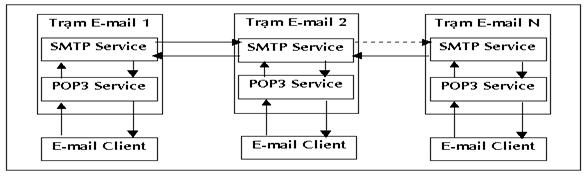
Tìm hiểu về giao thức SMTP

**1.TỔNG QUAN.**

**a. Khái niệm**

Là dịch vụ rất phổ biến và thông dụng trong mạng Internet/Intranet và hầu như không thể thiếu được trong Internet/Intranet hiện nay. Tuy nhiên không phải là dịch vụ “từ đầu - đến cuối” (end to end). Nghĩa là dịch vụ này không đòi hỏi hai máy tính gởi và nhận thư phải nối trực tiếp với nhau để thực hiện việc chuyển thư. Nó là dịch vụ kiểu lưu và chuyển tiếp (store and forward) thư được chuyển từ máy này sang máy khác cho tới khi máy đích nhận được. Người nhận cũng chỉ thực hiện một số thao tác đơn giản để lấy thư, đọc thư và nếu cần thì cho in ra. Cách liên lạc này thuận tiện hơn nhiều so với gởi thư thông thường qua bức điện hoặc Fax, lại rẻ và nhanh hơn. Cách thực hiện việc chuyển thư không cần phải kết nối trực tiếp với nhau để chuyển thư, thư có thể được chuyển từ máy này đến máy khác cho tới máy đích.. Giao thức truyền thống sử dụng cho hệ thống thư điện tử của Internet là SMTP(Simple Mail Transfer Protocol). Cơ chế hoạt động của thư điện tử(E-mail):



b. Gioi thiệu cơ bản về SMTP

**Simple Mail Transfer Protocol (SMTP)** là giao thức truyền tải e-mail qua mạng, **SMTP** cho phép chuyển thông điệp mail từ mail server của người gửi đến mail server của người nhận. Thông điệp mail ở định dạng ASCII 7 bit. SMTP được định nghĩa trong RFC 821, điều chỉnh trong RFC 1123 và mở rộng trong RFC 2821 với tên gọi **ESMTP** (Extended SMTP). SMTP sử dụng cổng 25 trên nền TCP.

Trong RFC 822 định nghĩa tiêu chuẩn của định dạng thông điệp maill, gồm 2 phần: dòng tiêu đề (header) và phần thân (body).

-    Mỗi tiêu đề có một dòng tiêu đề *from,to, subject* và một số dòng tiêu đề khác từ lệnh SMTP. Sau tiêu đề là một dòng trống.

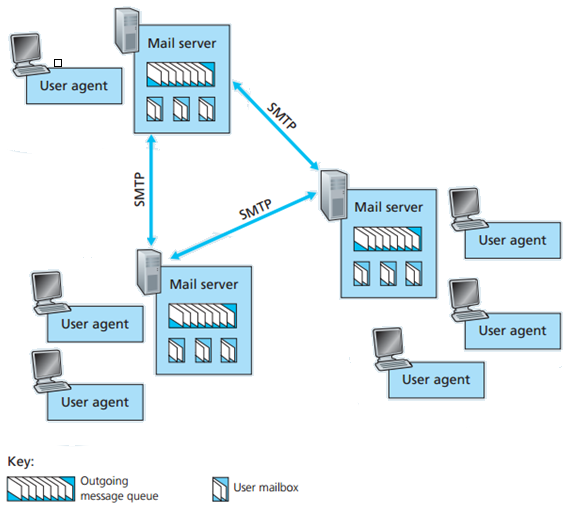
-    Phần thân chứa các thông điệp định dạng ở mã ASCII. Mỗi dòng thông thông điệp được kết thúc bằng một dấu chấm.

Một hệ thống e-mail có ba thành phần chính: user agents, mail server và giao thức SMTP.

-    User agentcho phép người dùng đọc, trả lời, chuyển tiếp, lưu và soạn thông điệp mail. User agent là những phần mềm gửi mail như Microsoft’s Outlook, Apple Mail (những phần mềm này sử dụng giao diện GUI).

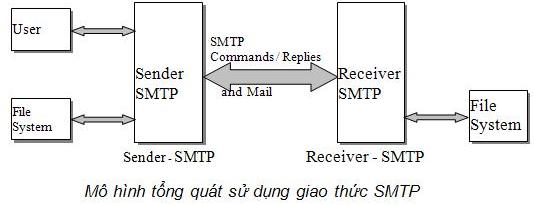
-    Mail server là thành phần cốt lỗi trong hạ tầng hệ thống e-mail. Khi người dùng soạn xong thông điệp thì user agent sẽ gửi thông điệp đến mail server và thông điệp được đặt trong hàng đợi (message queue), sau đó sẽ gửi đến mail server của người dùng khác và được lưu tại mailbox.

-    Giao thức SMTP đảm nhiệm việc truyền tải thông điệp từ mail server của người gửi đến mail server của người nhận. SMTP thiết lập kết nối TCP trên cổng 25.

[](http://2.bp.blogspot.com/-ga649HvovqM/U746LMWZ6zI/AAAAAAAAAHM/ioZCKrzYbak/s1600/smtp.PNG)

**2.GIAO THỨC SMTP(RFC821)**

Mục đích của giao thức SMTP là truyền mail một cách tin cậy và hiệu quả. Giao thức SMTP không phụ thuộc vào bất kỳ hệ thống đặc biệt nào và nó chỉ yêu cầu trật tự của dữ liệu truyền trên kênh truyền đảm bảo tính tin cậy.



a.*Ý nghĩa các lệnh SMTP:*

- Những lệnh SMTP định nghĩa sự truyền  mail hay chức năng của hệ thống mail được yêu cầu bởi  user. Những lệnh SMTP là những chuỗi ký tự kết thúc bằng <CRLF>. Bản thân mã lệnh là những ký tự chữ (alphabetic)  kết  thúc bởi <SP> nếu có những tham số  theo sau và nếu không có thì <CRLF>. Cú pháp của những mailbox phải tuân theo những qui ước của receiver.

**¨ HELLO (HELO)**

Lệnh này được dùng để xác định ra ai là người gởi mail. Vùng đối số chứa host name của bên gởi.

Bên nhận định danh cho nó đối với sender thông qua việc bắt tay trả lời kết nối.

Với lệnh này và sự trả lời OK để xác định rằng cả sender và reciever đang ở trạng thái khởi đầu, tất cả các bảng trạng thái và buffer đã được xoá sạch.

**¨ MAIL**

Lệnh này được dùng để khởi tạo quá trình trao đổi mail mà ở đó mail data được phân phát tới một hay nhiều mailbox.

**¨ RECIPIENT (RCPT)**

Lệnh này được sử dụng để định ra một người nhận mail, nhiều người nhận (cùng một nội dung mail) sẽ được xác định bằng cách  gởi nhiều lệnh này.

**¨ DATA**

Reciever sẽ xử lý những dòng theo sau lệnh khi mail data đến từ sender. Lệnh này tạo ra mail data để đặt vào mail data buffer. Mail data có thể chứa bất kỳ ký tự nào trong bộ mã ASCII. Mail data được kết thúc bởi một dòng mà nó chỉ chứa một dấu chấm “ .”

**¨ SEND**

Lệnh này được dùng để khởi tạo sự truyền mail mà ở đó maildata sẽ được truyền đi tới một hay nhiều người nhận.

**¨ SEND OR MAIL (SOML)**

          Lệnh này được sử dụng để khởi tạo sự truyền mail mà ở đó mail data một hay nhiều người nhận hoặc các mailbox.

**¨ RESET (RSET)**

Lệnh này xác định sự truyền mail hiện tại đã bị huỷ bỏ. Các sender, recipient, mail data đã lưu sẽ bị huỷ bỏ và tất cả các bảng trạng thái, các buffer bị xoá. Receiver phải gửi một reply OK.

**¨ VERIFY (VRFY)**

Lệnh này yêu cầu receiver xác nhận đối số là định danh một user. Nếu nó là một user name, full name của user đó (nếu receiver biết) và mailbox  đặc tả đầy đủ được trả về.Lệnh này không ảnh hưởng đến reverse-path buffer, forward-path buffer và data mail buffer.

**¨ EXPAND  (EXPN)**

Lệnh này yêu cầu receiver xác nhận đối số là một mailing list(danh sách địa chỉ) và trả về một thành phần trong danh sách đó. Full name của các user (nếu biết)  và những mailbox đã xác định đầy đủ được trả về trong một reply gồm nhiều dòng.

**¨ HELP**

Lệnh này cho receiver những thông tin giúp đỡ cho sender. Lệnh này có thể nhận một đối số (có thể là tên lệnh) và trả về thông tin chi tiết.  
Lệnh này không ảnh hưởng đến reverse-path buffer, forward-path buffer và data mail buffer.

**¨ NOOP**

Lệnh này không ảnh hưởng các tham số hay các lệnh được đưa vào trước nó, nó đặc tả không có một hành động nào khác hơn là receiver gửi một reply OK. Lệnh này không ảnh hưởng đến reverse-path buffer, forward-path buffer và data mail buffer.

**¨ QUIT**

Lệnh này định rõ receiver phải gửi một reply OK và sau đó đóng kênh truyền. Receiver sẽ không đóng kênh truyền cho đến khi nó nhận và trả lời cho  lệnh QUIT  (ngay cả nếu có một lỗi xảy ra).

    b. *Cú pháp của các lệnh*

        - Các lệnh bao gồm một mã lệnh theo sau là đối số của lệnh. Mã lệnh là 4 ký tự alphabetic. Không phân biệt chữ thường hoặc chữ hoa.

        - Giữa mã lệnh và đối số là một hoặc nhiều khoảng trắng. Tuy nhiên trong reverse-path và forward-path, kiểu chữ rất quan trọng. Đặc biệt, trên một số host, tên user cũng phân biệt kiểu chữ hoa và thường.

        - Đối số bao gồm một chuỗi ký tự có chiều dài biến đổi kết thúc bằng chuỗi ký tự “ <CRLF> “.

       - Dấu ngoặc vuông biểu diễn cho một vùng đối số tuỳ chọn.

       - Sau đây là những lệnh SMTP:

                 HELO <SP> <domain> <CRLF>

                  MAIL <SP> FROM:<reverse-path> <CRLF>

                  RCPT <SP> TO:<forward-path> <CRLF>

                  DATA <CRLF>

                  RSET <CRLF>

                  SEND <SP> FROM:<reverse-path> <CRLF>

                  SOML <SP> FROM:<reverse-path> <CRLF>

                  SAML <SP> FROM:<reverse-path> <CRLF>

                 VRFY <SP> <string> <CRLF>

                 EXPN <SP> <string> <CRLF>

                 HELP [<SP> <string>] <CRLF>

                 NOOP <CRLF>

                 QUIT <CRLF>

     c. *Ví dụ về một giao dịch của SMTP*

**1.** Server : 220 sample2 Simple Mail Transfer Service Ready

khi được kết nối qua nghi thức TCP/IP, máy nhận trả lời với mã 220 đầu báo cho máy gởi biết dịch vụ SMTP đã sẵn sàng.

**2.** Client : HELLO tmt01vn

Bên nhận đã sẵn sàng, bên gởi gởi HELLO và xưng tên người gởi

**3.** Server : 250 hello.

Trả với mã 250 báo cho biết bên nhận đã sẵn sàng

**4.** Client : MAIL FROM:<tmt01vn@tmt01vn.com>

Bên gởi dùng lệnh MAIL để khởi động phiên giao dịch. Cú pháp như trên cho bên nhận biết địa chỉ bên gởi ( mailbox của bên gởi ) để bên nhận gởi thông báo lỗi nếu có về bên gởi

**5.** Server : 250 OK

Trả lời với mã 250 cho biết sẵn sàng

**6.** Client : RCPT TO:<phungkhn@tmt01vn.com>

**7.** Server: 250 OK

**8.** Client : RCPT TO: phungkhn1@yahoo.com

 Muốn gởi cho bao nhiêu người dùng bấy nhiêu lệnh RCPT kèm theo địa chỉ nhận, bên nhận nếu đúng sẽ trả về mã 250 kèm theo OK

**9.** Server : 550 No such user here

Báo kèm theo mã 550 cho biết không có mailbox trên địa chỉ trên đối với nơi nhận

**10.**Client : DATA

                    Báo cho bên nhận biết dữ liệu bắt đầu từ sau từ DATA

**11.** Server : 354 Start mail input; end with <CRLF>.<CRLF>

Mã 354 báo cho biết đã sẵn sàng nhận mail, kết thúc mail với ký tự CRLF.CRLF

**12.** Client : Bắt đầu thân của mail

**13.**  …v..v..

**14.** Client : ( đến khi kết thúc nhấn CRLF.CRLF )

**15.** Server : 250 OK

**16.** Client : QUIT

Phát lệnh báo kết thúc phiên giao dịch

**17.** Server : 221 sample2 Service closing transmission channel

Mã 221 đóng kết nối đã thiết lập

Ví dụ trên sau phiên làm việc mail được gởi tới địa chỉ mail phungkhn@yahoo.com

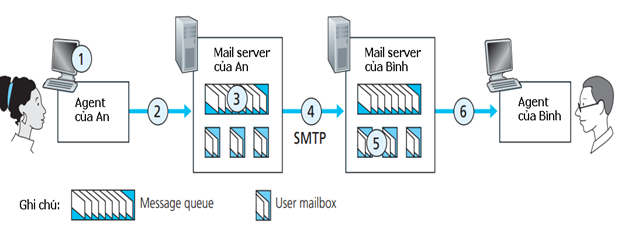
Đồ án tốt nghiệp - Xây dựng dịch vụ thư điện tử Mailling System - Java, jsp

- Xây dựng một dịch vụ thư điện tử khép kín hỗ trợ đa miền miễn phí và dịch vụ  
- Đóng vai trò trạm trung chuyển mail trên mạng thông qua dịch vụ DNS hay SMTP server khác  
- Sử dụng các giao thức truyền thông mail phổ biến nhất hiện nay  
- Chương trình được xây dựng bằng ngôn ngữ Java cho ứng dụng mail server và jsp cho ứng dụng webmail.

3. Phương thức hoạt động của SMTP

**Phương thức hoạt động của SMTP:**

-          Để mô tả hoạt động cơ bản của giao thức SMTP một cách dễ hiểu ta xem xét một hoạt cảnh phổ biến “An gửi thông điệp cho Bình” ở hình

[](http://2.bp.blogspot.com/-7p6PYmSy4H0/U746K5tvAcI/AAAAAAAAAHE/YtuLXvnSRyg/s1600/smtp2.PNG)

Hình 2. Hoạt cảnh gửi email của An và Bình

Ø  Bước 1: An khởi động useragent của mình, cung cấp địa chỉ e-mail của Bình, soạn thông điệp và chỉ thị user agent gửi mail.

Ø  Bước 2: User agent của An gửi thông điệp đến mail server của An và thông điệp được đặt trong hàng đợi.

Ø  Bước 3: SNMP client chạy trên mail server của An phát hiện ra thông điệp trong hàng đợi và tiến hành mở kết nối TCP đến **SMTP server** chạy trên mail server của Bình.

Ø  Bước 4: Sau khi thực hiện bắt tay chào hỏi (handshaking), **SMTP client** của An sẽ gửi thông điệp của An đến kết nối TCP.

Ø  Bước 5: Tại mail server của Bình, SMTP server nhận được thông điệp và lưu lại trên mailbox.

Ø  Bước 6: Khi Bình khởi động user agent của mình thì sẽ thấy mail của An trong mailbox.

-          Lưu ý: SMTP không sử dụng các mail server trung gian để gửi thư, mà chỉ sử dụng một kết nối TCP trực tiếp giữa hai mail server ngay cả khi hai mail server cách nhau một khoản cách rất xa. Ví dụ: mail server của An ở thành phố Hồ Chí Minh và mail server của Bình ởMát-xcơ-va (thủ đô của Nga) thì chỉ có các kết nối TCP trực tiếp giữa hai mail server. Khi mail server của Bình bận, thì thông điệp sẽ được lưu trong hàng đợi của mail server An và chờ đợi để gửi lại tin nhắn.

SMTP truyền thông điệp mail qua ba pha: handshaking (chào hỏi), truyền thông điệp, đóng kết nối.

-          Pha handshaking: đầu tiên, SMTP client (chạy trên mail server bên gửi) thiết lập kết nối TCP trên cổng 25 tới SMTP server (chạy trên mail server bên nhận). Nếu SMTP server gặp sự cố thì client sẽ cố gắng kết nối lại. Khi kết nối được thiết lập, SMTP server và SMTP client thực hiện một số chào hỏi, SMTP client sẽ cho biết địa chỉ email của người gửi và địa chỉ email của người nhận.

-          Pha truyền thông điệp: sau cuộc chào hỏi, SMTP client tiến hành gửi thông điệp trên đường truyền tin cậy TCP.

-          Pha đóng kết nối: Nếu client muốn gửi tiếp tục gửi thông điệp thì lặp lại quá trình trên trong cùng kết nối TCP, còn không thì chỉ thị TCP đóng kết nối

Ví dụ về hoạt cảnh của client (C) là An (với tải khoản anhutech@gmail.com) gửi thông điệp "Hi Binh”,“How are you?” từ mail server gmail.com đến mail server yahoo.com của server (S) là Bình (với tài khoản binhmoskva@yahoo.com)

*S: 220 yahoo.com*

*C: HELO gmail.com*

*S: 250 Hello gmail.com, pleased to meet you*

*C: MAIL FROM: <anhutech@gmail.com>*

*S: 250 an@gmail.com... Sender ok*

*C: RCPT TO: <binhmoskva@yahoo.com>*

*S: 250 binh123@yahoo.com ... Recipient ok*

*C: DATA*

*S: 354 Enter mail, end with "." on a line by itself*

*C: Hi Binh*

*C: How are you?*

*C: .*

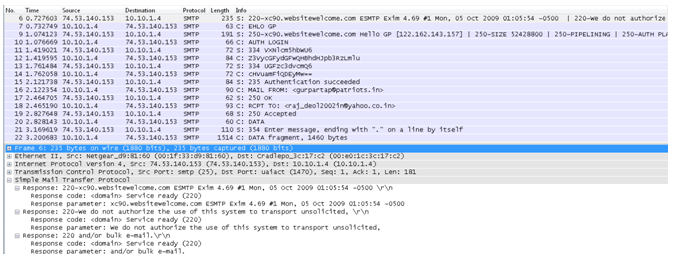
*S: 250 Message accepted for delivery*

*C: QUIT*

*S: 221 yahoo.com closing connection*

Client sử dụng các lệnh: HELO (viết tắt cho HELLO), MAIL FROM, RCPT TO, DATA, và QUIT với ý nghĩa giống tên của chúng. Server trả lời cho mỗi lệnh bằng một mã số trả lời và một số (tùy chọn) giải thích bằng tiếng Anh. Mỗi thông điệp được client gửi đi bắt đầu bằng lệnh HELO gmail.com và chỉ đưa ra QUIT sau khi tất cả các thông điệp đã được gửi và server thực hiện đóng kết nối TCP

Khi cần thực hiện cuộc thoại trực tiếp đến SMTP server, client có thể thực hiện lệnh “telnet<serverName> 25”. Sau khi làm điều này client sẽ nhận được trả lời 220 của server. Client có thể thực hiện telnet đến mail server của mình hay các mail server khác.

[](http://1.bp.blogspot.com/-SAxbHQAmvmM/U746KzhhSGI/AAAAAAAAAHI/PTZx2bJ8QBo/s1600/smtp3.PNG)