**Introdução**

O objetivo principal deste trabalho é descrever de forma simplificada os aspectos básicos do Simple Mail Transfer Protocol (SMTP) e do Post Office Protocol versão 3 (POP3).

**SMTP – Simple Mail Transfer Protocol**

Para que as mensagens de correio eletrônico sejam enviadas pelo protocolo SMTP são necessários dois componentes básicos: o remetente e destinatário-SMTP.

O remetente-SMTP é aquele agente que fica responsável de enviar ao destinatário-SMTP a mensagem através de uma comunicação por comandos definidos pelo próprio SMTP. O destinatário-SMTP é o agente final, no caso de ser o destino da mensagem de fato, ou cada agente intermediário, que irá retransmitir a mensagem recebida para o próximo destino, fazendo o papel de remetente-SMTP, até que ela chegue ao seu destinatário propriamente dito.

As mensagens de correio eletrônico são enviadas quando um host se conecta com a porta 25 de outro host via conexão TCP. Um daemon fica atento nessa porta aceitando as conexões e copiando as mensagens ao seu destino (caixas de correio eletrônico).

O SMTP é um protocolo ASC II bem simples, a transmissão ocorre de uma máquina (agindo como cliente) para com outra máquina (agindo como servidor), o servidor que fica responsável de iniciar a comunicação.

No processo de comunicação o servidor envia sua identidade e diz se está apto ou não a receber as mensagens, senão estiver o remetente (cliente) encerrará a comunicação, do contrário, se estiver apto, o cliente anunciará de quem veio a mensagem e informará para onde ela está indo, se o endereço de destino estiver presente no servidor, então será enviado uma mensagem do servidor dizendo que pode enviar.

Como mensagem pode ser enviada para vários endereços de e- mail, então o remetente – SMTP tenta-se conectar com a maior quantidade de destinatários possíveis, para assim uma fazer economia de conexões. Isto faz com que haja uma melhor distribuição de processamento e, consequentemente, uma economia maior de tempo de conexão a outros hosts.

Nesse modelo uma rota pela qual passou a mensagem até chegar ao seu destinatário pode ser definida, já que quando um destinatário-SMTP (agente intermediário) recebe a mensagem ele insere no cabeçalho o remetente anterior no início da mensagem.

Na comunicação SMTP não é necessária verificação, já que o TCP fornece um fluxo de bytes confiável, quando não houver mais mensagens a conexão entre as máquinas serão encerradas.

O interpretador SMTP não é case-sensitive, ou seja, os comandos Mail, mail, MaiL são considerados iguais, porém o mesmo não de vale para os endereços de e-mail e alguns hosts. Apesar da sintaxe dos comandos enviadas pelo cliente ser bem rígida, a sintaxe da resposta é mais flexível, onde apenas o código numérico é importante.

O uso das linhas de texto ASCII para comando deve-se ao fato de que a maioria dos protocolos da internet funcionarem desse modo, já que o seu uso torna os protocolos mais fáceis de testar e depurar, um dos meios testes pode ser enviando comandos manualmente e verificando as suas saídas.

Apesar de ser considerado um protocolo bem definido, o SMTP pode ter alguns problemas, como por exemplo o tamanho das mensagens (dependendo do modo de implementação). Outro problema que pode surgir é causado pelos timeouts, pois se o cliente e servidor possuírem timeouts diferentes, um deles poderá desistir enquanto o outro ainda está ocupado.

Para contornar alguns desses problemas, o STMP estendido (ESMTP) foi definido na RFC 2821. Os clientes que quiserem usá-lo deverão enviar inicialmente uma mensagem EHLO, em vez de HELO. Se essa mensagem for rejeitada, isso significa que o servidor é um servidor STMP comum, e o cliente deverá proceder da forma habitual. Se EHLO for aceita, novos comandos e parâmetros serão permitidos.

Com o tempo surgiu a necessidade de enviar mensagens para pessoas que estejam off-line (o que não era possível há um tempo atrás devido ao esquema de comunicação), como solução podemos ter um agente de transferência de mensagens em uma máquina do ISP aceitando mensagens de correio eletrônico para seus clientes e armazenando nas suas caixas de correio, uma vez que o agente (ISP) pode estar online o tempo todo.

**Comandos SMTP**

É importante saber que todo comando é feito no sentido remetente-destinatário, ou seja, somente o remetente requer os comandos, enquanto o destinatário os executa.

Como dito antes os comandos SMTP são compostos de caracteres do código ASCII. Como o código ASCII segue o padrão 7-bit, quando o serviço de transporte utilizado provê um canal baseado no padrão 8-bit, é necessária uma conversão do formato 7-bit para o formato 8-bit. Neste caso, para o bit de mais alta ordem é dado o valor zero (falso).

**HELO**

Este comando é usado pelo remetente para que possa identificar-se para o destinatário, este é o primeiro comando executado pelo protocolo SMTP. Sua sintaxe se constitui dos seguintes elementos:

HELO <espaço> <domínio-do-agente-remetente> <CRLF>

O elemento <domínio-do-agente-remetente> representa exatamente o agente que está fazendo o papel de remetente para a transmissão.

Para o exemplo a seguir e os subsequentes o destinatário será representado (D) remetente será representado por (R).

*D: 220 redes.com.br Simple Mail Transfer Service Ready*

*R: HELO ucsal.br D: 250 redes.com.br*

Neste exemplo supomos que o host “redes.com.br” funciona com o serviço SMTP, que está sendo acessado por um cliente (“ucsal.br”), que se identifica e recebe uma resposta positiva do destinatário.

**QUIT**

Este é o comando usado para encerrar a comunicação entre dois hosts que se comunicam pelo protocolo SMTP, sua sintaxe é dada por:

QUIT <CRLF>

Exemplo:

*R: QUIT*

*D: 221 redes.com.br Service closing transmission channel*

**MAIL**

Este comando SMTP serve basicamente para informar ao destinatário o endereço de correio do remetente da mensagem, e para casos onde a mensagens passa por agentes intermediários ele também informa o caminho reverso por onde a mensagem já passou.

No momento em que este comando é executado, uma nova transação está sendo iniciado, assim quando o comando MAIL é executado os buffers de dados e destinatários são apagados (“resetados”).

A sintaxe para este comando é a seguinte:

MAIL <espaço> FROM: <caminho-reverso> <CRLF>

Exemplos:

*R: MAIL FROM:<aluno01@ucsal.br*

*D: 250 OK*

Ou

*R: MAIL FROM:<@ucsal.br:joao@teleprocessamento.com.br*

*D: 250 OK*

**RCPT**

O comando RCPT é o responsável pelo direcionamento da mensagem para um destinatário.

O comando RCPT pode ser repetido várias vezes em uma só transação para ter a possibilidade de entregar a mesma cópia de mensagem para vários destinatários. A cada vez que o comando é executado o destinatário pode responder positivamente ou negativamente ao recebimento da mensagem.

Sua sintaxe se baseia nos seguintes elementos ordenados desta forma:

RCPT <espaço> TO: <caminho> <CRLF>

Tanto o elemento <caminho>, usado neste comando, como também o elemento <caminho-reverso>, usado no comando MAIL são formados por uma lista opcional de hosts e um endereço de correio, por exemplo:

<@ucsal.br,@teleprocess.com.br,@informatica.eti.br:jose@redes.com.br>

Caso este elemento seja considerado um caminho a mensagem passará no sentido do @ucsal.br até [jose@redes.com.br](mailto:jose@redes.com.br).

Caso contrário, sendo ele considerado um caminho reverso, o primeiro host passado no caminho será o último host pela qual a mensagem irá passar.

**DATA**

Após serem aceitos os comandos RCPT, o remetente poderá enviar o comando DATA para requerer a mensagem transmitida.

Um exemplo de uma requisição através desse comando:

*R: DATA*

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

*R: Bla bla bla bla bla bla bla bla ...*

*R: No no no no no no no no no no ...*

*R: <CRLF>.<CRLF>*

*D: 250 OK*

**SEND**

Este comando é semelhante ao comando MAIL, sendo que ele é feito para entregar mensagens a um ou mais terminais de usuário, caso eles estejam ativos. Caso contrário é retornada uma resposta de código 450 para o comando RCPT correspondente. Exemplo de uso do comando:

*R: SEND FROM:<aluno01@alunos.com.br>*

*D: 250 OK*

**SOML**

Também semelhante ao MAIL, este comando requer que o correio seja entregue a um ou mais terminais de usuário, caso este estejam ativos no host. Se não estiverem ativos ou não aceitarem mensagens de terminal, a mensagem é, então, entregue à caixa de mensagens do usuário. Exemplo:

*R: SOML FROM:[aluno01@alunos.com.br](mailto:aluno01@alunos.com.br)*

*D: 250 OK*

**SAML**

Este comando, como os demais, além de requerer que o correio seja entregue a um ou mais terminais de usuário, caso este estejam ativos no host, ele também exige que a mensagem seja entregue na caixa de mensagem de usuário. Independente de não estarem ativos ou não aceitarem mensagens de terminal, a mensagem é também entregue à caixa de mensagens do usuário.

*R: SAML FROM:<aluno01@alunos.com.br>*

*D: 250 OK*

**RSET**

Este comando indica que a transação corrente está sendo abortada. Neste caso, todos os buffers e tabelas de estado são limpados e o agente destinatário deve responder um OK.

**VRFY**

Este comando pede ao agente destinatário que confirme o argumento que está sendo passado que identifica um usuário. Se o argumento é o nome do usuário, então é retornado o nome completo do usuário e sua completa caixa postal.

**EXPN**

Parecido com o VRFY, o EXPN deve receber como argumento o nome de uma lista de e-mail (mailing list) e retornar, se for possível, todos os nomes de usuário e suas caixas postais correspondentes.

**HELP**

Este comando permite que o agente remetente receba algumas informações de ajuda. Acompanhado de algum argumento, este comando pode devolver uma resposta mais específica.

**NOOP**

Este comando não tem nenhuma funcionalidade para o sistema. Significa não fazer operação alguma e sempre terá como resposta um OK.

**TURN**

Este é um interessante comando do SMTP que faz a troca das funções dos dois agentes que estão conectados, ou seja, quando o agente remetente envia este comando para o destinatário e recebe uma resposta positiva, o remetente passa a ser o destinatário e, da mesma forma, o destinatário passa a ser o remetente de uma nova transação que é, neste momento, iniciada.

**POP3 – Post Office Protocol**

Para que as mensagens de correio eletrônico sejam resgatadas pelo protocolo POP3 são necessários dois componentes básicos: a máquina cliente e um servidor que oferece serviço POP3.

O cliente-POP3 é aquele agente que fica responsável de enviar ao servidor-POP3 a mensagem através de uma comunicação por comandos definidos pelo próprio POP3. O servidor-POP3 é o agente que irá retransmitir a mensagem recebida para o destinatário propriamente dito.

O serviço POP3 é de fácil acesso e simples de usar e foi definido na RFC 1939. Por ser um protocolo simples tem sua funcionalidade bastante limitada. O POP3 te inicio quando o agente usuário (cliente) abre uma conexão TCP como o servidor de correio (servidor) na porta 110 ou 995 se a ligação utilizar o protocolo SSL Após o estabelecimento da conexão, o servidor POP3 envia uma mensagem de saudação ao cliente, a qual confirma a ligação ao servidor.

A partir do momento em que a comunicação foi aberta o cliente de correio esta constantemente trocando mensagens com o servidor POP3 até a comunicação ser fechada.

Os comandos interpretados pelo POP3 são palavras-chave seguidas ou não de argumentos e o protocolo não faz distinção entre letras maiúsculas ou minúsculas sendo que todos os comandos terminam com um par CRLF, ou seja, mudança de linha. As palavras-chave e os argumentos são separados pelo caractere espaço. As palavras-chave têm de 3 a 4 caracteres e os argumentos no máximo 40 caracteres e ambos são formados pelos caracteres ASCII.

As respostas no POP3 são constituídas por um indicador de estado e uma palavra-passe, seguidos ou não de informações adicionais e terminadas por um CRLF. As respostas podem ter até 512 caracteres de comprimento, incluindo o CRLF na versão POP3 os indicadores são “+OK” e “-ERR”.

Com a conexão aberta TCP ativada, o protocolo passa por três fases: autorização, transação e atualização. Durante a primeira fase o cliente manda envia nome de usuário e senha (às claras) e obtém uma resposta do servidor. Após a validação do utilizador, o protocolo bloqueia a pasta de entrada para prevenir que mensagens possam ser alteradas ou removidas antes da entrada em modo de atualização. No modo de transação, o usuário pode efetuar comandos repetidos e também é nesta fase que o usuário pode marcar as mensagens que serão apagadas. A última fase ocorre logo após o cliente ter dado o comando QUIT que encerra a sessão POP3 no servidor e também apaga as mensagens marcadas.

Há um problema com o POP3, pois o destinatário pode ser um nômade, ou seja, ele pode querer acessar seu correio de muitas maquinas, por exemplo, do PC do seu escritório, do PC de sua casa e de seu computador portátil. Se ele utilizar o protocolo POP3 e se ler uma de suas mensagens no escritório ele não poderá ler esta mesma mensagem mais tarde em computador portátil.

**Comandos POP3**

**Comandos no estado de Autorização:**

**QUIT**

Este é o comando usado para encerrar a comunicação entre dois hosts que se comunicam pelo protocolo POP3, sua sintaxe é dada por:

QUIT <CRLF>

Exemplo:

*C: QUIT*

*S: +OK dewey POP3 server signing off*

**Comandos no estado de Transação:**

**STAT**

Este comando POP3 nos retorna uma resposta positiva seguido do número de mensagens na pasta de entrada e do espaço e tamanho da pasta de entrada em octetos.

A sintaxe para este comando é a seguinte:

STAT <CRLF>

Exemplos:

*C: STAT*

*S: +OK 6 480*

**LIST**

O comando LIST, quando enviado sem parâmetros, terá uma resposta positiva do servidor seguido do número de mensagens na pasta de entrada e efetua uma listagem em modo multilinha. Quando enviado com parâmetro (opcional) o número de uma determinada mensagem, este não podendo se referir a uma mensagem marcada como eliminada, o servidor responde com uma linha que contem a informação sobre a mensagem indicada.

A sintaxe para este comando é a seguinte:

LIST <N>

Exemplos:

*C: LIST*

*S: +OK 2 messages (320 octets)*

*S: 1 100*

*S: 2 210*

*S:.*

*…*

*C: LISTA 2*

*S: + OK 2 210*

*...*

*C: LISTA 3*

*S: -ERR no such message, only 2 messages in maildrop*

**RETR**

Ao executar o comando RETR , o servidor POP3 envia uma resposta positiva seguida da informação relativa à mensagem especificada na forma multilinha.

A sintaxe para este comando é a seguinte:

RETR <N>

Exemplos:

C: RETR 1

S: +OK 130 octets

S: <the POP3 server sends the entire message here>

S: .

**DELE**

Ao executar o comando RETR, o servidor POP3 marca a mensagem como eliminada, porem, só será removida definitivamente quando a sessão passar para o modo ATUALIZAÇÃO. E ao tentar executar qualquer comando antes de passar para o modo ATUALIZAÇÃO, referente a mensagem apagada, será gerada uma mensagem de erro.

A sintaxe para este comando é a seguinte:

DELE <N>

*Exemplos:*

*C: DELE 1*

*S: +OK message 1 deleted*

*...*

*C: DELE 2*

*S: -ERR message 2 already deleted*

**NOOP**

Ao executar o comando NOOP, o servidor POP3 apenas envia uma resposta positiva.

A sintaxe para este comando é a seguinte:

*DELE <CRLF>*

Exemplo:

*C: NOOP*

*S: +OK*

**RSET**

Ao executar o comando RSET, se existir mensagens marcadas como deletadas, o servidor POP3 enviará uma resposta positiva e irá desmarcá-las, informando o número de mensagens desmarcadas.

*Examples:*

*C: RSET*

*S: +OK maildrop has 3 messages (410 octets)*

***Comandos no estado de Atualização:***

**QUIT**

Este é o comando usado para encerrar a comunicação entre dois hosts que se comunicam pelo protocolo POP3, sua sintaxe é dada por:

QUIT <CRLF>

Exemplo:

*C: QUIT*

*S: +OK dewey POP3 server signing off*

***Comandos Opcionais:***

**TOP**

Ao executar este comando, o servidor POP3 envia o cabeçalho da mensagem pedida e o numero de linhas informado do corpo da mensagem em multilinhas, sua sintaxe é dada por:

TOP <N> <N>

Exemplo:

*C: TOP 1 10*

*S: +OK*

*S: <CABEÇALHO, LINHA EM BRANCO, 10 PRIMEIRAS LINHAS DO CORPO DA MENSAGEM>*

**UIDL**

Ao executar este comando seguido do número de uma determinada mensagem, se for válido, o servidor POP3 enviará uma confirmação e a lista de identificadores únicos para a mensagem indicada. Se não for informado o número da mensagem, o servidor POP3 enviará uma confirmação e listará os identificadores únicos em forma multilinha das mensagens disponiveis.

Sua sintaxe é dada por:

UIDL <N>

Exemplo:

C: UIDL

S: +OK

S: 1 whqtswO00WBw418f9t5JxYwZ

S: 2 QhdPYR:00WBw1Ph7x7

S: .

...

C: UIDL 2

S: +OK 2 QhdPYR:00WBw1Ph7x7

...

C: UIDL 3

S: -ERR no such message, only 2 messages in maildrop

**USER**

Este comando, em conjunto com o comando PASS é utilizado para fazer a autenticação no servidor, sendo necessário informar seu nome de usuário.

Sua sintaxe é dada por:

USER <Nome>

Exemplo:

C: USER frated

S: -ERR sorry, no mailbox for frated here

...

C: USER mrose

S: +OK mrose is a real hoopy frood

**PASS**

Após o retorno positivo do comando USER, este comando é utilizado para fazer a autenticação da senha do usuário no servidor.

Sua sintaxe é dada por:

PASS <Senha>

Exemplo:

CC: USER mrose

S: +OK mrose is a real hoopy frood

C: PASS secret

S: -ERR maildrop already locked

...

C: USER mrose

S: +OK mrose is a real hoopy frood

C: PASS secret

S: +OK mrose's maildrop has 2 messages (320 octets)

**APOP**

Também utilizado para autenticação, informando o nome de usuário e uma chave MD5, sendo mais seguro que o método USER/PASS, pois é feita de maneira encriptada. O servidor com esta funcionalidade irá incluir um carimbo de tempo, definido no RFC822 'msg-id', e a chave MD5 é composta por 16 octetos, será calculada aplicando um algoritmo definido na RFC1321 no carimbo de tempo seguido de uma palavra-chave partilhada entre usuário e servidor.

Sua sintaxe é dada por:

APOP <Nome> <Chave>

Exemplo:

S: +OK POP3 server ready <1896.697170952@dbc.mtview.ca.us>

C: APOP mrose c4c9334bac560ecc979e58001b3e22fb

S: +OK maildrop has 1 message (369 octets)

**Bibliografia**

Kurose, James F., Keith W. Ross, and Wagner Luiz Zucchi. *Redes de Computadores ea Internet: uma abordagem top-down*. Pearson, 2010.

TANENBAUM, Andrew. "S. Redes de Computadores. São Paulo: Ed." (2003).

Carvalho, Carlos Eduardo Teixeira. “SIMPLE MAIL TRANSFER PROTOCOL (SMTP)”, Universidade Católica do Salvador. Abril de 2001.

J. G. Myers, M. T. Rose, Post Office Protocol – Version 3 – Standards Track, RFC 1939, maio 1996.