

Universidade Católica de Salvador
Departamento de Engenharia e Informática
Disciplina: Teleprocessamento e Redes

Enviando e recebendo mensagens através dos protocolos SMTP e POP3

João Gilberto Magalhães

Objetivo do Trabalho

O objetivo desse trabalho é demonstrar como é possível enviar e receber e-mail através da utilização dos protocolos de comunicação SMTP e POP3 existentes na camada de software. Também tem como objetivo ilustrar rapidamente como funciona o mecanismo para definição de padrão (RFC) e suas aplicações práticas.

O Que é RFC?

A RFC (Request for Comments) é um documento que descreve os padrões para fazer a internet funcionar. Esses padrões estão na camada mais alta em relação a camada OSI estando bem próximo do entendimento humano. Essa camada nós chamamos de camada de Software. Aliás, nos primórdios da Internet, os usuários que quisessem fazer alguma tarefa na Internet deveria conhecer pelo menos boa parte das RFCs existentes.

Apesar do nome “Solicitação por comentários” – RFC em português – existem normas bastante rígidas para se escrever e publicar, sendo a sua aprovação e posterior utilização um processo bastante lento.

Como a internet utiliza um padrão aberto de comunicação, é importante que todos os softwares que desejem enviar algum tipo de informação pela grande rede, sigam corretamente a especificação definida em um RFC. A RFC abrange apenas a camada de software. Portanto você poderá encontrar nessa especificação apenas a definição de como um software poderá enviar dados utilizando a internet, mas não chega na definição de como será isso no nível do Hardware.

Enviando mensagens texto na Internet na camada de software

Qual seria o tipo de dado que trafega pela Internet? Por incrível que pareça o tipo de informação que trafega na Internet é fundamentalmente textual. Apesar de recebermos imagens, vídeos e o que mais imaginarmos, a informação em si deve, necessariamente, ser convertida para texto através da codificação Base64, enviada e logo depois decodificada no receptor.

É claro que para enviar e receber uma mensagem é importante que ambas as pontas (transmissor e receptor) sejam capazes de empacotar e ler a informação que está trafegando. Como foi visto anteriormente, A RFC determina esses padrões.

O padrão que servirá de base para o nosso estudo está definido na RFC822 e trata sobre como enviar mensagens texto através da Internet. Nesse padrão está definido qual é o conjunto mínimo de cabeçalhos que devem ser reconhecidos e a sua funcionalidade.

Por exemplo, uma mensagem na internet tem o seguinte formato:

```
Received: from [10.71.198.77] by sedeped03
        (ArGoSoft Mail Server Freeware, Version 1.8 (1.8.2.5)); Fri, 21 Mar
2003 09:07:45 -0300
Subject: Exemplo
Message-ID: <fbozyyzd3v8ajga.210320030907@sedeped03.unitech.net.br>
Date: Fri, 21 Mar 2003 09:07:45 -0300
```

Exemplo de um email

.

Note que os itens que estão em **negrito** correspondem ao cabeçalho, e, alguns deles serão vistos ainda nesse trabalho. Havendo um maior interesse é necessário consultar a bibliografia.

Mas existe um determinado padrão que é fácil de observar. Todos os cabeçalhos possuem um nome separado por dois pontos e o valor correspondente.

Enviando mensagens

Para enviar uma mensagem na Internet é necessário se conectar a um servidor que suporte algum dos protocolos de envio de mensagem. Nesse trabalho estaremos abordando o protocolo SMTP (Simple Mail Transfer Protocol [RFC821]).

Como regra na Internet, a maioria dos serviços exige que você abra um Socket e estabeleça uma conexão para envio e recebimento de textos.

No protocolo SMTP, a porta padrão é 25. Como estaremos abordando o processo de envio, não iremos tratar de softwares, mas sim do mecanismo definido na RFC. Para poder se conectar a essa porta e conversar com o servidor SMTP, é necessário utilizar o comando TELNET. A chamada seria:

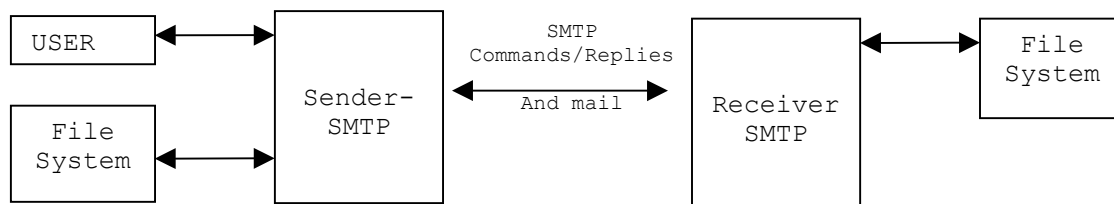
```
telnet servidor 25
```

Caso exista algum servidor SMTP nessa porta, será mostrada a mensagem de saudação e um cursor esperando a entrada de dados:

```
220 nomeservidor NomeSoftwareEmail, Version x.x (x.x.b.b)
```

Como funciona o SMTP? [RFC821]

O modelo SMTP se baseia no seguinte modelo de comunicação: Após o usuário solicitar o envio de email o emissor SMTP estabelece uma comunicação em dois sentidos com o receptor SMTP. Esse receptor pode ser o destino esperado, bem como pode ser algum receptor intermediário. Os comandos SMTP são enviados pelo emissor e enviadas ao receptor. Este por sua vez confirma o recebimento.



Os comandos e respostas do SMTP não são sensíveis a minúsculo e maiúsculos.

Os procedimentos SMTP podem ser divididos em várias partes, porém nesse trabalho somente serão abordados os comandos básicos necessários ao envio

de uma mensagem de forma simples, não sendo considerado o RELAY, ou configurações de servidores.

HELO

Para iniciar o envio do e-mail é necessário que o usuário se identifique informando o nome da sua máquina. Esse comando é uma alusão a “HELLO, I’m <machine name>” (ou, “Oi, eu sou <nome da máquina>”). Atualmente, a grande maioria dos softwares SMTP é capaz de identificar a origem (IP) da máquina ignorando o valor digitado no comando Helo.

```
helo machinename
```

Caso a mensagem tenha sido aceita com sucesso, o servidor deverá retornar um código 250. Na prática, não é necessária nenhuma outra informação adicional, já que o software deve interpretar apenas os três primeiros dígitos da resposta. O restante é considerado informação adicional a ser exibida para o usuário.

MAIL FROM

A próxima etapa é informar quem está enviando o e-mail. Esse nome de usuário será visível no campo FROM do seu leitor de e-mail. Um exemplo está exibido abaixo:

```
Mail from: nome usuario <email@servidor.com>
```

Novamente se a resposta for satisfatória deverá ser devolvido uma mensagem com o código 250.

RCPT TO

Após informar o email do emissor é necessário informar o destinatário da mensagem. Para fazer isso é necessário utilizar o comando:

```
Rcpt to: nome usuario <email@servidor.com>
```

Novamente se a resposta for satisfatória deverá ser devolvido uma mensagem com o código 250.

DATA

Após digitar essa cláusula se dará início à composição da mensagem. Essa mensagem é dividida em duas partes:

- Informações de cabeçalho;
- Corpo da mensagem.

As informações de cabeçalho devem ser as primeiras informações digitadas e seguem as regras definidas na RFC822.

O corpo da mensagem é separado do cabeçalho através de uma linha em branco e é encerrado com um ponto em um linha.

```
Subject: Assunto
Date: Tue, 25 mar 2003 19:00:00 -0300
From: Complete Name and mail
To: Recipients Name and mail

Esta é uma mensagem simples.
.
```

Vendo um exemplo completo

A seguir veremos um exemplo completo de como é uma “conversação” de envio de e-mail. Os símbolos R: e S: correspondem a Response e Send respectivamente.

```
S: MAIL FROM: info@byjg.com.br
R: 250 ok
S: RCPT TO: joao@byjg.com.br
R: 250 ok
S: DATA
R: 354 send the mail data, end with .
S: Date: Tue, 25 mar 2003 19:00:00 -0300
S: From: Informações <info@byjg.com.br>
S: To: Joao Gilberto <joao@byjg.com.br>
S: Subject: Re: Problema de Envio
S:
S:   Recebemos corretamente a sua mensagem e estamos
S:   encaminhando para o departamento correto.
S: .
R: 250 Message send to delivery
```

Recebimento de Mensagens

Na verdade o termo recebimento de mensagens não é adequado, pois, como visto anteriormente quem recebe a mensagem é o servidor SMTP receptor. A intenção do protocolo Post Office Protocol versão 3 (ou simplesmente, POP3) [RFC1081] é possibilitar que uma estação de trabalho com poucos recursos, possa ler as mensagens remotamente, sem ter que possuir um servidor SMTP em sua máquina.

Como funciona o POP3?

O cliente deve se conectar ao servidor POP3 através da porta 110, utilizando o comando telnet.

```
telnet servidor 110
```

Em resposta o servidor devolverá a seguinte mensagem:

```
+OK NomeSoftwareEmail, Version x.x (x.x.b.b)
```

No POP3 as respostas são bem diretas e devolvem apenas +OK e –ERR. Na verdade não era necessária nenhuma mensagem após essa mensagem.

No POP3 os comandos estão divididos em duas camadas: Camada de Autenticação e Camada de Transação. Os comandos de uma camada não podem ser utilizados em outra.

Camada de Autenticação

Na camada de autenticação o usuário deverá fornecer as informações necessárias para se autenticar no servidor POP3. Para que isso seja possível, são utilizados dois comandos:

Comando User

Através da utilização desse comando o usuário estará fornecendo o seu login na conta POP3.

```
User NomeLogin
```

Comando Pass

Após digitar o usuário é necessário informar a senha correspondente ao usuário digitado anteriormente.

```
Pass SenhaSecreta
```

Camada de Transação

Uma vez validado o nome e a senha do usuário, este passa a ter acesso à sua caixa POP3 e sai da camada de autenticação e vai para a camada de transação. Nesta camada são feitas as operações que permitem ler, listar e apagar mensagens dentre outras.

A seguir uma lista dos principais comandos desta camada.

List

Comando sem argumentos que exibe uma listagem contendo todos os Ids de mensagens e o tamanho correspondente.

Top

O comando TOP lista as n primeiras de uma determinada mensagem. Se n for igual a zero, será exibido apenas o cabeçalho.

```
Top 5 30
```

Retr

O comando Retr permite recuperar totalmente uma mensagem.

```
Retr 5
```

Dele

O comando Dele permite marcar para exclusão uma determinada mensagem. Essa mensagem será apagada no momento que o usuário encerrar normalmente a conexão. Caso a conexão com o servidor tenha sido abortada por qualquer motivo essa mensagem não será excluída.

```
Dele 5
```

Quit

O comando Quit permite encerrar uma conexão POP3.

Bibliografia

[RFC822]

<http://rfc.net/rfc822.html>

Formato de como as mensagens são enviadas através da internet

[RFC821]

<http://rfc.net/rfc821.html>

Protocolo de envio de mensagens SMTP (Obsoleta pela rfc2821)

[RFC1081]

<http://rfc.net/rfc1081.html>

Protocolo para recebimento de mensagens (POP3).