

T202 – B/C Redes de Computadores

03 – Protocolos de Aplicação I (HTTP e SMTP/IMAP/POP3)

Prof. Edson J. C. Gimenez
soned@inatel.br

Referência principal:

- ✓ Kurose & Ross. Redes de Computadores e a Internet: uma abordagem top-down. Capítulo 2.



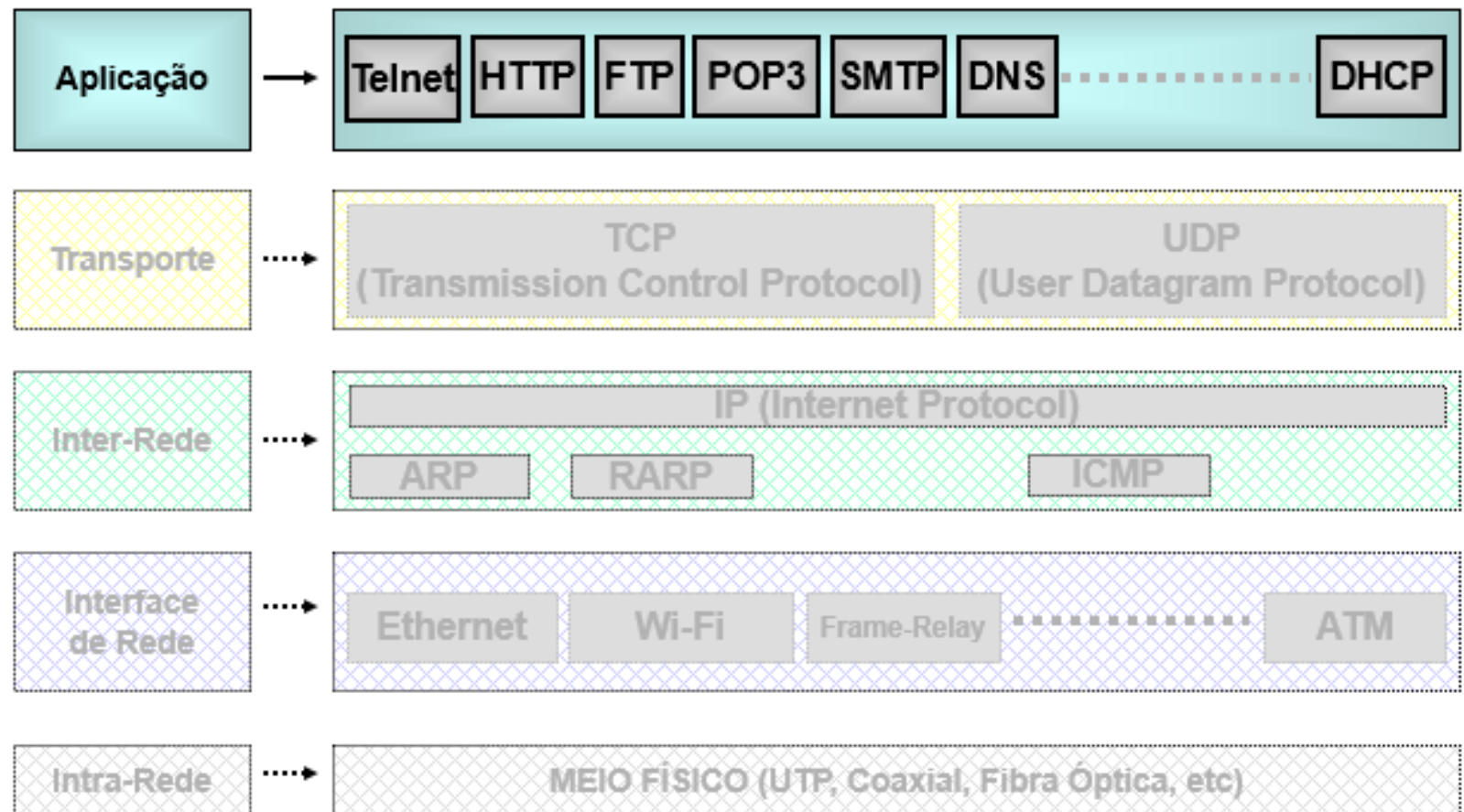
Outras referências:

- ✓ Tanenbaum & Wetherall. Redes de Computadores; capítulo 7.
- ✓ Farrel. A Internet e seu Protocolos: uma Análise Comparativa; capítulo 12.
- ✓ Forouzan & Mosharraf. Redes de Computadores: uma abordagem top-down; capítulo 2.
- ✓ Comer. Interligação de Redes com TCP/IP; volume 1; capítulos 23 a 27 e 29.

- ✓ A camada de aplicação da Internet contém os aplicativos que são disponibilizados para os usuários, e que utilizam os protocolos subjacentes (TCP ou UDP, IP, camada 2) para trocar informações.
- ✓ Assim, um aplicativo interage com pelo menos um dos protocolos da camada de transporte (TCP ou UDP) para enviar e/ou receber dados.
- ✓ Cada aplicativo escolhe o serviço de transporte necessário (UDP ou TCP), de acordo com as suas necessidades.

- ✓ Definem como os processos de uma aplicação, em sistemas finais diferentes, trocam mensagens entre si, estabelecendo:
 - Os tipos de mensagens trocadas.
 - A sintaxe dessas mensagens.
 - Regras para envio e resposta dessas mensagens.

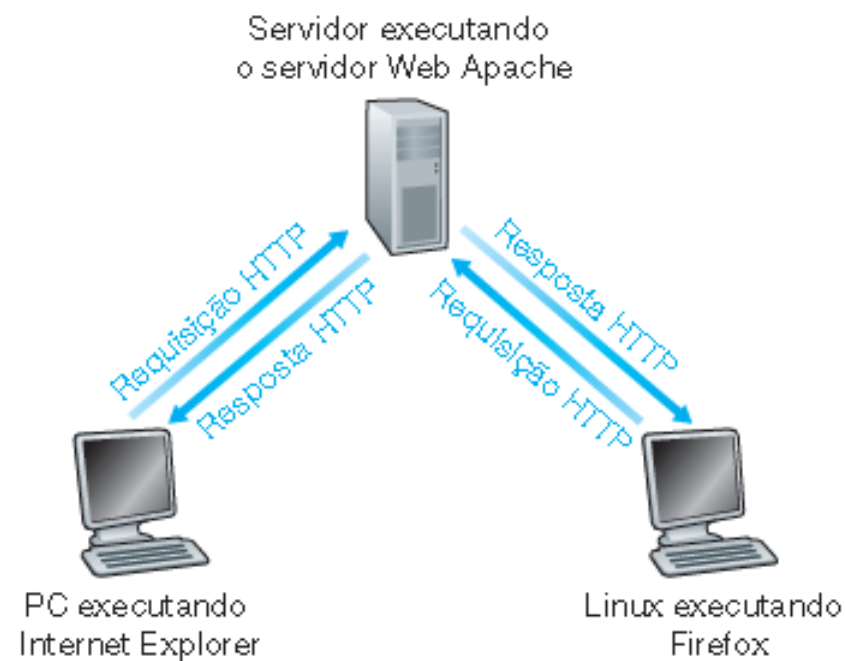
✓ Exemplos:



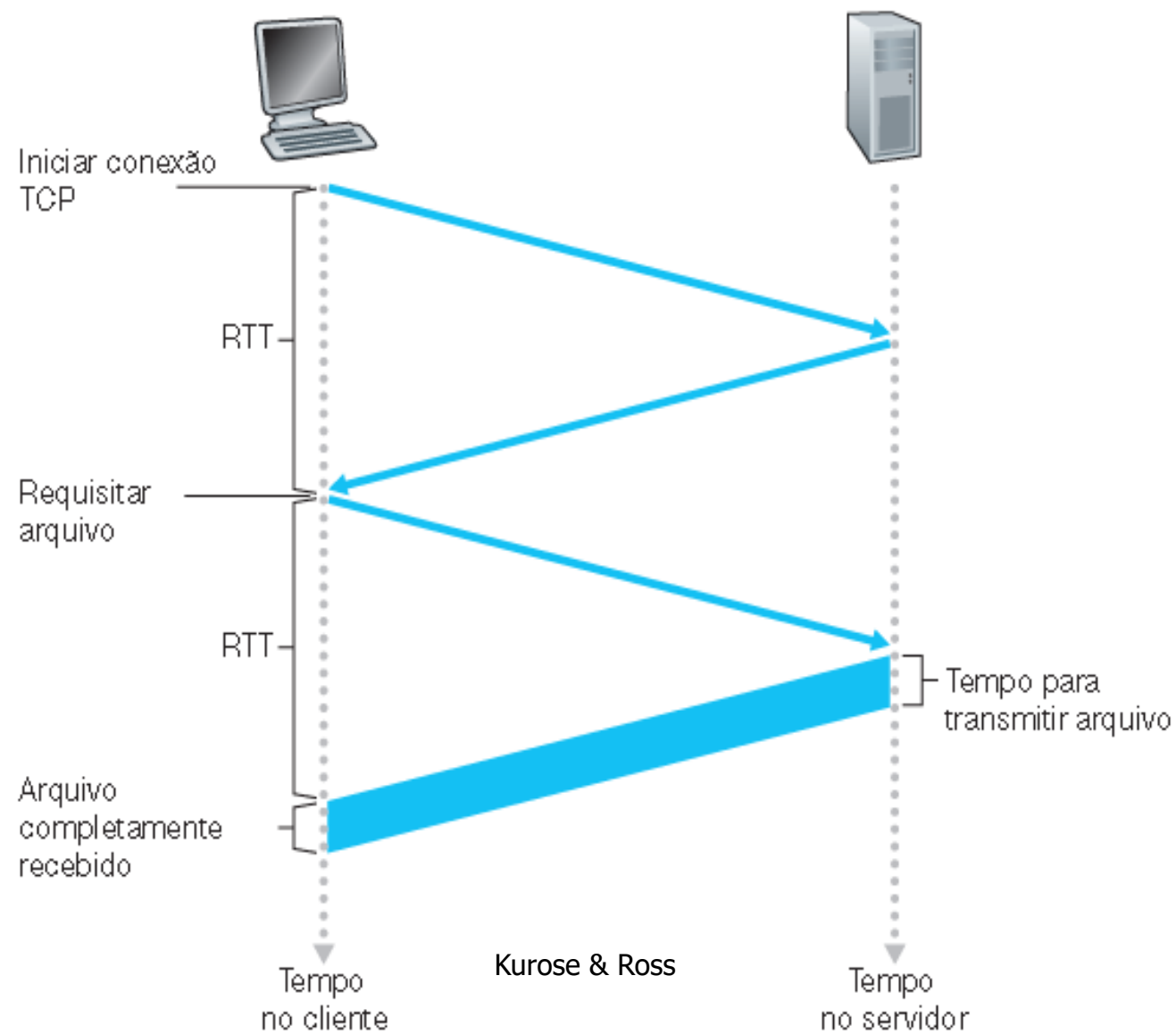
Protocolo de Transferência de Hipertexto.

- HTTP/1.0 (RFC 1945, May/1996)
 - HTTP/1.1 (RFC 2616, June/1999).
 - HTTP/2 (RFC 7540, May/2015)
 - HTTP/3 (RFC 9114, June/2022)
- ✓ Protocolo padrão para comunicação Web
- ✓ Define como as páginas Web são requisitadas pelos clientes e enviadas pelos servidores.
- Em geral, as páginas Web são constituídas de um arquivo-base HTML e diversos objetos referenciados.
 - Cada objeto é um arquivo, tal como arquivo html, imagem jpeg, arquivo de vídeo, etc., que se pode acessar com um único URL (Uniform Resource Locator).
 - Cada URL tem dois componentes básicos: o **nome de hospedeiro** (hostname) do servidor que abriga o objeto e o **nome do caminho** do objeto.
 - <http://www.someschool.edu/somedepartment/picture.gif>
 - <http://cisco.com/index.html>

- ✓ Utiliza o TCP como seu protocolo de transporte.
- ✓ Trabalha no modo cliente / servidor.
 - Cliente envia requisições, servidor responde às requisições



Kurose & Ross



Kurose & Ross

Requisição/Resposta HTTP

✓ Conexões não persistentes:

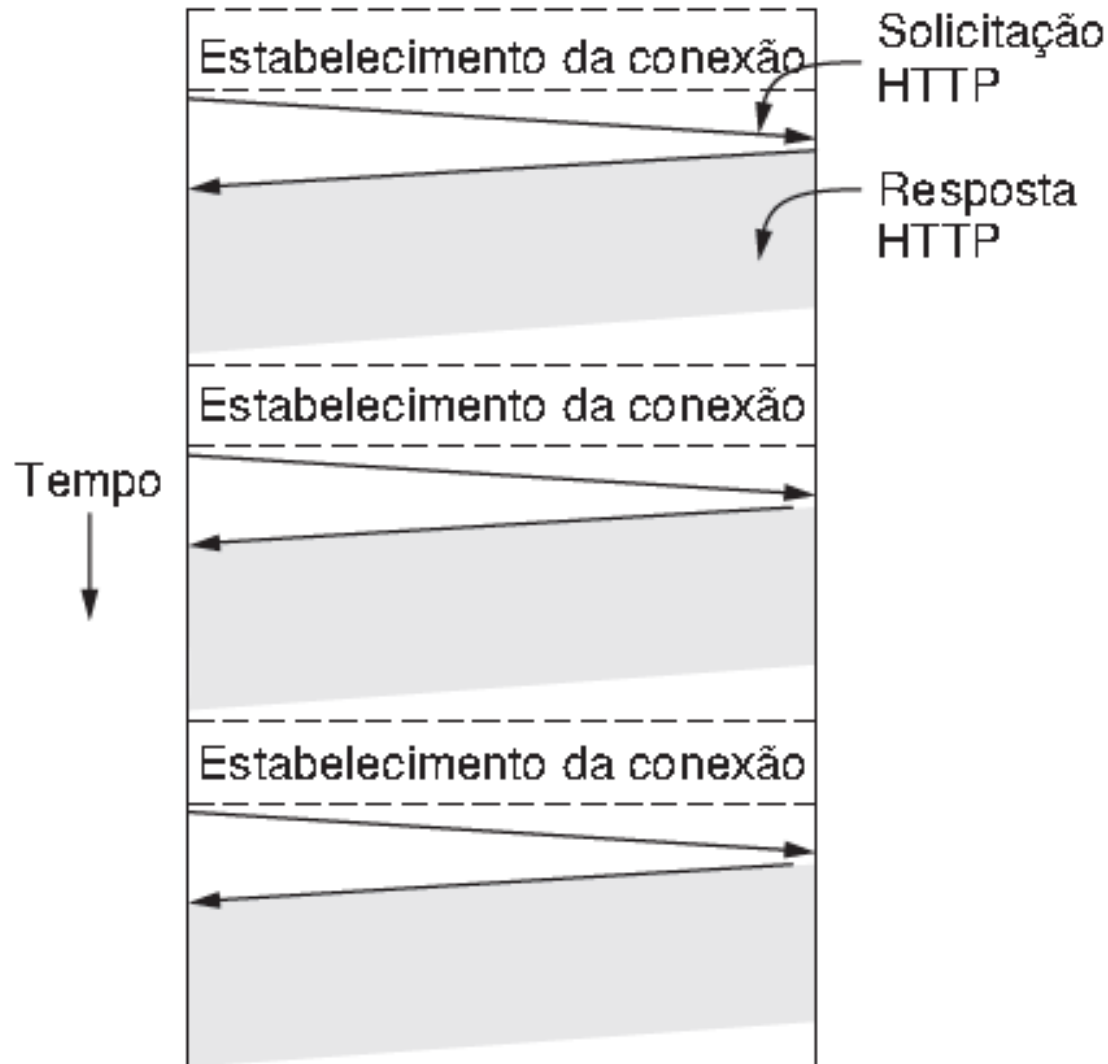
- Cada par de requisição/resposta deve ser enviado por uma conexão TCP distinta
- No máximo um objeto é enviado em cada conexão TCP.
- HTTP/1.0

✓ Conexões persistentes:

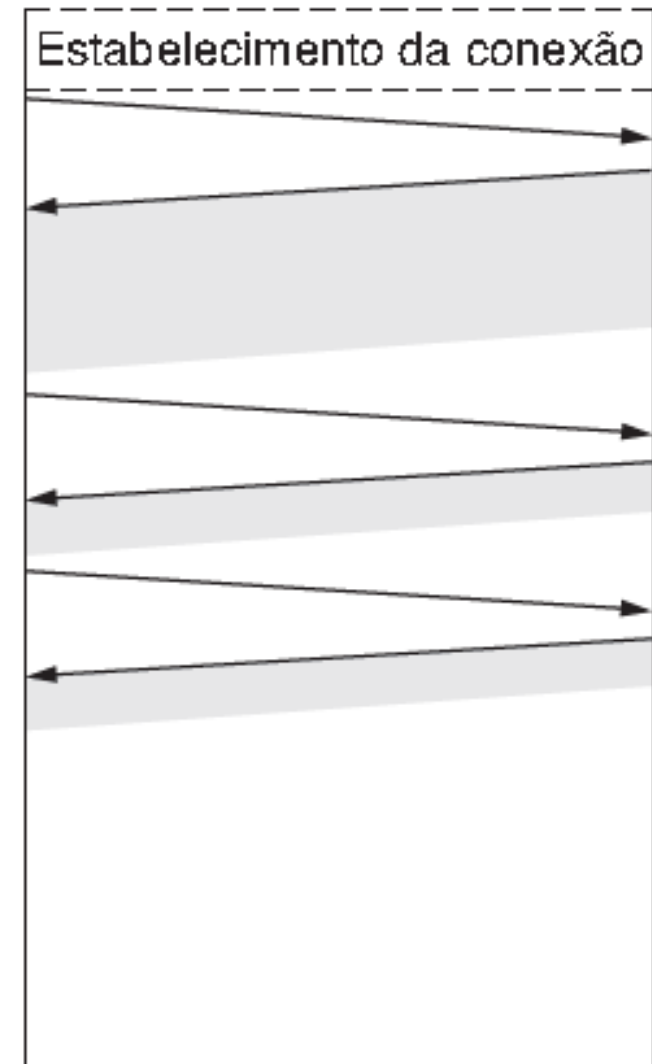
- Todas as requisições e suas respostas são enviadas por uma mesma conexão TCP.
- Múltiplos objetos podem ser enviados em uma única conexão.
- HTTP/1.1 e HTTP/2

Requisição/Resposta HTTP → Atrasos

Conexões não persistentes



Conexões persistentes



Exemplo: supondo uma página com 01 arquivo-base HTML e 10 imagens JPEG

1. O processo cliente HTTP inicia uma conexão TCP para o servidor `www.someschool.edu` na porta numero 80. Associados à conexão TCP, haverá um *socket* no cliente e um *socket* no servidor.
2. O cliente HTTP envia uma mensagem de requisição HTTP ao servidor por meio de seu *socket*. Essa mensagem inclui o nome de caminho `/someDepartment/home.index`.
3. O processo servidor HTTP recebe a mensagem de requisição por meio de seu *socket*, extrai o objeto `/someDepartment/home.index` de seu armazenamento (RAM ou disco), encapsula-o em uma mensagem de resposta HTTP e a envia ao cliente pelo *socket*.
4. O processo servidor HTTP ordena ao TCP que encerre a conexão TCP.
5. O cliente HTTP recebe a mensagem de resposta e a conexão TCP é encerrada. A mensagem indica que o objeto encapsulado é um arquivo HTML. O cliente extrai o arquivo da mensagem de resposta, analisa o arquivo HTML e encontra referencias aos dez objetos JPEG.
6. As etapas são repetidas para cada um dos objetos JPEG referenciados.
 - Abre conexão / requisita objeto / recebe objeto / fecha conexão.

Exemplo: supondo uma página com 01 arquivo-base HTML e 10 imagens JPEG

1. O processo cliente HTTP inicia uma conexão TCP para o servidor `www.someSchool.edu` na porta numero 80. Associados à conexão TCP, haverá um *socket* no cliente e um *socket* no servidor.
2. O cliente HTTP envia uma mensagem de requisição HTTP ao servidor por meio de seu *socket*. Essa mensagem inclui o nome de caminho `/someDepartment/home.index`.
3. O processo servidor HTTP recebe a mensagem de requisição por meio de seu *socket*, extrai o objeto `/someDepartment/home.index` de seu armazenamento (RAM ou disco), encapsula-o em uma mensagem de resposta HTTP e a envia ao cliente pelo *socket*.
4. O cliente HTTP recebe a mensagem de resposta. A mensagem indica que o objeto encapsulado é um arquivo HTML. O cliente extrai o arquivo da mensagem de resposta, analisa o arquivo HTML e encontra referências aos dez objetos JPEG.
5. As etapas 2 a 4 são repetidas para cada um dos objetos referenciados (cada imagem JPEG), na mesma conexão aberta inicialmente.
6. Após o envio/recepção do último objeto, a conexão é fechada.

Exemplos de mensagens HTTP:

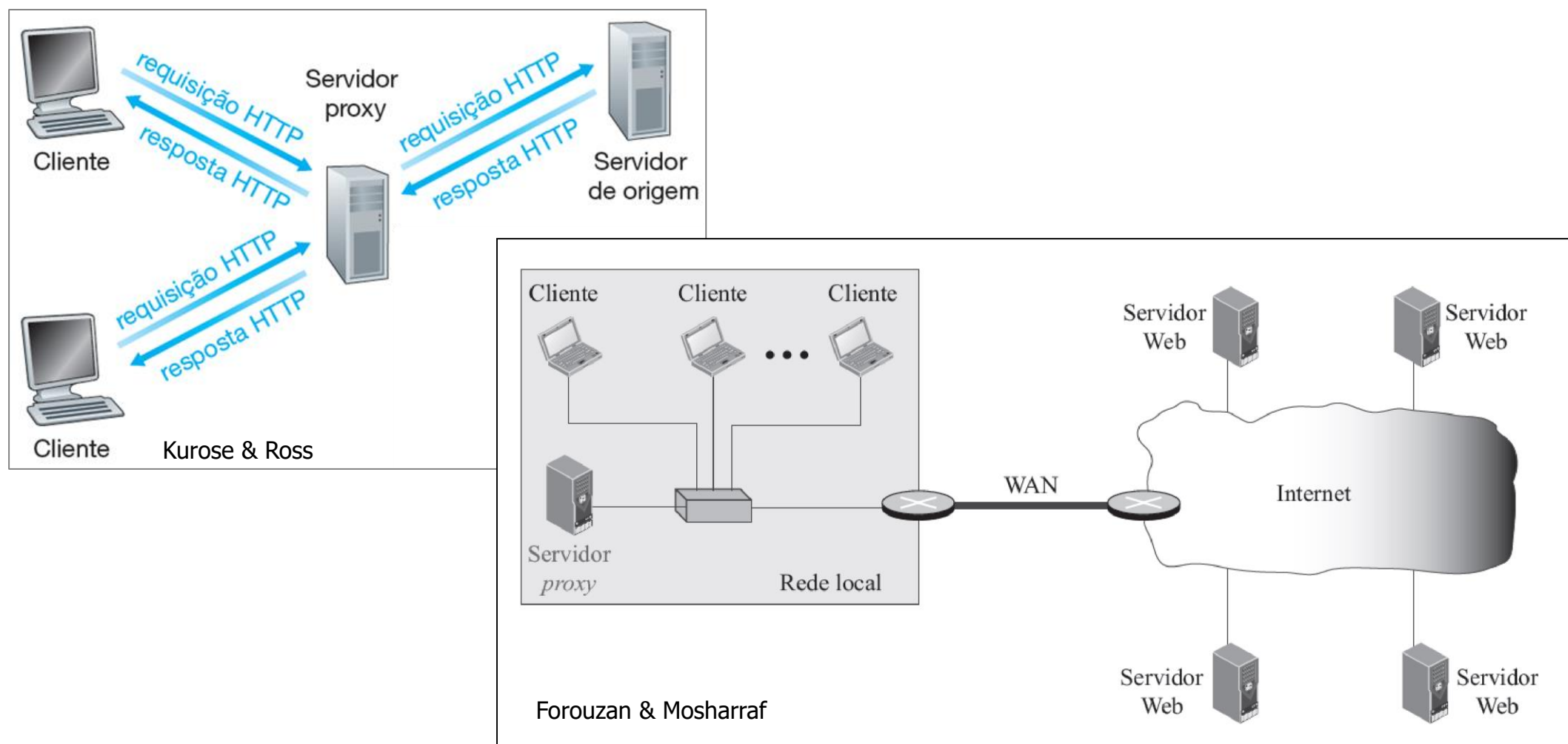
✓ Mensagens de requisição:

- GET – lê uma página Web.
- HEAD – lê um cabeçalho de página Web.
- POST – acrescenta algo a uma página Web.
- PUT – armazena uma página web.
- DELETE – remove uma página Web.

✓ Mensagens de resposta:

- 200 OK – solicitação realizada.
- 301 Moved Permanently – os dados solicitados foram removidos.
- 400 Bad Request – solicitação mal formulada ou impossível de validar.
- 404 Not Found – servidor não encontrou nada no endereço indicado.
- 501 Not Implemented – servidor não suporta o serviço solicitado.
- 505 HTTP Version Not Supported – versão do protocolo HTTP requisitada não é suportada pelo servidor.

- ✓ Um **cache Web** (servidor proxy) é uma entidade da rede que atende requisições HTTP em nome de um servidor Web de origem, mantendo cópias de objetos recentemente requisitados.
 - Mensagem “conditional GET” – mecanismo que permite que um *cache* verifique se seus objetos estão atualizados.

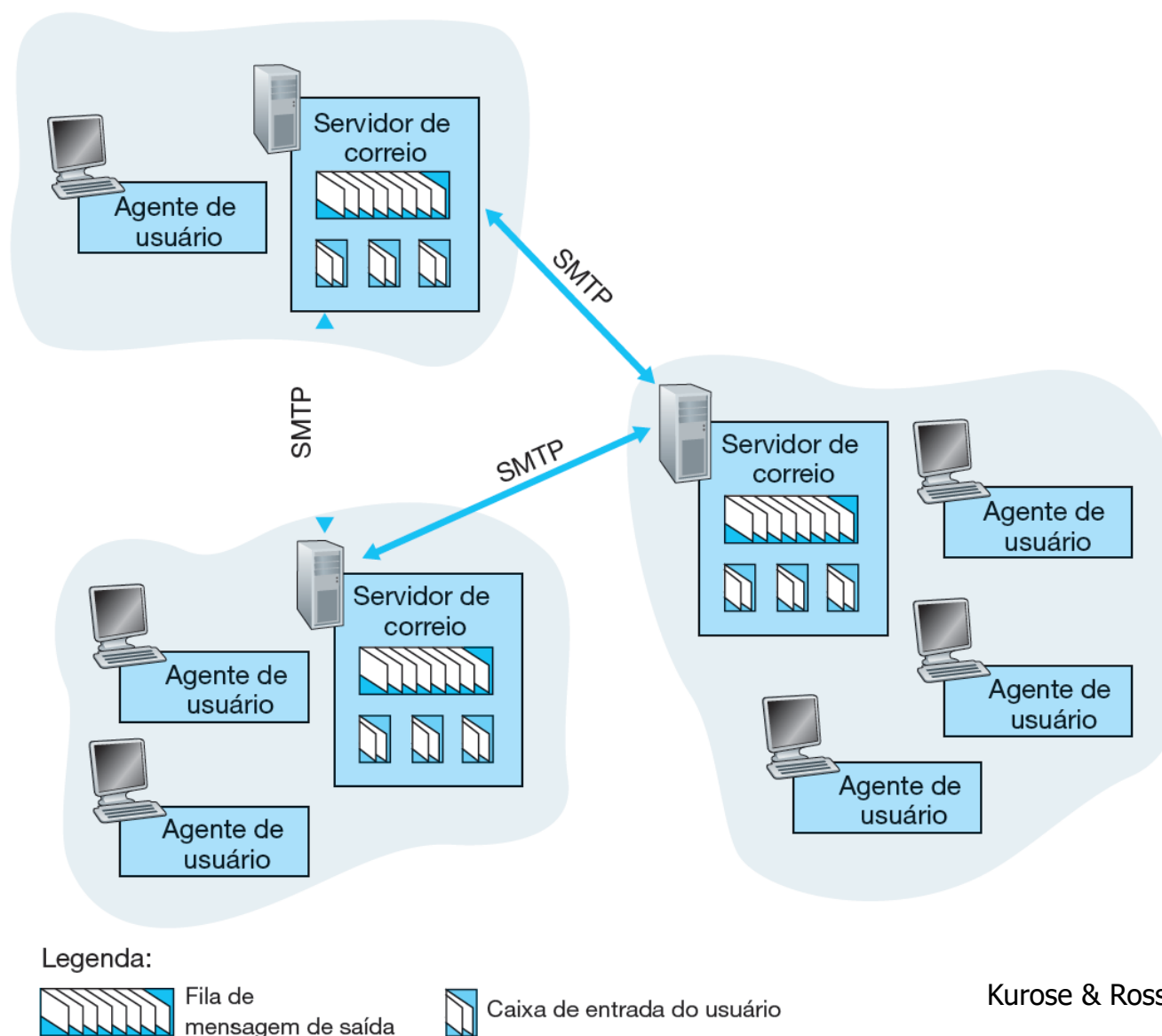


Exemplo: Supondo que o navegador requisita o objeto <http://www.someschool.edu/campus.gif> ao cache Web.

1. O navegador estabelece uma conexão TCP com o cache Web e envia a ele uma requisição HTTP para o objeto.
2. O cache Web verifica se tem uma cópia do objeto armazenada localmente.
 - a) **Se tiver o objeto:** envia o objeto ao navegador do cliente, dentro de uma mensagem de resposta HTTP.
 - b) **Se não tiver o objeto:** o cache Web abre uma conexão TCP com o servidor de origem www.someschool.edu; então, envia uma requisição HTTP do objeto para a conexão TCP. O servidor de origem então envia o objeto solicitado ao cache Web, dentro de uma resposta HTTP. Recebendo o objeto, o cache Web guarda uma cópia em seu armazenamento local e envia outra cópia, dentro de uma mensagem de resposta HTTP, ao navegador do cliente, pela conexão TCP existente entre o navegador do cliente e o cache Web.

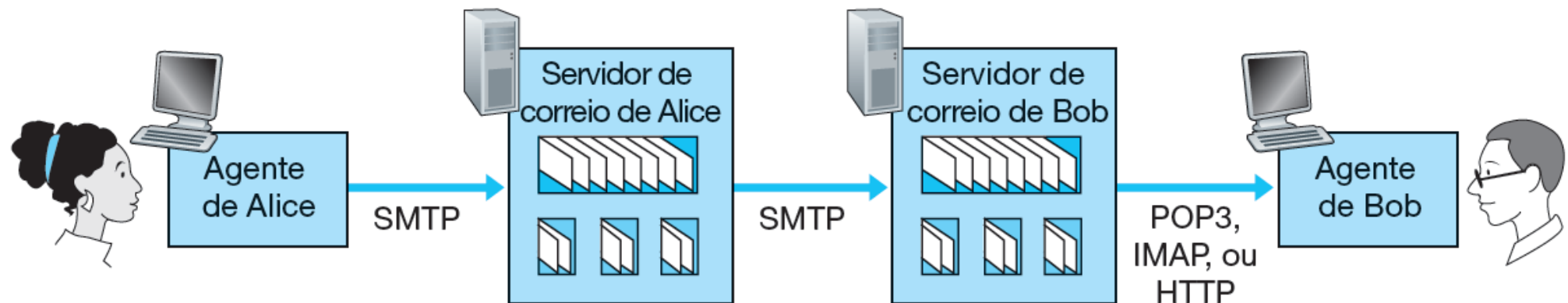
- ✓ Embora o HTTP seja flexível, não é um protocolo seguro.
 - As mensagens requisitadas enviam informações ao servidor em modo texto simples.
 - As respostas do servidor, normalmente páginas HTML, também não são criptografadas.
- ✓ O HTTPS possibilita conversações seguras na Internet, fazendo uso de autenticação e criptografia para proteger os dados trocados entre o cliente e o servidor.
 - O HTTPS usa o mesmo processo de requisição do cliente e resposta do servidor do HTTP, porém o fluxo de dados é criptografado antes de ser transportado através da rede.
 - SSL (Secure Sockets Layers) ou TLS (Transport Layer Security).

- ✓ Permite o envio e recebimento de mensagens de modo assíncrono.
- ✓ Possui três componentes principais: agente de usuário, servidor de correio e um protocolo para transferência das mensagens.



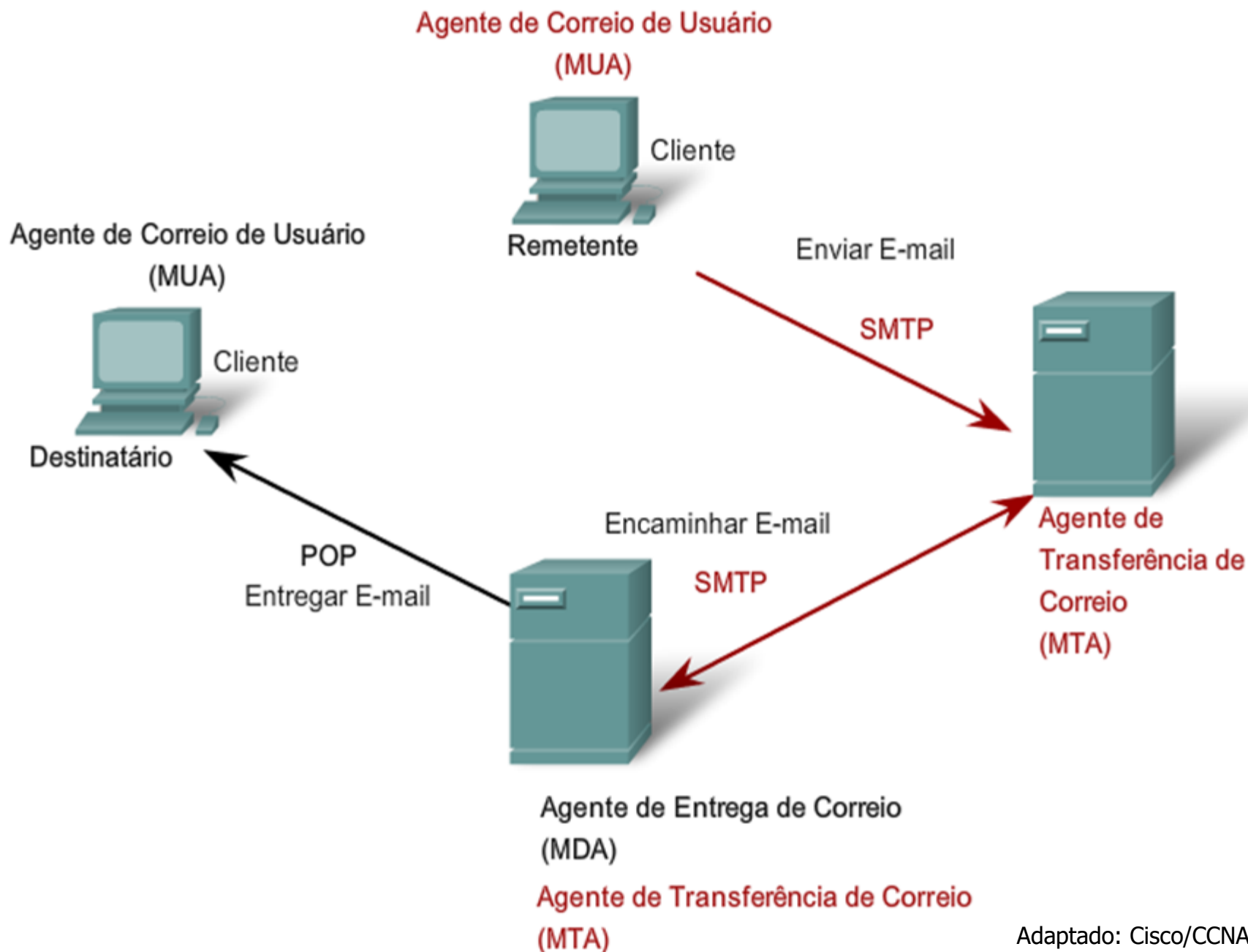
SMTP (Simple Mail Transfer Protocol) [RFC 5321 (out/2008)]

- ✓ Permite transporte de mensagens de e-mail, em formato ASCII, através do TCP, para e entre servidores.
- ✓ Quando um servidor de correio eletrônico recebe uma mensagem destinada a um cliente local, armazena-a e espera que ela seja coletada pelo cliente.
- ✓ O cliente podem coletar sua correspondência usando um dos muitos protocolos de rede existentes: POP3 (Post Office Protocol versão3), IMAP (Internet Mail Access Protocol) ou HTTP.



- ✓ Faz uso dos processos MUA (Mail User Agent), MTA (Mail Transport Agent) e MDA (Mail Delivery Agent).
 - Quando criadas, as mensagens de e-mail normalmente utilizam o processo MUA, ou cliente de e-mail.
 - O MUA permite que mensagens sejam enviadas e coloca as mensagens recebidas na caixa de correio do cliente, sendo processos diferentes.
 - O processo MTA é usado para encaminhar os e-mails.
 - O MTA recebe mensagens do MUA ou de outro MTA em outro servidor de e-mail.
 - Com base no cabeçalho da mensagem, ele determina como encaminhar a mensagem. Se o e-mail for endereçado a um usuário cuja caixa de correio fique no servidor local, ele será passado para o MDA. Se o e-mail for para um usuário fora do servidor local, o MTA o encaminha para o MTA no servidor em questão.
 - O MDA aceita um e-mail de um MTA e faz a entrega real.
 - O MDA recebe toda correspondência que chega no MTA e a coloca nas caixas de correio dos usuários adequados.

Processos MUA, MTA e MDA:



Atividade 04 – Aplicações I