

# **Correio Eletrônico**

## **SMTP, POP3, IMAP**

**Profª Angélica da Silva Nunes**

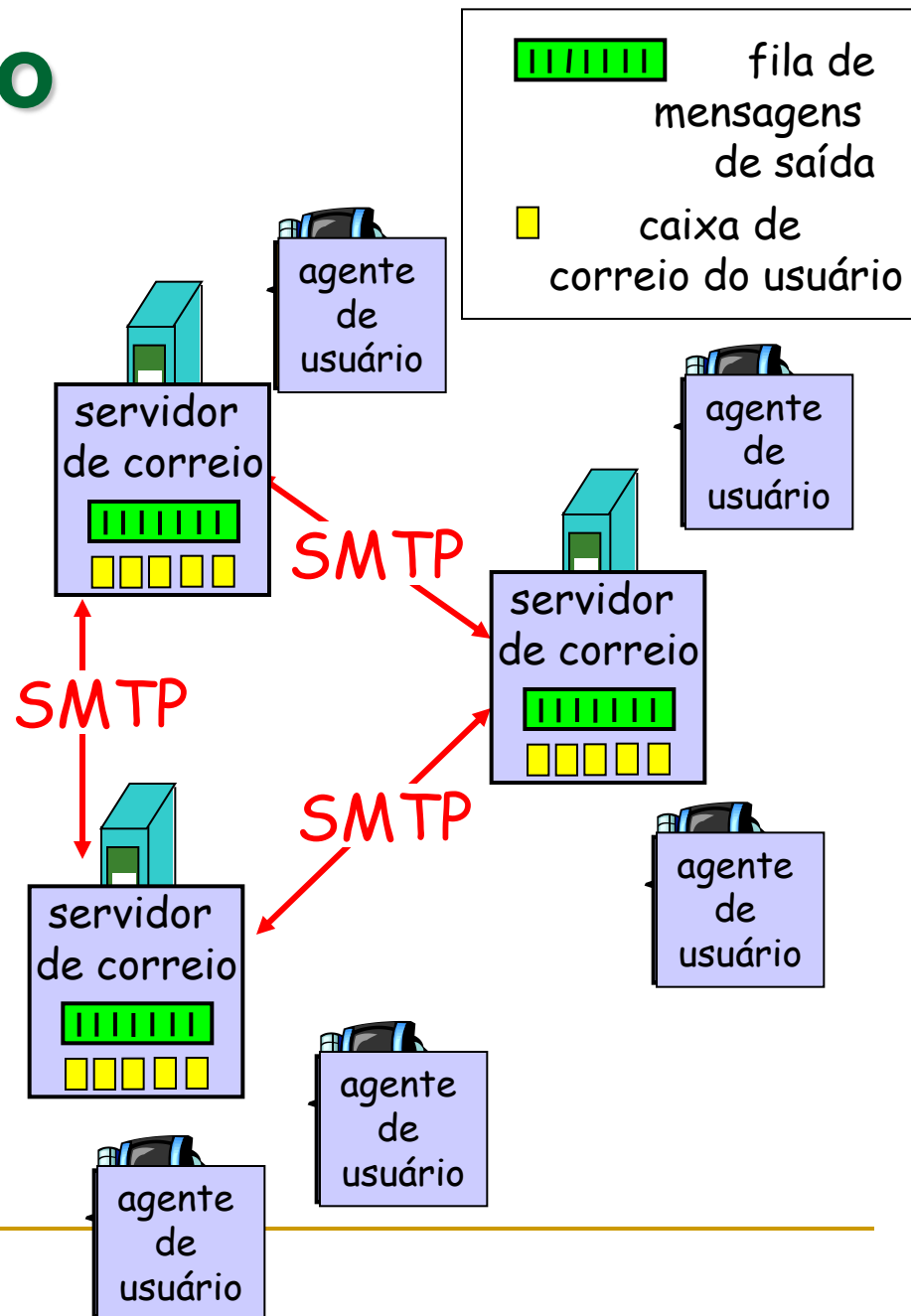
# Correio Eletrônico

## Três grandes componentes:

- agentes de usuário (UA)
- servidores de correio
- simple mail transfer protocol: SMTP

## Agente de Usuário

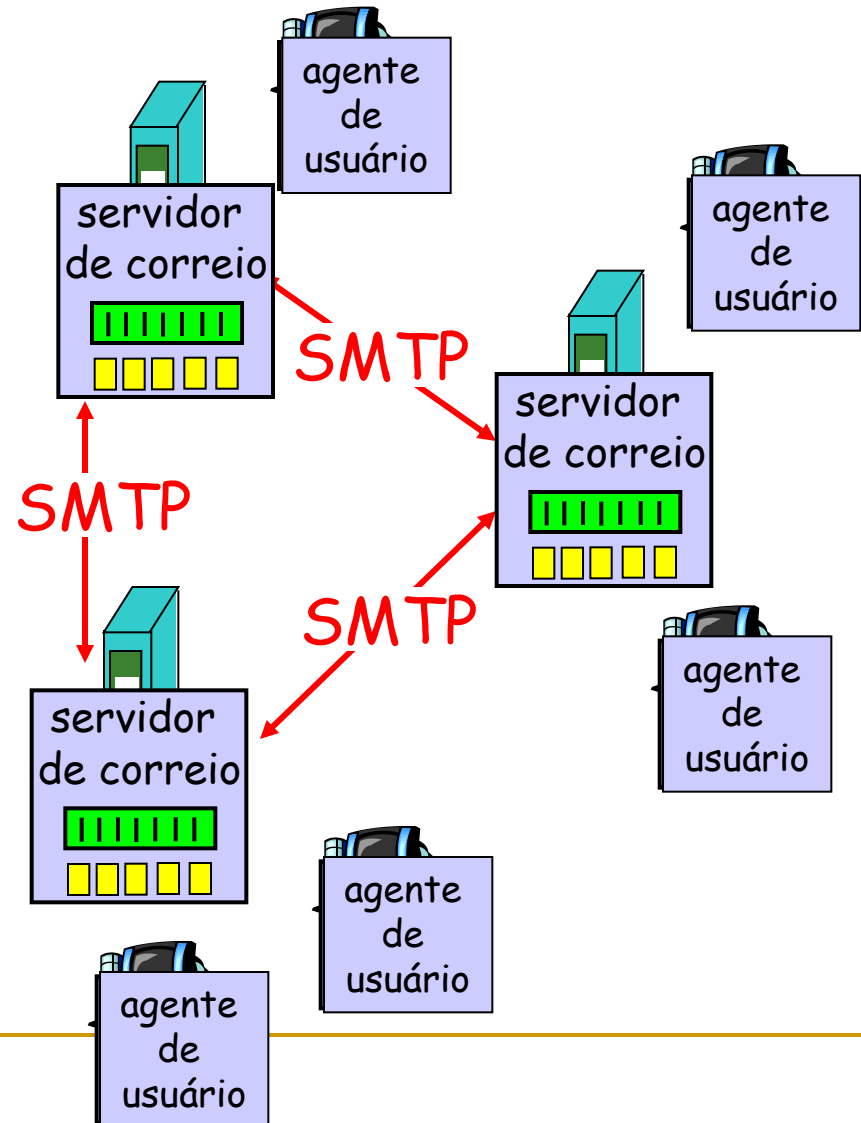
- “leitor de correio”
- compor, editar, ler mensagens de correio
- p.ex., Eudora, Outlook, elm, Netscape Messenger
- mensagens de saída e chegando são armazenadas no servidor



# Servidores de correio

## Servidores de correio

- **caixa de correio** contém mensagens de chegada (ainda não lidas) p/ usuário
- **fila de mensagens** contém mensagens de saída (a serem enviadas)
- **protocolo SMTP** entre servidores de correio para transferir mensagens de correio
  - ❑ cliente: servidor de correio que envia
  - ❑ “servidor”: servidor de correio que recebe

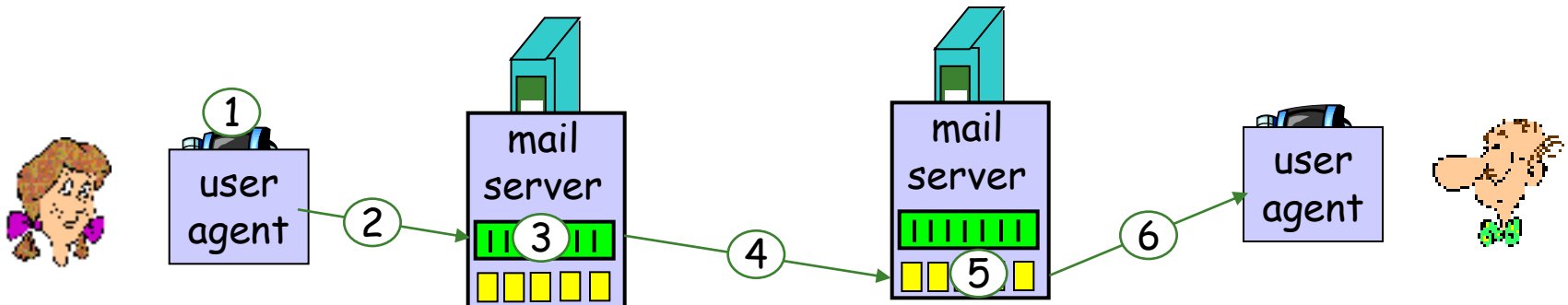


# SMTP [RFC 2821]

- usa TCP para a transferência confiável de msgs do correio do cliente ao servidor, porta 25
- transferência direta: servidor remetente ao servidor receptor
- três fases da transferência
  - ❑ *handshaking* (cumprimento)
  - ❑ transferência das mensagens
  - ❑ encerramento
- interação comando/resposta
  - ❑ **comandos**: texto ASCII
  - ❑ **resposta**: código e frase de status
- mensagens precisam ser em ASCII de 7-bits

# Ex: Alice envia uma msg para Bob

- 1) Alice usa o UA para compor uma mensagem “para” bob@someschool.edu
- 2) O UA de Alice envia a mensagem para o seu servidor de correio; a mensagem é colocada na fila de mensagens
- 3) O lado cliente do SMTP abre uma conexão TCP com o servidor de correio de Bob
- 4) O cliente SMTP envia a mensagem de Alice através da conexão TCP
- 5) O servidor de correio de Bob coloca a mensagem na caixa de entrada de Bob
- 6) Bob chama o seu UA para ler a mensagem



# Interação SMTP típica

```
S: 220 doces.br
C: HELO consumidor.br
S: 250 Hello consumidor.br, pleased to meet you
C: MAIL FROM: <ana@consumidor.br>
S: 250 ana@consumidor.br... Sender ok
C: RCPT TO: <bernardo@doces.br>
S: 250 bernardo@doces.br ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: Voce gosta de chocolate?
C: Que tal sorvete?
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 doces.br closing connection
```

# SMTP: últimas palavras

- SMTP usa conexões persistentes
- SMTP requer que a mensagem (cabeçalho e corpo) sejam em ASCII de 7-bits
- servidor SMTP usa `CRLF.CRLF` para reconhecer o final da mensagem

## Comparação com HTTP

- HTTP: *pull* (puxar)
- SMTP: *push* (empurrar)
- ambos têm interação comando/resposta, códigos de status em ASCII
- HTTP: cada objeto é encapsulado em sua própria mensagem de resposta
- SMTP: múltiplos objetos de mensagem enviados numa mensagem de múltiplas partes

# Formato de uma mensagem

SMTP: protocolo para trocar msgs de correio

RFC 822: padrão para formato de mensagem de texto:

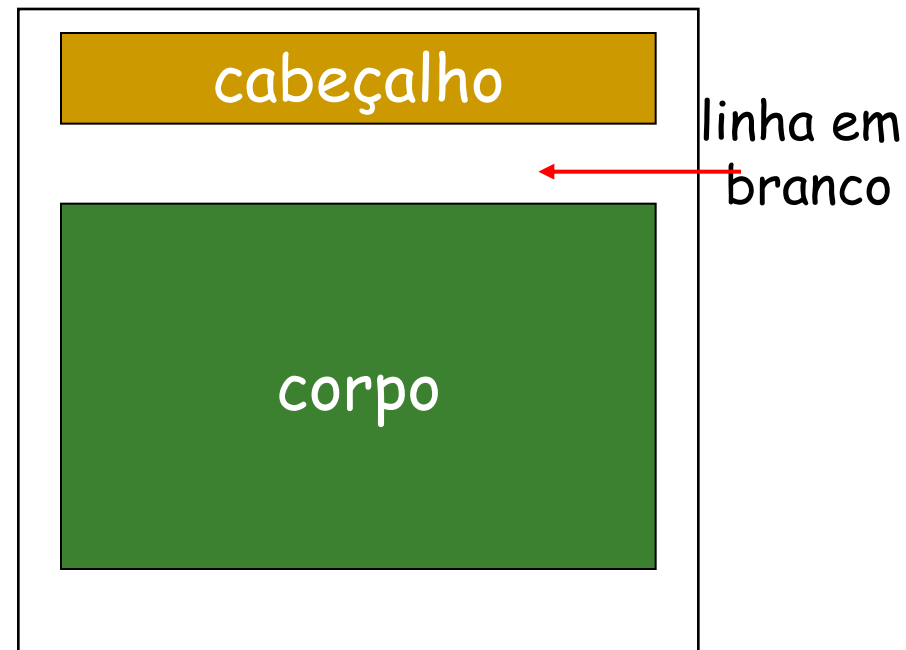
- linhas de cabeçalho, p.ex.,

- ❑ To:
- ❑ From:
- ❑ Subject:

*diferentes dos comandos de smtp!*

- corpo

- ❑ a “mensagem”, somente de caracteres ASCII





# Extensões para multimídia

- MIME: *multimedia mail extension*, RFC 2045, 2056
- linhas adicionais no cabeçalho da msg declaram tipo do conteúdo MIME

versão MIME

método usado  
p/ codificar dados

tipo, subtipo de  
dados multimídia,  
declaração parâmetros

Dados codificados

```
From: ana@consumidor.br
To: bernardo@doces.br
Subject: Imagem de uma bela torta
MIME-Version: 1.0
Content-Transfer-Encoding: base64
Content-Type: image/jpeg

base64 encoded data .....
.....
.....base64 encoded data
```

# Tipos MIME

Content-Type: `tipo/subtipo; parâmetros`

## *Text*

- subtipos exemplos: `plain`,  
`html`
- `charset="iso-8859-1"`,  
`ascii`

## *Image*

- subtipos exemplos : `jpeg`,  
`gif`

## *Video*

- subtipos exemplos : `mpeg`,  
`quicktime`

## *Audio*

- subtipos exemplos : `basic` (8-bit codificado mu-law),  
`32kadpcm` (codificação 32 kbps)

## *Application*

- outros dados que precisam ser processados por um leitor para serem “visualizados”
- subtipos exemplos : `msword`,  
`octet-stream`

# Tipo Multipart

From: alice@crepes.fr  
To: bob@hamburger.edu  
Subject: Picture of yummy crepe.  
MIME-Version: 1.0  
Content-Type: multipart/mixed; boundary=98766789

--98766789

Content-Transfer-Encoding: quoted-printable  
Content-Type: text/plain

Dear Bob,  
Please find a picture of a crepe.

--98766789

Content-Transfer-Encoding: base64  
Content-Type: image/jpeg

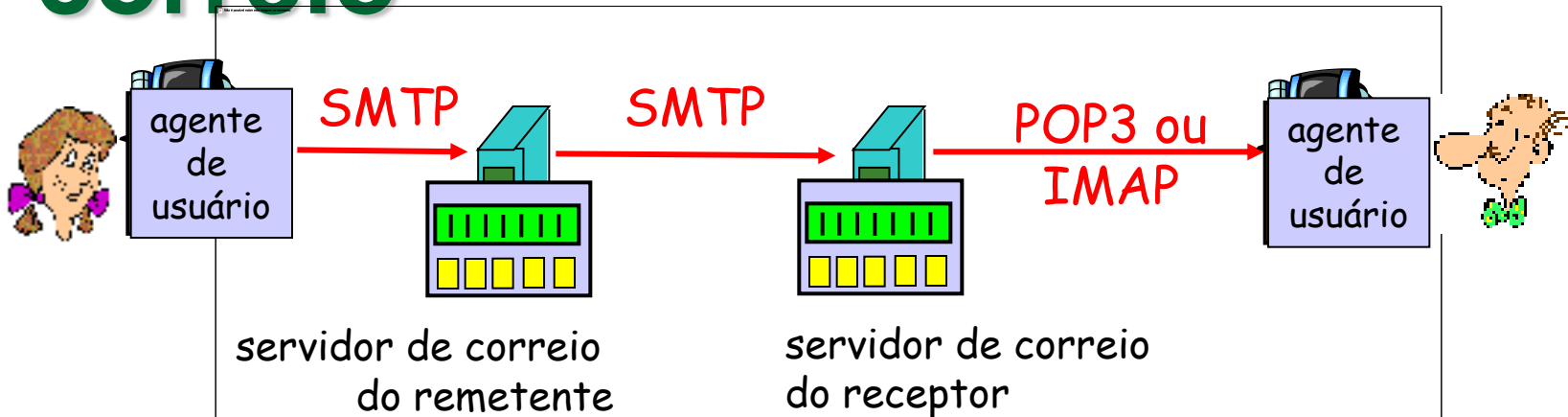
base64 encoded data .....

.....

.....base64 encoded data

--98766789--

# Protocolos de acesso ao correio



- SMTP: entrega/armazenamento no servidor do receptor
- protocolo de acesso ao correio: recupera do servidor
  - POP: Post Office Protocol [RFC 1939]
    - autorização (agente <-->servidor) e transferência
  - IMAP: Internet Mail Access Protocol [RFC 1730]
    - mais comandos (mais complexo)
    - manuseio de msgs armazenadas no servidor
  - HTTP: Hotmail , Yahoo! Mail, Webmail, etc.

# Protocolo POP3

## fase de autorização

- comandos do cliente:
  - **user**: declara nome
  - **pass**: senha
- servidor responde
  - **+OK**
  - **-ERR**

## fase de transação, cliente:

- **list**: lista números das msgs
- **retr**: recupera msg por número
- **dele**: apaga msg
- **quit**

```
S: +OK POP3 server ready
C: user ana
S: +OK
C: pass faminta
S: +OK user successfully logged on

C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: <message 1 contents>
S: .
C: dele 1
C: retr 2
S: <message 1 contents>
S: .
C: dele 2
C: quit
S: +OK POP3 server signing off
```

# POP3 (mais) e IMAP

## Mais sobre o POP3

- O exemplo anterior usa o modo “*download e delete*”.
- Bob não pode reler as mensagens se mudar de cliente
- “*Download-e-mantenha*”: copia as mensagens em clientes diferentes
- POP3 não mantém estado entre conexões

## IMAP

- Mantém todas as mensagens num único lugar: o servidor
- Permite ao usuário organizar as mensagens em pastas
- O IMAP mantém o estado do usuário entre sessões:
  - ❑ nomes das pastas e mapeamentos entre as IDs das mensagens e o nome da pasta