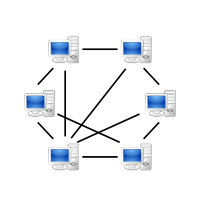
**Module P2P**

1. ***Ý tưởng và thực thi Module P2P:***
2. *Mô hình mạng Peer To Peer:*

Mô hình mạng Peer-to-Peer còn gọi là mạng đồng đẳng, là một mạng máy tính trong đó hoạt động của mạng chủ yếu dựa vào khả năng tính toán và băng thông của các máy tham gia chứ không tập trung vào một số nhỏ các máy chủ trung tâm như các mạng thông thường. Mạng đồng đẳng thường được sử dụng để kết nối các máy tính một cách trực tiếp với nhau thay vì các máy tính tương tác với một máy chủ cố định làm nhiệm vụ cung cấp dịch vụ như trong mô hình mạng truyền thống client-server. Một mạng đồng đẳng đúng nghĩa không có khái niệm mô hình khách chủ, tất cả các máy tham gia trong mạng đều bình đẳng và được gọi là một **peer node**, đóng vai trò đồng thời là máy chủ cung cấp dịch vụ cho những peer node khác, và là máy khách yêu cầu dịch vụ từ những peer node khác.



Mô hình mạng Peer-to-Peer (p2p)

Mạng đồng đẳng có rất nhiều ứng dụng bởi khả năng có thể mở rộng một cách dễ dàng và vô hạn của nó. Càng được mở rộng thì tài nguyên của mạng càng nhiều, và khả năng cung cấp tài nguyên của mạng càng lớn. Cách thức liên lạc trực tiếp của mạng P2P làm cho sự liên lạc trở nên bảo mật và riêng tư bởi vì kết nối chỉ diễn ra giữa hai peer ndoe, đồng thời chi phí cho chuyển giao dữ liệu giữa hai peer node ít tốn chi phí hơn so với mô hình client-server thông thường. Khái niệm đồng đẳng ngày nay được tiến hóa vào nhiều mục đích sử dụng khác nhau, không chỉ để trao đổi tệp mà còn khái quát hóa thành trao đổi thông tin giữa người với người, đặc biệt trong những tình huống hợp tác giữa một nhóm người trong cộng đồng. Ứng dụng thường gặp nhất là chia sẻ tập tin hoặc để truyền dữ liệu thời gian thực như điện thoại VoIP.

Trong khi hệ thống P2P trước đó đã được sử dụng trong nhiều ứng dụng, kiến trúc của mạng P2P chính thức được phổ biến bởi hệ thống chia sẻ file Napster vào năm 1999. Hệ thống P2P cho phép hàng triệu người sử dụng Internet kết nối trực tiếp, tạo nhóm và hợp tác với nhau để trở thành cỗ máy tìm kiếm người dùng được tạo ra bởi số lượng user tham gia, máy chủ ảo và hệ thống chia sẻ file. Trong mô hình này sử dụng mô hình máy chủ - máy khách cho một số tác vụ và mô hình đồng đẳng cho những tác vụ khác, những mô hình mạng kiểu này thuộc thế hệ mạng P2P thứ nhất. Trong khi đó, các mạng khác như Gnutella hay Freenet (thế hệ thứ 2) sử dụng mô hình đồng đẳng cho tất cả các tác vụ, nên các mạng này thường được xem như là mạng đồng đẳng đúng nghĩa (thực ra Gnutella vẫn sử dụng một số máy chủ để giúp các máy trong mạng tìm kiếm địa chỉ IP của nhau). Do đó mạng đồng đẳng tuỳ thuộc vào cấu trúc có thể chia ra làm 2 loại:

1. *Mô hình mạng đồng đẳng thuần tuý:*

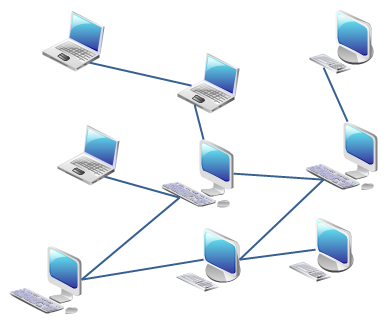
Mô hình mạng này có những đặc điểm đặc trưng như sau:

* + Các peer node vừa có vai trò là máy khách, vừa là máy chủ.
  + Không có máy chủ trung tâm quản lý mạng.
  + Không có máy định tuyến (bộ định tuyến) trung tâm và các peer node có khả năng định tuyến:

Trong mô hình này các peer node khi tham gia vào mạng cần đều có khả năng tìm vị trí của một peer node khác trong mạng dựa vào một thuật toán có được sử dụng để đánh địa chỉ và tìm thông tin về những peer node khác.

1. *Mạng đồng đẳng lai:*
   * Có một máy chủ trung tâm dùng để lưu trữ dữ thông tin của các máy trạm và trả lời các thông tin truy vấn này. Máy chủ trung tâm này có mục đích chính là xử lý và trả lời các thông tin truy vấn từ các nút mạng khác như tham gia mạng, rời mạng, cập nhật thông tin của nút mạng. Nút mạng này được gọi là Bootstrap Node. Vì sự cần thiết và tác dụng của Bootstrap Node nên Bootstrap Node luôn luôn phải được chạy trước tiên và chạy liên tục trong quá trình hoạt động của mạng.
   * Sử dụng các trạm định tuyến để xác định địa chỉ IP của các máy trạm.
2. *Những thách thức cần giải quyết khi tạo nên một mạng P2P:*

Thách thức của việc thiết kế một phần mềm hoạt động cho mạng P2P, đó là làm thế nào để các peer có thể giao tiếp được với nhau. Hình bên dưới minh họa một cấu trúc mạng không có tổ chức. Tất cả các peer đều kết nối với Internet, nhưng làm cách nào để những peer này có thể biết địa chỉ của nhau, cũng như là thế nào để chắc chắn rằng một peer giao tiếp đúng với peer mà nó cần?



Vấn đề cốt lõi trong một mạng, mà cụ thể ở đây là mạng P2P là làm sao để những Peer Node có thể tìm thấy và giao tiếp chính xác với Peer Node khác mà nó muốn cần phải được giải quyết. Để những peer trong mạng P2P có thể giao tiếp được với nhau, cần giải quyết hai vấn đề sau:

* + - Nhận dạng Peer trong mạng: một Peer Node trong mạng phải được phân biệt với các Peer Node khác và mang một định danh duy nhất. Khi đó một Peer Node muốn liên lạc với Peer Node khác sẽ sử dụng ID để liên lạc, đảm bảo được điều kiện liên lạc chính xác node mà nó muốn.
    - Tìm vị trí của Peer trong mạng: Một liên kết tồn tại giữa hai nút mạng khi một nút mạng biết vị trí của nút mạng kia. Do đó cần phải đưa ra giải pháp làm sao cho Peer Node này có thể chứa hoặc tìm kiếm được thông tin của những Peer Node khác nằm trong mạng, và sử dụng thông tin có được này tạo kết nối và liên lạc với những Peer Node còn lại.

1. *Ý tưởng giải quyết vấn đề trong đề tài:*

Trong ngữ cảnh nghiên cứu của đề tài này, nghiên cứu sẽ tập trung vào việc implement một phần mềm tạo sự tương tác giữa các peer trong cùng một mạng LAN nhỏ và số lượng peer node tham gia không lớn (khoảng 50 – 100 peer). Trong ngữ cảnh này, từng vấn đề nêu trên sẽ được giải quyết theo những ý tưởng sau.

* 1. *Giải quyết vấn đề thứ nhất – Định danh Peer Node:*
* Mỗi Peer Node tham gia vào mạng phải được gán một định danh duy nhất và định danh này sẽ được biết đến bởi các Peer Node khác. Điều này giúp một Peer Node khi muốn liên lạc với một Peer Node khác trong mạng thì chỉ cần gọi định danh của Peer Node đó ra và sử dụng thông tin của Peer Node đó để thực hiện kết nối, vì định danh là duy nhất nên đảm bảo rằng Peer Node liên lạc chính xác với node mà nó muốn.
* Trong phạm vi đề tài này là thực thi một thư viện phần mềm hiện thực hóa mô hình P2P trong cùng một mạng LAN nên để đơn giản thì chương trình sẽ lấy luôn địa chỉ IP của thiết bị trong mạng để làm định danh cho Peer Node. Vì địa chỉ IP của một thiết bị trong một mạng là duy nhất nên điều này thỏa mãn điều kiện thứ nhất là ID của Peer Node phải là duy nhất.
* Trong trường hợp Peer Node thay đổi IP thì IP trong một mạng LAN cũng là duy nhất, do đó khi trường hợp này xảy ra thì điều kiện trên vẫn được đảm bảo.
  1. *Vấn đề thứ hai – Tìm vị trí của một Peer Node trong mạng:*

Để giải quyết vấn đề này thì chương trình sử dụng cấu trúc mạng đồng đẳng lai với mô hình Bootstrap Node đã nêu ở trên. Bootstrap Node có nhiệm vụ lưu trữ thông tin của tất cả các Peer Node trong mạng. Bootstrap Node sẽ xử lý và trả lời các yêu cầu thông tin truy vấn với các Peer Node khi Peer Node phát sinh các sự kiện liên quan đến liên kết mạng. Đồng thời Bootstrap Node cũng thực hiện gửi các tác vụ đến Peer Node khi cần thiết để đảm bảo mạng hoạt động ổn định và các Peer Node luôn luôn có thông tin cập nhật về tất cả những Peer Node khác trong mạng.

Ý tưởng cơ bản của mô hình này như sau: một mạng P2P được hình thảnh bởi các nút mạng và liên kết nút mạng giữa chúng. Một Peer Node được gọi là liên kết với một Peer Node khác trong mạng khi Peer Node đó có chứa thông tin về định danh và vị trí của Peer Node khác. Quy luật này được áp dụng cho tất cả những Peer Node tham gia vào mạng, đó là Peer Node sẽ có tất cả các thông tin về các Peer Node khác có trong mạng. Vấn đề cần giải quyết ở đây là làm sao để Peer Node có thể có được những thông tin này, và phải đảm bảo rằng những thông tin này luôn được cập nhật liên tục khi có một Peer Node tham gia mạng hoặc một Peer Node rời mạng.

* Khi mới tham gia vào mạng, một Peer Node không thể biết chính xác nó phải liên kết với nút mạng nào để có thể có được những thông tin này vì địa chỉ IP tự phát sinh khi kết nối vào mạng LAN. Điều này được giải quyết khi sử dụng Bootstrap Node có địa chỉ IP cố định, cung cấp dịch vụ xử lý yêu cầu truy vấn thông tin Peer Node có trong mạng. Bootstrap Node được chạy trước tiên, cung cấp dịch vụ và hoạt động cho đến khi mạng dừng lại để liên tục xử lý việc truy vấn thông tin cho những Peer Node tham gia vào mạng, cập nhật thông tin khi có một Peer Node rời mạng hoặc cập nhật lại thông tin của node đó. Mỗi khi có một Peer Node tham gia vào mạng, địa chỉ IP Bootstrap Node sẽ được Peer Node này sử dụng để gửi yêu cầu tham gia vào mạng (thông điệp JOIN) đến dịch vụ mà Bootstrap Node cung cấp đồng thời Peer Node này cũng cung cấp một dịch vụ lắng nghe sự kiện thay đổi của mạng từ Bootstrap Node.
* Sau khi nhận được yêu cầu từ Peer Node, Bootstrap Node sẽ xử lý thông tin về Peer Node gửi tới, lấy thông tin của Peer Node đó và chèn vào ***bảng định tuyến (routing table)***, bao gồm tất cả những thông tin về Peer Node có trong mạng, đến đây thông tin Peer Node đã có trong bảng định tuyến. Tiếp đến, Bootstrap Node sẽ gửi bảng định tuyến này cho Peer Node vừa phát sinh yêu cầu qua dịch vụ mà Peer Node đã mở để lắng nghe sự kiện thay đổi của mạng, Peer Node sẽ ghi nhận lại tất cả những thông tin này và tạo ra một bảng định tuyến ***clone*** từ Bootstrap Node. Đến lúc này Peer Node đã chính thức trở thành một node trong mạng P2P và Peer Node sẽ sử dụng bảng định tuyến để thực hiện kết nối và tương tác với các Peer Node khác trong mạng và cung cấp các chức năng chính của chương trình.
* Tuy rằng phiên giao tiếp của Peer Node gửi yêu cầu tham gia mạng đến lúc này đã xong (nếu chỉ xét Peer Node). Nhưng đến lúc này công việc Bootstrap Node vẫn chưa dừng lại, nó tiếp tục gửi broadcast đến tất cả những Peer Node có trong bảng định tuyến thông điệp một Peer Node vừa được thêm vào mạng (thông điệp NODE\_ADD) và yêu cầu các Peer Node phải cập nhật lại bảng định tuyến của riêng mình cùng với thông tin mà Bootstrap Node kèm theo. Sau khi thực hiện công việc này xong thì yêu cầu tham gia mạng gửi từ Peer Node trên Bootstrap Node mới thật sự kết thúc.
* Khi một Peer Node muốn rời mạng, nó cũng thực hiện quy trình giống như trên nhưng nó sẽ gửi yêu cầu khác (thông điệp LEAVE) đến Bootstrap Node. Khi Bootstrap Node nhận được thông điệp LEAVE từ Peer Node, nó sẽ thực hiện truy vấn trong bảng định tuyến, tìm thông tin về Peer Node được gửi lên và xóa thông tin đó khỏi bảng định tuyến. Sau đó Bootstrap Node sẽ gửi broadcast đến tất cả các Peer Node trong mạng thông điệp NODE\_LEAVE để tất cả các Peer Node có thể cập nhật lại bảng định tuyến của mình. Sau khi tác vụ này diễn ra thì Peer Node sẽ không còn là một node thành viên của mạng nữa vì mọi thông tin liên lạc với Peer Node đã không còn trong bảng định tuyến của các Peer Node khác nữa.
* Khi một Peer Node thay đổi thông tin của mình (thay đổi IP) thì Peer Node sẽ phát sinh yêu cầu đến Bootstrap Node (thông điệp UPDATE) và thực hiện quy trình giống như trên. Sau khi Bootstrap Node nhận được thông điệp này, nó sẽ truy vấn trong bảng định tuyến của mình thông tin của Peer Node vừa được gửi lên và cập nhật lại bảng định tuyến. Sau đó, Bootstrap Node sẽ phát sinh broadcast đến tất cả những Peer Node có trong mạng yêu cầu cập nhật Peer Node (thông điệp NODE\_UPDATE). Các Peer Node sẽ cập nhật lại bảng định tuyến của riêng mình.

Ý tưởng trên đã thực thi được yêu cầu thứ hai là làm thế nào để một Peer Node trong mạng có thể tìm thấy được Peer Node khác. Ý tưởng này cung cấp một giải pháp đảm bảo các Peer Node luôn luôn có được thông tin cập nhật về trạng thái của tất cả các Peer Node trong mạng. Từ đó khi một Peer Node muốn liên lạc với một Peer Node khác, với bảng định tuyến nó có thể tạo kết nối và liên lạc một cách dễ dàng.

Ý tưởng này tạo nên một mô hình mạng đồng đẳng lai với Bootstrap Node là một máy chủ dữ liệu trung tâm chuyên xử lý và thực hiện các thông tin truy vấn từ các Peer Node trong mạng, đảm bảo rằng dữ liệu của tất cả các thành viên trong mạng đều được cập nhật mới nhất đến bảng định tuyến và thông tin này cũng được cập nhật đến tất cả bảng định tuyến của các Peer Node trong mạng. Bootstrap Node ngoài đóng vai trò xử lý và cung cấp thông tin về các Peer Node trong mạng thì không còn vai trò gì khác nữa. Trong mô hình mạng kiểu này, Bootstrap Node chỉ đóng vai trò xử lý, phân phối dữ liệu, cập nhật thông tin về tình trạng mạng P2P cho các Peer Node để khởi tạo và duy trì hoạt động của mạng. Còn các tác vụ chính thì Peer Node sau khi gia nhập vào mạng sẽ sử dụng mạng P2P để thực hiện tác vụ như chia sẻ file hoặc và VoIP.

1. *Các thành phần trong mạng P2P:*

Phần ý tưởng đã nêu rõ hai thực thể chính trong mô hình được thực thi, đó là Bootstrap Node và Peer Node. Sự tương tác giữa các Bootstrap Node và Peer Node tạo cơ sở hình thành và mở rộng mạng các node. Sau khi đã hình thành nên mạng lưới, các Peer Node sẽ kết nối các Peer Node khác mà nó muốn tương tác để thực hiện tác vụ khác. Phần trên cũng đã nêu bật rõ vai trò cơ bản của Bootstrap Node và Peer Node trong mạng cùng những yêu cầu cơ bản của những thực thể này. Phần này sẽ đi vào sâu hơn cấu trúc và các yêu cầu cần thiết phải có của Bootstrap Node và Peer Node.

1. *Bootstrap Node:* Như đã nêu ở trên, Bootstrap Node là máy chủ trung tâm đảm trách vai trò tiếp nhận yêu cầu truy vấn dữ liệu, phân phối và cập nhật thông tin tất cả các node trong mạng đến tất cả các Peer Node, đảm bảo tình trạng mới nhất của mạng đến với các Peer Node. Để làm được điều này thì Bootstrap Node phải:

* Có cơ chế lưu trữ, cập nhật thông tin của tất cả các Peer Node có trong mạng khi có yêu cầu tham gia mạng, update thông tin, hoặc rời mạng của một Peer Node nào đó. Cơ chế này được thực thi bằng cách Bootstrap Node tạo ra một bảng định tuyến và lưu tất cả các thông của các Peer Node còn active (còn hoạt động và chưa rời khỏi mạng). Khi có yêu cầu phát sinh từ bất kỳ một Peer Node nào trong mạng, thông tin của Peer Node sẽ được cập nhật vào bảng định tuyến này. Trong chương trình bảng định tuyến này được thực thi bằng file text định dạng JSON, có tên là “***list\_peer.json***”.
* Cung cấp dịch vụ lắng nghe và xử lý các yêu cầu từ các Peer Node như tham gia, cập nhật thông tin và rời mạng. Trong chương trình này, Bootstrap Node sẽ lắng nghe và xử lý yêu cầu ở port 6868. Mỗi khi một Peer Node muốn phát sinh yêu cầu thì nó sẽ phát sinh vào cổng này và Bootstrap Node sẽ xử lý.
* Cơ chế ***broadcast*** đến tất cả các Peer Node có trong mạng về một sự kiện nào xảy ra với bất kỳ Peer Node trong mạng. Trong chương trình, trong pha giao tiếp giữa Peer Node và Bootstrap Node, sau khi xử lý yêu cầu của Peer Node, Bootstrap Node sẽ phát sinh một thông điệp tương đương với yêu đã được xử lý cùng thông tin đến tất cả các Peer Node trong mạng.

1. *Peer Node:* là thành phần cốt lõi tạo nên mạng P2P và đảm trách vai trò thực thi mục đích chính của chương trình. Để trở thành một node trong mạng thì cần phải đảm bảo những thành phần sau đây:
   * Thực thi phương thức giao tiếp với Bootstrap Node để thực hiện phương thức giao tiếp đã được quy định trong chương trình (sẽ được diễn tả ở phần kế tiếp). Phương thức đó bao gồm các thực thi về yêu cầu tham gia (JOIN), yêu cầu cập nhật (UPDATE), yêu cẩu rời mạng (LEAVE).
   * Cơ chế quản lý thông tin của các Peer Node khác trong mạng nhận được bảng định tuyến từ Bootstrap Node sau khi thực hiện yêu cầu JOIN.
   * Cung cấp dịch vụ lắng nghe sự thay đổi trạng thái của mạng khi bất kỳ Peer Node nào trong mạng thay đổi thông tin và Bootstrap Node gửi broadcast đến. Trong chương trình của đề tài, Peer Node sẽ tạo một dịch vụ xử lý yêu cầu của Bootstrap Node ở port 8686.
2. *Phương thức giao tiếp giữa các thành phần trong mạng:*

Các node trong mạng cần phải giao tiếp được với nhau để có thể thông báo tình trạng và dữ liệu cần trao đổi, cụ thể ở đây là sự trao đổi thông tin giữa Bootstrap Node và Peer Node. Cách thức giao tiếp giữa các node trong chương trình được sử dụng bằng thông điệp. Mỗi một thông điệp mang một yêu cầu tác vụ và thông tin thông điệp mà hai bên hiểu được và dựa vào tác vụ đó bên nhận sẽ có cách thức đáp ứng tương ứng.

Một quá trình bắt đầu gửi thông điệp đến khi kết thúc gửi thông điệp giữa hai node được gọi là một ***phiên giao tiếp***.

Một phiên giao tiếp bao gồm những pha:

* Khi muốn giao tiếp, client node sẽ phát sinh một thông điệp yêu cầu servere node bắt đầu một phiên giao tiếp.
* Server node chấp nhận và gửi lại thông điệp chấp nhận phiên giao tiếp.
* Tiếp đến, client node sẽ gửi thông điệp đến server node. Server node sẽ nhận thông điệp và giữ lại.
* Client node gửi yêu cầu kết thúc session giao tiếp.
* Server node chấp nhận và gửi lại thông điệp chấp nhận kết thúc phiên giao tiếp.

Sau khi phiên giao tiếp kết thúc, server node sẽ xử lý thông điệp. Nếu có phản hồi lại client node thì server node sẽ tạo ra một phiên giao tiếp với node mà nó cần và quy trình lặp lại như quy trình vừa trên và server node lúc này sẽ đóng vai trò là client node yêu cầu phiên giao tiếp.

1. *Cách thức gửi thông điệp giữa các node trong mạng:*

* Mỗi khi một node muốn giao tiếp với một node khác, nó sẽ thiết lập một phiên gửi message tới node mà nó muốn giao tiếp. Node nhận thông điệp sẽ đóng vai trò là server node và node gửi thông điệp sẽ đóng vai trò là client node. Quá trình thiết lập phiên gửi thông điệp sẽ được diễn ra bởi client node và server node bằng các request và respone.
* ***Request*** là một yêu cầu được gửi từ client node lên server node thông báo cho server node biết rằng client node yêu cầu một tác vụ giao tiếp với server node. Request trong chương trình được xây dựng trên định dạng text, nghĩa là request này là một chuỗi từ mà client node gửi lên cho server node một cách riêng biệt, khi server node gặp những từ này thì theo quy định server node sẽ có những cách đáp ứng phù hợp. Có các loại request sau:
  + START\_MSG (“start\_msg”): gửi yêu cầu tác vụ thông báo cho server node biết rằng client node bắt đầu phiên giao dịch gửi message. Sau khi nhận được request này, server node sẽ chấp nhận và gửi respone SERVER\_OK về client. Sau đó client sẽ gửi message và server sẽ chấp nhận message.
  + END\_MSG (“end\_msg”): gửi yêu cầu tác vụ thông báo cho server node biết rằng client node đã gửi xong message và muốn kết thúc phiên gửi message. Sau khi nhận được request này, server node gửi respone thông báo SERVER\_OK cho client node. Tiếp đó, server node sẽ xử lý thông điệp mà client node vừa gửi.
  + Sau khi phiên giao tiếp thông điệp giữa server node và client node được thiết lập, client node sẽ dùng request sẽ tiếp tục gửi nội dung của thông điệp lên server node.
* Respone là một đáp trả từ server node đến client node thông báo rằng server node đã chấp nhận request của client node. Respone cũng được xây dựng dựa trên định dạng text giống như request, khác ở đây là cách thức sử dụng của repone khác với request.
  + SERVER\_OK (“ok”): thông điệp đáp trả từ server node tới client node rằng yêu cầu của client node đã được chấp nhận bởi server node.
* Hình dưới đây mô tả một phiên giao tiếp thông điệp đầy đủ giữa các node, node gửi request yêu cầu thiết lập phiên gửi thông điệp là client node và node đáp ứng yêu cầu là server node.



1. *Thông điệp (message):*

Thành phần quan trọng trong phiên giao tiếp và trao đổi thông tin giữa các node chính là thông điệp. Thông điệp là một yêu cầu tác vụ và thông tin về tác vụ đó được tạo ra theo một quy định từ trước được hiểu bởi tất cả các node trong mạng, và các node sẽ dựa trên đó để xử lý các thông điệp. Trong chương trình thông điệp được gởi sau khi phiên giao tiếp được thiết lập.

Thông điệp trong chương trình có định dạng json. Thông điệp gồm 2 thành phần: tác vụ thông điệp (msg\_type) và nội dung thông điệp (msg\_data). Cấu trúc thông điệp như sau:

{“msg\_type” : “loại thông điệp”, “msg\_data” : “nội dung thông điệp”}

1. *Các loại message (msg\_type):*

Loại message được mô tả bởi phần đầu của thông điệp – phần “msg\_type

”, mô tả tác vụ được yêu cầu của client node lên server node. Trong đặc tả của chương trình sử dụng các loại thông điệp sau:

* Thông điệp JOIN (“join”): thông điệp này thông báo yêu cầu tác vụ một Peer Node muốn tham gia vào mạng. Khi một Peer Node muốn tham gia vào mạng, nó sẽ gửi thông điệp này đến Bootstrap Node thông báo rằng nó muốn tham gia vào mạng.
* Thông điệp LEAVE (“leave”): thông điệp này thông báo yêu cầu tác vụ một Peer Node muốn rời khỏi mạng đến Bootstrap Node.
* Thông điệp UPDATE (“update”): trong chương trình sau khi một Peer Node trở thành một node trong mạng thì sẽ sử dụng địa chỉ IP của mình để làm định danh và liên lạc. Do đó nếu thông tin Peer Node có thay đổi thì Peer Node sẽ sử dụng thông điệp này để thông báo cho Bootstrap Node.
* Thông điệp TRANSFER\_LIST (“transfer\_list”): thông điệp này được phát sinh bởi Bootstrap Node đến Peer Node sau khi nhận được thông điệp JOIN từ Peer Node đó. Thông điệp này thông báo với Peer Node rằng Bootstrap Node đã xử lý xong thông điệp JOIN của Peer Node đó, đã thêm Peer Node đó vào bảng định tuyến và nội dung thông điệp mang danh sách bảng định tuyến.
* Thông điệp NODE\_ADD (“node\_add”): thông điệp này được Bootstrap Node gửi đến cho tất cả các Peer Node trong mạng thông báo rằng có một node vừa được thêm vào mạng. Và nội dung của thông điệp chính là thông tin node vừa thêm. Sau khi nhận được thông điệp, các Peer Node sẽ xử lý và thêm thông tin Peer Node vào trong bảng định tuyến của mình.
* Thông điệp NODE\_UPDATE (“node\_update”): thông điệp này được Bootstrap Node gửi đến cho tất cả các Peer Node trong mạng thông báo rằng có một node vừa được cập nhật thông tin. Và nội dung thông điệp là thông tin node vừa update. Các Peer Node sẽ tìm trong bảng định tuyến của mình và cập nhật lại thông tin.
* Thông điệp NODE\_LEAVE (“node\_leave”): thông điệp này được Bootstrap Node gửi đến cho tất cả các Peer Node trong mạng thông báo rằng có một node vừa rời khỏi mạng. Nội dung thông điệp là thông tin node vừa rời đi. Các Peer Node sẽ tìm trong bảng định tuyến của mình và xóa thông tin node đó đi.
* Thông điệp PING (“ping”): thông điệp này được phát sinh từ một node đến một node kiểm tra Peer Node có đang hoạt động hay không.

1. *Nội dung message (msg\_data):*

Nội dung của thông điệp được mô tả bởi phần sau của thông điệp – phần “msg\_data”. Nội dung thông điệp bao gồm thông tin cần được gửi bởi client node đến server node. Trong chương trình nội dung thông điệp là danh sách thông tin các Peer Node được lưu dưới định dạng JSON. Mỗi Peer Node đều có thông tin của riêng mình gọi là thông tin Peer Node. Trong chương trình thông tin Peer Node gồm có địa chỉ IP của Peer Node và tên người dùng (user name) người dùng đặt cho node. Tên người dùng này được sử dụng trong địa chỉ SIP. Nội dung của thông điệp giống như sau:

{"list\_peer":[{"address":"192.168.0.104","username":"SangNguyen"}]}

1. *Các pha giao tiếp giữa các thành phần trong mạng:*

Phần bên trên đã mô tả các message được dùng để giao tiếp giữa các node trong mạng. Phần này sẽ trình bày rõ chi tiết tác vụ của từng message.

1. *Message JOIN – TRANSFER\_LIST:* trong quá trình giao tiếp thông điệp này, Peer Node muốn tham gia vào mạng sẽ gửi thông điệp JOIN đến Bootstrap Node. Quá trình cụ thể như sau:

* Peer Node muốn tham gia vào mạng gửi message JOIN đến Bootstrap Node.
* Bootstrap Node xử lý thông điệp, lấy danh sách thông tin Peer Node trong nội dung thông điệp, thêm thông tin vào trong bảng định tuyến.
* Bootstrap Node phát sinh thông điệp TRANSFER\_LIST đến Peer Node kèm theo bảng định tuyến vào nội dung thông điệp.
* Peer Node nhận thông điệp trên, xử lý thông điệp và lấy bảng định tuyến được gửi từ Bootstrap Node, lưu vào trong dữ liệu của mình. Đến lúc này Peer Node đã trở thành một node trong mạng.
* Đến lúc này phiên giao tiếp giữa Peer Node gửi thông điệp JOIN với Bootstrap Node đã kết thúc.
* Bootstrap Node sẽ broadcast message NODE\_ADD đến tất cả các node trong mạng để thông báo rằng có một Peer Node vừa tham gia vào mạng cùng với nội dung thông điệp là thông tin Peer Node đó.
* Các Peer Node nhận được broad cast NODE\_ADD từ Bootstrap Node sẽ lấy thông tin Peer Node vừa được thêm vào mạng thêm vào bảng định tuyến của mình.



1. *Message UPDATE:* thông điệp này được phát sinh giữa Peer Node và Bootstrap Node để thông báo rằng thông tin Peer Node được cập nhật. Các hoạt động cụ thể như sau:
   * Peer Node gửi message UPDATE kèm theo nội dung là thông tin Peer Node cập nhật đến Bootstrap Node.
   * Bootstrap Node nhận message và lấy thông tin của Peer Node cập nhật ra từ nội dung message. Sau đó Bootstrap Node truy vấn trong bảng định tuyến của mình tìm thông tin Peer Node được cập nhật và cập nhật thông tin.
   * Bootstrap Node broadcast message NODE\_UPDATE đến tất cả các Peer Node trong mạng kèm theo thông tin của Peer Node được cập nhật.
   * Các Peer Node nhận được broadcast sẽ xử lý thông tin, lấy thông tin của Peer Node được cập nhật, truy vấn trong bảng định tuyến của mình thông tin Peer Node đó và cập nhật.
   * Đến đây thông tin Peer Node đã được cập nhật.



1. *Message LEAVE:* thông điệp này được phát sinh giữa Peer Node đến Bootstrap Node khi Peer Node muốn rời khỏi mạng. Các hoạt động diễn ra như sau:
   * Peer Node gửi message LEAVE kèm theo nội dung là thông tin Peer Node đến Bootstrap Node.
   * Bootstrap Node nhận message và lấy thông tin của Peer Node từ nội dung message. Sau đso Bootstrap Node sẽ truy vấn trong bảng định tuyến của mình để tìm thông tin Peer Node muốn rời mạng. Sau khi tìm thấy Bootstrap Node sẽ xóa thông tin Peer Node ra khỏi bảng định tuyến.
   * Bootstrap Node broadcast message NODE\_LEAVE đến tất cả Peer Node trong mạng kèm theo thông tin của Peer Node muốn rời khỏi mạng.
   * Các Peer Node nhận được broadcast sẽ xử lý thông tin, lấy thông tin của Peer Node muốn rời mạng, truy vấn trong bảng định tuyến của mình thông tin Peer Node và xóa khỏi bảng định tuyến.
   * Đến lúc này Peer Node đã thực sự rời khỏi mạng.



1. *Message NODE\_ADD:* thông điệp này được gửi từ Bootstrap Node đến Peer Node trong quá trình broadcast message sau khi một Peer Node tham gia vào mạng. Các hoạt động diễn ra như sau:
   * Bootstrap Node sẽ gửi message NODE\_ADD cùng thông tin Peer Node tham gia vào mạng đến Peer Node.
   * Peer Node nhận được message này sẽ lấy thông tin Peer Node trong thông điệp ra, thêm vào bảng định tuyến của mình.



1. *Message NODE\_UPDATE:* thông điệp này được gửi từ Bootstrap Node đến Peer Node trong quá trình broadcast message sau khi một Peer Node muốn cập nhật thông tin của mình. Các hành động tương tự như broadcast message NODE\_ADD.
   * Bootstrap Node sẽ gửi message NODE\_UPDATE cùng thông tin Peer Node cập nhật đến Peer Node.
   * Peer Node sẽ xử lý message và sẽ lấy thông tin muốn cập nhật trong message, truy vấn thông tin Peer Node cập nhật trong bảng định tuyến và cập nhật thông tin Peer Node đó.



1. *Message NODE\_LEAVE:* thông điệp này đượcgửi từ Bootstrap Node đến Peer Node trong quá trình broadcast message sau khi một Peer Node muốn rời khỏi mạng. Các hành động tương tự như broadcast message NODE\_ADD.
   * Bootstrap Node sẽ gửi message NODE\_LEAVE cùng thông tin Peer Node muốn rời khỏi mạng đến Peer Node.
   * Peer Node sẽ xử lý message và sẽ lấy thông tin Peer Node muốn rời khỏi mạng trong message, truy vấn thông tin Peer Node muốn rời khỏi mạng trong bảng định tuyến và xóa thông tin Peer Node đó.



1. *Message PING:* thông điệp PING được gửi giữa các Peer Node với nhau để kiểm tra tình trạng của Peer Node. Quy trình diễn ra như sau:
   * Peer Node muốn kiểm tra Peer Node khác phát sinh message PING.
   * Peer Node phát message chờ Peer Node nhận message trong 30s.
   * Nếu có sự hồi đáp từ Peer Node nhận message gửi message PING lại cho Peer Node phát message thì quá trình ping thành công.
   * Nếu trong thời gian chờ mà Peer Node nhận message không có phản hồi lại cho Peer Node phát message thì quá trình ping sẽ có kết quả là time out.
   * Các hình sequence dưới đây mô tả về hai quá trạng thái ping của Peer Node.



PING thành công



PING không thành công

1. *Ưu điểm và nhược điểm của mô hình:*
2. *Ưu điểm:*

* Mô hình dễ thực thi vì có thiết kế đơn giản.
* Cách thức giao tiếp thông điệp dễ dàng, đơn giản.
* Một node tham gia vào mạng thì sẽ có thể liên lạc ngay tức thời với các node khác trong mạng vì Peer Node đã sao chép bảng định tuyến từ Bootstrap Node.
* Các tình trạng của mạng ngay lập tức được cập nhật tới các Peer Node thông qua Bootstrap Node nên Peer Node luôn luôn biết được tình trạng mạng hiện thời.

1. *Nhược điểm:*

* Vì phải sử dụng Bootstrap Node cho tác vụ quản lý mạng nên vẫn phải phụ thuộc vào Bootstrap Node. Nếu Bootstrap Node không hoạt động hoặc hoạt động không ổn định thì mạng sẽ bị ảnh hưởng theo.
* Khi mạng trở nên lớn với nhiều Peer Node tham gia hơn thì cơ chế sử dụng bảng định tuyến và update bảng định tuyến không khả thi vì công việc này chiếm rất nhiều tài nguyên xử lý cũng như băng thông của mạng.

1. *Hướng phát triển và cách khắc phục:*

* Để giải quyết vấn đề không phụ thuộc vào Bootstrap Node nữa, mạng sẽ phát triển thành mạng p2p đúng nghĩa, nghĩa là không có nút xử lý trung tâm nữa. Thay vào đó các Peer Node đều có khả năng tự định tuyến.
* Để thực thi mô hình này mô hình sẽ phát triển và sử dụng kỹ thuật distributed hash table (DHT). Với mô hình này, các Peer Node sẽ có khả năng tự định tuyến và mỗi tác vụ sẽ sử dụng tài nguyên mạng thấp hơn.