chuyển.
- Số hiệu cổng: RTP không thao tác trên các cổng mặc định như TCP và UDP. Tuy nhiên nó thường dùng các cổng có số hiệu chẵn, có giá trị từ 16384- 32767, dải địa chỉ này đang dần được mở rộng. Việc này gây khó khăn trong trường hợp có firewall và NATs. Để giải quyết vấn đè này, nó cần khởi tạo STUN server.

3.2 Hoạt động của giao thức:

3.2.1 . Sender

-Bên gửi có nhiệm vụ lưu trữ và biến đổi dữ liệu dạng nghe nhìn để truyền tải. RTP tham gia vào quá trình sửa lỗi và điều phối tránh tắc nghẽn bằng cách thêm vào dòng dữ liệu truyền tải thông tin phản hồi của bên nhận.
-Các khung truyền được nạp vào các gói RTP và sẵn sàng gửi. Nếu khung truyền quá lớn, nó sẽ được chia nhỏ thành nhiều gói RTP. Ngược lại, nó sẽ được tập hợp lại thành một gói RTP. Tuỳ thuộc vào lược đồ sửa lỗi được dùng, 1 chanel coder được dùng để tạo ra các gói sửa lỗi hoặc các gói lặp lại trước khi truyền.
- Sau khi các gói RTP được gửi bộ đệm dữ liệu tương ứng với nó được giải phóng. Bên gửi không huỷ các dữ liệu cần cho quá trình sửa lỗi hoặc mã hoá. Có nghĩa là bên gửi phải lưu trữ dữ liệu trong một khoảng thời gian sau khi các gói tin tương ứng được gửi, tuỳ thuộc vào sơ đồ mã hoá và sửa lỗi được dùng.
- Bên gửi có trách nhiệm tạo các bản báo cáo trạng thái định kỳ về dòng dữ liệu để đồng bộ dữ liệu. Nó cũng nhận thông tin phản hồi từ các bên tham gia và dùng thông tin đó để tham gia vào quá trình truyền.

3.2.2 Receiver

- Bên nhận có trách nhiệm thu nhận các gói RTP từ mạng, sửa lỗi, recovering the timing, cắt bớt dữ liệu và hiển thị kết quả cho người dùng. Nó gửi phản hồi chất

lượng cho bên gửi. Cơ sở dữ liệu của các bên được duy trì trong một phiên truyền.

3.3 Đặc điểm của cơ chế

- RTP không có sẵn các cơ chế để đảm bảo việc truyền theo thời gian hay các kỹ thuật về QoS mà dựa vào các dịch vụ ở lớp dưới để thực hiện những khả năng này. RTP không đảm bảo an toàn hay thứ tự các packet khi truyền, số thứ tự trong RTP packet cho phép bên nhận sắp xếp lại các packet thứ tự khi truyền của người gửi. Ngoài ra số thứ tự cũng có thể được tận dụng để xác định vị trí thích hợp của một packet, ví dụ trong các việc giải mã video, mà không cần phải giải mã các packet theo thứ tự.

Các gói tin truyền trên mạng Internet có trễ và jitter không đoán được. Nhưng các ứng dụng đa phương tiên yêu cầu một thời gian thích hợp khi truyền các gói dữ liệu và phát lại. RTP cung cấp các cơ chế bảo đảm thời gian, số thứ tự và các cơ chế khác liên quan đến thời gian.

Bằng các cơ chế này RTP cung cấp sự truyền tải dữ liệu thời gian thực giữa các đầu cuối qua mạng. Tem thời gian (time-stamping) là thành phần quan trọng nhất trong các ứng dụng thời gian thực. Người gửi thiết lập các "tem thời gian" tăng dần theo thời gian với mọi gói. Sau khi nhận được gói dữ liệu, bên thu sử dụng các "tem thời gian" này nhằm khôi phục thời gian gốc để chạy các ứng dụng này với tốc độ thích hợp. Ngoài ra nó được sử dụng để đồng bộ các dòng dữ liệu khách nhau (chẳng hạn như giữa hình và tiếng). Tuy nhiên RTP không thực hiện đồng bộ mà các ứng dụng phía trên sẽ thực hiện sự đồng bộ này. Bộ phận dạng tải xác định kiểu định dạng của tải tin cũng như các phương cách mã hóa nén. Từ các bộ phận định dạng này, các ứng dụng phía thu biết cách phân tích và chạy các dữ liệu tải tin. Tại một thời điểm bất kỳ trong quá trình truyền tin, các bộ phát RTP chỉ có thể gửi một dạng của tải tin dạng tải tin có thể thay đổi trong thời gian truyền (thay đổi để thích ứng với sự tắc nghẽn của mạng).

Một chức năng khác mà RTP có là xác định nguồn. Nó cho phép các ứng dụng thu biết được dữ liệu từ đâu. Ví dụ thoại hội nghị, từ thông tin nhận dạng nguồn một người sử dụng có thể biết được ai đang nói.

Hình 10 - Mã hóa gói tin RTP trong gói IP

Các cơ chế hoạt động nêu trên được thực hiện thông qua mào đầu của RTP. Cách mã hoá gói tin RTP trong các gói tin IP được mô tả trên hình RTP nằm ở phía trên UDP, sử dụng các chức năng ghép kênh và liểm tra của UDP. UDP và TCP là 2 giao thức là 2 giao thức sử dụng chủ yếu trên Inernet. TCP cung cấp kết nối định hướng và các dòng thông tin với độ tin cậy thấp với hai trạm chủ. Sở dĩ UDP được sử dụng làm thủ tục truyền tải cho RTP là bởi hai lý do:

- Thứ nhất, RTP được thiết kế chủ yếu cho việc truyền tin đa đối tượng, các kết nối có định hướng, có báo nhận không đáp ứng tốt điều này.
- Thứ hai, đối với dữ liệu thời gian thực, độ tin cậy không quan trọng bằng truyền đúng theo thời gian. Hơn nữa sự tin cậy trong TCP là do cơ chế báo phát lại không thích hợp cho RTP. Ví dụ khi mạng bị tắc nghẽn một số gói dữ liệu sẽ bị mất, chất lượng dịch vụ thấp nhưng vẫn chấp nhận được. Nếu thực hiện việc phát lại sẽ gây ra độ trễ lớn chất lượng thấp gây ra sự tắc nghẽn của mạng.

3.4 Kiến trúc gói dữ liệu

bit offset	0-1	2	3	4-7	8	9-15	16-31
0	Ver.	P	X	CC	M	PT	Sequence Number
32	Timestamp						
64	SSRC identifier						
96	CSRC identifiers (optional) ...						
96+(CC×32)	Extension header (optional).						
96+(CC×32) + (X×((EHL+1)×32))	Data						

Hình 11: Kiến trúc gói dữ liệu

RTP header có kích thước tối thiểu là 12 octet

- Ver (2 bit) : Cho biết version của giao thức. Hiện tại đang là version 2.
- P (Padding): được dùng để cho biết có byte mở rộng thêm vào cuối gói RTP.
- X(Extension): (1 bit) cho biết sự có mặt của Extension header và payload data.
- CC (CSRC Count_4 bit): tổng số CSRS đã được định nghĩa thêm vào header.
- M(Marker_ 1 bit): được dùng trong các mức ứng dụng và được định nghĩa bởi 1 profile. Nếu nó được khởi tạo dữ liệu có liên quan đặc biệt đến ứng dụng.
- PT (Payload Type): Cho biết khuôn dạng payload và xác định giới hạn của nó trong ứng dụng.
- Sequence Number : số thứ tự tăng dần trong mỗi ứng dụng RTP đã được gửi và được dùng để biên nhận sửa lỗi và lưu trữ lại số thứ tự gói tin.
- Time Stamp: phản ánh sampling instant của dữ liệu trong gói RTP. Nó tăng dần và tuyến tính để đồng bộ dữ liệu và tính jitter. Số liệu đó phải đủ để đồng bộ chính xác và tính được jitter, thông thường 1 tick trên 1 khung truyền video là không đủ để tính. Tần số của đồng hồ phụ thuộc vào khung dữ liệu. Giá trị mặc định khi khởi tạo là 0.
- SSRC (32 bits): Đồng bộ nguồn dữ liệu xác định duy nhất một dòng dữ liệu vào.

CSRC: Đếm số các nguồn vào từ nhiều nguồn khác nhau.

Extension header: 32 bit đầu bao gồm 16 bit định danh chương trình và 16 bit xác định độ dài phần mở rộng (EHL=extension header length), thêm 32 bit mở rộng.

Chương 4: RCTP

RCTP (Real-time Transport Control Protocol)là giao thức hỗ chợ cho RTP cung cấp các thôngtin phản hồi về chất lượng truyền dữ liệu. Cácdich vụ mà RCTP cung cấp là:

- Giám sát chất lượng và điều khiển tắc nghẽn: Đây là chức năng cơ bản của RCTP. Nó cung cấp thông tin phản hồi đến một ứng dụng về chất lượng phản hồi dữ liệu. Thông tin điều khiển này rất hữu ích cho các bộ phát, bộ thu và giám sát. Bộ phát có thể điều chỉnh cách thức truyền dữ liệu dựa trên các thông báo phản hồi của bộ thu. Bộ thu có thể xác định được tắc nghẽn là cục bộ, từng phần hay toàn bộ. Người quản lý mạng có thể đánh giá được hiệu suất mạng.
- Xác định nguồn: Trong các gói RTP, các nguồn được xác định bởi các số ngẫu nhiên có độ dài là 32 bit. Các số này không thuận tiện với người sử dụng RCTP cung cấp thông tin nhận dạng cụ thể hơn dạng văn bản. Nó có thể bao gồm tên người sử dụng, số điện thoại, địa chỉ E-mail và các thông tin khác.
    - Đồng bộ môi trường: Các thông báo của bộ phận phát RTCP chứa thông tin để xác định thời gian RTP tương ứng.
    - Chúng có thể được sử dụng để đồng bộ giữa âm thanh và hình ảnh.
    - Điều chỉnh thông tin điều khiển: Các gói RTCP được gửi theo chu kỳ giữa những người tham dự. Khi số người tham dự tăng lên, cần phải cân bằng giữa việc nhận thông tin điều khiển mới nhất và hạn chế dung lượng điều khiển. Để hỗ trợ một nhóm người điều khiển lớn, RCTP phải chấm dứt điều khiển rất lớn đến từ các tài nguyên của mạng. RTP chỉ cho phép tối đa 5% lưu lượng cho điều khiển toàn bộ lưu lượng của phiên làm việc. Điều này được thực hiện bằng cách điều chỉnh tốc độ phát của RCTP theo số lượng người tham dự.

Chương 5: Secure Realtime Transport Protocol (SRTP)

5.1 Giới thiệu

- Secure Realtime Transport Protocol (SRTP): là 1 phần của RTP có thêm cơ chế mã hoá, xác thực thông điệp và kiểm tra tính toàn vẹn và khôi phục lại dữ liệu RTP trong cả các ứng dụng dạng unicast và multicast.

- Nó được phát triển bởi 1 đội đã từng nghiên cứu giao thức IP và thành thạo mã hoá đến từ Cissco và Ericssion bao gồm David Oran, David McGrew,…Nó được giới thiệu lần đầu tiên vào tháng 3 năm 2004.

- Cũng giống như RTP, SRTP cũng có giao thức điều khiển là SRTCP (Secure Realtime Transport Control Protocol).

5.2 Cách mã hoá dữ liệu:

1) Để mã hoá và giải mã luồng dữ liệu, SRTP dùng Integer Counter Mode: cho phép truy cập ngẫu nhiên vào block bất kỳ. Điều này cho phép RTP chạy trên các mạng không đáng tin cậy có khả năng mất gói dữ liệu. Thông thường bất kỳ chức năng nào cũng có thể dùng cơ chế "couter" nhưng nó tỏ ra không thể lặp lại nhiều lần. Chuẩn mã hoá của RTP thường là bộ đếm số nguyên tăng dần. AES dùng trong cơ chế này là thuật toán mã hoá mặc định với độ dài khoá mã là 128 bit, khoá giải mã là 112 bit
2) Cơ chế f8 hay còn gọi là output feedback mode: phát triển để có khả năng tìm kiếm và thay thế chức năng mặc định. Giá trị khoá mã và khoá giải mã tương tự cơ chế trên. (Cơ chế này của AES được dùng cho mạng điện thoại 3G).

-Bên cạnh việc dùng AES,SRTP còn dùng một cơ chế mã hoá đặc biệt là "NULL Cipher". Trong trường hợp này, "NULL Cipher" không thực hiện bất kỳ mã hoá nào. Điều bắt buộc đối với cơ chế này là phải thực thi trên hệ thống có tương thích SRTP. Độ tin cậy hoàn toàn được bảo đảm. Trong khi các chức năng khác của SRTP như chứng thực, xác nhận thông điệp… vẫn được sử dụng.

- SRTP dễ dàng điều chỉnh thuật toán mã hoá mới. Những chuẩn SRTP luôn là những thuật toán mã hoá mới được thêm vào trong thực thi của giao thức.
- Những cơ chế mã hoá trên không đảm bảo tính toàn vẹn của thông điệp.Attacker có thể giả mạo thông tin và phát lại. SRTP cung cấp cơ chế đảm bảo tính toàn vẹn dữ liệu và an toàn trong phát lại.
- Thuật toán HMAC- SHA1 được dùng Để xác thực thông điệp và đảm bảo tính toàn vẹn. Nó dùng 160 bit để mã hoá, sau đó cắt ra 80 hoặc 32 bit để xác thực, tuỳ thuộc vào gói dữ liệu. HMAC được tính toán dựa trên gói tin payload và thông tin phần header là thứ tự gói dữ liệu.
- Để tránh tấn công lặp lại, bên nhận phải giữ lại các thông điệp đã nhận trước đó, so sánh chỉ số với các thông điệp mới được nhận. và chỉ nhận nếu chưa được nhận.

Chương 6. Các hàm RTP API

6.*1 Giới thiệu chung.*

-RTP Library cung cấp giao diện để phát triển các ứng dụng sử dụng RTP. Thư viện này dựa trên phiên bản mới nhất của những đặc tả và tích hợp những chức năng mới nhất bao gồm cả những thuật toán RCTP.

6.2 Một số hàm RTP

6.2.1 Hàm khởi tạo

RTPCreat() tạo ra một context. Context là định danh được thư viện dùng để chỉ ra phiên RTP nào được tích hợp cùng. Một ứng dụng có thể chạy nhiều phiên tại cùng một thời điểm. Mỗi lời gọi RTPCreat riêng biệt sẽ tạo context khác nhau. Hầu hết các hàm trong thư viện chấp nhận context là đối số đầu tiên.

-RTPCreat() được gọi để khởi tạo một phiên làm việc, các địa chỉ phiên phải được khởi tạo. Thư viện hỗ trợ cả unicast (single point to point), multi-unicast (multiple unicast point to point), multicast và hybrids. Có 2 cách khởi tạo địa chỉ.

- Cách 1là "send set": đây là danh sách các địa chỉ unicast và/hoặc multicast, số hiệu cổng và giá trị ttl (chỉ dành cho multicast). Khi một gói tin được gửi bởi ứng dụng, thư viện sẽ truyền gói tin tới tất cả các địa chỉ trong danh sách. Cách này cho phép truyền theo kiểu unicast bằng cách tạo một địa chỉ unicast trên một port. Theo multicast bằng cách tạo một địa chỉ multicast trên 2 cổng. Theo multi-unicast bằng cách tạo nhiều địa chỉ unicast trên 2 cổng, và hybrids.

- Nếu tạo một địa chỉ trên cặp cổng thì có thể là multicast hoặc unicast. Cách phân biệt như sau:

1. Nếu địa chỉ là unicast nhưng không đồng nhất trên giao diện cục bộ, INADDR_ANY sẽ chấp nhận gói tin trên giao diện bất kỳ.

2. Nếu địa chỉ unicast trên một giao diện cục bộ, thư viện sẽ chỉ chấp nhận gói tin trên giao diện này.

3. Nếu địa chỉ là multicast, thư viện sẽ chuyển tới INADDR_ANY và "join" vào nhóm multicast. Cách này, nó sẽ chấp nhận các gói tin cả unicast hoặc multicast trên cổng xác định.

4. Nếu địa chỉ là NULL, thư viện sẽ chuyển tới INADDR_ANY.

5. Nếu cổng là 0, thư viện sẽ dùng một cổng động. Số hiệu cổng RTP và RTCP phải thay đổi liên tục, thư viện sẽ thử ngẫu nhiên một cặp cổng trong các cổng đã được chỉ định (trên 49152) cho đến khi tìm được cặp cổng. Nếu không tìm được nó sẽ báo lỗi.

- Tất cả các địa chỉ đều viết dưới dạng chuỗi. Nó có dạng "A.B.C.D" hoặc một hostname "machine.domain". Nếu là hostname, thư viện sẽ chuyển đổi tên sang địa chỉ dùng DNS.

Ngoài ra còn một số hàm khởi tạo khác.

- RTPSessionSetBandwidth(): Các gói tin RTCP được gửi với tốc độ phụ thuộc vào băng thông của phiên truyền. Đây là thuộc tính của phiên RTP. Các ứng dụng nên khởi tạo giá trị này trước khi gọi RTPOpenConnection() nhằm tăng tốc độ truyền của RTCP. Nếu không khởi tạo, tốc độ mặc định là 120 kbps.
- RTPMemberInfoSetSDES(): là một kiểu gói tin RTCP, SDES chứa thông tin về mỗi người dùng. Nó bao gồm tên, email vàCNAME của người dùng. Tuỳ từng ứng dụng để khởi tạo giá trị phù hợp. Thông thường trường CNAME phải được khởi tạo trước khi gọi RTPOpenConnection. Tất cả các trường này đều không bắt buộc.

Mỗi lần địa chỉ của phiên được khởi tạo, hàm RTPOpenConnection() được gọi. Thông thường nó phụ thuộc vào socket nhận. Vì vậy, thư viện phải "join" vào nhóm multicast. Sau đó, thư viện sẵn sàng chấp nhận và gửi gói tin.

6.2.2 Các hàm gửi, nhận

- Hàm RTPSend() được dùng để gửi gói tin RTP. Nó cần người dùng chỉ ra bộ đệm, độ dài, giá trị của trường marker trong RTP header, số gia của tem thời gian và context. Thư viện sẽ lấy từ bộ đệm thêm vào RTP header, thực hiện các thao tác cần thiết và gửi gói tin. Tem thời gian ban đầu và số thứ tự được lựa chọn ngẫu nhiên.

Nếu ứng dụng cần để gửi một gói tin RTP đã lưu ở những cấu trúc phân tán, nó có thể dùng RTPSendVector, một dạng tương đương của RTPSend.

- Các gói tin nhận là một tập hợp nhỏ. Để biết, nếu một gói tin có thể đọc, một tiến trình có thể dừng, nó có thể poll hoặc dùng một kỹ thuật khác. Thư viện không đề ra các luật này, nó gửi cho người dùng để xác định xem khi nào dữ liệu được đọc. Để làm điều này,hàm RTPSessionGetRTPSocket và RTPSessionGetRTCPSocket được dùng cho phép người dùng truy cập vào socket nhận. Nó lấy context vào và chỉ tới một socket. Socket được điền đầy.

- Khi gói tin nằm trong socket, ứng dụng gọi hàm RTPReceiver() lấy context, con trỏ bộ đệm và một con trỏ tới giá trị length. Nó nên được khởi tạo phù hợp với bộ đệm. Thư viện sẽ đọc và xử lý các gói RTP hoặc RTCP. Đối với RTCP, nó sẽ thực hiện tất cả lựa chọn. Đối với gói RTP, thư viện sẽ chỉ cập nhật một vài thống kê và giá trị.

- Gói tin trong bộ đệm thông thường được dùng trong RTPReceiver vẫn chứa trong header. Để truy cập lại vào trường header và payload, hàm RTPGetRTPPacket() được dùng. Hàm này truy cập bộ đệm lấy dữ liệu, và trả về con trỏ gói dữ liệu.

6.2.3. Hàm đóng kết nối.

- Khi ứng dụng thoát khỏi phiên RTP, nó thực hiện hai thao tác.

- Hàm RTPCloseConnection được gọi. Nó đóng tất cả các socket nhận đang hoạt động và gửi gói tin BYE và đóng tất cả các socket gửi. Nó không xoá dữ liệu đã

lựa chọn. RTPCloseConnection chấp nhận một chuỗi lý do. Chuỗi này được gửi trong gói BYE. Nếu đặt NULL nghĩa là không có lý do.

- Hàm RTPDestroy dùng để huỷ tất cả thông tin về phiên. Nó lấy lại context và giải phóng bộ nhớ.

6.2.4 Hàm truy cập thông tin thành viên

Các hàm RTP hỗ trợ truy cập thông tin của các thành viên trong mỗi phiên. Mỗi người dùng đã được xác định bằng một số thứ tự bởi thư viện, type peron. Không giống như SSRC có thể gây xung đột, số thứ tự là không đổi. Thông tin thành viên được truy cập thông qua id. Các bước:

1. Hàm CallBack: dùng khi sự kiện chắc chắn xuất hiện.Như thành viên mới, timeouts,… Hàm này dùng để cấp id cho người đó. Nó cũng cho phép ứng dụng truy cập thông tin thành viên dễ dàng.
2. ListIterators: cho phép ứng dụng lấy ra danh sách thành viên nhóm hiện tại.
3. Most Recent: thư viện cung cấp hai hàm để lấy ra thành viên gửi các gói tin RTPvà RCTP là RTPMostRecentRTPPerson và RTPMostRecentRTCPPerson.
4. Local. Các thành viên local thường được biết bằng lệnh unique identifier 0.
5. Finding. Thư viện cung cấp hàm RTPFindMember cho phép một ứng dụng lấy id thành viên dựa trên giá trị cảu trường SDES. Hàm này chấp nhận một context, trường SDES và giá trị. Nó cũng chấp nhận con trỏ tới một thành viên để khởi tạo ID cho thành viên đầu tiên. Hàm này rất hữu ích để tìm thành viên dựa vào CNAME. Khi có nhiều phiên, CNAME của người dùng trong các phiên gần đó rất cần cho ứng dụng.

Định danh id còn được dùng để truy cập tới các thông tin khác:

1. SDES. Hàm RTPMemberInforGetSDES() lấy id,trường SDES và một con trỏ bộ đệm. Thư viện sẽ copy mục SDES vào bộ đệm.
2. SR Infor. Đây là một hàm của host để lấy thông tin trong báo cáo gửi của thành viên. Hàm này có một dạng RTPMemberInfoGetX với X là một trong các giá trị (NTPStamp, RTPStamp, PktCount, OctCount), phù hợp với báo cáo. Nếu thành viên không phải là bên gửi, nó sẽ trả lại giá trị zero.
3. SSRC. Hàm RTPMemberInforGetSSRC(): trả về SSRC cho thành viên.
4. RR. Mỗi thành viên trong nhóm gửi báo các nhận cho mỗi bên gửi. Thư viện sẽ ghi lại tất cả các thông tin trong báo các nhận từ mọi thành viên nhóm. Để lấy thông tin từ báo cáo nhận, ứng dụng phải cung cấp id thành viên gửi báo cáo và SSRC cho bên gửi
5. Member Status: Hàm RTPMemberInforGetstatus(): trả về trạng thái thành viên.

Chương 7: Phân tích chương trình thực nghiệm

7.1. Phân tích chương trình

7.1.1. Phát biểu bài toán:

- Xây dựng chương trình cho phép truyền hình ảnh từ thiết bị thu như Webcam, Camera… tới các máy tính khác trong cùng mạng LAN.

7.1.2. Nền tảng thiết bị và hệ điều hành:

- Máy tính cấu hình từ Pen III trở lên.
- Hệ điều hành: Win XP.
- Hệ thống mạng LAN tốc độ 10Mbps trở lên.
- Tối thiểu 1 Webcam, Camera được cài trên máy gửi.

7.1.3. Cơ sở dữ liệu

- Dữ liệu đầu vào: hình ảnh từ thiết bị thu.
- Dữ liệu đầu ra: đồng bộ với dữ liệu đầu vào và gần như là tức thời.

7.1.4. Yêu cầu của hệ thống

Hình ảnh truyền với thời gian thực.

Chất lượng hình ảnh tốt

7.1.5. Ngôn ngữ lập trình:

Dùng ngôn ngữ C# trên nền .NET do có hỗ trợ các thư viện thực thi.

7.1.6. Các yêu cầu về chức năng của hệ thống

- Truyền hình ảnh từ 1 máy tính có gắn thiết bị thu hình ảnh đến nhiều máy tính trong mạng LAN.
- Quản lý được các máy tính cùng tham gia vào hệ thống.
- Xem được hình ảnh trước khi truyền.
- Tạm thời không truyền hình ảnh khi bên gửi không muốn gửi hình ảnh.
- Bật/ tắt thiết bị thu hình ảnh: Camera…..

7.1.7. Yêu cầu giao diện hệ thống

Thao tác chức năng thông qua các nút bấm và khung hình hiển thị dễ hiểu.

7.2. Thiết kế chương trình

7.2.1 Thiết kế chức năng

Stt	Chức năng	Đặc tả
1	Kết nối	Khi người dùng muốn truyền hình ảnh thì phải kết nối tới một nhóm có địa chỉ IP Multicast
2	Quản lý hình ảnh	Cho phép truyền hoặc dừng truyền hình ảnh
3	Quản lý Camera	Để bật/ tắt Camera

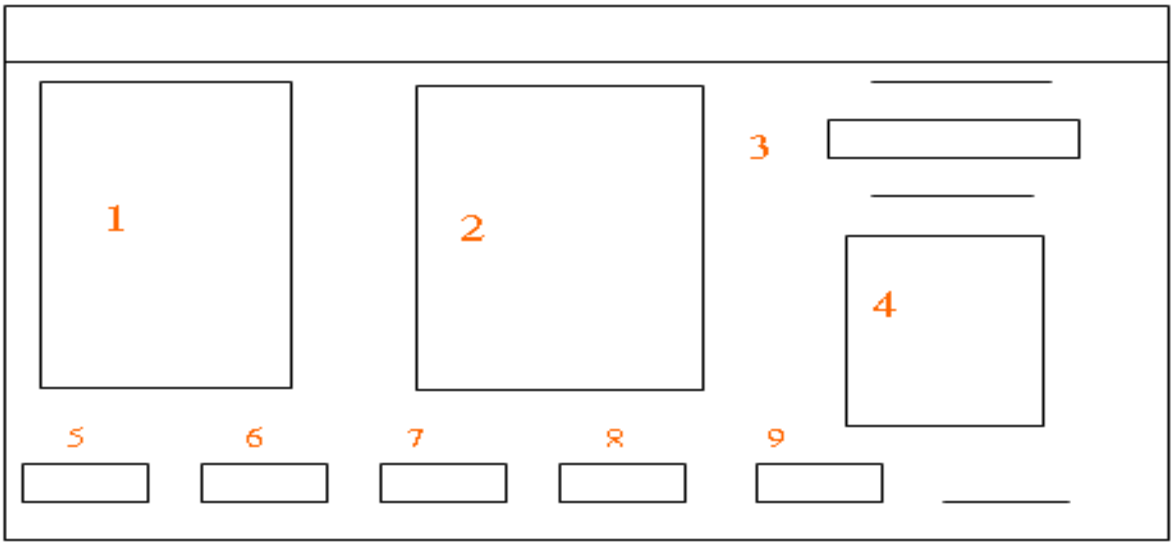
1. Kết nối
1.1 Kết nối vào nhóm 1.2 Hiển thị danh sách máy tính đã kết nối

2. Quản lý hình ảnh
2.1 Xem hình ảnh trước khi truyền 2.2 Truyền hình ảnh 2.3 Dừng truyền hình ảnh

3. Quản lý Camera
3.1 Bật Camera 3.2 Tắt Camera

7.2.2 Thiết kế giao diện

Giao diện chính của chương trình



(1): PictureBox Receive_ Hình ảnh nhận được

(2): PictureBox Send_ Hình ảnh thu từ camera

(3): TextBox_ Nhập địa chỉ nhóm Multicast

(4): ListBox_ Chứa danh sách các máy tính kết nối vào hệ thống

(5): Button _ Kết nối hoặc dừng kết nối trong từng trường hợp

(6): Button _ Gửi hình ảnh thu được từ Camera

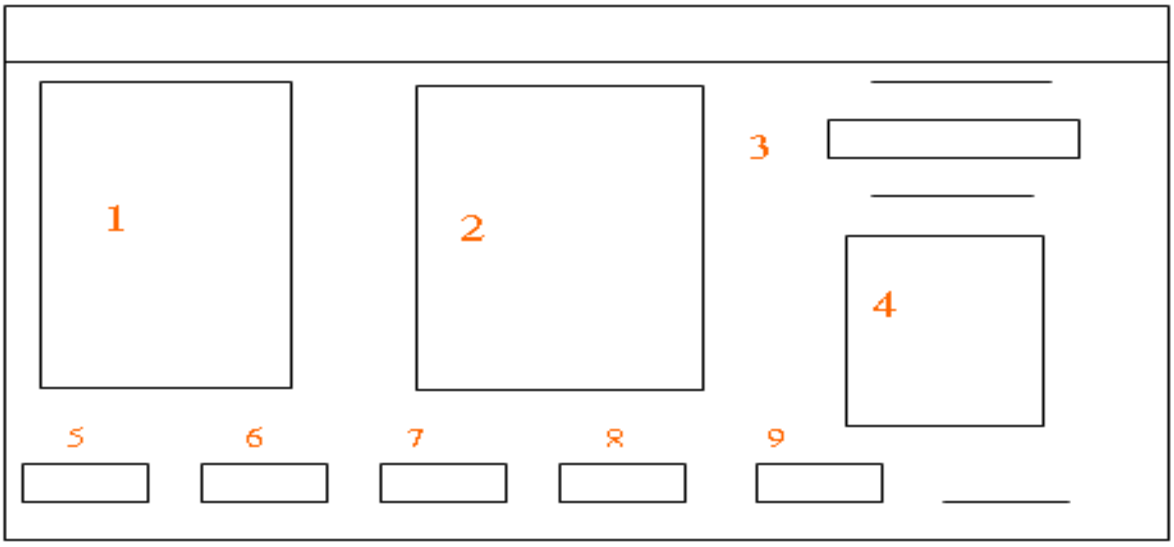
(7): Button_ Dừng gửi hình ảnh

(8): Button _ Bật Camera đã cài đặt trên thiết bịi gửi

(9): Button _ Tắt Camera

Các thay đổi về giao diện khi chương trình hoạt dộng

1. Người dùng nhấn nút Kết nối để tham gia vào nhóm.Tên máy tính tham gia được hiển thị trong Danh sách kết nôi.



TextBox (3) nhập địa chỉ IP multicast

ListBox (4) hiển thị danh sách các máy tính đã kết nối vào hệ thống.

Label của Button 5 chuyển từ Kết nối sang Ngắt.

2. Người gửi hình ảnh nhấn nút Bật Camera.

- Hình ảnh thu được từ Camera hiển thị trong PictureBox 2

3. Người gửi nhấn nút Gửi hình ảnh

- Hình ảnh từ PictureBox 2 và PictureBox 1 đồng nhất với nhau.

4. Nếu muốn tạm dừng gửi hình ảnh, nhấn nút Dừng gửi

- Hình ảnh thu từ Camera vẫn hiển thị trong PictureBox 2 nhưng không được truyền đi.

5. Nhấn nút Tắt Camera để không truyền hình ảnh

- PictureBox 2 không hiển thị hình ảnh,

7.2.3 Thiết kế Module

1.Kết nối

```
private void btnJoinLeave_Click(object sender, System.EventArgs e)
      {
         ep = new IPEndPoint(IPAddress.Parse(text_IP_Multicast.Text), 5000);
         if (btnJoinLeave.Text == "Kết nối")
         {
               HookRtpEvents(); // 1
               JoinRtpSession(Dns.GetHostName()); // 2

               //Thay đổi giao diện từ Kết nối sang Ngắt
               btnJoinLeave.Text = "Ngắt";
               btnSend.Enabled = true;
               button2.Enabled = true;
               text_IP_Multicast.Enabled = false;
         }
         else
         {
            Cleanup(); // 6

            btnJoinLeave.Text = "Kết nối";
            btnSend.Enabled = false;
            text_IP_Multicast.Enabled = true;
            button2.Enabled = false;
         }
      }
```

2. Bật Camera

```
private void button1_Click(object sender, EventArgs e)
      {
         iDevice = 0;
         OpenPreviewWindow();
      }
```

```
private void OpenPreviewWindow()
      {
         int iHeight = 320;
         int iWidth = 200;
          // Mở hình ảnh trong picturebox
         Hwnd = capCreateCaptureWindowA(iDevice.ToString(), (WS_VISIBLE
| WS_CHILD), 0, 0, 640, 480, pictureBox_sender.Handle.ToInt32(), 0);
         //   Kết nối với Webcam
         if (SendMessage(hHwnd, WM_CAP_DRIVER_CONNECT, iDevice, 0)
== 1)
         {
            //
            // Đặt mức độ hiển thị
            //
            SendMessage(hHwnd, WM_CAP_SET_SCALE, 1, 0);
            //
            // Đặt tốc độ hiển thị trên 1 giây
            //
            SendMessage(hHwnd, WM_CAP_SET_PREVIEWRATE, 66, 0);
            // Khởi động Camera
            SendMessage(hHwnd, WM_CAP_SET_PREVIEW, 1, 0);
            //  điều chỉnh khung cửa sổ vừa với picturebox
            //
            SetWindowPos(hHwnd, HWND_BOTTOM, 0, 0, iWidth, iHeight,
(SWP_NOMOVE | SWP_NOZORDER));
         }
         else
         {
            //  Lỗi kết nối thiết bị
            DestroyWindow(hHwnd);
         }
      }
```

3. Tắt Camera

```
 private void button3_Click(object sender, EventArgs e)
      {
         ClosePreviewWindow();
      }

  private void ClosePreviewWindow()
      {
         //
         //  Ngắt kết nối với Webcam
```

```
            //
            SendMessage(hHwnd, WM_CAP_DRIVER_DISCONNECT, iDevice,
0);
            //
            // Đóng Window
            //
            DestroyWindow(hHwnd);
        }
```

4. Gửi hình ảnh

```
private void btnSend_Click(object sender, System.EventArgs e)
        {
            timer1.Enabled = true;

        }
```

5. Dừng gửi hình ảnh

```
private void button2_Click(object sender, EventArgs e)
        {
            timer1.Enabled = false;
        }
```

7.2.4 Kết quả chương trình thực nghiệm

- Địa điểm: phòng F103 trường ĐH Dân lập Hải Phòng

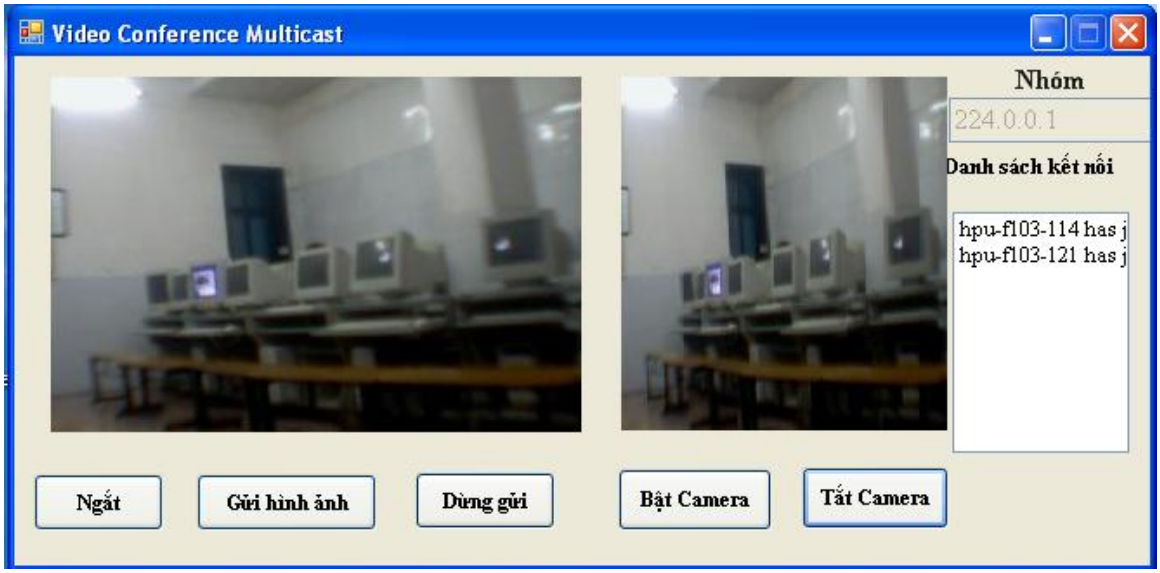
- Cấu hình thiết bị: Máy tính PenIV, Switch Planet 24 port, Lan 100Mbps

- Một số hình ảnh thực nghiệm:

Bước 1: Người dùng nhấn nút Kết nối để tham gia vào nhóm.Tên máy tính tham gia được hiển thị trong Danh sách kết nối.

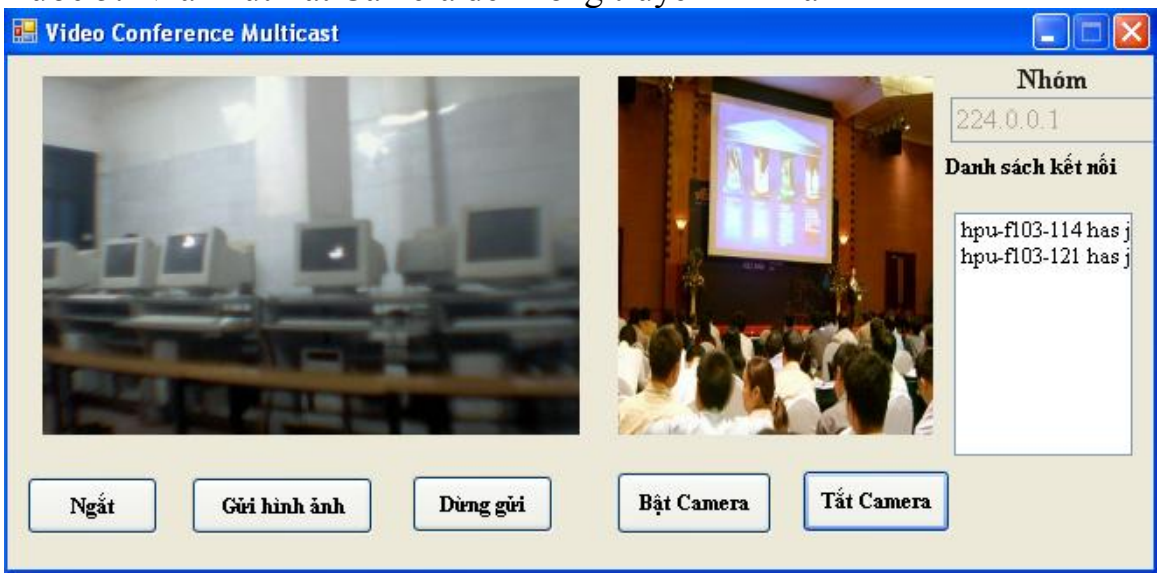
Bước 2: Người gửi hình ảnh nhấn nút Bật Camera.

Bước 3: Người gửi nhấn nút Gửi hình ảnh



Bước 4: Nếu muốn tạm dừng gửi hình ảnh, nhấn nút Dừng gửi

Bước 5: Nhấn nút Tắt Camera để không truyền hình ảnh



Tổng kết

1. Tự đánh giá khoá luận

- Trong đồ án này, em đã nghiên cứu được một số vấn đề:
  - Công nghệ Multicast
  - Giao thức RTP truyền tải hình ảnh thời gian thực.

a. Kết quả đạt được:

- Nghiên cứu khá sâu về công nghệ Multicast và giao thức RTP trong truyền tải hình ảnh thời gian thực.
- Xây dựng thành công chương trình có tính ứng dụng cao trong nhiều lĩnh vực như: giáo dục, hội nghị truyền hình…

b. Vấn đề còn tồn tại:

- Cần tìm hiểu thêm về C# để xây dựng thêm nhiều tiện ích chương trình.
- Giao diện chương trình đơn giản, tính chuyên nghiệp chưa cao.

2. Hướng phát triển của đề tài

- Tích hợp thêm Chat Voice, truyền Desktop… đáp ứng tốt hơn cho hệ thống hội nghị truyền hình.
- Có thể thương mại hoá sản phẩm.

Tài liệu tham khảo

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

2.http://www.csharphelp.com

3.http://www.codeproject.com

4.http://www.itgate.com.vn

5.http://www.cs.columbia.edu/irt/software/rtplib.