UNIVERSIDADE DO ESTADO DE MATO GROSSO (UNEMAT)
FACULDADE DE CIÊNCIAS EXATAS E TECNOLÓGICAS
CURSO DE SISTEMAS DE INFORMAÇÃO

LUCAS PEREIRA RIBEIRO CORRÊA
MARCOS JUNIOR DA SILVA

PROTOCOLO TCP/IP
CAMADA DE APLICAÇÃO

SINOP-MT
2023

LUCAS PEREIRA RIBEIRO CORRÊA
MARCOS JUNIOR DA SILVA

PROTOCOLO TCP/IP
CAMADA DE APLICAÇÃO

Trabalho apresentado para a Disciplina Introdução às redes de computadores, pelo Curso de Sistemas de Informação da Universidade do Estado de Mato Grosso (UNEMAT), ministrada pelo Prof. João Ricardo dos Santos Rosa.

SINOP-MT
2023

SUMÁRIO

1 INTRODUÇÃO

Este relatório foi elaborado a partir de pesquisas realizadas acerca do tema "Camada de Aplicação", juntamente com seus subtemas. Esse conteúdo aborda a importância da camada de aplicação no contexto das redes de computadores, enfatizando seu papel fundamental ao oferecer uma interface “comum” aos usuários e aplicativos que utilizam a rede para comunicação. Em relação à camada da pilha de protocolos, o foco está na implementação dos serviços da aplicação, considerando que os protocolos inferiores já têm garantido a segurança necessária para a transmissão de dados. Para o usuário comum, os serviços de aplicação são frequentemente os aspectos mais reconhecíveis e visíveis do uso da rede. Ao longo deste relatório, abordaremos alguns exemplos notáveis de aplicações de rede, como o DNS, a World Wide Web, o e-mail, o Protocolo de Transferência de Arquivos (FTP) e o Telnet.

2 OBJETIVO

O objetivo deste artigo é examinar a Camada de Aplicação nas redes de computadores, destacando sua importância ao fornecer uma interface comum para usuários e aplicativos que utilizam a rede para comunicação. Vamos explorar a implementação dos serviços de aplicação, com ênfase na relevância prática da Camada de Aplicação, e analisar exemplos notáveis de aplicações de rede, como os subtemas mencionados acima, na introdução.

3 CAMADA DE APLICAÇÃO

Depois de passarmos por todas as camadas preliminares, chegamos à camada em que são encontradas todas as aplicações. As camadas situadas abaixo da camada de aplicação têm a função de oferecer um serviço de transporte confiável, mas, na verdade, elas não executam qualquer tarefa para os usuários. Neste capítulo, estudaremos algumas aplicações reais de redes.

No entanto, mesmo na camada de aplicação existe a necessidade de protocolos de suporte, a fim de permitir que as aplicações funcionem. Da mesma forma, examinaremos um desses protocolos antes de iniciarmos o estudo das aplicações em si. O item em questão é o DNS, que cuida da nomenclatura na Internet. Depois disso, examinaremos três aplicações reais: correio eletrônico, a World Wide Web e, por fim, multimídia.

3.1 DNS — Domain Name System

3.1.1 Introdução

Em sistemas distribuídos nomes são utilizados para se referir a uma grande variedade de recursos do sistema, como computadores, portas, serviços e outros objetos do sistema. Tais nomes são necessários para a comunicação entre componentes do sistema e para compartilhamento de recursos (Marcial Porto Fernández, 2019). A autora apresenta a seguir alguns conceitos importantes.

Serviços de nomes

Permite a ligação de um nome a um conjunto de atributos relacionados a este nome. A mais frequente operação que é solicitada a um serviço de nomes é a resolução de um nome, i.e.,a procura dos atributos relacionados a um determinado nome.

Espaço de nomes

É uma coleção de nomes sintaticamente válidos reconhecidos por um sistema de resolução de nomes. Ex.: /usr/home em sistema de arquivos Unix.

Contextos

A resolução de um determinado nome nem sempre se dá de maneira direta, isto é, não solicitamos ao serviço de nomes a simples resolução de um nome absoluto (plano). Geralmente o nome é identificado dentro de um contexto. Um contexto funcionaria de maneira análoga ao sistema de diretório: um diretório definiria um contexto para para a resolução dos nomes. Assim: /home/ jose/arquivo1 (a) /home/maria/arquivo1 (b) O nome arquivo1, quando apresentado ao serviço de resolução de nomes com o contexto /home/jose retornaria uma referência ao objeto do sistema indicado por (a) que, não necessariamente, seria o mesmo objeto que (b). Como podemos ver um mesmo nome pode aparecer em contextos diferentes referenciando objetos diferentes.

Um serviço de nomes que não permite a definição de mais que um contexto para o seu espaço de nomes é dito possuir um espaço de nomes flat. Para um espaço de nomes flat existe somente um único contexto. De volta a nossa analogia com o sistema de diretório, um sistema flat seria um sistema de diretório que só possuísse um único diretório: o raiz. Neste caso, todos os nomes são resolvidos de maneira global, absoluta, sempre em relação ao único contexto existente.

Domínio de Nomes

É um espaço de nomes para o qual existe uma única e geral autoridade administrativa. Esta autoridade determina quais nomes podem ser inseridos/ removidos dentro de seu espaço de nomes. Resolução de Nomes Em geral, a resolução de um nome é um processo iterativo em que um nome é apresentado repetidas vezes a diferentes contextos de nomes. Assim para resolvermos o nome/home/jose/arquivo1, teríamos os seguintes passos:

• Apresentamos o nome /home/jose/arquivo1 ao sistema.

• O nome home é resolvido então no contexto raiz, retornando um identificador válido, ou então uma condição de erro.

• Caso o valor de retorno seja um identificador válido, apresentamos o nome “jose” ao contexto /home.

• Novamente, se o valor de retorno for um identificador válido, prosseguimos com a resolução de nosso nome. Apresentamos arquivo1 ao contexto /home/jose e,

finalmente, nos é retornado um identificador (ou outro atributo) para o nome /home/jose/arquivo1.

3.1.2 Histórico

Divisão do espaço de nomes em contextos. Inicialmente, o espaço de nomes da Internet era flat e administrado por uma única entidade centralizadora, responsável pelo único contexto então existente. Como o número de nomes cresceu muito, ficou impossível para tal entidade administrar um espaço de nomes gigantesco, bem como prestar serviços a todos os demais usuários. Surgiu então o DNS, um sistema de resolução de nomes distribuído, onde existiam vários domínios: subespaços de nomes administrados localmente.

Assim, por exemplo, uma universidade americana era responsável pelo seu domínio, podendo ela determinar a inclusão e remoção de nomes de seu espaço de nomes, bem como incumbida de ajudar na solução de nomes que refere-se a um dos contextos pertencentes a seu domínio. Todo domínio possui uma única autoridade sobre as operações de pesquisa e atualização de seu espaço de nomes.

3.1.3 Domínio

Como posso mandar uma carta para você? Este tipo de pergunta resulta numa resposta do tipo: Rua, Número, Bairro, Cidade, Estado, País e um Número (CEP) para facilitar a identificação do logradouro, para envio de cartas sem contar com o seu nome que aparece na carta. Para que você possa se comunicar com outro usuário existe também um endereço eletrônico; assim sendo a estrutura apresentada para endereços é: Usuário@domínio.

O domínio pode ser dividido em várias partes chamadas subdomínios. Estas partes estão separadas por ponto (“.”) Assim se tivessem o seguinte endereço eletrônico: joao@servidor.empresax.com.br.

O subdomínio mais à direita é o domínio de maior nível, conforme você ler os subdomínios mais à esquerda eles tornarão mais específicos o possível; O domínio de maior nível indica em que país se encontrará, a ausência do domínio de maior nível, indica que a máquina estará no EUA (isto é óbvio, foi onde iniciou-se a Internet). No nosso caso o domínio de maior nível é ***.br*** (Brasil).

O subdomínio define o tipo de instituição ao qual pertence. No nosso exemplo, o ***.com*** define um domínio comercial (pertencente a uma empresa). Veja na tabela os tipos de subdomínio padronizados. O próximo subdomínio empresa x seria a indicação da empresa que mantém este endereço eletrônico e o último subdomínio servidor seria o nome de uma máquina específica que a empresa mantém onde reside o programa de e-mail. Você poderá ver domínios divididos em mais de 3 subdomínios porém, não existem na Internet domínios divididos em menos de 2 subdomínios.

3.1.4 DNS

Imagine agora a sua necessidade de se comunicar com uma máquina ou com um usuário. Como sabemos, os computadores trabalham com números e nós não somos capazes de decorar todos os números possíveis. Um exemplo 197.168.8.30, 149.82.34.11, 149.82.34.4, deste modo fica difícil saber onde estão estas máquinas (se no Brasil ou no Exterior).

Por isso foi criado o FQDN, ou seja, Nome do Domínio Completamente Qualificado, ou mais conhecido como DNS. Assim, o número 197.168.8.30 é da máquina server.empresax.com.br, o endereço 149.82.34.11 é da marte. escola.edu.br, o endereço 149.82.34.4 é da vênus.escola.edu.br, agora ficou mais fácil pois vemos que o primeiro endereço é de uma máquina na empresa x e os outros dois são de máquinas da escola.

O último domínio indicará o país em que esta máquina está situada, o segundo domínio indica a instituição que detém o domínio e o primeiro nome indica uma máquina, ou mais especificamente, um endereço IP, o que chamamos de Endereços Internet Protocol (IP). Os IP 's são constituídos por quatro números unidos por pontos (’.’) chamados de quádrupla ou dotted quad e cada pedaço deste número é chamado de octeto. As informações na rede são passadas pelo números IP’ s e não pelo FQDN. O programa DNS faz todo o “trabalho árduo” para passar de FQDN para IP ou de IP para FQDN.

3.2 World Wide Web (WWW)

O WWW (World Wide Web), é o maior aplicativo existente na Internet. Sua facilidade de uso e simplicidade permitiram o grande crescimento da Internet no mundo. De fato é tedioso utilizar serviços como FTP, Telnet, conhecer e aprender todos os requisitos e comandos. Com o acesso a WWW você pode escolher o que pesquisar, sem muitas complicações e sem ao menos saber em que local da rede ele se encontra, o WWW utiliza o método de Hipertextos, são textos que em certas palavras podemos “clicá-las” e assim navegando-se dentro da rede, podendo salvar telas, imprimi-las, copiar programa, ou seja, é um dos estágios finais de aglutinação de tudo o que é possível fazer e se obter dentro da rede.

O WWW teve início no CERN (Centre Européen Recherche Nucléaire), a ideia era padronização de tudo o que é possível conectar-se à rede Internet, podendo assim utilizar a rede como um todo e um enorme banco de dados, com a existência de vários tipos de dados,daí a sua implementação ao suporte de multimídia. Foi então criado o protocolo HTML (Hyper Text Mark-up Language) e os dados em relação a um servidor e cliente como HTTP (Hypertext Transfer Protocol).

A ideia é utilizar páginas com textos, imagens e ícones fazendo com que da combinação destes elementos, obtenhamos uma tela visível, com as opções de mover-se entre estas páginas. Resumindo, seria como se você tivesse virado folhas de um livro.

Normalmente utilizamos um software específico chamado de cliente WWW ou Navegador (browser). Como exemplo, podemos destacar o Mozilla, o Opera, o Chrome, o Internet Explorer, dentre outros. Existem implementações para qualquer plataforma, seja ela PC Windows, MAC ou o ambiente Windows do UNIX.

Os componentes do modelo Web são mostrados na Figura 3.2.1.

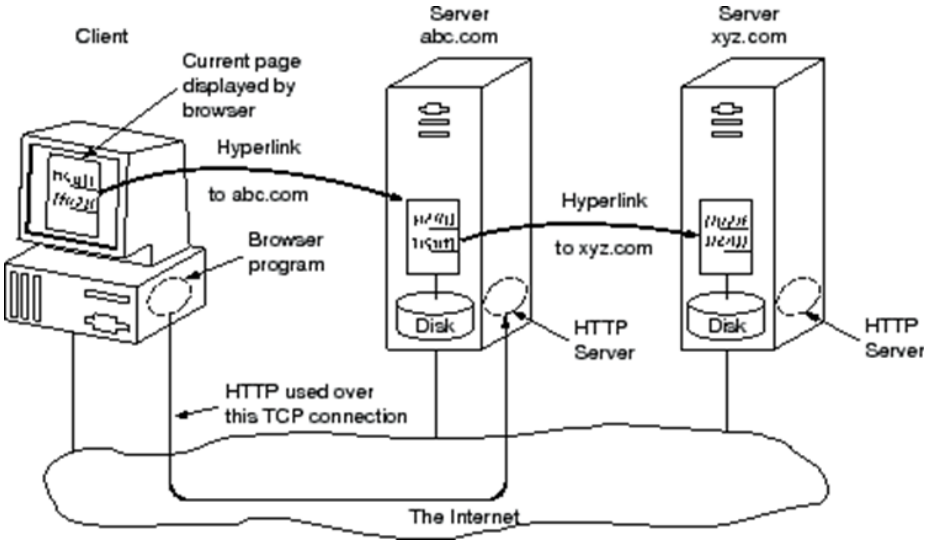

Figura 3.2.1: Componentes do Modelo Web.

3.2.1 Protocolo HTTP (Hypertext Transfer Protocol)

O protocolo HTTP é um protocolo do nível de aplicação que possui objetividade e rapidez necessárias para suportar sistemas de informação distribuídos, cooperativos e de hipermídia. Suas principais características são as seguintes:

• propiciam busca de informação e atualização

• as mensagens são enviadas em um formato similar aos utilizados pelo correio eletrônico da Internet e pelo MIME (Multipurpose Internet Mail Extensions)

• comunicação entre os agentes usuários e gateways, permitindo acesso a hipermídia a diversos protocolos do mundo Internet, tais como, SMTP, NNTP, FTP, etc

• pode ser implementado em cima de qualquer protocolo Internet

• obedece ao paradigma de pedido/resposta: um cliente estabelece uma conexão com um servidor e envia um pedido ao servidor, o qual o analisa e responde. A conexão deve ser estabelecida antes de cada pedido de cliente e encerrada após a resposta. As mensagens seguem o formato da RFC822 e se apresentam na forma de:

• Pedidos enviados pelo cliente ao servidor

• Respostas enviadas pelo servidor para o cliente

3.2.2 Métodos

Método indica a forma a ser aplicada para requisitar um recurso. Os métodos aceitos por um determinado recurso podem mudar dinamicamente. O cliente é notificado

com o código 501 quando o método é desconhecido ou não implementado. Os métodos são sensíveis ao caso. Os principais métodos são:

GET Recupera todas as informações identificadas no recurso da rede. Se o recurso for um processo executável, ele retornará a resposta do processo e não o seu texto. Existe o GET condicional que traz o recurso apenas se o mesmo foi alterado depois da data da última transferência. HEAD Semelhante ao método GET, só que neste caso não há a transferência da entidade para o cliente. Este método é utilizado para testar a validade e acessibilidade dos links de hipertexto. POST Utilizado para solicitar que o servidor destino aceite a entidade constante no pedido como um novo subordinado ao recurso constante no URI. Suas principais funções são:

1. anotações de recursos existentes
2. postar uma mensagem em um bulletin board, newsgroup, mailing list
3. abastecer um processo com um bloco de dados
4. estender uma base de dados com uma operação de append

A entidade é subordinada da mesma forma que um arquivo é subordinado ao diretório, o registro da base de dados PUT coloca a entidade abaixo do recurso especificado no pedido. Se esta entidade não existe, é criada. Se existe, apenas é atualizada DELETE Solicita que o servidor origem apague o recurso identificado no URI LINK Estabelece uma ou mais relações de links entre o recurso identificado pelo URI e outros recursos existentes, não permitindo que o corpo da entidade enviada seja subordinada ao recurso UNLINK Remove uma ou mais relações de links existentes entre o recurso identificado no URI.

URI (Uniform Resource Identifiers) URI identifica o recurso da rede. Por exemplo: server.escola.edu.br/pub/rfc/rfc822.txt Quando o caractere “*” aparece antes do recurso da rede, o mesmo indica que a requisição não se aplica ao recurso de rede especificado e sim ao seu servidor.

3.2.3 Cabeçalhos

O pedido possui três cabeçalhos distintos:

- Cabeçalho Geral São dados complementares não relacionados com as partes que estão se comunicando nem com o conteúdo sendo transferido. Exemplo: data, versão do MIME.
- Cabeçalho de Resposta Informações adicionais sobre o pedido e o cliente, como por exemplo, o intervalo de respostas esperadas no processamento da requisição.
- Cabeçalho da Entidade: Informações adicionais sobre a entidade, tais como: título da entidade, tamanho, linguagem utilizada.

3.2.4 Códigos de Resposta

O status representa o resultado do processamento executado pelo servidor. O status possui o seguinte formato: 9XX, onde 9 representa a classe da resposta e XX representa a categoria da resposta.

Atualmente o protocolo possui cinco classes de respostas:

1XX

Não utilizada, reservada para utilização futura

2XX Sucesso:

A ação foi recebida, entendida e executada com sucesso. Com o método GET, a entidade correspondente é enviada com a resposta. Com o método HEAD, a resposta contém o cabeçalho da informação. Com o método POST, a resposta descreve/contém o resultado da ação. Nos restantes, a resposta descreve o resultado da ação. Exemplos:

201 - Um novo recurso foi criado

202 - O pedido foi aceito para processamento, mas o mesmo não foi concluído

3XX Redirecionamento:

indicam que as ações devem ser efetuadas em ordem para completar o pedido. Exemplos:

300 - O recurso requisitado está disponível em mais de um local e o local preferido não pode ser determinado via negociação

302 - O recurso requisitado reside temporariamente em outro URI

4XX Erro no cliente:

O pedido contém erro de sintaxe ou não pode ser efetuado. Exemplos:

401 - O recurso requisitado necessita autenticação do usuário

404 - O servidor não encontrou o recurso definido no URI

5XX Erro no servidor:

O servidor falhou ao executar um pedido aparentemente válido. Exemplos:

500 - Erro interno no servidor

501 - Recurso solicitado não implementado no servidor

3.3 Correio Eletrônico (Eletronic Mail ou E-Mail)

A troca de correspondência eletrônica é o segundo recurso mais utilizado na Internet. Como em cada máquina existe um programa diferente que controla suas cartas, estes programas além de usar o mesmo protocolo TCP/IP, devem receber, enviar, reenviar, salvar, imprimir e apagar as cartas lidas. Apresentaremos apenas os comandos genéricos, existentes em qualquer programa.

Um e-mail é semelhante a uma carta escrita em papel. Uma carta é constituída por um envelope e um conteúdo (a carta propriamente dita). O e-mail tem um “envelope” que contém o nome e endereço do destinatário, o nome e endereço do remetente e uma linha opcional para escrever o assunto. O conteúdo é onde escrevemos a carta, e geralmente é como um editor de textos.

O nome e endereço é simplesmente o endereço padrão Internet, isto é, nome@domínio. Somente com essa sigla podemos identificar o destinatário e o remetente. Geralmente o endereço do remetente é definido no próprio programa de e-mail e não necessita ser escrito quando se envia um e-mail. Também não é necessário datar, pois o programa já inclui a hora e data local.

Se quisermos enviar uma carta para o joão que está no endereço empresax.com no Brasil fica assim:

Mail to: joao@empresax.com.br

Cc:

Subject: Saudações

Olá, João, como vai?

Abraços,

Maria

Observe que acentos da língua portuguesa geralmente não são usados no endereço, pois não são compreendidos por muitos equipamentos (apesar de cada vez mais ser aceito). Além disso, não se deve colocar acentos no texto escrito, mesmo que o programa de e-mail permita. Alguns equipamentos na Internet (por exemplo, proxies), não entendem os caracteres ASCII estendidos, e trocam os caracteres acentuados por outros caracteres que poderão tornar o e-mail incompreensíveis pelo destinatário. Assim, principalmente se for enviar um e-mail para o exterior, é recomendado não colocar acentos.

O campo CC serve para enviar uma cópia do e-mail para outra pessoa. Neste caso, tanto quem receber a mensagem como Mail to ou CC saberá que o outro recebeu também a mensagem. Alguns programas também apresentam um campo Bcc, de Blind Copy ou cópia secreta, que indica uma pessoa que receberá o e-mail, sem que as outras, Mail to e Cc, saibam que ela recebeu uma cópia.

O João receberá a seguinte mensagem quando ele conectar o computador à Internet.

Subject: Saudações

Date: Sun, 10 Jun 2007 12:34:22 -0745

From: maria@escola.edu.br

To: joao@empresax.com.br

Olá, João, como vai? Abraços,

Mario

Se João quer responder o e-mail ele simplesmente clica em Reply. Esse comando copia o endereço do remetente para ser destinatário de um novo e-mail e, se ele quiser, copiar a carta recebida. Para indicar quais partes foram copiadas do e-mail original é colocado o sinal “>”na frente de cada linha copiada. Se esta mensagem sofrer outro reply serão colocados “>na frente, de tal forma que podemos saber a sequência de envio de mensagem.

> Subject: Saudacoes

> Date: Sun, 10 Jun 2007 12:34:22 -0745

> From: maria@escola.edu.br

> To: joao@empresax.com.br

>

> Olá, João, como vai ?

>

> Abraços,

>

> Mario

> Ola’ Maria,

Tudo bem comigo. E você ?

João

Se João além de responder para Mario quiser enviar uma cópia para outra pessoa, ele pode usar o comando Forward, isto é, encaminhar para.

3.3.1 SMTP (Simple Mail Transfer Protocol)

Permite enviar, receber e armazenar mensagens eletrônicas para usuários de outros computadores (correio), observando os endereços eletrônicos.

O programa de e-mail na máquina do usuário abre a conexão para o servidor de e-mail. O programa dá o nome da máquina, o remetente e o conteúdo da mensagem. Então envia um comando dizendo que está iniciando a mensagem neste ponto, o outro lado termina o tratamento que é visto como comando e começa a receber a mensagem. A ponta remetente começa então a enviar o texto da mensagem.

No final, uma marca especial é enviada. Após isto, ambas as pontas compreendem que a ponta remetente está novamente enviando comandos.

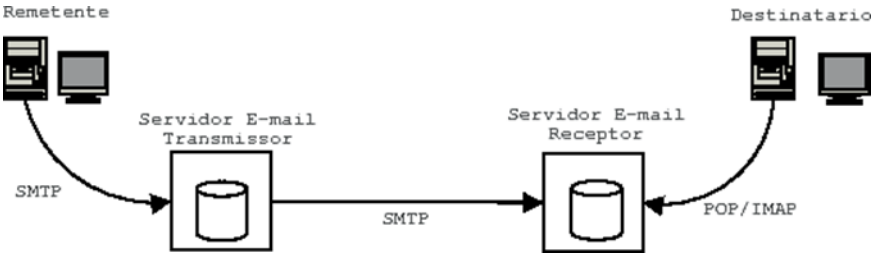

Figura 3.3.2: Componentes de um sistema de correio eletrônico.

1. cliente estabelece conexão com servidor e espera o servidor enviar a mensagem 220 READY FOR MAIL

2. após, cliente envia o comando HELO, o fim da linha marca o fim do comando

3. o servidor responde identificando-se

4. com a conexão estabelecida, o remetente pode transmitir uma ou mais mensagens de e-mails, terminar a conexão, ou solicitar que o servidor troque as regras de enviar e receber, assim, mensagens podem fluir na direção oposta.

5. a transmissão começa com o comando mail que dá a identificação do remetente

6. a partir do comando DATA, o receptor responde com a mensagem “start mail input”

Exemplo: Smith deseja mandar uma mensagem para Jones (s - servidor, c - cliente)

s: 220 escola.edu.br SMTP ready

c: HELO empresax.com.br

s:250 escola.edu.br

c: mail from: <@escola.edu.br>

s:250 ok

c: RCPT to:<empresax.com.br>

to: s:250 ok

c: DATA

s:354 start mail input; end with

<><>.<><>

c: .... envia mensagens de mail....

c: ..... mensagens....

c: <><>.<><>

s: 250 ok

c: QUIT

s: 221 beta.gov service closing trans

mission channel Enviar uma mensagem

para o usuário joão, no computador

empresax.com.br, observando o formato da mensagem :

% Mail joao@empresax.com.br

subject: Saudações

Olá João, como vai?

Abraços,

Maria

^d

Características

Possui basicamente três entidades: Agente do Usuário, Emissor-SMTP e Receptor-SMTP.

• É orientado a conexão, sendo transmitido sobre TCP.

• A comunicação entre Emissor-SMTP e Receptor-SMTP é feita através de comandos formados por sequências de caracteres no padrão ASCII.

• Apenas alguns dos comandos tem implementação obrigatória em um servidor básico: HELO, MAIL, RCPT, DATA, NOOP, QUIT e RSET

• Para cada comando enviado do Emissor-SMTP para o Receptor- -SMTP ocorrerá uma resposta do Receptor, através de um Código Numérico de Resposta.

Funcionamento Básico

É estabelecido a conexão entre Emissor-SMTP e Receptor-SMTP, onde este último pode ser o destino final da mensagem ou apenas um retransmissor.

1. O Emissor-SMTP envia a identificação do Remetente da mensagem, que o Receptor-SMTP responde com um OK.

2. Após, identifica-se o destinatário da mensagem, então, o Receptor- -SMTP verifica se este existe e retorna o código apropriado.

3. Estando identificado o destinatário o Emissor-SMTP começa o envio da mensagem propriamente dita.

4. Ao seu término o Emissor-SMTP envia um sequência especial de finalização. 5. Então, a conexão entre o Emissor-SMTP e o Receptor-SMTP é desativada.

3.3.2 Comandos SMTP

Semântica dos Comandos

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>

TURN <CRLF>

Descrição dos Comandos

HELO (HELLO) (Obrigatório) Identifica o Emissor da mensagem para o Receptor.

MAIL (Obrigatório) Este comando inicializa uma transação de mail na qual uma mensagem é enviada a uma ou mais caixa de mensagens (mailbox).

RCPT (ReCiPienT) (Obrigatório) Este comando identifica o destinatário da mensagem;múltiplos destinatários são definidos por múltiplos usos desse comando.

DATA (Obrigatório) Inicializa a transmissão da mensagem, após seu uso é transmitido o conteúdo da mensagem, que pode conter qualquer um dos 128 caracteres ASCII. O seu término é especificado por uma sequência “.”.

RSET (ReSET) (Obrigatório) Este comando determina que a operação atual de e-mail deverá ser abortada. Todos os dados referentes são descartados.

SEND Este comando é usado para inicializar uma transação de e-mail na qual uma mensagem é enviada para um ou mais terminais onde estejam os destinatários e não para os seus mailboxes. É um comando alternativo ao comando MAIL

SOML (Send Or Mail) Este comando é usado para inicializar uma transação de e-mail na qual uma mensagem é enviada para um ou mais terminais onde estejam os destinatários ou a seus mailboxes. A mensagem é direcionada aos terminais dos destinatários ativos no momento(e aceitando mensagens) caso contrário é direcionada aos seus mailboxes. É alternativo ao comando MAIL.

SAML (Send And Mail) Este comando é usado para inicializar uma transação de e-mail na qual uma mensagem é enviada para um ou mais terminais dos destinatários e aos seus mailboxes. A mensagem é direcionada aos terminais dos destinatários ativos no momento (e aceitando mensagens) e a todos os mailboxes.

VRFY (VeRiFY) Este comando solicita ao Receptor-SMTP a confirmação de que o argumento identifica um usuário conhecido. Se for identificado é retornado o nome completo do usuário (se este possuir) e seu mailbox completo.

EXPN (EXPaNd) Este comando solicita ao Receptor-SMTP a confirmação de que o argumento identifica uma lista de usuários de e-mail (mailing list). Se for identificada serão retornados os membros desta lista no mesmo formato retornado pelo comando VRFY.

HELP Este comando faz com que o Receptor-SMTP envie informação de ajuda ao Emissor-SMTP.

NOOP (Obrigatório) Este comando não possui efeitos nem parâmetros. Apenas faz com que o receptor envie um OK.

QUIT (Obrigatório) Este comando determina que o Receptor-SMTP envie um OK e então feche o canal de comunicação com o Emissor-SMTP.

TURN Este comando faz com que o Receptor e o Emissor troquem de papéis, o Receptor fica como Emissor e o Emissor como Receptor.

3.3.3 Códigos de Resposta

211 System status, or system help reply

214 Help message (Informação de como usar o Receptor-SMTP ou algum comando não padronizado)

220 Service ready

221 Service closing transmission channel

250 Requested mail action okay, completed

251 User not local; will forward to

354 Start mail input; end with .

421 Service not available, closing transmission channel (É uma resposta que pode ser dada a qualquer comando; indica que a conexão foi desfeita)

450 Requested mail action not taken: mailbox unavailable (Ex.: mailxbox está em uso)

451 Requested action aborted: local error in processing

452 Requested action not taken: insufficient system storage

500 Syntax error, command unrecognized (Usado também para casos tal como linha muito longa)

501 Syntax error in parameters or arguments

502 Command not implemented

503 Bad sequence of commands

504 Command parameter not implemented

550 Requested action not taken: mailbox unavailable (ex.: mailbox não encontrado, sem acesso)

551 User not local; please try

552 Requested mail action aborted: exceeded storage allocation

553 Requested action not taken: mailbox name not allowed (ex.:sintaxe do mailbox errada)

554 Transaction failed

3.4 File Transfer Protocol (FTP)

3.4.1 Utilização do FTP

Volta e meia precisamos de um programa para nossas máquinas, tal como, um novo antivírus, uma fonte para Word, ou seja, versões atualizadas de programas ou novos programas. Para isso existe o serviço FTP, onde podemos pegar esses programas em vários lugares no mundo. Usaremos o comando FTP mais o local em que faremos a busca, utilizaremos o serviço do servidor.com.br.

1º Digita-se:

Ñí é=ë É ê î á Ç ç ê KÃç ã KÃê

Isto vai gerar algo do tipo: connected to servidor.com.br

â~ã É W

2º Entre com o login:

â~ã É W= ~â ç â ó ã ç i ë

é~ë ë ï ç ê ÇW

3º Entre com a senha, que pode ser qualquer uma, mas vamos usar o nosso e-mail: joao@empresax.com.br

â~ã É W=~â ç ä ó ã ç i ë

é~ë ë ï ç ê Ç W= ñ

OPM= ~ â ç â ó ã ç ì ë = ä ç ÖÖÉÇ= á â

4º Repare que estamos agora na máquina remota que é um servidor Unix. Então vamos tentar o comando "ls" para listar o diretório.

Ñí é > = ä ë

200= Pç ê í = Åç ã ã a å Ç= ë ì Åeë ë Ñì ä K

150=O é eå á å Ö=A S C I I =ã ç Çe=Ça í a =Åç å å eÅí á ç å =Ñç ê =Ñá ä e=ä á ë í

Çê ï Çê ï J ê ï 1024O Åí = 15= 12:00= K Çê ï Çê ï

J ê ï 1024O Åí = 15= 12:00KK

Çê ï Çê ï J ê ï 1024= N ç î = 23 = 13 :24= é ì Ä

J ê ï J ê J ê J 1056 0= D eÅ= 11= 11:50= ä eá a ã eKí ñ í

226 = T ê a å ë Ñeê = C ç ã é ä eí e

3 218= B ó í eë = ê eÅeá î eÇ= 1K2= ë eÅç å Çë = (2K6 = K Ä/ë) Ñí é > =

Repare que onde existe a letra "d" é um diretório. O nome do diretório está na última coluna, os dois primeiros não são diretório, são do sistema (o mesmo tipo que existe numa máquina Unix convencional). O PUB é um dire- tório, normalmente este onde encontraremos os arquivos do nosso interesse. O leiame.txt é um arquivo, repare na 1º coluna "-" (pode também ser "a") indica que é um arquivo. Suponha que primeiramente você queira saber o que tem dentro deste arquivo leiame.txt, basta enviar para a sua máquina(lembre-se que no momento é como se você estivesse lá).

5º Então deve usar o comando get da seguinte forma: get leiame. txt. Lembre-se você está numa máquina Unix, então se comporte como se estivesse usando uma:

Ñí é > = Öeí = ä eá a ã eKí ñ í

200= Pç ê í = Åç ã ã a å Ç= ë ì ÅÅeë ë Ñì ä

150=O é eå á å Ö=A S C I I =ã ç Çe=Ça í a =Åç å å eÅí á ç å =Ñç ê =ä eá aã eK í ñ í = (1056 0= B ó í eë)

226 = T ê a å ë Ñeê = Åç ã é ä eí e

50= B ó í eë = ê eÅeá î eÇ= 6 K9= ë eÅç å Çë = (1K047= K Ä/ë)

Ñí é >

6º Agora suponha-se que você queira entrar num subdiretório, no caso, no diretório "PUB", basta usar o comando cd pub e de um novo ls, isto vai gerar algo do tipo:

Ñí é > = ÅÇ= é ì Ä

200= Pç ê í = Åç ã ã a å Ç= ë ì ÅÅeë ë Ñì ä 250= C W D = Åç ã ã a å Ç= ë ì ÅÅeë ë Ñì ä Ñí é > = ä ë

200= Pç ê í = Åç ã ã a å Ç= ë ì ÅÅeë ë Ñì ä

150=O é eå á å Ö=A S C I I =ã ç Çe=Çaí a=Åç å å eÅí á ç
å =Ñç ê =Ñá ä e=ä á ë í K Çê ï Çê ï J ê ï 1024O Åí =
15= 12:00= K
Çê ï Çê ï J ê ï 1024O Åí = 15= 12:00KK

J ê ï J ê J ê J 13 444N ç î = 23 = 21:50= a ê è ì á
î ç 1Kò á é J ê ï J ê J ê J 1707= N ç î = 23 =
21:55= a ê è ì á î ç 2Kò á é

226 = T ê a å ë Ñeê = Åç ã é ä eí e

5414= B ó í eë = ê eÅeá î eÇ= 1K1= ë eÅç å Çë = (4K9=
K Ä/ë) Ñí é >

7º Suponha-se que queremos o arquivo arquivo2.zip, note que ele não é um diretório e nem um arquivo texto, portanto para seu melhor modo envio ele tem que ser transferido por conjuntos de byte's.

Devemos usar o comando binary (bin) antes de usar o comando get, para informar que o arquivo que está sendo transferido é binário. Depois da

transferência destes arquivos, sairemos imediatamente do servidor usando o comando quit.

é ì Ä> = Äá å

200= T ó é e= ë eí = í ç = I

é ì Ä> = Öeí = a ê è ì á î ç 2Kò á é

200= Pç ê í = Åç ã ã a å Ç= ë ì ÅÅeë ë Ñì ä

353 =O é eå á å Ö=Äá å aê ó =ã ç Çe=Åç å å eÅí á ç å =Ñç ê =a ê è ì á î ç 2K ò á é (1707= B ó í eë)
456 = Äó í eë = í ê a å ë Ñeê
411= Äó í eë = í ê a å ë Ñeê
422= Äó í eë = í ê a å ë Ñeê

456 = T ê a å ë Ñeê = Åç ã é ä eí e

1707=B ó í eë =ê eÅeá î eÇ==6 K1=ë eÅç å Çë =(0K27=K Ä/ë)

é ì Ä> = è ì á í
200= Pç ê í = Åç ã ã a å Ç= ë ì ÅÅeë ë Ñì ä
221= Äó e

3.4.2 Principais comandos

ASCII

Alterna para o modo ASCII. O Modo ASCII é o modo default usado para transferência de arquivos textos;

BIN

Alterna para o modo Binary. Para transferência de arquivos binários tipo .exe,.com, .zip entre outros. **CD**

Muda de diretório na máquina que você está acessando (O remoto);

DIR ou LIST

Lista os arquivos no diretório corrente do remoto;

GET

Copia arquivos do micro remoto para a sua conta;

HELP

Mostrar os comandos existentes e suas funções;

LCD

Muda de diretório na sua conta corrente (O L é de local);

MGET

Cópia múltiplos arquivos do computador remoto para o seu;

MPUT

Transfere múltiplos arquivos da sua conta para o computador remoto;

PUT

Transfere arquivos da sua conta corrente para a máquina remota (ftp anônimos não aceitam este comando);

O FTP também pode ser utilizado para acessar outra máquina ou conta de outro sistema, caso haja a necessidade de enviar arquivos ou de copiar arquivos de uma máquina ou conta para uma outra máquina e conta, basta saber o FQDN ou IP deste outra máquina o login e o password desta outra conta. É neste caso que podemos utilizar os comandos PUT e MPUT descritos anteriormente.

3.4.3 Protocolo FTP

O FTP é o protocolo de transferência de arquivos da Arquitetura Internet. Trata-se de um utilitário de uso interativo que pode ser chamado por programas para efetuar transferência de arquivos. Os seus principais objetivos são:

• motivar a utilização de computadores remotos;

• tornar transparentes ao usuário diferenças existentes entre sistemas de arquivos associados a estações de uma rede;

• transferir dados de maneira eficiente e confiável entre dois sistemas;

• promover o compartilhamento de arquivos, sejam programas ou dados.

O FTP não se preocupa em definir um sistema de arquivos virtual e sim em definir uma interface com os sistemas de arquivos nativos. Podemos dividir o entendimento do protocolo em quatro partes:

1. Modelo

2. Sistema de Arquivos

3. Processo de Transferência de Arquivos

4. Comandos.

Modelo

O FTP trabalha com o modelo CLIENTE-SERVIDOR. O modelo implementado possui uma característica interessante, que é a de utilizar duas conexões diferentes entre os sistemas envolvidos: uma denominada conexão de

controle, dedicada aos comandos FTP e suas respostas, e a outra denominada conexão de dados, dedicada à transferência de dados.

A parte executada no cliente (chamada de Cliente-FTP) pode ser dividida em três módulos que interagem por algum mecanismo interno. Esses módulos são:

1. Interface do Usuário

2. Interpretador de Protocolo do Cliente (Cliente-PI)

3. Processo de Transferência de dados (Cliente-DTP).

A parte executada no servidor (chamada de Servidor-FTP) é dividida em dois módulos com funções análogas aos seus equivalentes no cliente. Esses módulos são:

1. Servidor-PI

2. Servidor-DTP.

A conexão de controle, usada na transferência de comandos FTP e suas respostas, é realizada diretamente entre o Cliente-PI e o Servidor-PI, e a conexão de dados é estabelecida entre o Cliente-DTP e o Servidor-DTP.

3.5 Telnet

O aplicativo Telnet oferece o acesso de uma máquina remotamente. O Telnet dá a oportunidade de acessar sistemas e máquinas que estejam distante do nosso local atual. Para abrir um Acesso Remoto normalmente utilizamos telnet local.dominio porta onde local.domínio poderá estar em FQDN ou em IP e a porta normalmente é utilizado 23 como porta default.

A sessão remota inicia especificando em qual computador você deseja conectar-se.

A partir do momento que se inicia a sessão de trabalho remoto, qualquer coisa que é digitada é enviada diretamente para o computador remoto (note que você continua ainda no seu próprio computador, mas o programa telnet torna seu computador um terminal do outro computador).

Será solicitado um username e uma password para acessar o sistema remoto. Telnet oferece três serviços básicos:

1. define um terminal virtual de rede, que proporciona uma interface padrão para sistemas remotos; programas clientes não têm que compreender os detalhes de todos os possíveis sistemas remotos, eles são feitos para usar a interface padrão;

2. inclui um mecanismo que permite ao cliente e ao servidor negociarem opções e proporcionar um conjunto de opções padrão;

3. trata ambas as pontas da conexão simetricamente. Assim, ao invés de forçar o cliente a conectar-se a um terminal de usuário, Telnet permite um programa arbitrário tornar-se um cliente. Além disso, cada ponta pode negociar opções.

3.5.1 Funcionamento

Para logins remotos, há somente uma conexão, normalmente envia dados. Quando é necessário enviar comando (isto é, para setar o tipo de terminal ou trocar algum modo) um caractere especial é usado para indicar que o próximo caractere é um comando. O caminho dos dados em uma sessão remota é como uma viagem do terminal do usuário para o sistema operacional remoto

3.5.2 Usando Telnet

Para chamar o telnet:

$= í eä å eí ==

O programa telnet apresenta então o seu prompt (telnet>) ao usuário, para receber novos comandos. O telnet também tem o comando help.

í eä å eí > = Üeä é =

Mostramos agora um exemplo de uso do comando telnet para listar o diretório de uma máquina remota. Desejando acessar informações que se encontram no computador chamado server.escola.edu.br, que está localizado fisicamente longe do computador do usuário, cujo usuário seja João e a password xxx.

$= í eä å eí = ë eê î eê Keë Åç ä a KeÇì KÄê

ì ë eê := à ç a ç

é a ë ë ï ç ê Ç:ñ ñ ñ

í eä å eí > =

A conexão está estabelecida, portanto a partir de agora tudo o que for digitado será executado na máquina remota.

í eä å eí > = ä ë

Çê ï Çê ï J ê ï 1024O Åí = 15= 12:00= K

Çê ï Çê ï J ê ï 1024O Åí = 15= 12:00KK

J ê ï J ê J ê J 13 444N ç î = 23 = 21:50= a ê è ì á î ç 1Kò á é

J ê ï J ê J ê J 1707= N ç î = 23 = 21:55= a ê è ì á î ç 2Kò á é

Para encerrar a sessão de trabalho remota:

í eä å eí > = ä ç Öç ì í ==

O grande problema do Telnet é que ele não é criptografado, portanto, tudo o que é digitado no terminal (inclusive senha) pode ser lido por qualquer usuário ligado

nesta rede. Para resolver isso pode-se usar o aplicativo SSH (Secure Shell), de funcionamento semelhante ao telnet, porém toda a comunicação é criptografada, além de várias funcionalidades de segurança.

4 PONTOS POSITIVOS

A camada de aplicação atua como intermediário para ajudar diferentes apps a “conversarem” entre si. Exemplo, quando você envia uma mensagem pelo ***WhatsApp***,ela garante que a mensagem seja transmitida de seu celular para o celular do destinatário, permitindo a comunicação instantaneamente.

Ela torna os aplicativos mais fáceis de usar. Quando você assiste a ***Netflix***, por exemplo, a camada de aplicação garante que a experiência de streaming aconteça sem erros, para que você possa clicar e assistir, sem se preocupar com a infraestrutura de rede complexa por trás dela.

É responsável por fornecer uma variedade de serviços online, como a ***Amazon***, que oferece serviços de compras online, permitindo que os clientes naveguem e comprem produtos, tornando o processo de compra rápido e conveniente.

Também ajuda a proteger suas informações. Quando você faz login em sua conta bancária, usando o ***Internet Banking***, ela utiliza criptografia para garantir que suas transações e informações sejam mantidas em sigilo, protegendo-as contra ataques hackers.

5 REFERÊNCIAS BIBLIOGRÁFICAS

ANDREW S. TANENBAUM **Redes de Computadores** 4ª Ed. Editora: Campus, 2004. Livro de referência clássica com mais de 30 anos desde a primeira edição, proporcionando ao estudante uma visão histórica das arquitetura e protocolos de redes de computadores. São quase 1.000 páginas de texto com descrição detalhada dos sistemas e protocolos, além disso, o autor tem um ótimo senso de humor tornando a leitura muito agradável.

LARRY L. PETERSON & BRUCE S. DAVIE **Redes de Computadores: uma Abordagem de Sistemas.** 3ª Ed. Editora: Campus, 2004. Livro texto, tratando não apenas da descrição dos sistemas de redes de computadores mas fornecendo uma explicação do funcionamento. É um livro introdutório mas completo e coeso. Essa edição apresenta assuntos atuais como IPv6, redes peer-to-peer e redes móveis.

DOUGLAS E. COMER **Interligação em Rede com TCP/IP.** 5ª Ed. Editora: Campus, 2006. Livro texto completo sobre a arquitetura TCP/IP. Mostra os princípios de projeto da Internet, trata de endereçamento e roteamento IP, programação usando Sockets e exemplos de aplicações como e-mail e WWW. Mostra também assuntos avançados como IP móvel, VPN e MPLS.

KEITH W. ROSS & JAMES F. KUROSE. **Redes de Computadores e a Internet**: Uma Abordagem Top-down. 3ª Ed. Editora: Addison-Wesley, 2006. O grande diferencial desse livro, desde sua primeira edição, é a proposta inovadora da visão top-down no estudo dos conceitos de redes de computadores, isto é, começando na camada de aplicação e descendo até a camada física. Mas independentemente da visão adotada, é um excelente livro com conteúdo detalhado e leitura agradável. Quem preferir a visão tradicional, bottom-up, pode começar pelo último capítulo.

DOUGLAS E. COMER. **Redes de Computadores e Internet.** 4ª Ed. Editora: Bookman, 2007. Livro clássico sobre TCP/IP de fácil leitura e apropriado para leitor iniciante. O livro apresenta uma visão superficial mas completa de todos os protocolos da pilha TCP/IP.

MARCIAL P. FERNÁNDEZ. Ed: EdUECE, 2019. **Livro Redes de Computadores.** https://educapes.capes.gov.br/bitstream/capes/432642/2/Livro%20%20Redes%20de%20Computadores.pdf