Tecnologias de Redes de Computadores - 90398

Apresentação 15 – Camada de rede

Pedro Gonçalves
pasg@ua.pt

Sumário

- Serviço de email:
  - Funcionamento dos serviços de email
  - Protocolos de email
  - Formato das mensagens de email
  - Portos usados
  - Questões de segurança
- Serviços P2P
  - Funcionamento básico
  - Análise do BitTorrent

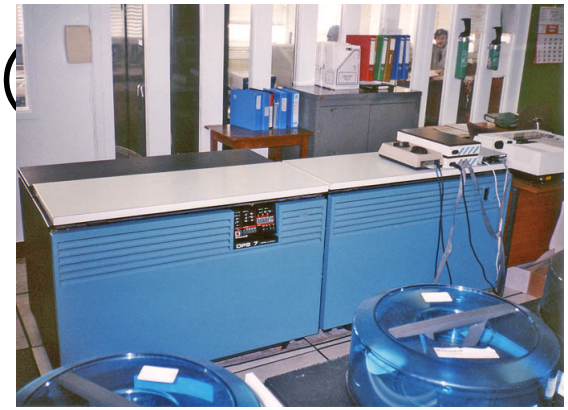
Funcionamento

Serviço de email

- Começou por ser mecanismo de comunicação entre utilizadores da mesma mainframe.
- Foi estendido para comunicação entre utilizadores de mainframes com mesmo SO.
- Interoperabilidade proposta em 1973 (RFC 561)
- SMTP proposto em 1982 (RFC 821)

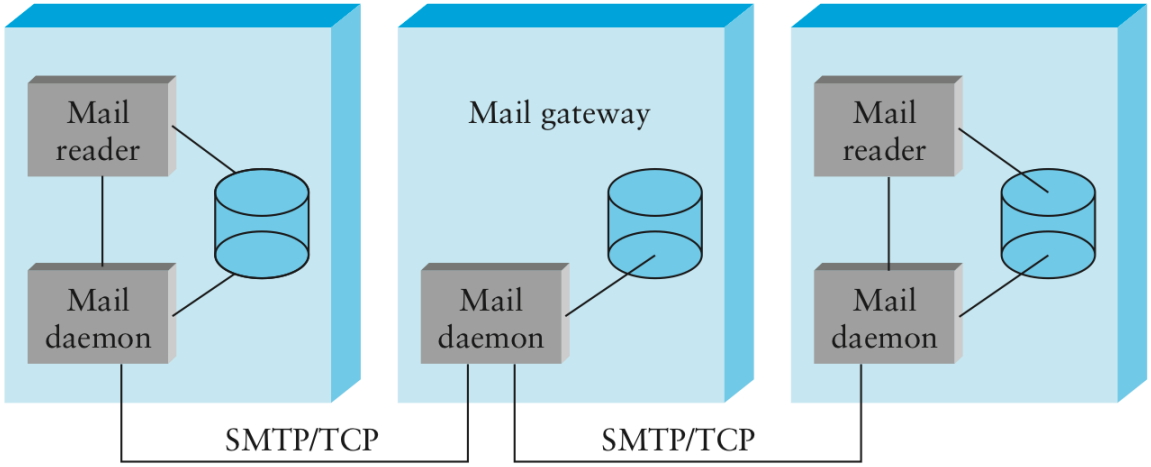
História

- Começou por ser mecanismo de comunicação entre utilizadores da mesma mainframe.
- Foi estendido para comunicação entre utilizadores de mainframes com mesmo SO.
- Interoperabilidade proposta em 1973 (
- m 1982 (RFC 821)

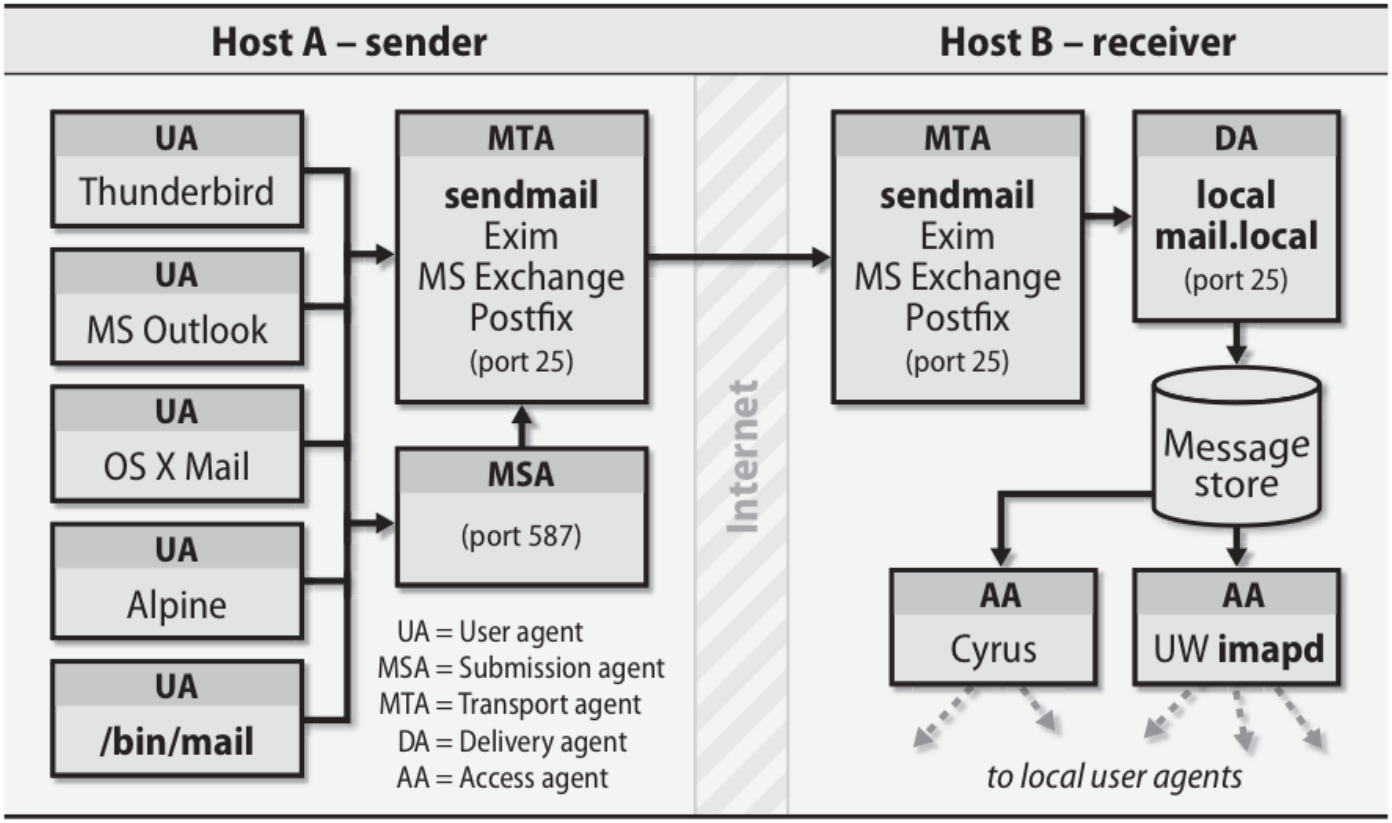


Funcionamento

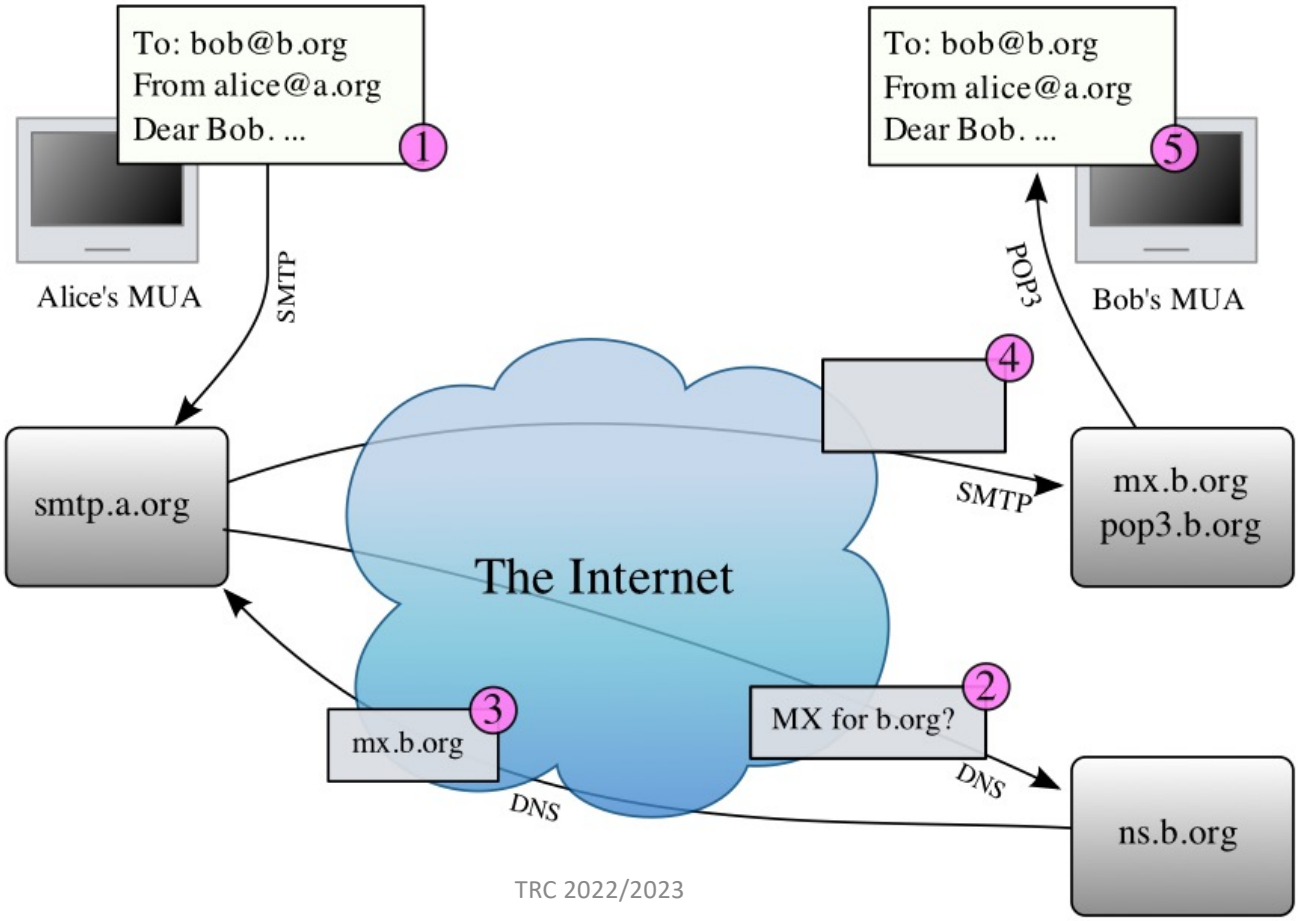
- Utilizadores servem-se de aplicações que comunicam com os seus Mail User Agent (MUA).
- Serviço efectua transporte de correio até destino.



Detalhes de funcionamento



Sequencia de operações

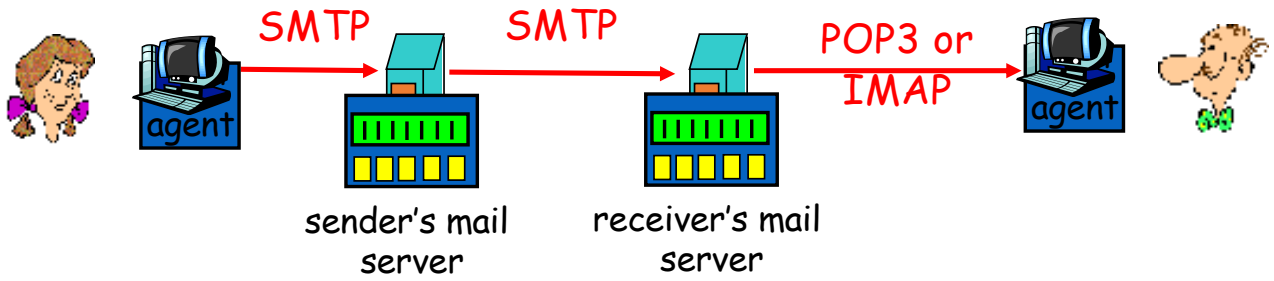


Protocolos de transporte de informação de correio

Protocolos de transporte de correio

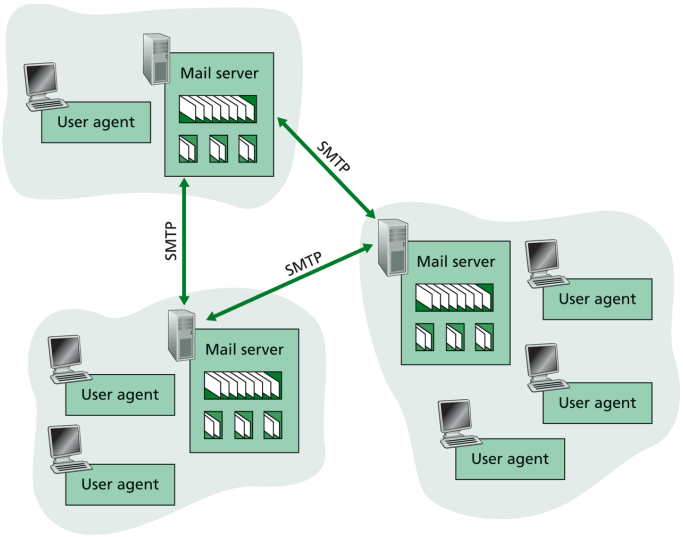
- Envio:
  - Simple Mail Transport Protocol:
    - Inicialmente proposto na RFC 821 actualmente na RFC 5321
- Recepção:
  - Post Office Protocol (POP)
  - IMAP

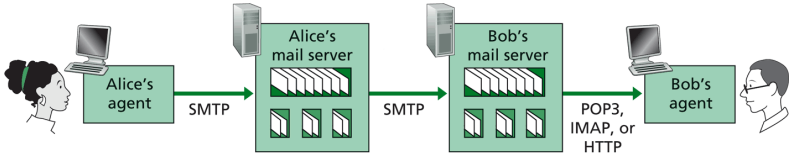
Protocolos de email



- SMTP: delivery/storage to receiver's server
- Mail access protocol: retrieval from server
  - POP: Post Office Protocol [RFC 1939]
    - authorization (agent <-->server) and download
  - IMAP: Internet Mail Access Protocol [RFC 1730]
    - more features (more complex)
    - manipulation of stored msgs on server
  - HTTP: Hotmail , Yahoo! Mail, etc.

SMTP: envio de correio electrónico






```
S: 220 mr1.its.yale.edu
C: HELO cyndra.yale.edu
S: 250  Hello cyndra.cs.yale.edu, pleased to meet you
C: MAIL FROM: <spoof@cs.yale.edu>
S: 250 spoof@cs.yale.edu... Sender ok
C: RCPT TO: <yry@yale.edu>
S: 250 yry@yale.edu ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: Date: Wed, 23 Jan 2008 11:20:27 -0500 (EST)
C: From: "Y. R. Yang" <yry@cs.yale.edu>
C: To: "Y. R. Yang" <yry@cs.yale.edu>
C: Subject: This is subject
C:
C: This is the message body!
C: Please don't spoof!
C:
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 mr1.its.yale.edu closing connection
```

POP and IMAP para recepção de correio

- These are protocols for how to deal with a mailbox server
- To SEND mail, both POP and IMAP clients use SMTP

- POP and IMAP clients need configuration:
  - mailbox server
  - SMTP server

Client/Server – 1 de 3 modelos

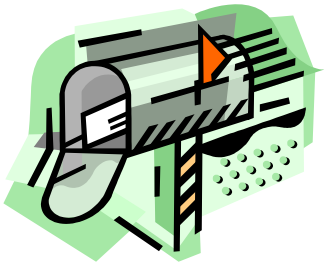
- Offline (POP3)
  - Cliente liga-se ao servidor e puxa todo o email
  - Tudo fica alojado no cliente
- Online (IMAP original)
  - Client liga-se ao servidor em cada transacção
  - Tudo fica no servidor
- Desligado (IMAP)
  - Armazenamento feito no cliente e no servidor
  - Server é sempre prevalente e cliente tem que se sincronizar com ele.

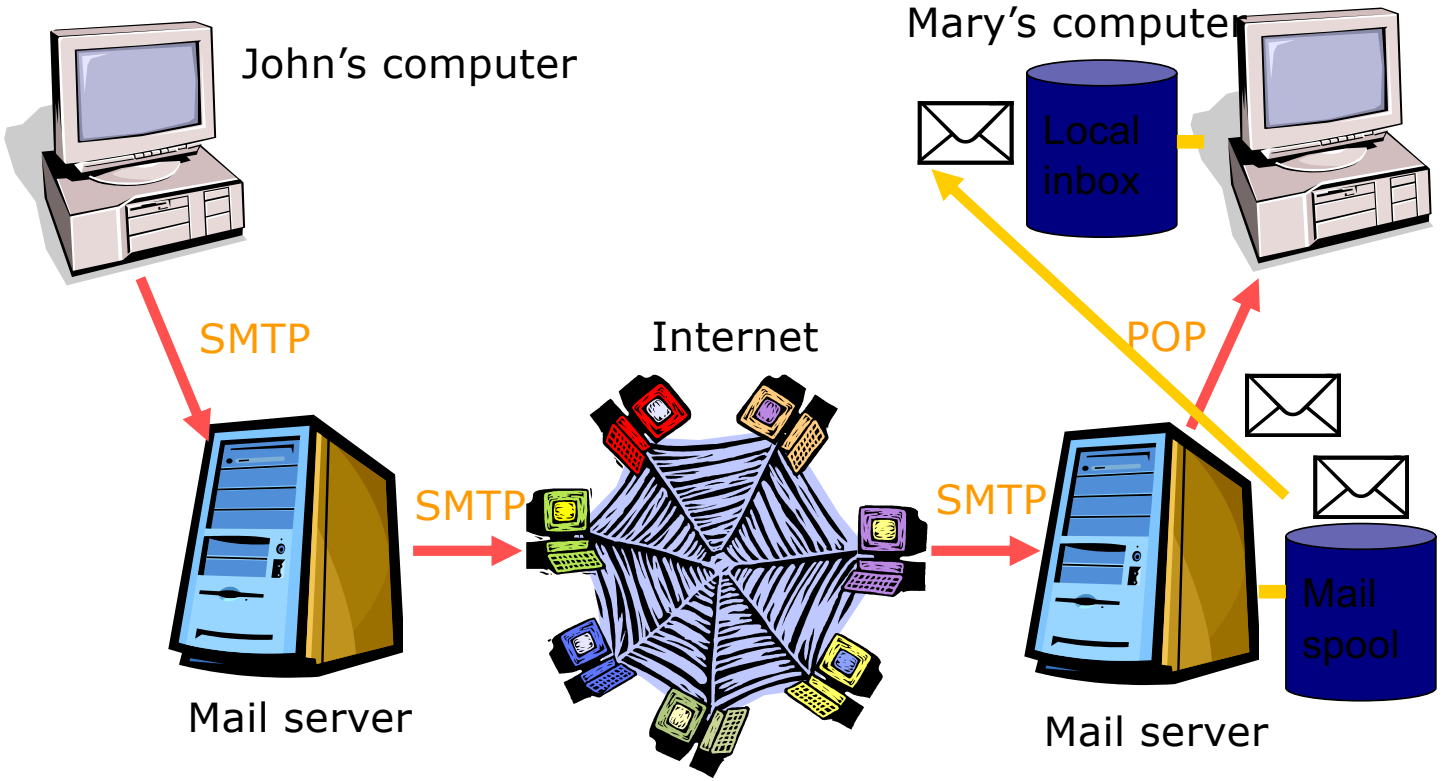
POP - Post Office Protocol

- POP client liga-se ao servidor e copia tudo para repositório local.
- Suporta leitura de correio offline
- Interacção típica com o servidor:
  - Liga-se ao servidor
  - Recebe todas as mensagens
  - Armazena mensagens em repositório local
  - Apaga mensagens do servidor
  - Desliga-se do servidor

- Pode ser configurado para manter mensagens no servidor.

Funcionamento do POP





Sessão POP

```
$ telnet/port=110 mail.opus1.com
Trying... Connected to MAIL.OPUS1.COM.

+OK cello.Opus1.COM MultiNet POP3 Server Process V4.0(1) at Fri 20-
Sep-96 3:21PM-MST
user trumbo
+OK User name (trumbo) ok. Password, please.
pass thisismypasswordincleartext
+OK 3 messages in folder NEWMAIL (V4.0)
list 2
+OK 2 7124
stat
+OK 3 14749
last
+OK 0
quit
+OK POP3 MultiNet cello.Opus1.COM Server exiting (3 NEWMAIL messages
left)

Connection closed by Foreign Host
$
```

‘list’ gives individual message size in bytes

‘stat’ gives total message size in bytes

Protocolo POP3: acesso ao email

Authorization phase

- client commands:
  - **user**: declare username
  - **pass**: password
- server responses
  - **+OK**
  - **-ERR**

Transaction phase, client:

- **list**: list message numbers
- **retr**: retrieve message by number
- **dele**: delete
- **quit**

```
S: +OK POP3 server ready
C: user alice
S: +OK
C: pass hungry
S: +OK user successfully logged on
```

```
C: list
S: 1 498å
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
```

```
%telnet <netid>.mail.yale.edu 110
%openssl s_client -connect pop.gmail.com:995
```

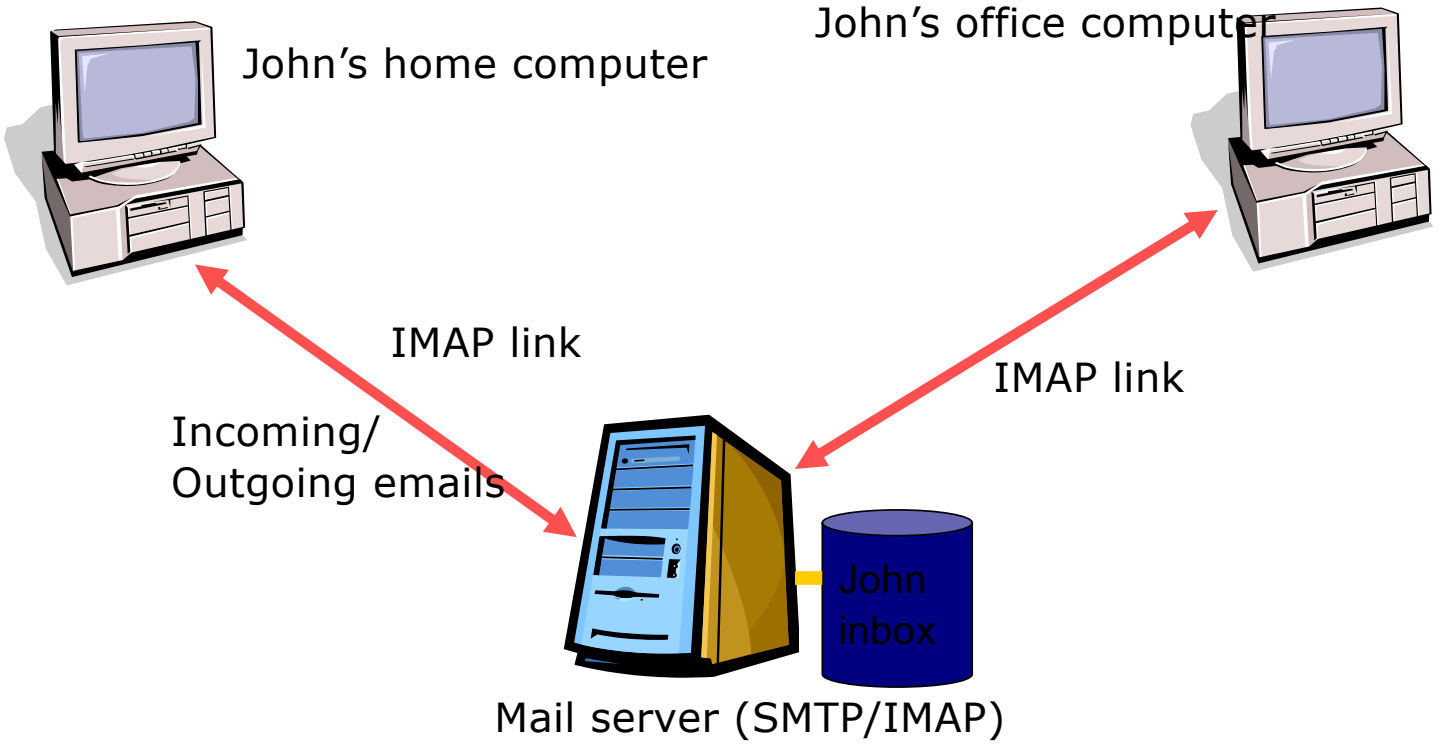
Interactive Mail Access Protocol IMAP

- Aceita os modos: On-line, off-line, or disconnected mode operation
- Permite o controlo de pastas de qualquer local
- Permite multiplas caixas num mesmo servidor
- Permite a criação e alteração de pastas no servidor

Permite procuras em cima do servidor

Permite acesso ao servidor a múltiplos clientes

Leitura de correio IMAP



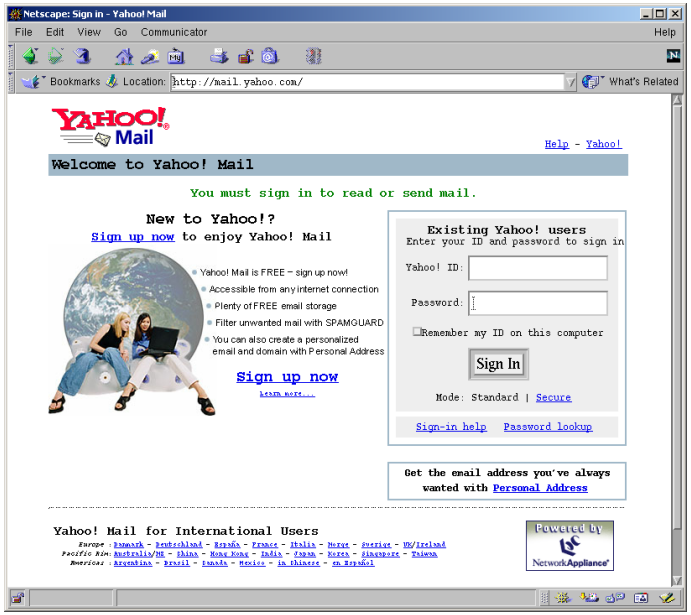
Leitura de correio IMAP

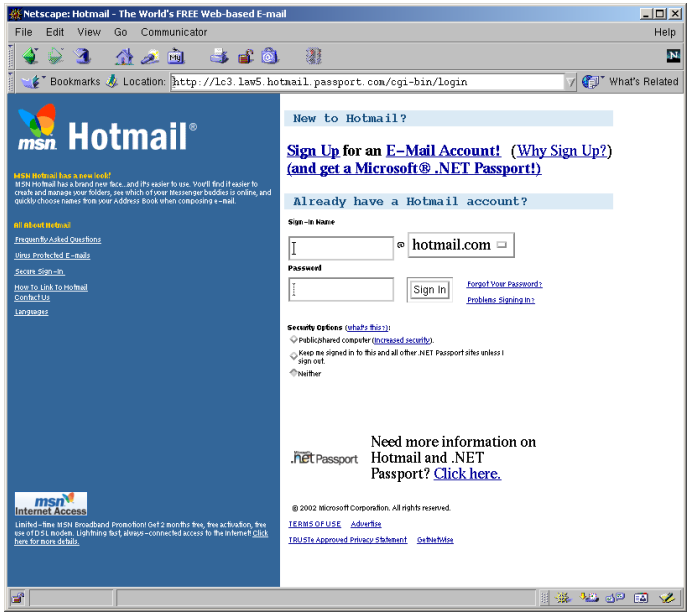
```
Trying 127.0.0.1...
Conn ected to localhost.
Escape character is '^]'.
OK Dovecot ready.
1 login john@example.com summersun
1 OK Logged in.
list "" "*”
* LIST (\HasNoChildren) "." "INBOX"
2 OK List completed. 3 select "INBOX” * FLAGS (\Answered \Flagged \Deleted \Seen \Draft)
* OK [PERMANENTFLAGS (\Answered \Flagged \Deleted \Seen \Draft \*)] Flags permitted.
* 1 EXISTS
* 0 RECENT
* OK [UIDVALIDITY 1180039205] UIDs valid
* OK [UIDNEXT 3] Predicted next UID
3 OK [READ-WRITE] Select completed.
```

```
4 fetch 1 all
* 1 FETCH (FLAGS (\Seen) INTERNALDATE .....
4 OK Fetch completed.
5 fetch 1 body[]
* 1 FETCH (BODY[] {474}
Return-Path: <steve@example.com>
X-Original-To: john@example.com
Delivered-To: john@example.com
Received: from example.com (localhost [127.0.0.1])
      by ... (Postfix) with ESMTP id 692DF379C7
      for <john@example.com>; Fri, 18 May 2007 22:59:31 +0200 (CEST)
Message-Id: <...>
Date: Fri, 18 May 2007 22:59:31 +0200 (CEST)
From: steve@example.com
To: undisclosed-recipients:;

Hi John,

just wanted to drop you a note.
)
5 OK Fetch completed.
```

Web-based e-mail - HTTP

- Can deliver mail message in web page format.
- More reliable to use POP and IMAP than HTTP mail account.





Portos email

- SMTP
  - 25
  - Sec SMTP: 465
- POP:
  - 110
  - sPOP: 995
- IMAP:
  - 143
  - secIMAP: 993
- Webmail: 80

Share de Mail Transport Agents

- December 2009 survey

MTA	Source	Market share 2009	2007	2001
Exim	exim.org	30%	20%	8%
Postfix	postfix.org	20%	15%	2%
MS Exchange	microsoft.com/exchange	20%	22%	4%
sendmail	sendmail.org	19%	29%	60%
All others	–	<3% each	<3% each	< 3% each

http://securityspace.com

Formato das mensagens

Anatomia de mensagem de email

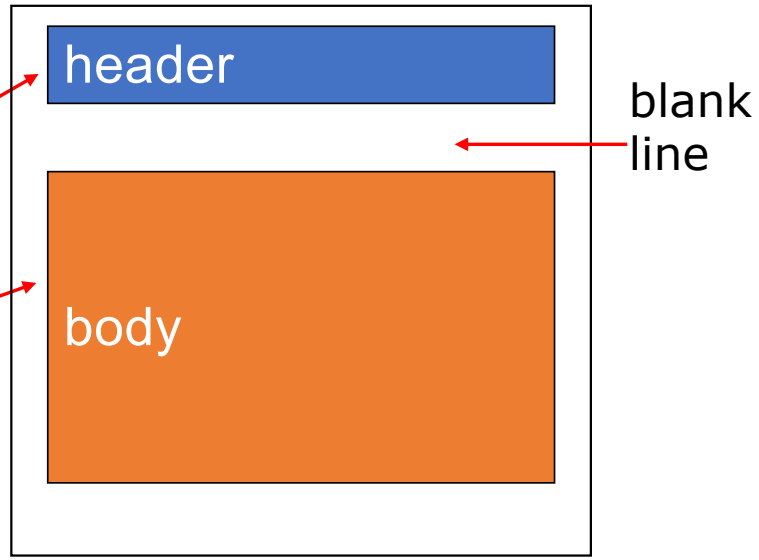
- Contém:
  - Envelope (nem sempre visível)
  - Cabeçalho: Definido na RFC 5322
    - Campos : From, To, Subject, Date, Message-ID
  - Corpo: Definido nas RFC 2045 a 2049
    - Inicialmente em texto (ASCII 7 bits)
    - Pode incluir corpo escrito em HTML
    - Inclui um conjunto de elemtos segundo uma norma designada de MIME
    - Formato HTML é muitas vezes usado como técnica de phishing

Formato das mensagens de Mail

SMTP: protocolo para troca de mensagens de email

RFC 822: standard para formato da message:

- Header,
  - To:
  - From:
  - Subject:
- Body
  - A mensagem em caracteres ASCII



Formato da Mensagem : Multimedia Extensions

- MIME: extensão multimedia para email, RFC 2045, 2056
- Linhas adicionais no header declaram o MIME content type

MIME version

multimedia data type, subtype, parameter declaration

method used to encode data

encoded data

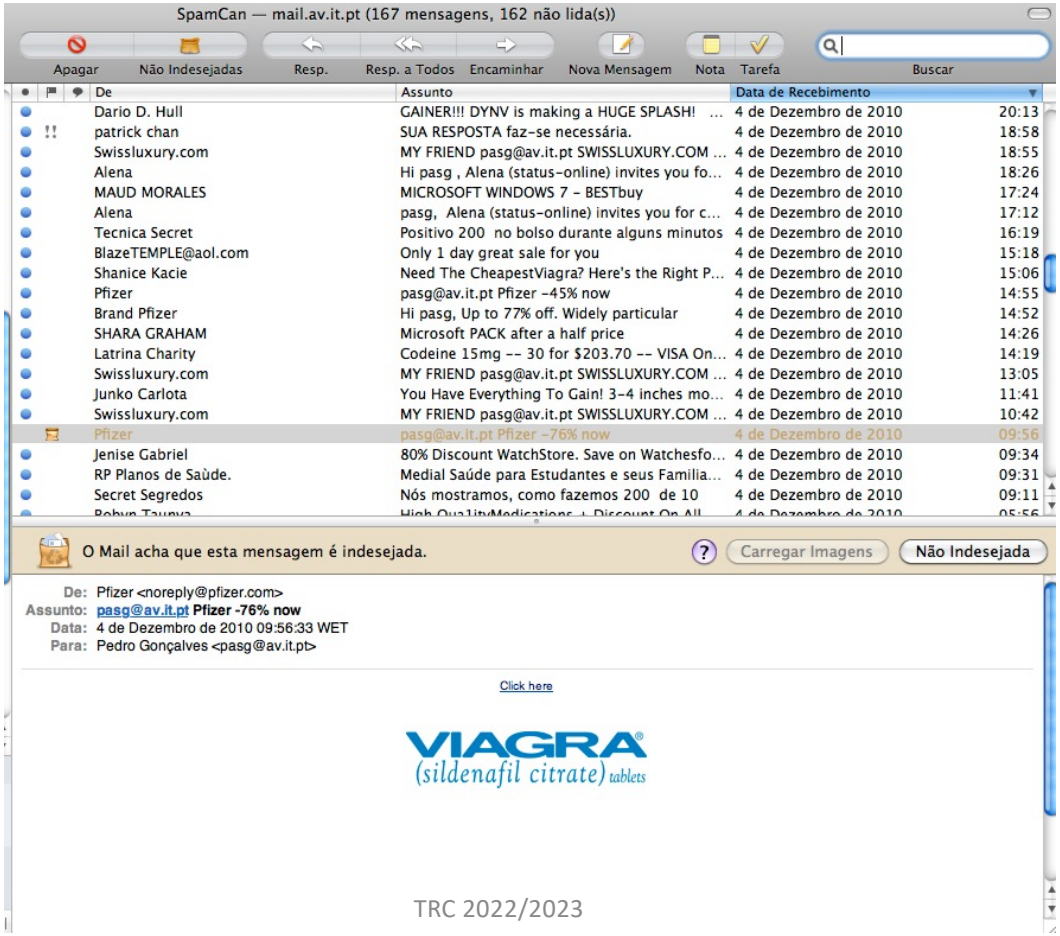
```
From: yry@cs.yale.edu
To: cs433@cs.yale.edu
Subject: Network map.
MIME-Version: 1.0
Content-Type: image/jpeg
Content-Transfer-Encoding: base64

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

Multipart Type: como funciona o Attachment

```
From: yry@cs.yale.edu
To: cs433@cs.yale.edu
Subject: Network map.
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary=98766789

--98766789
Content-Transfer-Encoding: quoted-printable
Content-Type: text/plain

Hi,
Attached is network topology map.
--98766789
Content-Transfer-Encoding: base64
Content-Type: image/jpeg

base64 encoded data .....
.........................
......base64 encoded data
--98766789--
```

Problemas de segurança

Cabeçalho email

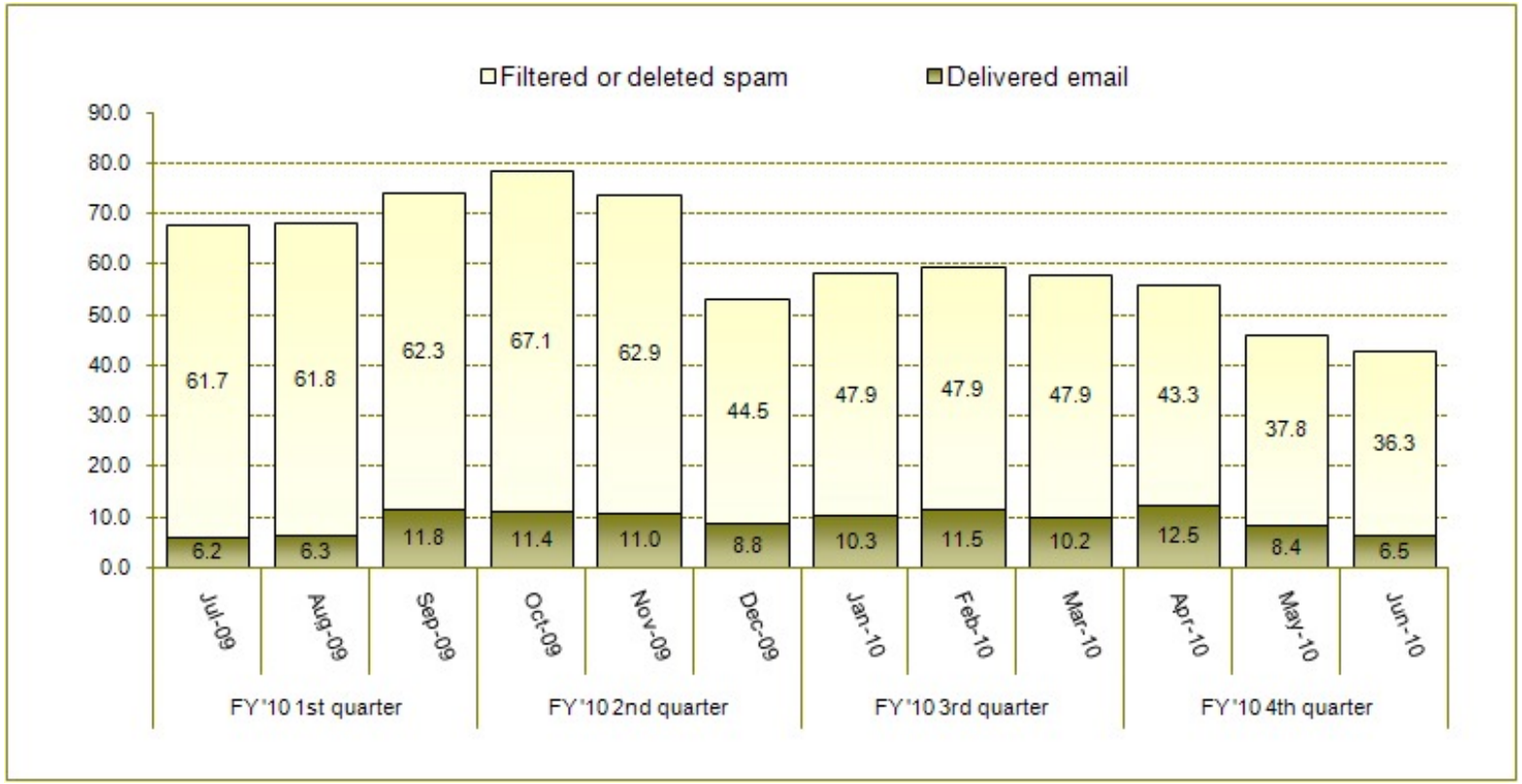


Envelope

Return-Path: <**eiabe7907@spitfireuk.net**>Received: by av.it.pt (CommuniGate Pro PIPE 5.2.11) with PIPE id 55737884; Sat, 04 Dec 2010 09:56:46 +0000
Received: from [**82.136.38.110**] (HELO spitfireuk.net)
by av.it.pt (CommuniGate Pro SMTP 5.2.11)
with ESMTP id 55737877 for pasg@av.it.pt; Sat, 04 Dec 2010 09:56:32 +0000
Received-SPF: fail
receiver=av.it.pt; client-ip=82.136.38.110; envelope-from=eiabe7907@spitfireuk.net
From: **Pfizer <noreply@pfizer.com>**
To: pasg@av.it.pt
Subject: pasg@av.it.pt Pfizer -76% now
Mime-Version: 1.0
Content-Type: text/html; charset="UTF-8"
Content-Transfer-Encoding: ISO-8859-1
Date: Sat, 04 Dec 2010 09:56:33 +0000
Message-ID: auto-000055737877@av.it.pt

De: Pfizer <noreply@pfizer.com>
Assunto: pasg@av.it.pt **Pfizer -76% now**
Data: 4 de Dezembro de 2010 09:56:33 WET
Para: Pedro Gonçalves <pasg@av.it.pt>

Estatísticas de spam



http://www.yale.edu/its/metrics/email/index.html

Problemas de segurança POP

```
$ telnet/port=110 mail.opus1.com
Trying... Connected to MAIL.OPUS1.COM.

+OK cello.Opus1.COM MultiNet POP3 Server Process V4.0(1) at Fri 20-Sep-96
3:21PM-MST
user trumbo
+OK User name (trumbo) ok. Password, please.
pass thisismypasswordincleartext
+OK 3 messages in folder NEWMAIL (V4.0)
list 2
+OK 2 7124
stat
+OK 3 14749
last
+OK 0
quit
+OK POP3 MultiNet cello.Opus1.COM Server exiting (3 NEWMAIL messages left)

Connection closed by Foreign Host
$
```

You can test passwords by connecting to the POP port

Medidas

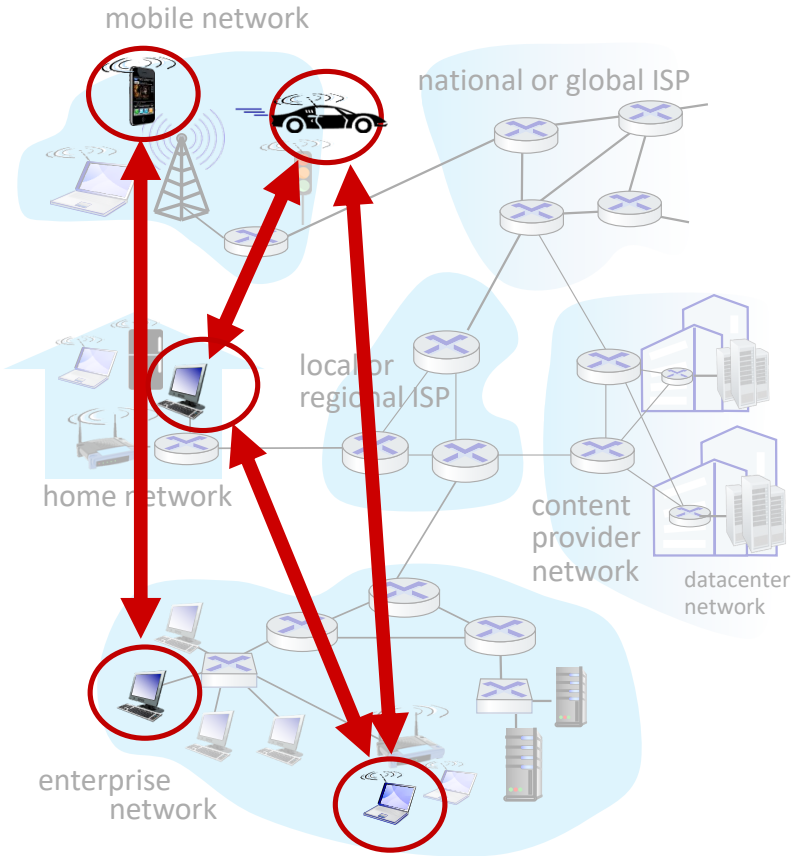
- Requer sempre autenticação do utilizador
- Usar SSL
  - sSMTP
  - sPOP
  - sIMAP

- Filtar SPAM
- Filtrar Virus
- Não usar clientes de email propensos a infecções

Aplicações P2P

Peer-to-peer (P2P) architecture

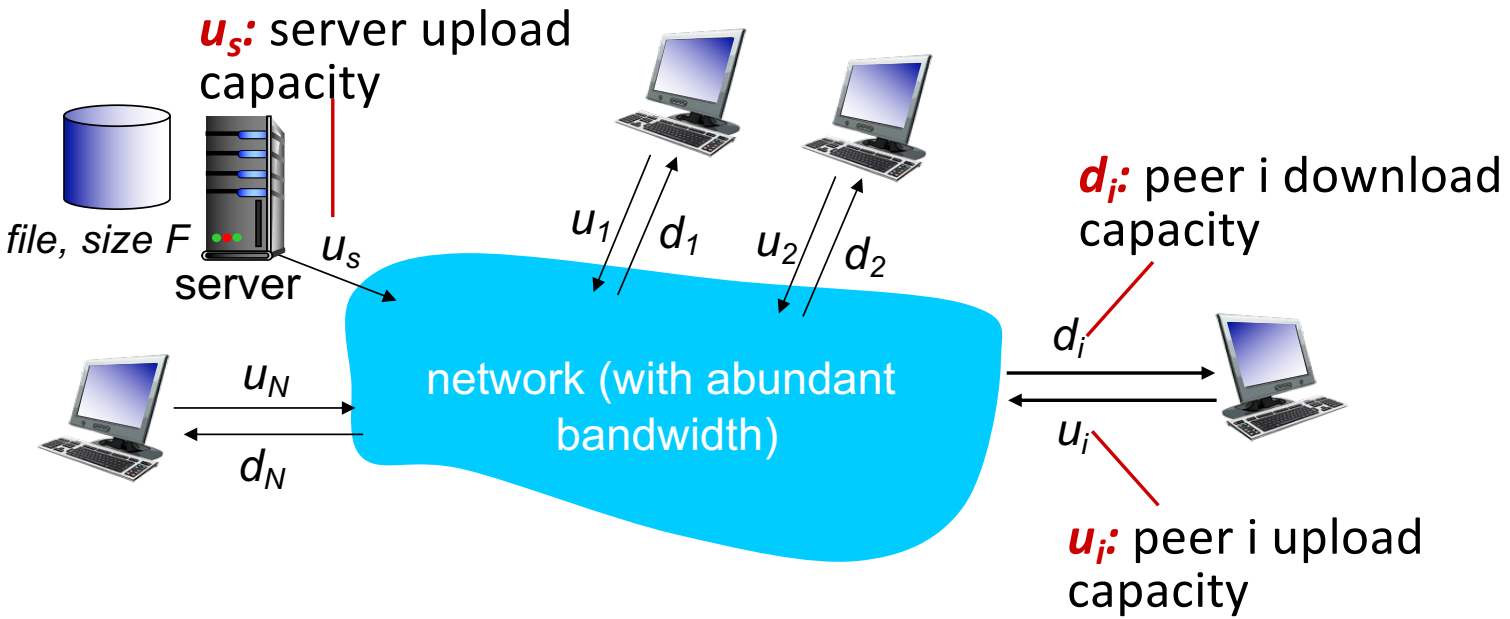
- *nenhum servidor sempre ligado*
- *sistemas finais arbitrários comunicam-se diretamente*
- *pares solicitam serviço de outros pares, fornecem serviço em troca de outros pares*
- *auto escalabilidade - novos pares trazem nova capacidade de serviço e novas procuras de serviço*
- *os pares estão conectados de forma intermitente e mudam os endereços IP*
- *gestão complexa*
- *exemplos: compartilhamento de arquivos P2P (BitTorrent), streaming (KanKan), VoIP (Skype)*



Distribuição de ficheiros: cliente-servidor vs P2P

Q: quanto tempo para distribuir o arquivo (tamanho F) de um servidor para N peers?

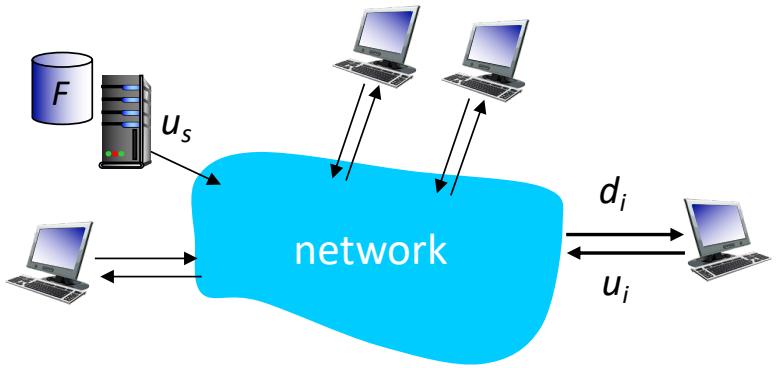
- a capacidade de upload / download de pares é um recurso limitado



Distribuição de ficheiros: client-server

- *server transmission:* must sequentially send (upload) *N* file copies:
  - time to send one copy: F/u_s
  - time to send *N* copies: NF/u_s

- *client:* each client must download file copy
  - d_{min} = min client download rate
  - min client download time: F/d_{min}



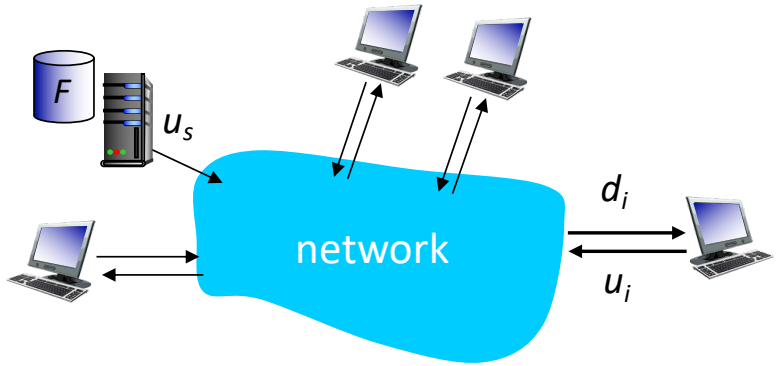
time to distribute F to N clients using client-server approach

$$D_{c\text{-}s} \geq max\{NF/u_s, F/d_{min}\}$$

increases linearly in N

Distribuição de ficheiros: P2P

- *server transmission:* must upload at least one copy:
  - time to send one copy: F/u_s
- *client:* each client must download file copy
  - min client download time: F/d_{min}
- *clients:* as aggregate must download NF bits
  - max upload rate (limiting max download rate) is $u_s + \Sigma u_i$



time to distribute F to N clients using P2P approach

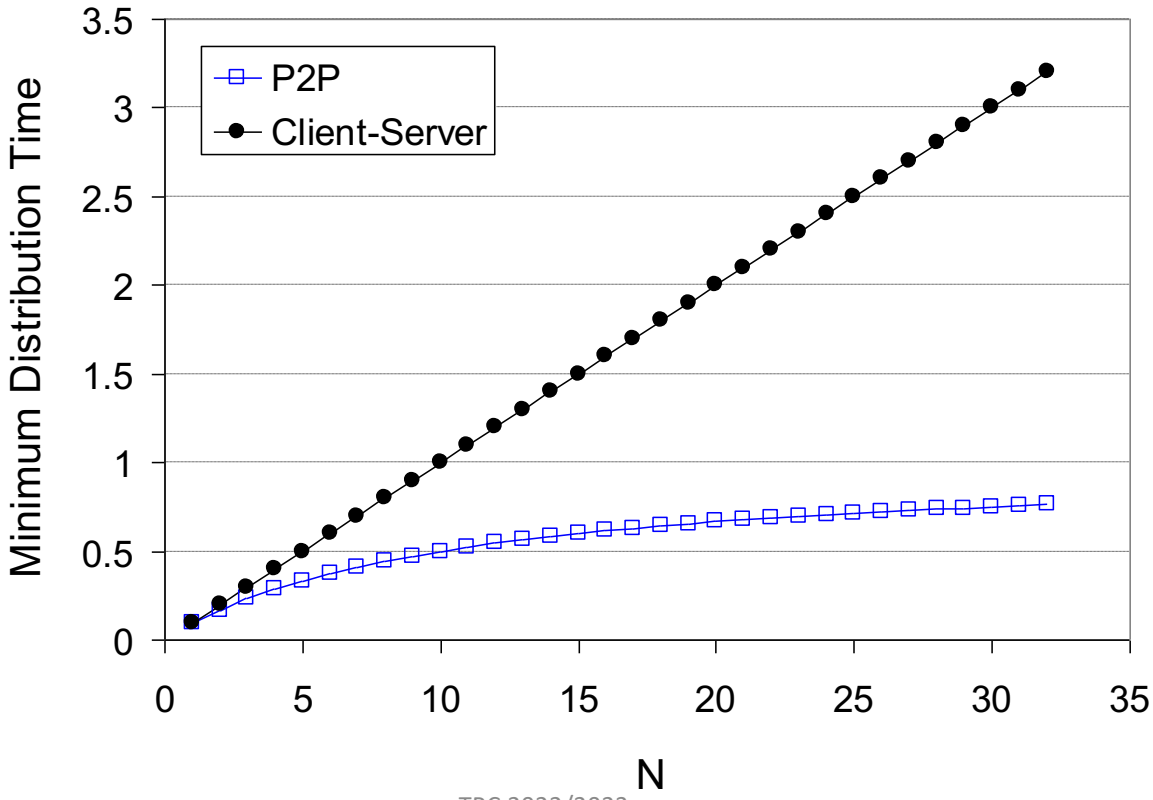
$$D_{P2P} \geq max\{F/u_s, F/d_{min}, NF/(u_s + \Sigma u_i)\}$$

increases linearly in N …

… but so does this, as each peer brings service capacity

Client-server vs. P2P: exemplo

client upload rate = u, F/u = 1 hour, $u_s = 10u$, $d_{min} \geq u_s$

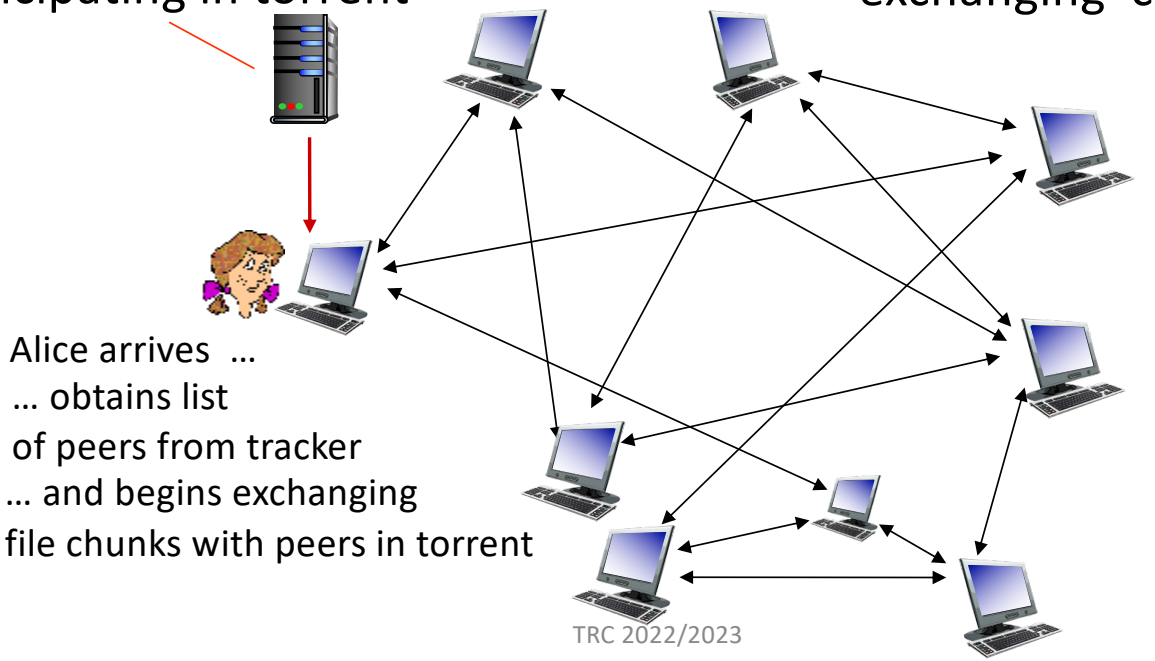


P2P file distribution: BitTorrent

- file dividido em pedaços de 256Kb
- peers da torrent enviam/recebem pedaços

tracker: tracks peers participating in torrent

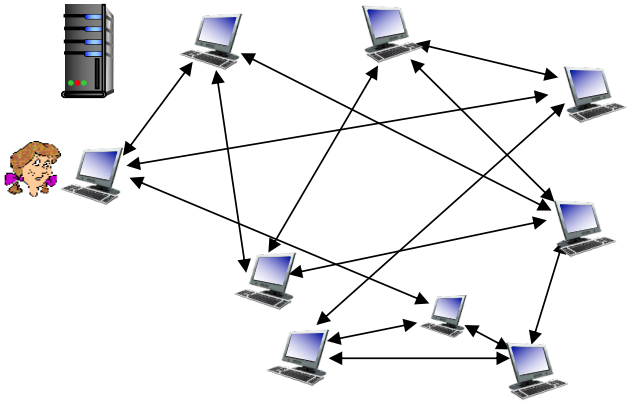
torrent: group of peers exchanging chunks of a file



Alice arrives …
… obtains list
of peers from tracker
… and begins exchanging
file chunks with peers in torrent

P2P file distribution: BitTorrent

- peer joining torrent:
  - não tem pedaços, mas irá acumulá-los ao longo do tempo de outros pares
  - registra-se com o rastreador para obter a lista de pares, conecta-se ao subconjunto de pares ("vizinhos")



- durante o download, o par carrega pedaços para outros pares
- o par pode mudar os pares com quem troca pedaços
- peers podem entrar e sair
- uma vez que o peer tenha o arquivo inteiro, ele pode (egoisticamente) sair ou (altruisticamente) permanecer no torrent

BitTorrent: requesting, sending file chunks

Pedindo pedaços:

- a qualquer momento, diferentes pares têm diferentes subconjuntos de blocos de arquivos
- periodicamente, Alice pede a cada par uma lista de pedaços que eles têm
- Alice solicita pedaços em falta de colegas, os mais raros primeiro

Enviando pedaços:

- Alice envia blocos para aqueles quatro pares que estão enviando blocos com a taxa mais alta
- outros colegas são sufocados por Alice (não receba pedaços dela)
- reavalie os 4 principais a cada 10 segundos
- a cada 30 segundos: seleciona aleatoriamente outro par, começa a enviar pedaços
- “Desmarque de forma otimista” este colega
- o par recém-escolhido pode se juntar aos 4 primeiros

Reflexão

- Qual a intervenção do serviço de nomes no funcionamento dos serviços de correio?
- Qual a função do registo do tipo MX do DNS? Qual a consequência de um erro no valor do registo MX?

- Proponha um mecanismo simples para detectar forged emails.
- Em que consistem as blacklists dos serviços de emails? Que pode fazer para não ser incluido nessas listas?
- O que torna os sistemas de P2P tão atractivos para a distribuição de ficheiros?

E é tudo...

- Questões?
- Comentários?