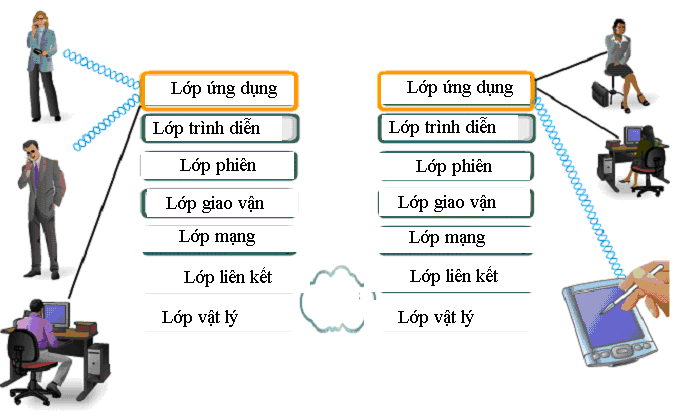
**Truyền thông trong hệ thống phân tán**

Trao đổi thông tin giữa các tiến trình là trọng tâm của tất cả các hệ thống phân tán, do đó cần phải nghiên cứu kỹ lưỡng cách thức các tiến trình trao đổi thông tin với nhau. Thực chất trao đổi thông tin trong hệ thống phân tán là chuyển thông điệp do  mạng máy tính đảm nhiệm, quá trình đó phức tạp hơn rất nhiều so với việc trao đổi thông tin trên một máy tính. Các hệ thống phân tán hiện đại bao gồm hàng triệu tiến trình trao đổi thông tin với nhau qua mạng Internet không tin cậy, nếu không thay đổi các phương thức truyền thông nguyên thủy thì sẽ rất khó phát triển các ứng dụng phân tán. Về bản chất, trao đổi thông tin giữa các tiến trình vẫn sử dụng các giao thức truyền tin truyền thống đã qui định trong từng lớp mạng. Các giao thức này được ứng dụng để xây dựng các mô hình truyền tin khác như Gọi thủ tục từ xa (RPC), gọi đối tượng từ xa (RMI),  phần mềm trung gian hướng thông điệp (MOM). Mô hình truyền tin đầu tiên trong hệ thống phân tán là RPC, bản chất của nó là ẩn những thủ tục phức tạp trong việc truyền thông điệp và đó là cách lý tưởng trong các ứng dụng khách/chủ. Về sau, mô hình này được cải tiến dựa trên việc cài đặt các đối tượng phân tán.

1. Cơ sở truyền thông

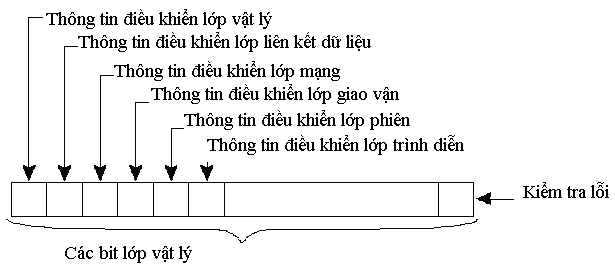
Trước khi thảo luận về truyền thông trong các hệ thống phân tán chúng ta sẽ nhắc lại một số kiến thức cơ bản về các giao thức mạng và sau đó sẽ đề cập đến một số cách tiếp cận áp dụng chúng  để giải quyết vấn đề truyền thông trong các hệ thống phân tán.

* 1. Giao thức mạng
* Các tiến trình trong hệ thống phân tán không sử dụng chung bộ nhớ, do đó việc trao đổi thông tin phải dựa hoàn toàn bằng phương thức truyền thông điệp.
* Khi một tiến trình A muốn trao đổi thông tin với tiến trình B, nó tạo một thông điệp trong vùng nhớ riêng của mình và thực hiện lời gọi hệ thống, khi đó hệ điều hành sẽ thực hiện chức năng chuyển thông điệp đó đến tiến trình B qua mạng.
* Thực tế quá trình này khá phức tạp bởi trong hệ thống phân tán có thể có các máy tính thuộc các nhà sản xuất khác nhau và sử dụng tiêu chuẩn mã hóa thông tin khác nhau.
* Để khắc phục vấn đề này, tổ chức chuẩn hóa Quốc tế ISO đã đưa ra mô hình liên kết hệ thống mở (OSI). Mặc dù các giao thức trong mô hình OSI ít được sử dụng, tuy nhiên đó là mô hình khá tốt để hiểu về mạng máy tính. Mô hình OSI được phân thành 7 tầng, mỗi tầng bao gồm các giao thức qui định khuôn dạng dữ liệu và các thủ tục xử lý như cách gửi/nhận đơn vị dữ liệu, cách xử lý khi xảy ra lỗi. Xét trên khía cạch cách thiết lập liên kết giữa các thực thể truyền thông người ta phân ra hai loại giao thức:
* Giao thức có liên kết: Cần phải thiết lập liên kết trước khi truyền dữ liệu, sau khi truyền xong thì phải hủy bỏ liên kết.
* Giao thức không liên kết: Không cần phải thiết lập liên kết khi truyền dữ liệu



Mỗi tầng trong mô hình OSI thực hiện một số chức năng nhất định:

* Tầng ứng dụng: Cung cấp giao diện phục vụ cho người sử dụng và các ứng dụng khác
* Tầng trình diễn: Thực hiện mã hóa/giải mã, nén/giải nén và bảo mật dữ liệu.
* Tầng phiên: Tạo ra các phiên làm việc.
* Tầng giao vận: Tạo liên kết giữa đầu cuối với đầu cuối, điều khiển tốc độ truyền dữ liệu, xử lý lỗi truyền tin.
* Tầng mạng: Quản lý địa chỉ logic của các đối tượng tham gia vào mạng, tìm đường đi tốt nhất cho mỗi gói tin.
* Tầng liên kết: Thiết lập liên kết giữa hai thiết bị vật lý kề cạnh nhau
* Tầng vật lý: Biến đổi các bit dữ liệu thành các tín hiệu phù hợp với môi trường truyền dẫn và thực hiện thu phát các tín hiệu đó.

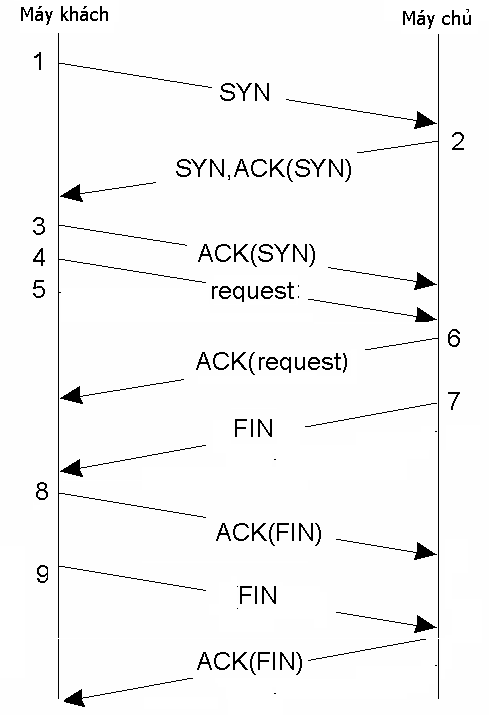


Khi một tiến trình trên máy tính muốn gửi dữ liệu cho tiến trình trên máy tính khác, nó tạo một bản tin tại tầng ứng dụng và bản tin đó lần lượt được chuyển đến các tầng  dưới trên máy tính đó. Khi đi qua mỗi tầng , thông tin điều khiển của giao thức tương ứng sẽ được thêm vào bản tin, quá trình đó gọi là bao đóng dữ liệu.

* + Các giao thức mức thấp

Các giao thức mức thấp hàm ý chỉ các giao thức thuộc ba lớp thấp nhất của mô hình OSI: tầng vật lý, tầng liên kết dữ liệu và tầng mạng.

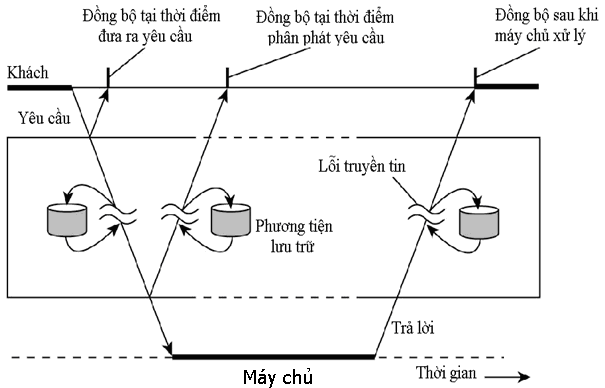
* Tầngvật lý liên quan tới việc chuyển các bít dữ liệu, qui định các tiêu chuẩn về điện, cơ và các giao diện kết nối đến đường truyền mạng, phương pháp truyền các bit.
* Tầngliên kết dữ liệu tập hợp các bit thành từng nhóm gọi là khung dữ liệu, phát hiện và sửa lỗi khi truyền các khung dữ liệu đó. Trong mạng diện rộng, việc trao đổi thông tin giữa các máy tính phải chuyển qua nhiều thiết bị định tuyến, nhiệm vụ chính của các thiết bị này là duy trì bảng định tuyến và tìm đường đi tốt nhất cho mỗi gói tin. Hiện nay giao thức đang được sử dụng phổ biến nhất tại tầngmạng là giao thức IP.
  + Các giao thức tầng vận tải
  + Tầng vận tải là tầng thấp nhất trong ngăn xếp các giao thức cơ sở để người lập trình viên sử dụng khi phát triển các ứng dụng mạng.
  + Chức năng cơ bản của tầng vận tải là quản lý việc trao đổi thông tin giữa các tiến trình trên các thiết bị đầu cuối của người sử dụng. Đối với truyền số liệu có liên kết, tầng vận tải thực hiện các chức năng phát hiện, sửa lỗi và điều khiển tốc độ kênh truyền logic, do đó bên gửi và bên nhận cần phải thiết lập liên kết để thống nhất các tham số điều khiển.



* + Các giao thức mức cao
* Các giao thức mức cao thuộc các tầng phía trên tầng vận tải, theo mô hình nó gồm ba tầng: tầng phiên, tầng trình diễn và tầng ứng dụng.
* Tầng phiên quản lý các phiên hội thoại giữa các tiến trình trên thiết bị đầu cuối mạng. Tại tầng này người ta thiết kết các điểm kiểm tra nhằm hạn chế việc phải truyền lại toàn bộ dữ liệu khi xảy ra sự cố mất dữ liệu trên mạng.
* Tầng trình diễn thực hiện các nhiệm vụ mã hóa/giải mã dữ liệu nhằm thống nhất cách thể hiện các loại dữ liệu khác nhau của người sử dụng, ví dụ dùng bảng mã ASCII để mã hóa văn bản. Ngoài ra tầng này cũng thực hiện các chức năng khác như  nén/giải nén dữ liệu, mã hóa/giải mã bảo mật dữ liệu.
* Tầng ứng dụng bao gồm các giao thức phục vụ trực tiếp cho các dịch vụ của người sử dụng như: thư điện tử, truyền tập tin, truy nhập trang thông tin điện tử….
  + Giao thức tầng trung gian
* Các giao thức tầng trung gian thường nằm tại tầng ứng dụng, nó bao gồm nhiều giao thức đa năng để đảm bảo tính độc lập với các ứng dụng, chúng thường được thể hiện dưới dạng các dịch vụ.
* Phương pháp truyền tập tin là mức đơn giản nhất trong các hệ thống phân tán, thông tin cần trao đổi giữa các đối tượng trong hệ thống được lưu dưới dạng tập tin, các máy tính phải cùng sử dụng một giao  thức để gửi và nhận tập tin.
* Các giao thức tầng trung gian đơn giản hoá sự phức tạp trong việc truyền dữ liệu trên mạng, nhờ có các giao thức này mà việc gọi các thủ tục từ xa sẽ được thể hiện tương tự như gọi các thủ tục trên máy cục bộ.
* Đối chiếu với mô hình phân tầng OSI, tầng trung gian thể hiện các tính năng của nó trong tầng trình diễn và tầng phiên. Hiện nay có nhiều kiến trúc khác nhau dùng để thể hiện giao thức tầng trung gian như: Gọi thủ tục từ xa (RPC), kiến trúc môi trường yêu cầu đối tượng chung (CORBA), mô hình đối tượng thành phần phân tán (DCOM), gọi phương thức từ xa trong Java (RMI). Giải pháp RMI áp dụng riêng cho ngôn ngữ lập trình Java, nó sử dụng nhiều đặc tính của phương pháp gọi thủ tục từ xa RPC và kiến trúc CORBA. Tuy nhiên khi kiến trúc CORBA được đưa vào lõi của ngôn ngữ Java thì kiến trúc này sẽ dần thay thế RMI. Mô hình DCOM chỉ được ứng dụng trong các hệ thống cài đặt hệ điều hành của Microsoft và có hiều hạn chế trong việc thiết kế hệ thống.

Tầng trung gian đóng vai trò quan trọng trong việc xây dựng các hệ thống phân tán, việc thiết lập các tiêu chuẩn cho tầng trung gian nhằm mục đích cung cấp khả năng tương thích và tính mềm dẻo của các ứng dụng phân tán, nghĩa là phải chuẩn hoá giao diện lập trình ứng dụng và các giao thức trao đổi thông tin. Theo hai tiêu chí trên, tầng trung gian được chia thành bốn loại sau:

* Tầng trung gian mở hoàn toàn: sử dụng giao diện lập trình ứng dụng và giao thức chung để trao đổi thông tin giữa các thành viên phân tán, các sản phẩm loại này đảm bảo khả năng tương tác với nhau mà không phụ thuộc nền tảng của mỗi thành viên trong hệ thống., ví dụ DCE RPC, CORBA, OpenDoc.
* Tầng trung gian với giao diện lập trình ứng dụng mở: Cho phép các thành viên giao tiếp với nhau không phụ thuộc nền tảng hệ thống, các thành viên chỉ sử dụng chung các giao thức, ví dụ sản phẩm ODBC của Microsoft
* Tầng trung gian với giao thức mở: sử dụng giao thức chung nhưng cần phải chuyển đổi giao diện lập trình ứng dụng, ví dụ kiến trúc cơ sở dữ liệu quan hệ phân tán DRDA của IMB.
* Tầng trung gian riêng: các ứng dụng phân tán chỉ giao tiếp với nhau khi sử dụng sản phẩm của một nhà cung cấp, ví dụ ActiveX/OLE.
  1. Phân loại mạng truyền thông
* Để hiểu về các loại truyền thông mà tầng trung gian cung cấp cho các ứng dụng, chúng ta coi nó như một dịch vụ phụ trợ trong mô hình khách/chủ. Ví dụ hệ thống thư điện tử, bản chất của hệ thống này là dịch vụ truyền thông trung gian, trên mỗi máy của người dùng cài đặt phần mềm cho phép biên soạn, gửi và nhận thư điện tử. Người dùng biên soạn thư, gửi lên hệ thống phân phát thư điện tử và chờ đợi kết quả phân phát thư đó đến người nhận. Tương tự như vậy, người nhận kết nối đến hệ thống thư điện tử, kiểm tra xem có thư của mình hay không, nếu có thì hệ thống thư điện tử sẽ chuyển các bức thư đó tới máy của người dùng.
* Truyền thông bền bỉ (persistent), các thông điệp của người dùng được lưu trữ trong hệ thống cho đến khi chuyển thành công đến người nhận, bên gửi và bên nhận hoạt động hoàn toàn độc lập với nhau. VD Thư điện tử
* Truyền thông tạm thời (transient) chỉ lưu giữ thông điệp trong thời gian gửi và nhận, nghĩa là bên gửi và bên nhận phụ thuộc lẫn nhau, nếu bên nhận không hoạt động thì các thông điệp sẽ bị hủy bỏ.  
  Truyền thông cũng có thể được thực hiện dưới hình thức đồng bộ hoặc không đồng bộ.
* Truyền thông đồng bộ, bên gửi sẽ bị phong tỏa cho đến khi biết chắc chắn yêu cầu của mình đã được bên nhận xử lý. Phương pháp này đánh dấu ba thời điểm: Thời điểm thứ nhất bên gửi sẽ bị phong tỏa cho đến khi hệ thống trung gian tiếp nhận xong yêu cầu, thời điểm thứ hai hệ thống trung gian thông báo đã chuyển yêu cầu cho bên nhận và thời điểm thứ ban bên gửi sẽ tiếp nhận kết quả bên nhận xử lý.
* Truyền thông không đồng bộ cho phép bên gửi tiếp tục thực hiện công việc của mình sau khi đã gửi thông điệp đến hệ thống trung gian



* Trong thực tế người ta thường kết hợp hai loại truyền thông trên để trao đổi thông tin giữa các tiến trình:
* Truyền bền bỉ và đồng bộ thường được áp dụng trong các trường hợp cẩn phải chuyển một lượng lớn dữ liệu
* Truyền thông tạm thời và đồng bộ  sẽ chỉ phù hợp với lượng dữ liệu nhỏ, phươg pháp này thường sử dụng trong kỹ thuật gọi thủ tục từ xa.
* Bên cạnh tính bền bỉ và tính đồng bộ người ta còn phân biệt tính rời rạc hay liên tục của truyền thông, những hệ thống mà mỗi thông điệp được truyền đi là những đơn vị dữ liệu độc lập sẽ được xếp vào nhóm rời rạc, nếu các thông điệp được truyền lần lượt và liên tục gọi là phương pháp truyền tin theo luồng.

1. Gọi thủ tục từ xa

Ý tưởng của phương pháp vẫn dựa vào qui trình gọi thủ tục, tuy nhiên điểm khác biệt nằm ở chỗ thủ tục đó được cài đặt trên một máy tính khác, người lập trình chỉ cần chuyển các giá trị vào các tham số của thủ tục.

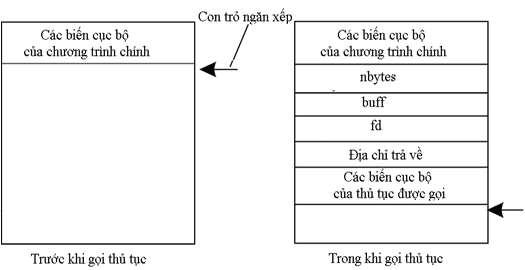
Phương pháp này khá đơn giản cho việc cài đặt, tuy nhiên trong thực tế nảy sinh khá nhiều vấn đề như:  thực thi mã lệnh được thực hiện trên các vùng nhớ khác nhau hoặc nếu một trong hai máy tính bị lỗi trong quá trình thực thi mã lệnh cũng nảy sinh nhiều vấn đề phức tạp. Mặc dù vậy, phương pháp gọi thủ tục từ xa vẫn là phương pháp được áp dụng phổ biến nhất trong các hệ thống phân tán

* 1. Cơ chế hoạt động của các phương pháp gọi thủ tục từ xa

Phương pháp gọi thủ tục từ xa cho phép cài đặt các hệ thống phân tán theo mô hình khách/chủ:

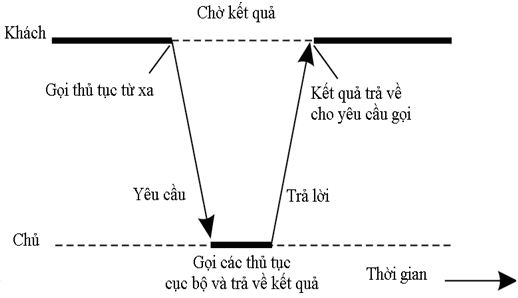
Các ứng dụng khách kết nối với máy chủ và sử dụng các dịch vụ do máy máy chủ cung cấp. Các bước gọi thủ tục trên máy chủ được thực hiện tương tự như gọi thủ tục trên máy cục bộ, máy khách chuyển các tham số đầu vào khi gọi thủ tục và dịch vụ trên máy chủ sẽ kiểm tra trính hợp lệ của các tham số đó, thực hiện tính toán và trả về các giá trị theo yêu cầu của ứng dụng máy khách. Để hiểu về phương pháp gọi thủ tục từ xa, trước hết chúng ta cần phải nhắc lại qui trình thực hiện khi gọi một thủ tục truyền thống trên một máy tính. Giả sử có một thủ tục đọc tập tin đơn giản như sau:

count = read(*fd, buff, nbytes*);  
trong đó *fd* là con trỏ tập tin, *buff* là vùng đệm, *nbytes* là số lượng byte cần đọc.



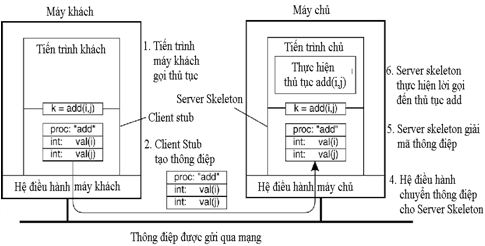
Khi chương trình gọi thủ tục, nó tạo ra một ngăn xếp dành cho thủ tục đó. Để thực thi lời gọi thủ tục, nó truyền các tham số của thủ tục vào ngăn xếp theo thứ tự ngược (tham số đầu tiên sẽ được chuyển cuối cùng).

Sau khi thực hiện xong thủ tục, giá trị trả về sẽ được chuyển tới các thanh ghi, giải phóng vùng nhớ và chuyển quyền điều khiển cho chương trình gọi, chương trình gọi sẽ loại bỏ tham số ra khỏi ngăn xếp, trả ngăn xếp về trạng thái như trước khi gọi thủ tục. Việc truyền tham số có thể được thực hiện bằng một trong ba phương pháp: Truyền giá trị, truyền con trỏ và truyền tham chiếu.  
Con trỏ đơn giản là địa chỉ của một đối tượng trong bộ nhớ, thông thường các đối tượng có thể được truy xuất trong hai cách: trực tiếp bởi tên đại diện hoặc gián tiếp thông qua con trỏ. Các biến con trỏ được định nghĩa trỏ tới các đối tượng của một kiểu cụ thể sao cho khi con trỏ hủy thì vùng nhớ mà đối tượng chiếm giữ được thu hồi. Các con trỏ thường được dùng cho việc tạo ra các đối tượng động trong thời gian thực thi chương trình. Không giống như các đối tượng bình thường (toàn cục và cục bộ) được  cấp phát lưu trữ trên ngăn xếp trong thời gian chạy (runtime stack), một đối tượng động được cấp phát vùng nhớ từ vùng lưu trữ khác được gọi là heap.  
Truyền tham chiếu cung cấp một tên tượng trưng khác gọi là bí danh (alias) cho một đối tượng, truy xuất một đối tượng thông qua một tham chiếu giống như là truy xuất thông qua tên gốc của nó. Tham chiếu nâng cao tính hữu dụng của các con trỏ và sự tiện lợi của việc truy xuất trực tiếp các đối tượng, chúng được sử dụng để hỗ trợ các kiểu gọi thông qua tham chiếu của các tham số hàm đặc biệt khi các đối tượng lớn được truyền tới thủ tục.



Ý tưởng phương pháp gọi thủ tục từ xa là che giấu quá trình thực hiện thủ tục trên máy tính khác, điều này được thực hiện bằng cách che giầu quá trình trao đổi thông tin giữa các máy tính. Khi thực hiện gọi  thủ tục từ xa, một thành phần trên máy khách gọi là Stub sẽ bao bọc các tham số vào thông điệp và thông điệp đó phải được chuyển đến máy chủ. Tại máy chủ, một thành phần tương ứng với Stub gọi là Skeleton sẽ giải mã thông điệp đã nhận được và thực thi các mã lệnh như phương pháp gọi thủ tục truyền thống, kết quả thực hiện lại được chuyển vào thông điệp để trả về cho Stub. Stub trên máy khách sẽ giải mã thông điệp và trả về cho chương trình gọi các giá trị hoặc các tham số theo yêu cầu,  quá trình xử lý giữa Stub và Skeleton hoàn toàn trong suốt đối với lời gọi thủ tục.  
Để sử dụng tính năng gọi thủ tục từ xa, một chương trình trên máy chủ phải cung cấp các dịch vụ mô tả trong ngôn ngữ RPC. Mỗi máy chủ được gán tên và số chương trình, tất cả các dịch vụ được khai báo trong danh sách với đầy đủ các tham số thể hiện dịch vụ. Với nguyên tắc này, ngôn ngữ RPC cho phép thực hiện các kiểu dữ liệu đơn giản cũng như các dữ liệu phức tạp như: struct, enum... Qui trình thực hiện bao gồm các bước sau:

* Máy khách sử dụng tính năng gọi thủ tục cục bộ trong Stub.
* Stub trên máy khách gửi các tham số đến máy chủ bằng cách gửi yêu cầu gọi thủ tục từ xa
* Yêu cầu của Stub được Skeleton trên máy chủ phân tích.
* Thực hiện thủ tục đã được phân tích trên máy chủ.
* Máy chủ trả về kết quả thực hiện cho máy khách



Phương pháp gọi thủ tục từ xa thể hiện quan điểm tách biệt giữa giao diện và phần cài đặt, xuất phát từ việc khai báo giao diện, phần mềm trung gian tạo các mã lệnh hỗ trợ cho việc xử lý phân tán bằng cách thể hiện các ứng dụng dưới dạng ngữ nghĩa truyền thống trên máy cục bộ, tuy nhiên nó phải đảm bảo nhiều nhiệm vụ phức tạp như: truyền dữ liệu, đồng bộ tiến trình. Quá trình thực hiện gọi thủ tục từ xa được thực hiện qua mười bước sau:

* Thủ tục trên máy khách gọi stub như phương pháp gọi thủ tục truyền thống.
* Stub tạo thông điệp và chuyển đến hệ điều hành của máy khách .
* Hệ điều hành của máy khách gửi thông điệp đến hệ điều hành của máy chủ.
* Hệ điều hành của máy chủ chuyển thông điệp đến Skeleton .
* Skeleton giải mã thông điệp thành các tham số và gọi thủ tục xử lý tương ứng.
* Máy chủ thực hiện lời gọi thủ tục và trả về giá trị cho Skeleton.
* Skeleton đóng gói tham số giá trị trả về thành thông điệp và chuyển đến hệ điều hành của máy chủ.
* Hệ điều hành của máy chủ chuyển thông điệp đến hệ điều hành của máy khách.
* Hệ điều hành của máy kháchnhận thông điệp và chuyển cho Stub.
* Stub giải mã kết quả và trả về các tham số theo yêu cầu của thủ tục đã gọi.
  1. Vấn đề truyền tham số

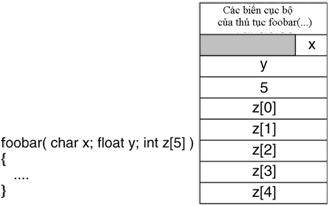
Gọi thủ tục truyền thống có ba cách truyền tham số: truyền giá trị, truyền tham chiếu và truyền con trỏ.

Đối với việc gọi thủ tục từ xa, không thể áp dụng phương pháp truyền con trỏ do đó chỉ có thể thực hiện bằng cách truyền giá trị hoặc truyền tham chiếu.

Truyền giá trị trở nên phức tạp đối với hệ thống phân tán không đồng nhất, các bộ vi xử lý của Intel theo định dạng **little endian**, trong khi các bộ vi xử lý của Sun theo định dạng  **big endian**.



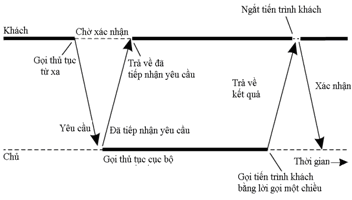
Hình 2.8 minh họa ví dụ thủ tục từ xa gồm 2 thám số (5, “JILL”), các bit dữ liệu lần lượt được chuyển từ máy tính của hãng Intel sang máy tính của hãng Sun, khi đến máy Sun giá trị của các tham số theo thứ tự ngược và máy Sun sẽ hiểu số 5 là  5 x 224, như vậy chỉ cần đảo ngược thứ tự là sẽ nhận được kết quả ban đầu. Tuy nhiên, kiểu số nguyên mới cần đảo vị trí các byte nhưng dữ liệu kiểu xâu ký tự lại không cần thiết, do đó phải bổ sung thêm thông tin về kiểu dữ liệu của tham số.  
Con trỏ là kiểu dữ liệu đặc biệt, nó lưu trữ địa chỉ của một biến số mà địa chỉ đó chỉ có ý nghĩa bên trong một máy tính, như vậy không thể áp dụng phương pháp này để truyền tham số trong gọi thủ tục từ xa. Tuy nhiên, giải pháp truyền tham chiếu có thể thực hiện bằng cách tạo một biến tham chiếu tương ứng trên Skeleton và thủ tục trên máy chủ sẽ sử dụng địa chỉ của biến này như phương pháp truyền tham chiếu của một thủ tục thông thường.



Phương pháp gọi thủ tục từ xa che giấu quá trình trao đổi thông tin trên mạng bằng cách đưa các tham số vào thông điệp. Như vậy, cả phía máy khách và máy chủ đều phải tuân thủ quy tắc về định dạng tham số, đó là giao thức gọi thủ tục từ xa. Để đơn giản hóa quá trình tạo Stub cho máy trạm và Scheleton cho máy chủ, một ngôn ngữ mới được sử dụng gọi là ngôn ngữ định nghĩa giao diện IDL. Lập trình viên chỉ việc viết các hàm và các thủ tục theo qui định của ngôn ngữ, sau đó dùng chương trình dịch để chuyển sang ngôn ngữ lập trình tương ứng.

* 1. Gọi thủ tục từ xa bằng phương pháp không đồng bộ

Phương pháp gọi không đồng bộ cho phép tiến trình gọi trên máy khách gửi yêu cầu đến máy chủ, sau khi máy chủ xác nhận đã nhận được yêu cầu, tiến trình trên máy khách có thể tiếp tục xử lý các tác vụ khác mà không cần chờ đợi kết quả xử lý của máy chủ, như vậy sẽ rút ngắn thời gian phong tỏa hệ thống. Như vậy quá trình thực thi thủ tục trên máy chủ sẽ độc lập với tiến trình trên máy khách, sau khi hoàn thành, tiến trình trên máy chủ sẽ gọi tiến trình trên máy khách để trao kết quả trả về.



* 1. Mô hình đối tượng phân tán

Hiện nay việc xây dựng hệ thống phân tán dựa trên quan điểm xây dựng các đối tượng phân tán, mỗi ứng dụng phân tán bao gồm tập các đối tượng. Đặc điểm cơ bản của lập trình hướng đối tượng là việc che giấu xử lý bên trong, do đó những thay đổi bên trong đối tượng sẽ không ảnh hưởng đến các đối tượng khác. Một đặc tính quan trọng của đối tượng là tính bao đóng các thuộc tính và phương thức xử lý, thao tác với các thuộc tính của đối tượng đều phải được thực hiện thông qua các hàm giao diện. Khi máy khách gọi đối tượng từ xa, giao diện của đối tượng đó sẽ được nạp vào không gian địa chỉ của máy khách và đảm nhiệm vai trò tương tự như Stub trong phương pháp gọi thủ tục từ xa, nó đảm nhiệm chức năng chuyển đổi lời gọi các hàm giao diện thành dạng thông điệp để hệ điều hành chuyển đến máy chủ và ngược lại chuyển đổi thông điệp từ máy chủ thành kết quả trả về của hàm giao diện. Kế thừa ưu điểm của kỹ thật lập trình hướng đối tượng, để thuận tiện cho việc xây dựng hệ thống phân tán người ta đã nâng tầm phương pháp gọi  thủ tục từ xa thành phương pháp gọi đối tượng từ xa, ví dụ Java RMI, Microsoft DCOM, CORBA và dịch vụ web.  
Đối tượng thực nằm ở phía máy chủ, nó có giao diện giống như trên máy khách và đồng thời một thành phần Skeleton có nhiệm vụ chuyển đối thông điệp từ phía máy khách thành lời gọi thủ tục tương ứng và sau đó chuyển kết quả thực hiện thành thông điệp để hệ điều hành chuyển tới máy khách. Các đối tượng phân tán liên quan đến vấn đề biên dịch, mà ngôn ngữ biên dịch lại phụ thuộc vào ngôn ngữ lập trình, do đó chúng phải xây dựng một cách tường minh, như vậy mới có thể xây dựng các ứng dụng dựa trên các ngôn ngữ lập trình khác nhau. Vấn đề cài đặt bên trong các đối tượng phân tán là công việc của các lập trình viên, điều quan trọng cần phải đảm bảo các tính thống nhất về mặt giao diện để các máy tính khác có thể gọi đối tượng đó. Giải pháp cho vấn đề này là xây dựng một thành phần thích nghi đối tượng để thể hiện các đối tượng phân tán trên môi trường mạng.  
Việc gọi các đối tượng phân tán tương tự như gọi đối tượng trên một máy tính, điểm khác biệt cơ bản là trước khi gọi hàm của đối tượng phân tán cần phải nhúng đối tượng cục bộ vào đối tượng phân tán. Có thể thực hiện bằng cách gọi tường minh hoặc không tường minh. Khi gọi tường minh, cần phải khai báo biến con trỏ trên máy tính máy khách, sau đó gọi hàm bind để nhúng đối tượng phân tán vào biến con trỏ cục bộ.

Distr\_object\* obj\_ref;    // Khai báo đối tượng tham chiếu toàn hệ thống  
obj\_ref = ...;            // Khởi tạo biến tham chiếu cho đối tượng phân tán  
obj\_ref->do\_something( ); // Nhúng không tường minh, gọi hàm  
Distr\_object\* obj\_ref;    // Khai báo đối tượng tham chiếu toàn hệ thống  
Localobject\* obj\_ptr;     // Khai báo biến con trỏ cho đối tượng cục bộ  
obj\_ref =...;             // Khởi tạo biến tham chiếu cho đối tượng phân tán  
obj\_ptr = bind(obj\_ref);  // Nhúng đối tượng bằng phương pháp tường minh  
obj\_ptr->do\_something( ); // Gọi hàm

1. Truyền thông hướng thông điệp
   1. Tính bền bỉ và tính đồng bộ trong trao đổi thông tin

Mỗi máy tính trên mạng cung cấp giao diện trao đổi thông điệp, khi cần chuyển thông điệp đến một máy tính khác, tiến trình bên gửi chỉ cần chuyển các thông điệp đó đến giao diện, công việc tiếp theo sẽ do hệ điều hành thực hiện và đảm bảo sẽ chuyển tới giao diện thông điệp của bên nhận.  Xét về độ tin cậy, truyền thông điệp có thể thực hiện bằng hai phương pháp:

* Truyền tạm thời: Thông điệp chỉ được lưu trong thời gian thực hiện chuyển thông tin, nếu không thành công thì thông điệp đó sẽ bị hủy bỏ.
* Truyền bền bỉ: Thông điệp được lưu tại bộ đệm cho đến khi toàn bộ thông điệp đã gửi thành công cho bên nhận. Chuyển thông điệp bền bỉ tương tự như tổ chức mạng lưới bưu chính,  bưu phẩm được chuyển đến các bưu cục và người đưa thư     có nhiệm vụ chuyển các bưu phẩm đó đến người nhận, quá trình này có thể phải được chuyển qua một hoặc nhiều bưu cục.

Xét về cách thức gửi/nhận, truyền thông điệp có thể được thực hiện bằng hai phương pháp:\

* Truyền đồng bộ: Bên gửi sẽ tạm ngừng tiến trình cho đến khi toàn bộ thông điệp được chuyển thành công đến  vùng đệm của bên nhận.
* Truyền không đồng bộ: Bên gửi chỉ bị tạm ngừng trong thời gian chuyển thông điệp đến vùng đệm gửi.

Trong thực tế, để truyền thông điệp có thể tổ hợp các phương pháp trên, như vậy có bốn khả năng xảy ra:

* Truyền đồng bộ, tạm thời
* Truyền đồng bộ, bền bỉ
* Truyền không đồng bộ, tạm thời
  + Truyền không đồng bộ, tạm thời, dựa trên vùng đệm của bên nhận
  + Truyền không đồng bộ, tạm thời, dựa trên vùng đệm của bên gửi
  + Truyền không đồng bộ, tạm thời, dựa trên phản hồi của bên nhận
* Truyền không đồng bộ, bền bỉ

Hầu hết các hệ thống phân tán lựa chọn phương pháp truyền tin không đồng bộ, tạm thời, và dựa trên phản hồi của bên nhận, như vậy có thể đảm bảo tính tin cậy nhưng không làm suy giảm hiệu năng của hệ thống.

* 1. Truyền tin nhanh hướng thông điệp

Nhiều ứng dụng trên các hệ thống phân tán được xây dựng dựa trên dịch vụ truyền thông điệp của tầng vận tải.  Theo mô hình mạng TCP/IP, mạng máy tính được phân làm 4 tầng, để đảm bảo tính duy nhất mỗi thông điệp của một tiến trình sẽ được chuyển đến cổng đã cấp cho tiến trình, số hiệu cổng và địa chỉ logic của máy tính gọi là socket. Để tạo điều kiện thuận lợi cho việc phát triển phần mềm trên hệ thống phân tán, mô hình TCP/IP đã qui định một số hàm nguyên thủy.

* Hàm tạm thời

|  |  |
| --- | --- |
| **Hàm** | **Ý nghĩa** |
| Socket | Tạo Socket |
| Bind | Gắn địa chỉ cục bộ vào socket |
| Listen | Thông báo sẵn sạng nhận thông tin |
| Accept | Tiếp nhận yêu cầu liên kết |
| Connect | Thiết lập liên kết |
| Send | Gửi dữ liệu |
| Receive | Nhận dữ liệu |
| Close | Hủy bỏ liên kết |

* Hàm bền bỉ

|  |  |
| --- | --- |
| **Hàm** | **Ý nghĩa** |
| MPLbsend | Thêm thông địp vào vùng đệm gửi |
| MPI-send | Gửi thông điệp và chờ cho đến khi toàn bộ thông điệp đã được chuyển đến vùng đệm của bên nhận |
| MPLssend | Gửi thông điệp và chờ cho đến khi bên nhận bắt đầu thực hiện việc nhận thông điệp |
| MPLsendrecv | Chuyển thông điệp và chờ cho đến khi có xác nhận của bên nhận |
| MPUsend | Tham chiếu đến thông đang chở gửi |
| MPLissend | Tham chiếu đến thông đang chở gửi và chờ cho đến khi bên nhận bắt đầu |
| MPLrecv | Nhận thông điệp, phong tỏa nếu không có thông điệp |
| MPLirecv | Nhận thông điệp, không phong tỏa nếu không có thông điệp |

1. Truyền thông hướng luồng

* Truyền thông bằng cách gọi thủ tục từ xa hay truyền thông điệp mới chỉ chú trọng đến việc chuyển dữ liệu mà chưa đề cập đến loại dữ liệu cần chuyển, thời gian thực hiện, tính liên tục và yêu cầu đồng bộ.
* Truyền tin yêu cầu thời gian thực phải được thực hiện với độ trễ thấp nhất có thể nhưng vẫn phải đáp ứng các tiêu chuẩn về chất lượng dịch vụ.
* Các dịch vụ đa phương tiện bao gồm nhiều luồng dữ liệu khác nhau như văn bản, âm thanh, hình ảnh…, chúng đòi hỏi tính đồng bộ giữa các luồng dữ liệu và đồng thời sự liên tục của thông tin cũng là thước đo của chất lượng dịch vụ.

1. Truyền thông theo nhóm

* Truyền thông theo nhóm là phương pháp truyền dữ liệu từ một thành viên đến nhiều thành viên khác trong cùng một nhóm chỉ với một lần thực hiện.
* Truyền thông theo nhóm được sử dụng trong rất nhiều lĩnh vực của ứng dụng phân tán như: dịch vụ phát thanh, truyền hình, hội thảo….