

Disciplina: Redes de Computadores

Nível de Aplicação

DNS, Correio Eletrônico, WWW

Profa. Débora Muchaluat Saade

debora@midia.com.uff.br

Protocolos de Nível de Aplicação

Redes de Computadores



- ➔ Incorporam a funcionalidade das camadas de Sessão, Apresentação e Aplicação do modelo OSI
- ➔ As aplicações trocam dados através dos serviços providos pela camada de transporte
- ➔ Utilizam em grande parte o paradigma cliente-servidor e a interação por troca de mensagens
 - Clientes solicitam serviços
 - Servidores atendem os pedidos de serviço solicitados

Protocolos de Nível de Aplicação TCP/IP

Redes de Computadores

➤ Resolução de Nomes

- DNS (Domain Name System)

➤ Transferência de arquivos e documentos

- FTP (File Transfer Protocol)
- TFTP (Trivial FTP)
- HTTP (HyperText Transfer Protocol)
- NFS (Network File System)

➤ Emulação de terminal

- TELNET
- RLOGIN

➤ Correio Eletrônico

- SMTP (Simple Mail Transfer Protocol)
- POP (Post Office Protocol)
- IMAP (Internet Message Access Protocol)

➤ Gerência

- SNMP (Simple Network Management Protocol)

...

DNS

Domain Name System

DNS

➔ Utilização de uma estrutura hierárquica para mapear nomes em recursos

- máquinas (end. IP), caixas de correio, servidores, ...
- Um nome é um conjunto de rótulos separados por “.”
- Em um nome, cada nível de hierarquia é descrito por um sufixo (conjunto de rótulos) precedido de “.”
- Nomes são case-insensitive (edu ou EDU ou Edu)

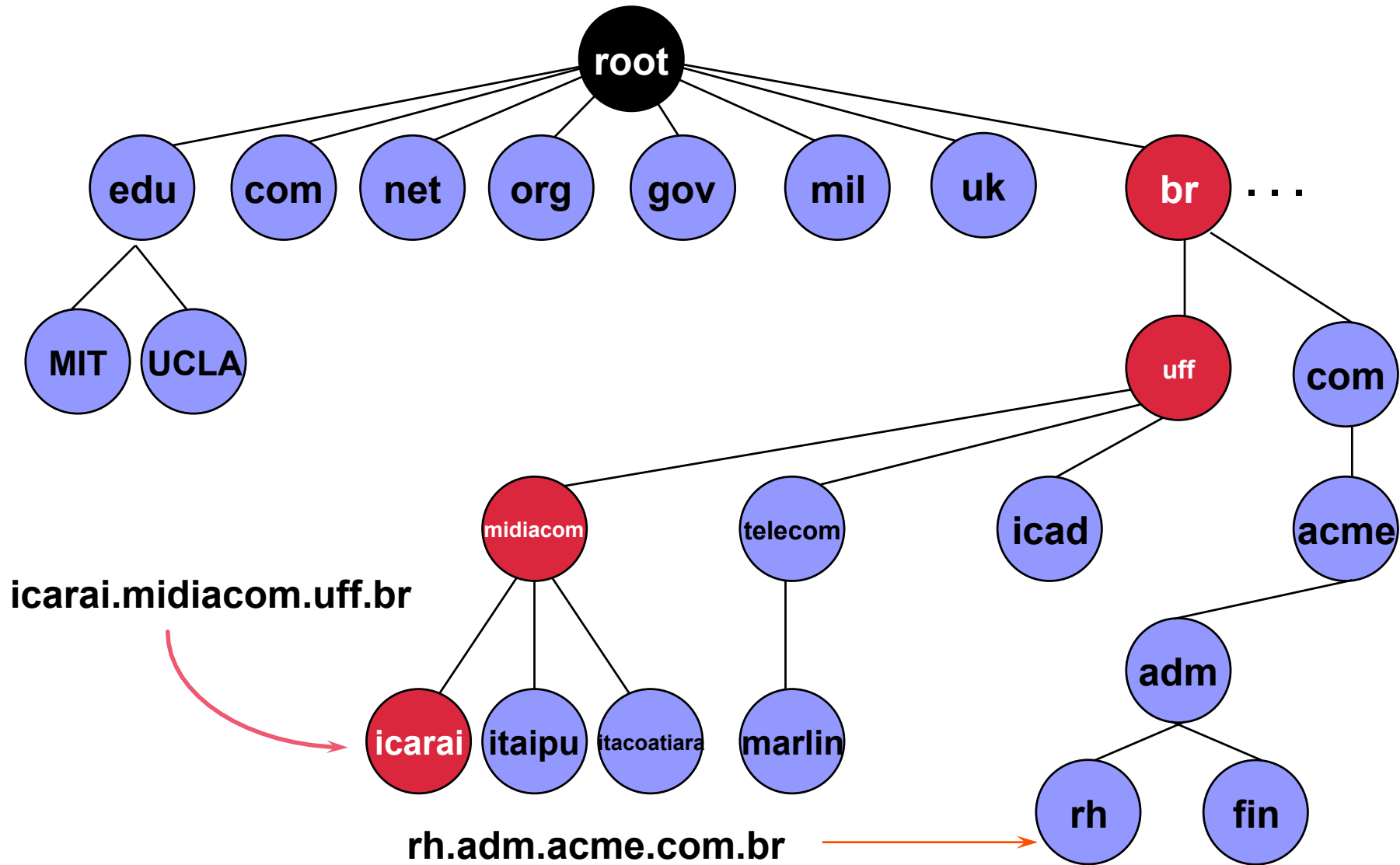
➔ Exemplos de nomes no DNS:

- odeon.csd.nasa.gov
 - mail.iis.com.br
 - www.cursos.telemidia.puc-rio.br
- 1o. nível
- 2o. nível
- 

Nomes Hierárquicos

- Domínio é um conjunto de rótulos que especifica um nome. Ex.: serv1.acme.com.br
- Cada rótulo tem no máximo 63 caracteres e nomes completos 255
- Um domínio é responsável por uma parte do espaço de nomes (espaço que define todos os nomes possíveis no DNS)
 - acme.com.br, com.br, br também são domínios
 - A responsabilidade pelas partes do espaço de nomes é delegada entre domínios
- Domínio principal: “ ” (branco), “.” (raiz)
- Domínios secundários: com, gov, net, br, fr, uk, ...
- Domínios terciários: nasa.gov, puc-rio.br, com.br, net.br, co.uk, co.jp, ...

Nomes de Domínio



DNS

➔ DNS abrange:

- as regras de sintaxe para os nomes de domínio e a delegação de autoridade sobre nomes
- o mecanismo de mapeamento de nomes em end. IP

➔ DNS é implementado como um sistema distribuído

- Paradigma cliente-servidor (servidor na porta TCP 53 e UDP 53)
- Resolução de nomes usa UDP
- Replicação da base de dados em servidores secundários usa TCP

➔ DNS possui três componentes:

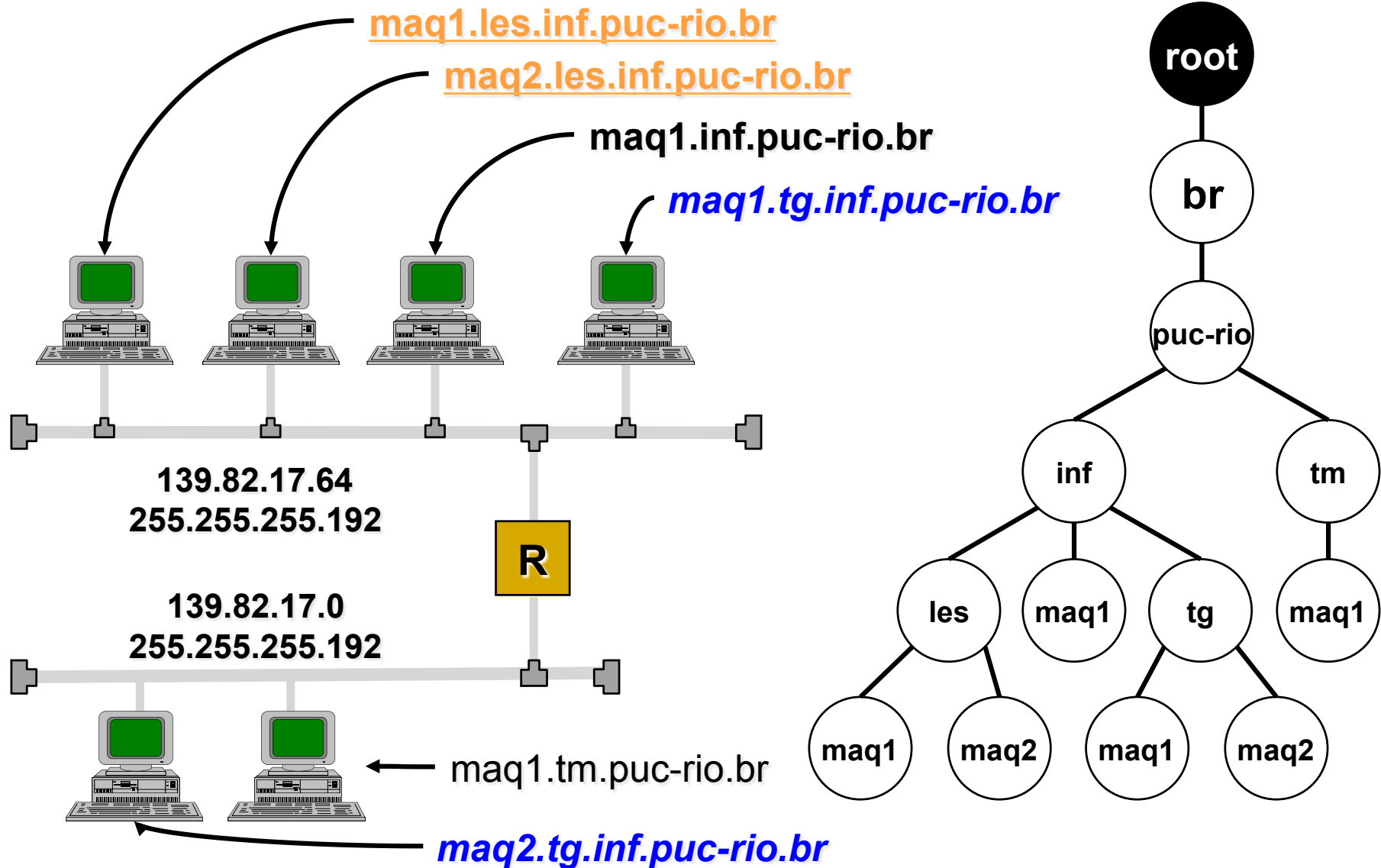
- Base de dados distribuída
- Servidores de nomes
- Clientes (resolvedores)

Organização do DNS

- ➔ **Nomes são completamente independentes da organização de redes físicas e de endereços IP**

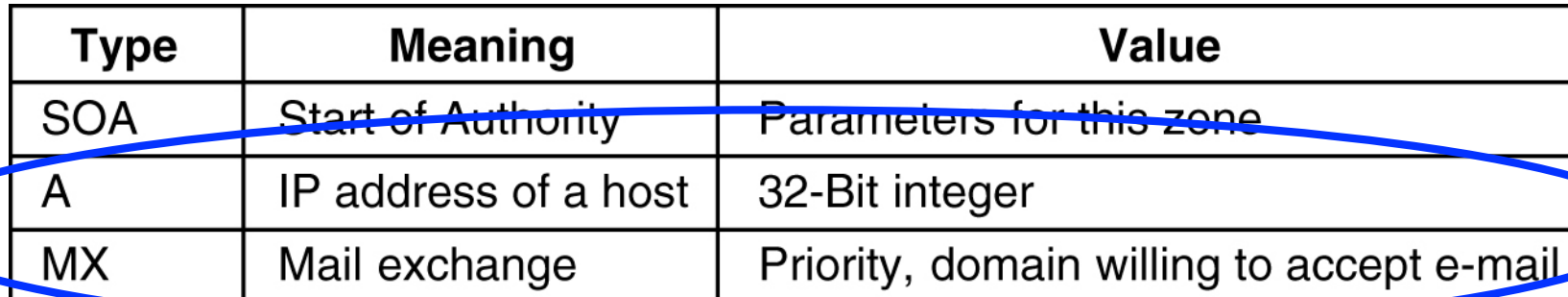
- ➔ **É possível ter:**
 - mais de uma rede em um mesmo domínio
 - Uma rede com máquinas em domínios diferentes
 - mais de um nome para uma mesma máquina (podendo ser nomes de domínios diferentes)

Exemplo de Organização do DNS



Resource Records

Tipos mais importantes: A e MX



Type	Meaning	Value
SOA	Start of Authority	Parameters for this zone
A	IP address of a host	32-Bit integer
MX	Mail exchange	Priority, domain willing to accept e-mail
NS	Name Server	Name of a server for this domain
CNAME	Canonical name	Domain name
PTR	Pointer	Alias for an IP address
HINFO	Host description	CPU and OS in ASCII
TXT	Text	Uninterpreted ASCII text

The principal DNS resource records types.

Resource Records (2)

```
; Authoritative data for cs.vu.nl
cs.vu.nl.      86400  IN  SOA   star boss (952771,7200,7200,2419200,86400)
cs.vu.nl.      86400  IN  TXT   "Divisie Wiskunde en Informatica."
cs.vu.nl.      86400  IN  TXT   "Vrije Universiteit Amsterdam."
cs.vu.nl.      86400  IN  MX    1 zephyr.cs.vu.nl.
cs.vu.nl.      86400  IN  MX    2 top.cs.vu.nl.

flits.cs.vu.nl. 86400  IN  HINFO Sun Unix
flits.cs.vu.nl. 86400  IN  A     130.37.16.112
flits.cs.vu.nl. 86400  IN  A     192.31.231.165
flits.cs.vu.nl. 86400  IN  MX    1 flits.cs.vu.nl.
flits.cs.vu.nl. 86400  IN  MX    2 zephyr.cs.vu.nl.
flits.cs.vu.nl. 86400  IN  MX    3 top.cs.vu.nl.
www.cs.vu.nl.   86400  IN  CNAME star.cs.vu.nl
ftp.cs.vu.nl.   86400  IN  CNAME zephyr.cs.vu.nl

rowboat         IN  A     130.37.56.201
                IN  MX    1 rowboat
                IN  MX    2 zephyr
                IN  HINFO Sun Unix

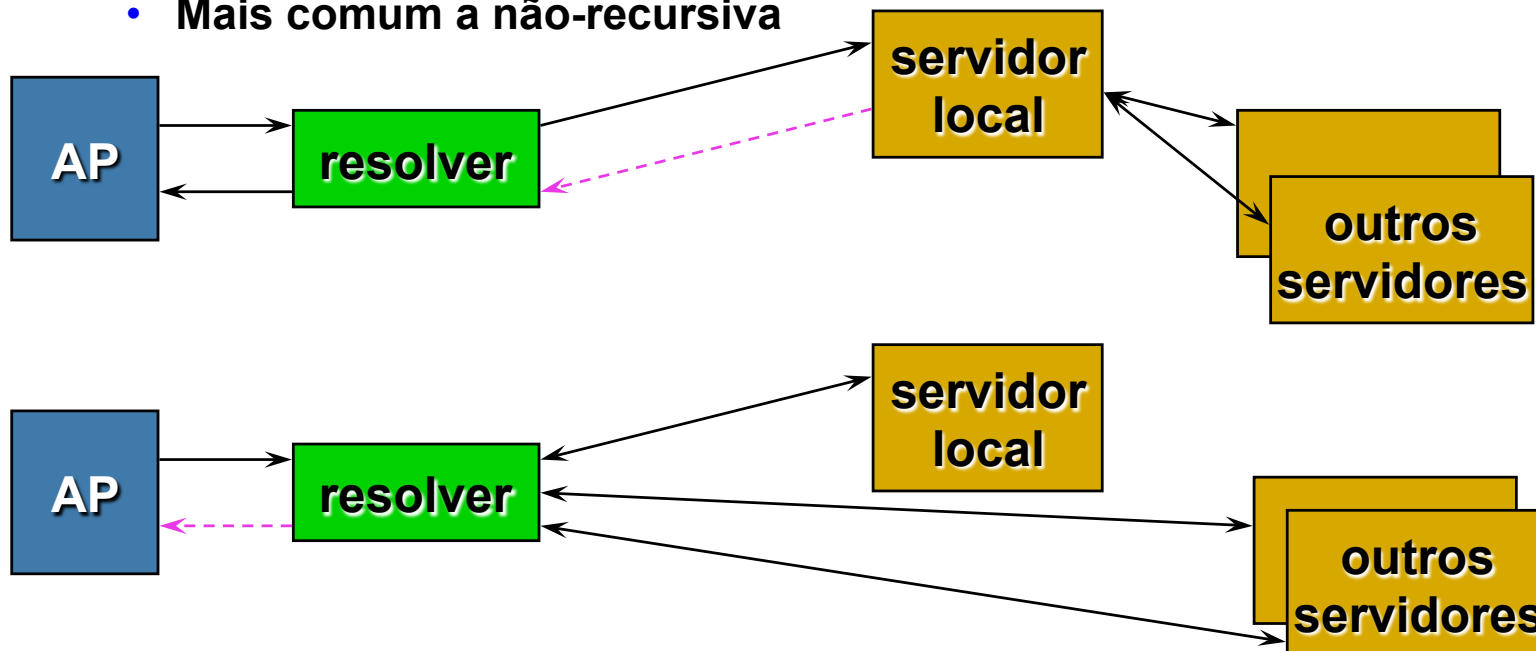
little-sister   IN  A     130.37.62.23
                IN  HINFO Mac MacOS

laserjet        IN  A     192.31.231.216
                IN  HINFO "HP Laserjet III Si" Proprietary
```

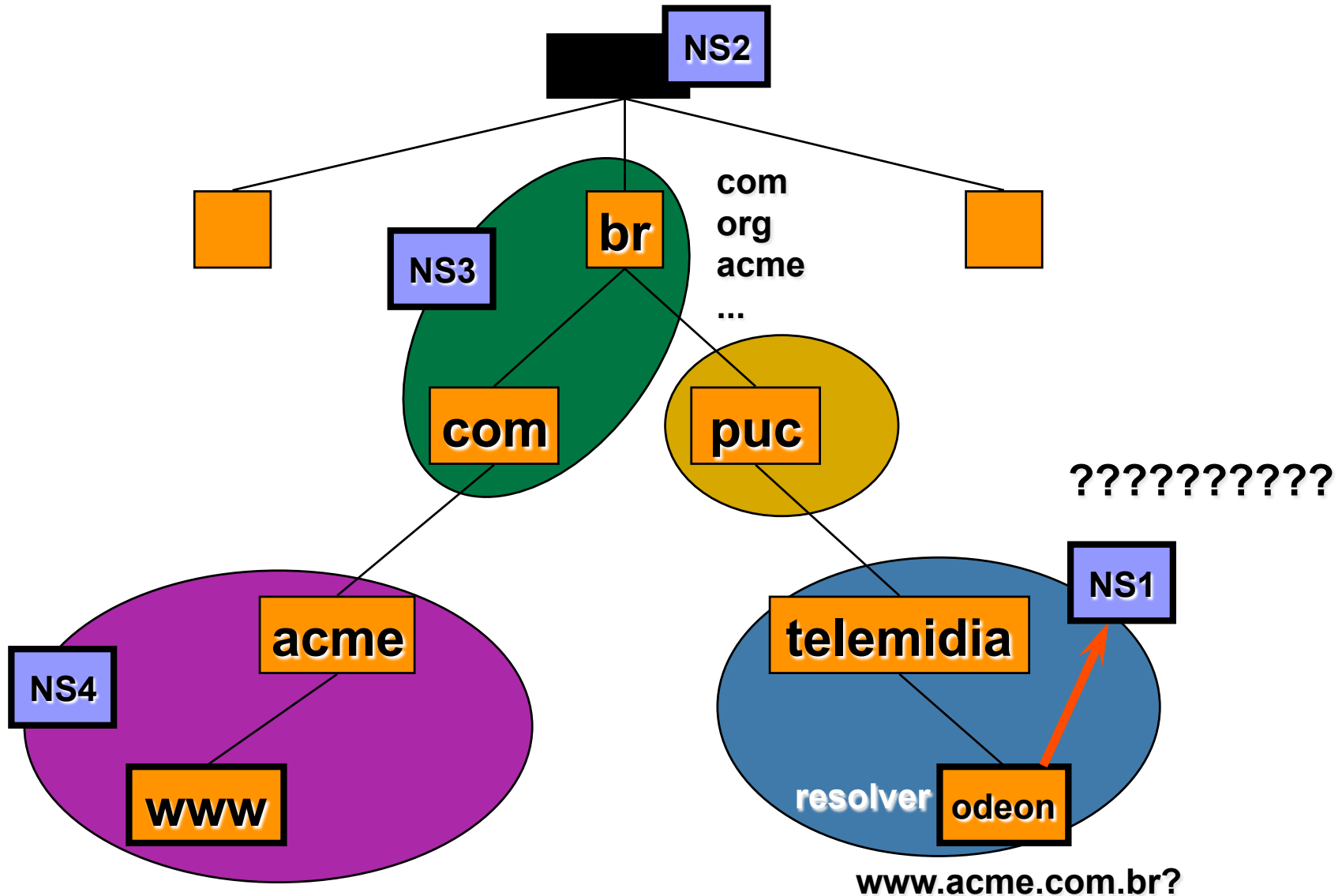
Exemplos de registros de uma base de dados de DNS para *cs.vu.nl*.

Resolução de Nomes

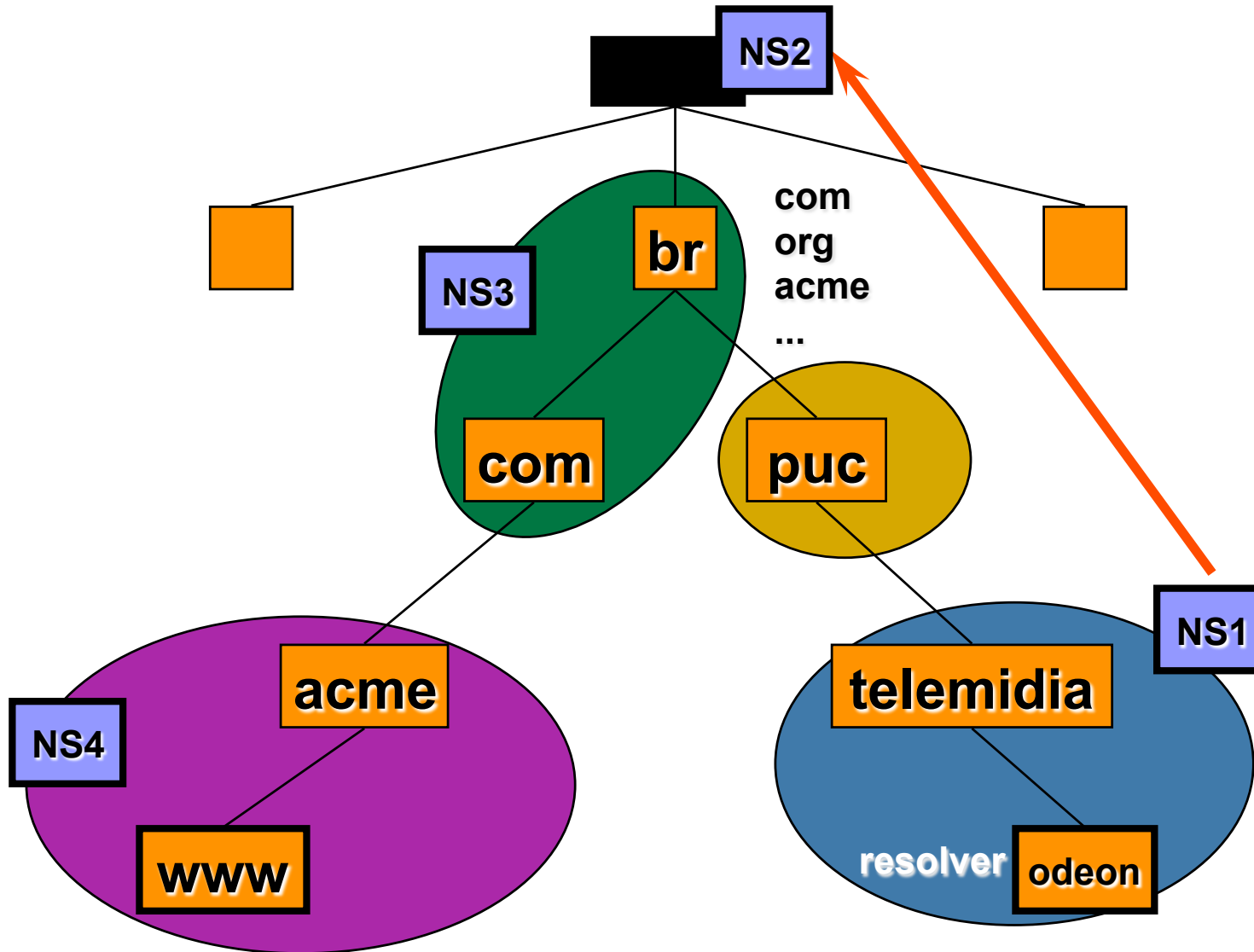
- **Clientes podem fazer dois tipos de consulta:**
 - recursiva: servidor de nomes se encarrega da consulta (mais comum)
 - não-recursiva: clientes realizam a consulta a cada servidor
- **Servidores também podem fazer os dois tipos de consulta**
 - Mais comum a não-recursiva



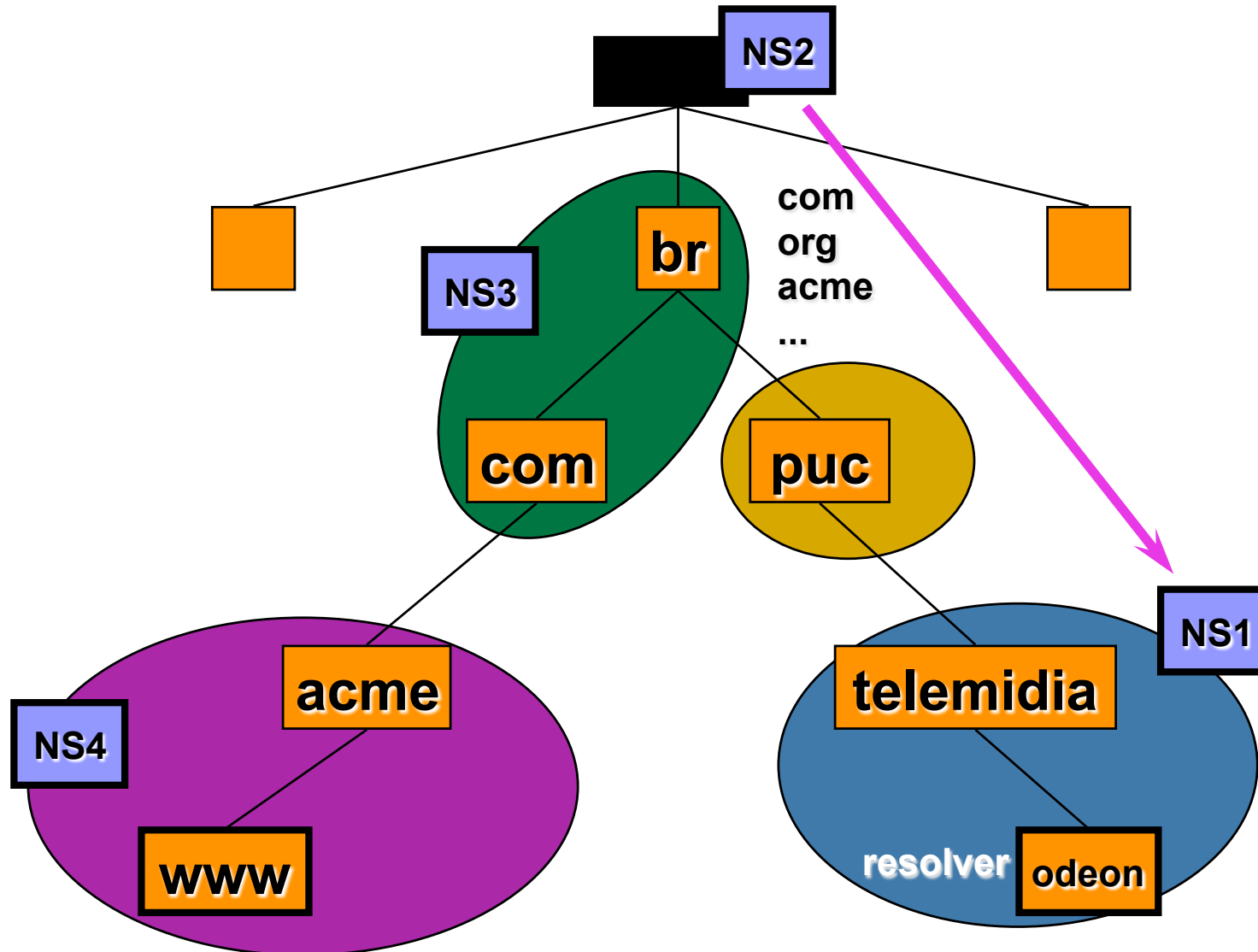
Exemplo de Resolução de Nomes



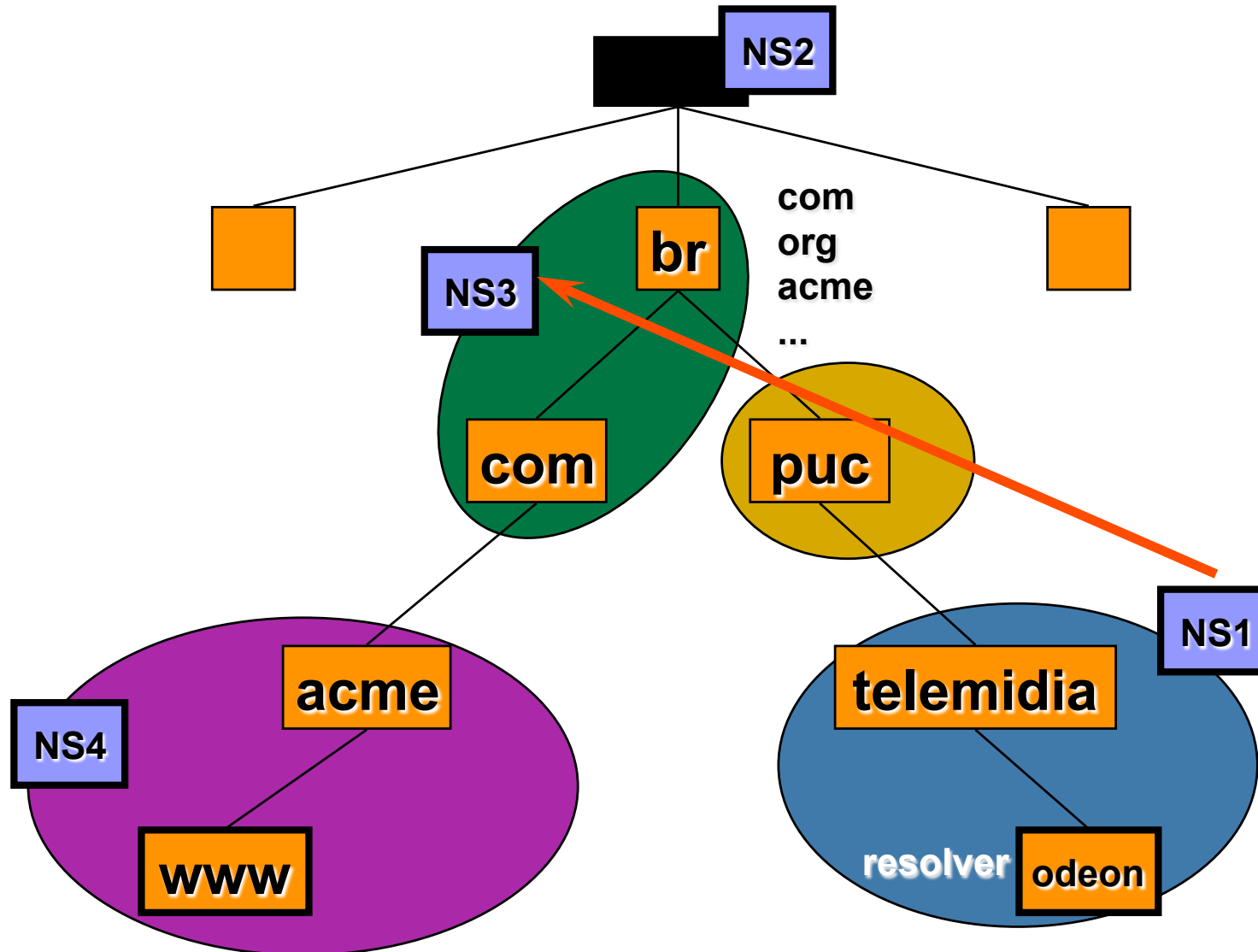
Exemplo de Resolução de Nomes



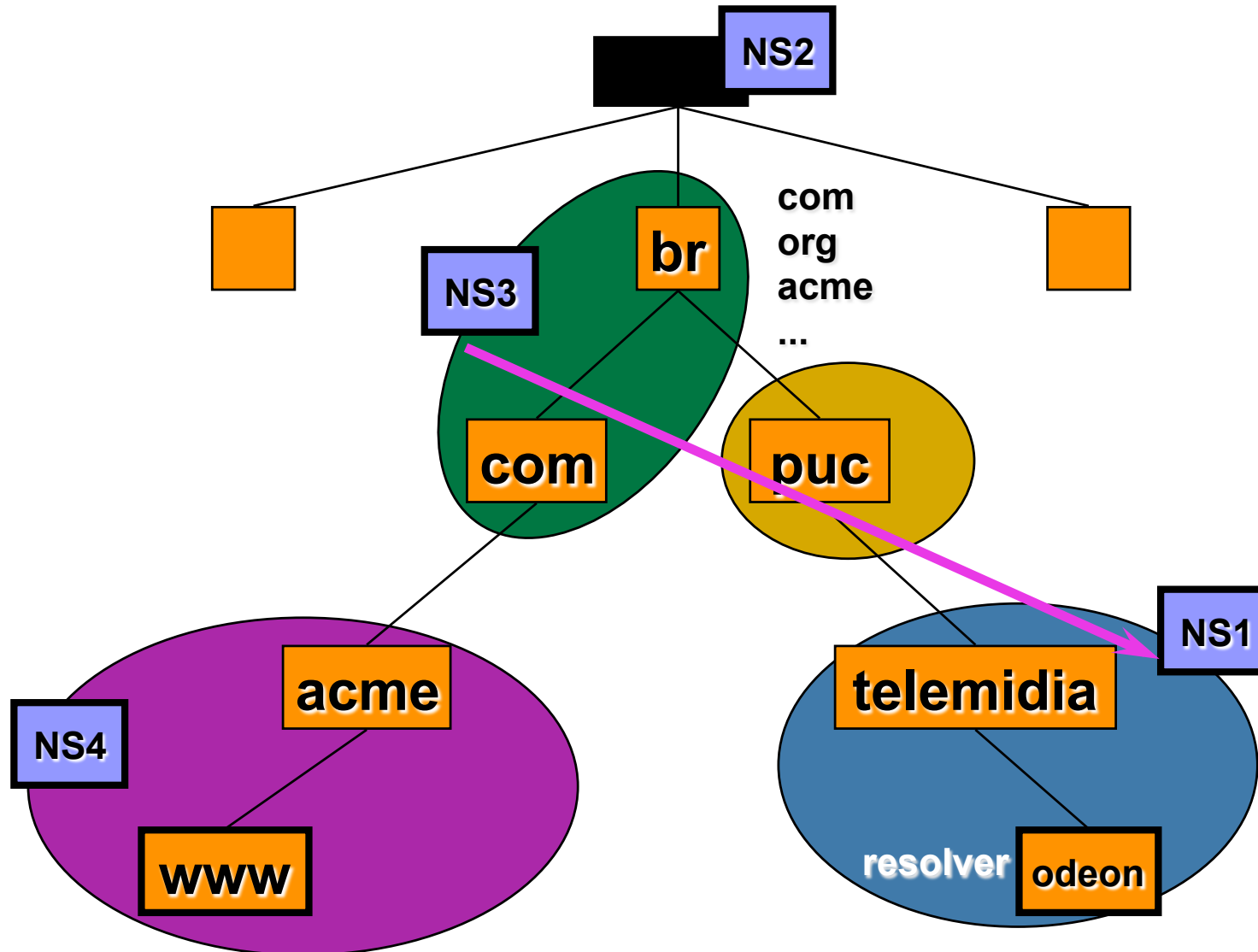
Exemplo de Resolução de Nomes



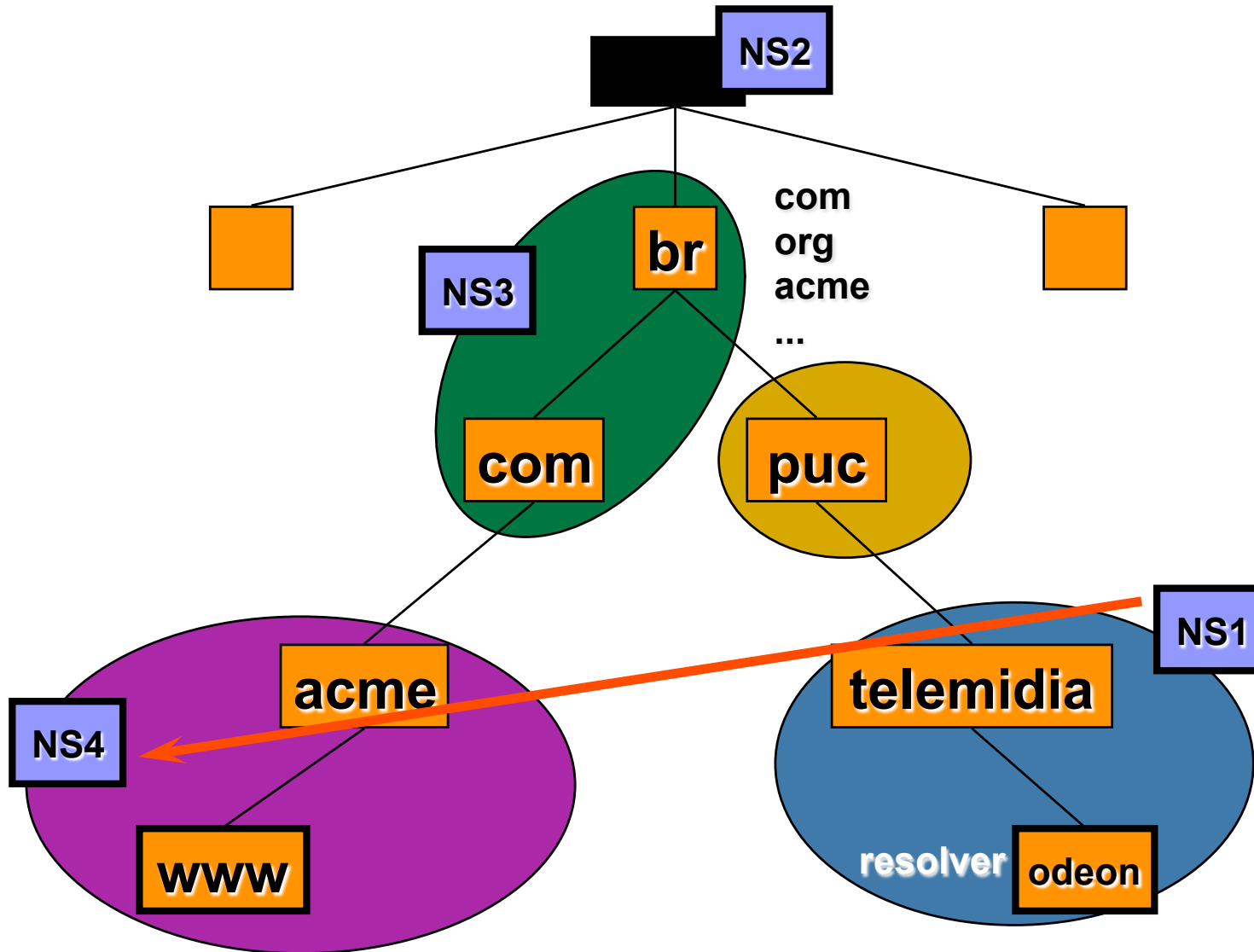
Exemplo de Resolução de Nomes



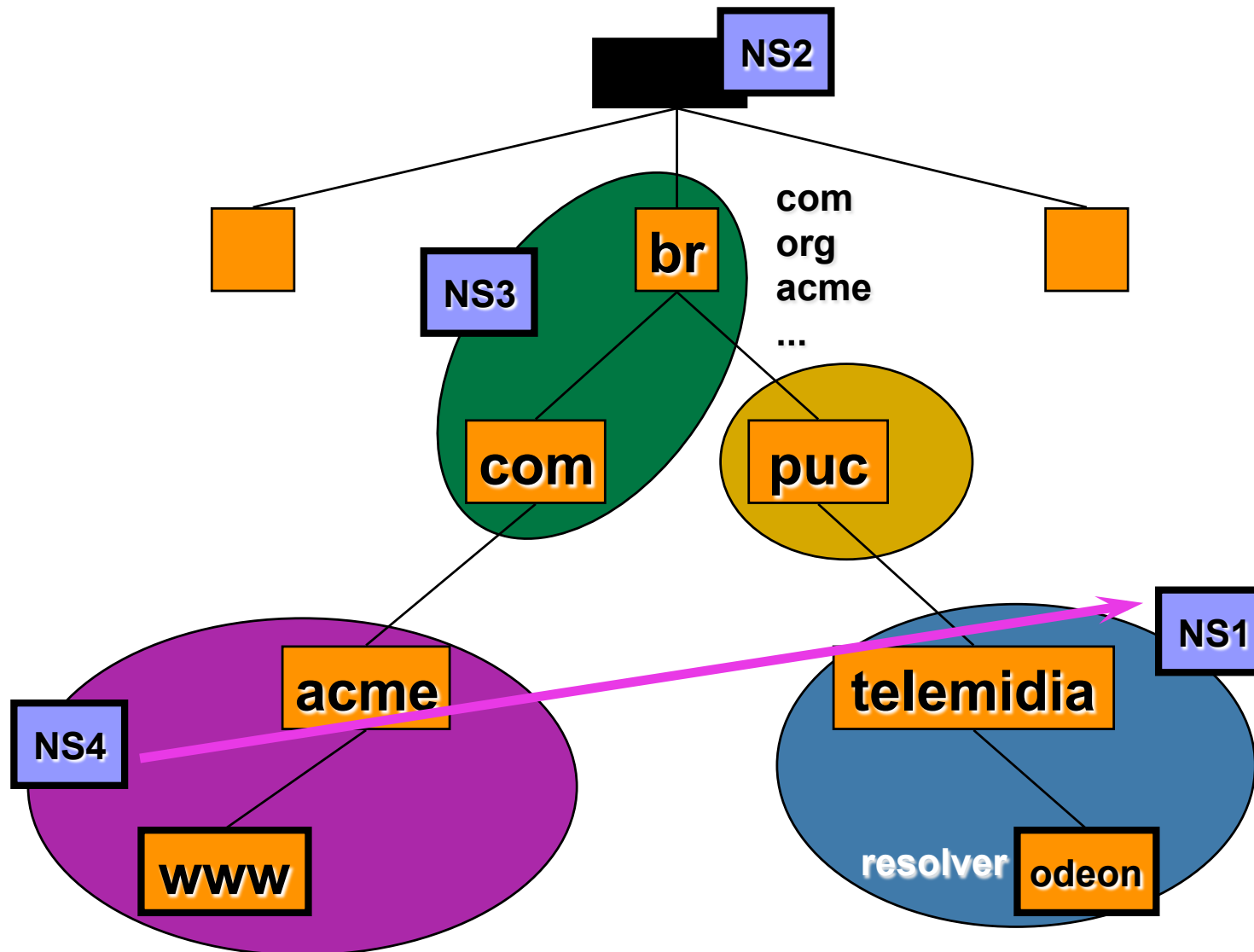
Exemplo de Resolução de Nomes



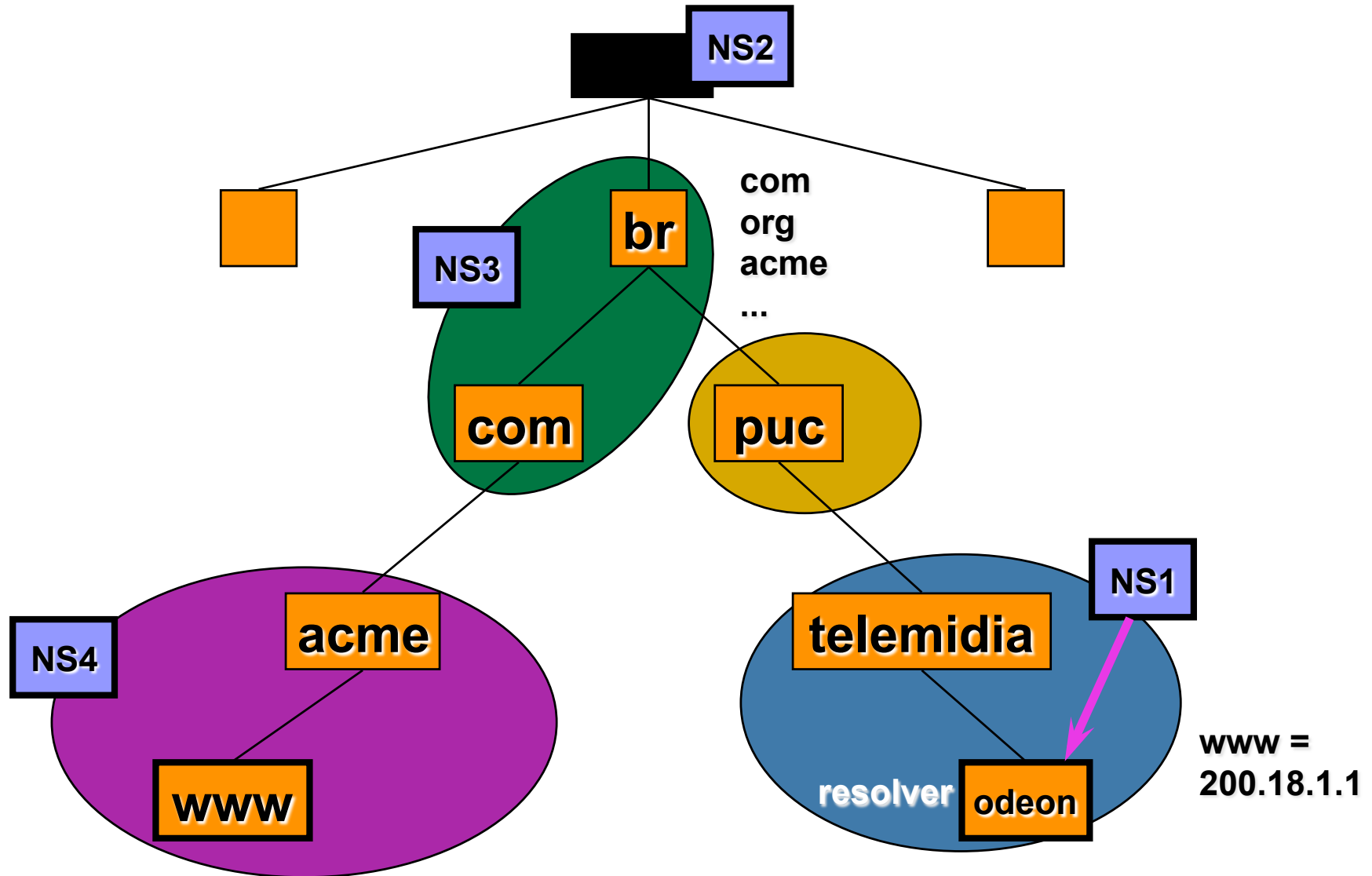
Exemplo de Resolução de Nomes



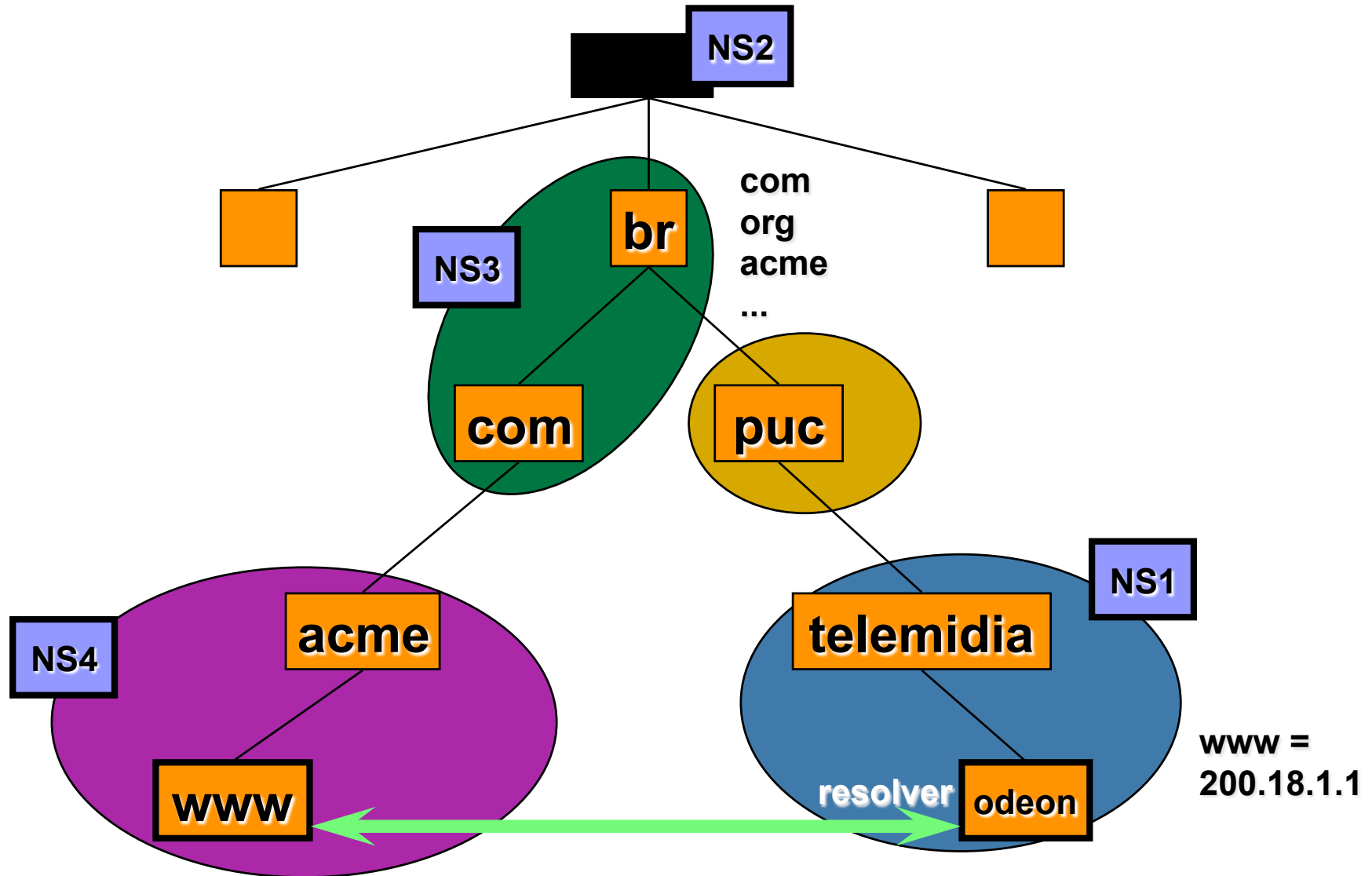
Exemplo de Resolução de Nomes



Exemplo de Resolução de Nomes



Exemplo de Resolução de Nomes



Exemplo de Resolução de Nomes (Resumo)

- ➔ **Cliente requisita mapeamento ao seu resolvedor**
 - Ex. resolução de `www.acme.com.br`, tipo A, a partir da máquina `odeon.telemidia.puc-rio.br`
- ➔ **Resolvedor consulta servidor de DNS do seu domínio**
 - domínio `.telemidia.puc-rio.br` (servidor NS1)
- ➔ **Não possuindo a informação, servidor NS1 consulta o servidor do domínio “ ” - ROOT SERVER - (servidor NS2)**
- ➔ **Servidor NS2 informa ao servidor NS1 o servidor do domínio `.br` (servidor NS3)**
- ➔ **Servidor NS3 também é servidor do domínio `.com.br`, por isso já informa ao servidor NS1 o servidor do domínio `.acme.com.br` (servidor NS4)**
- ➔ **Servidor NS1 consulta servidor NS4 e retorna ao resolvedor o endereço IP de `www.acme.com.br` (200.18.1.1)**

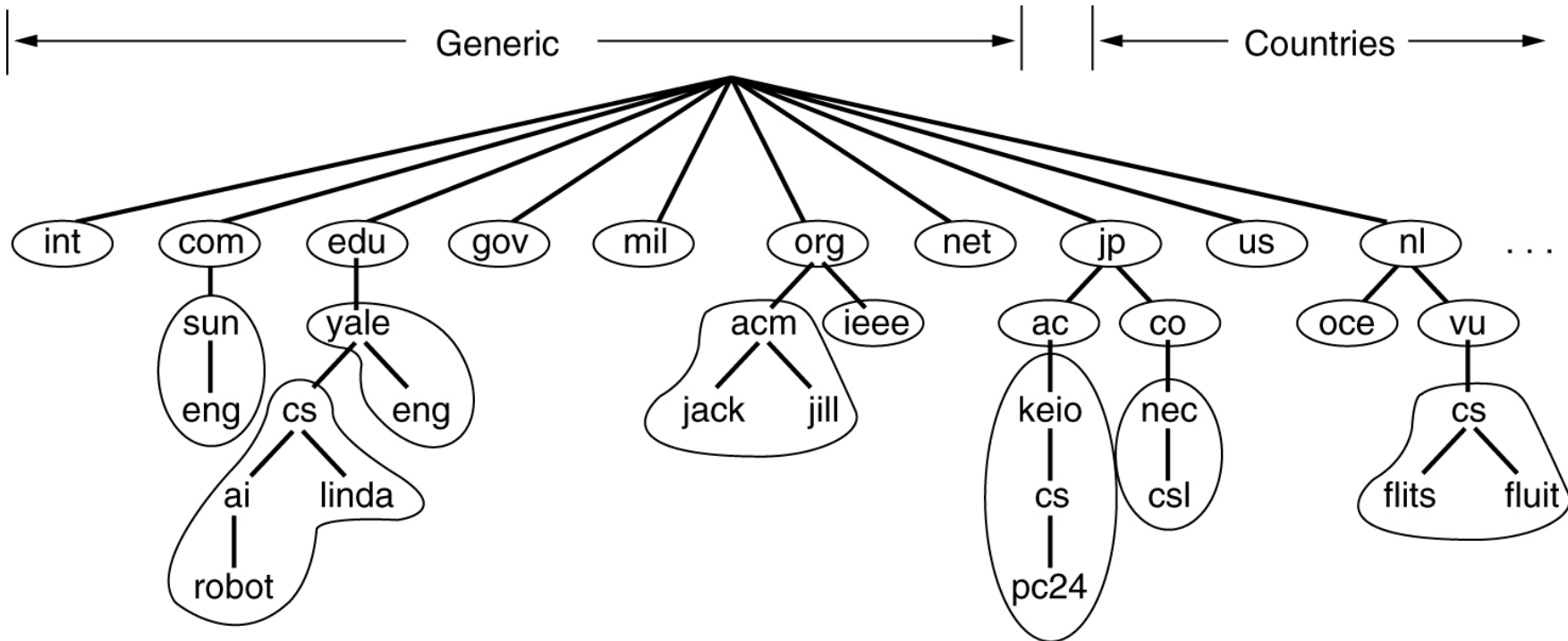
Exemplo de Resolução de Nomes (Uso de Cache)

- ➔ **Servidor do domínio telemidia.puc-rio.br mapeou endereço IP de www.acme.com.br**
- ➔ **Durante o processo, foram armazenados no cache desse servidor informações a respeito dos servidores de .br, .com.br e .acme.com.br, além do end. IP de www.acme.com.br**
- ➔ **Se um resolvedor nesse domínio pedir o end. IP de aloha.acme.com.br, o servidor desse domínio não precisa iniciar a consulta a partir do ROOT SERVER, mas sim a partir do servidor de .acme.com.br**

DNS

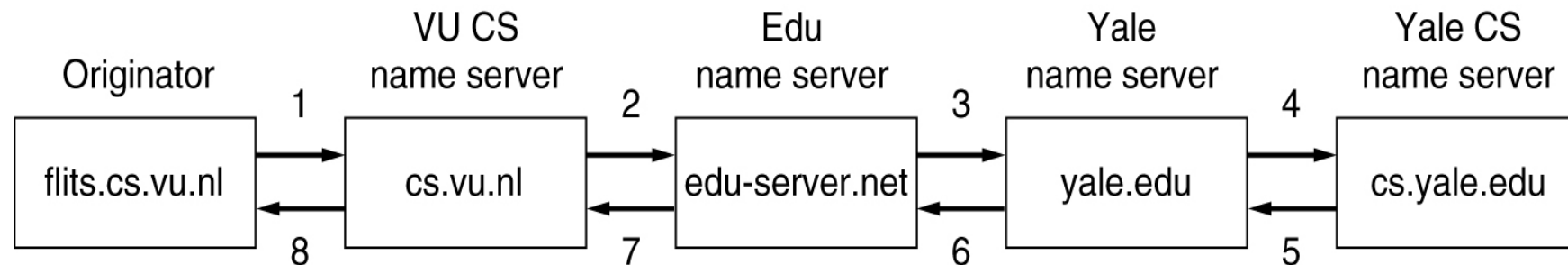
- ➔ **Os servidores que definem uma validade em cache (TTL) de seus registros de recursos em outros servidores**
 - default: 2 dias
- ➔ **A escolha do endereço IP do servidor raiz é feita usando round robin entre os endereços existentes (atualmente são 13 máquinas espalhadas pelo mundo)**

Name Servers



- **Parte do espaço de nomes DNS mostrando a divisão em zonas.**
- **Uma zona representa uma parte da árvore e o conjunto de servidores com informações sobre a zona**

Name Servers (2)

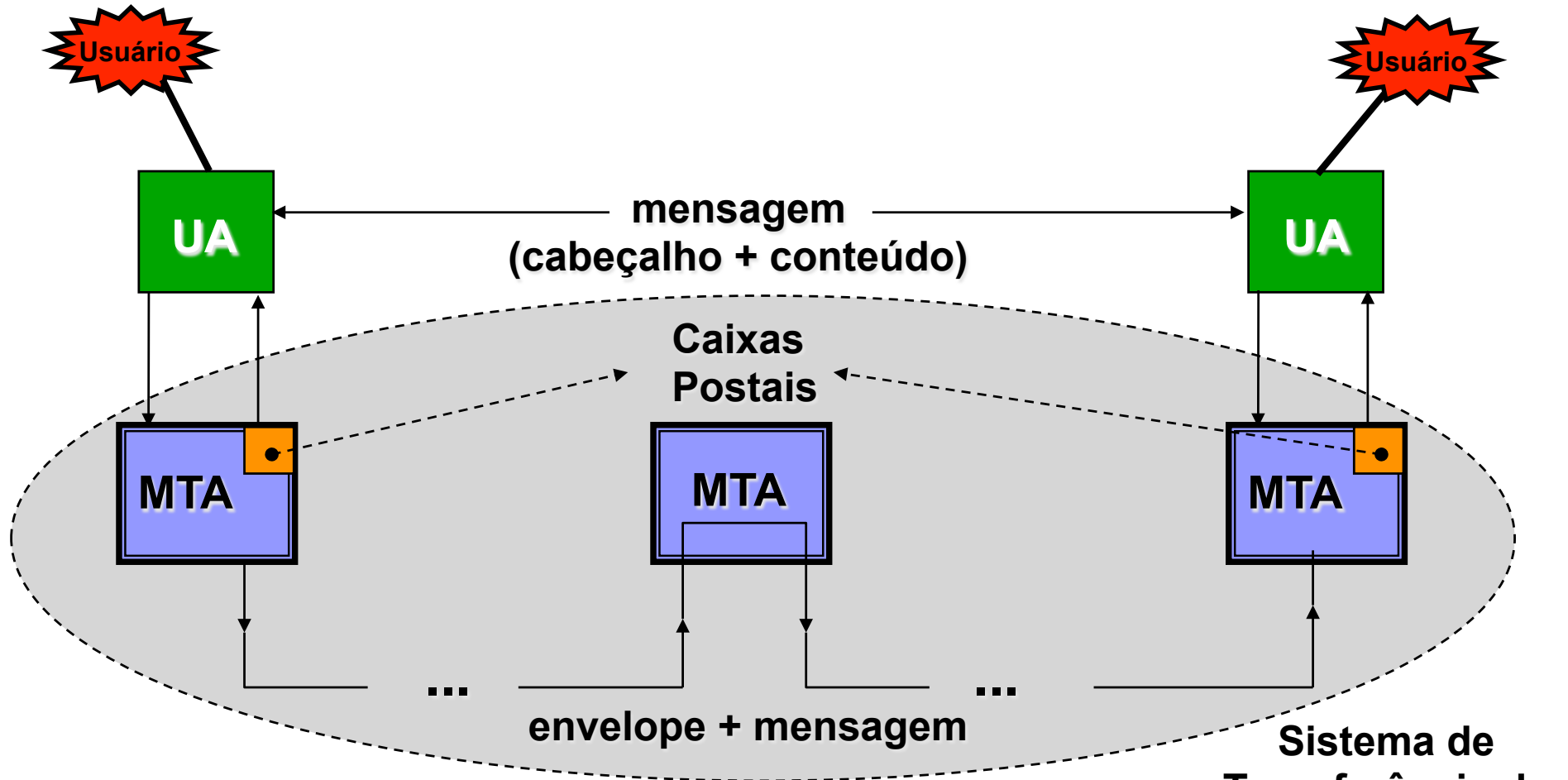


How a resolver looks up a remote name in eight steps (`linda.cs.yale.edu??`).

Correio Eletrônico

Arquitetura dos Sistemas de Correio Eletrônico

Redes de Computadores



Componente MTA (Message Transfer Agent)

Componente UA (User Agent)

Sistema de Transferência de Mensagens

Componentes de um Sistema de Correio Eletrônico

Redes de Computadores

➔ UA - User Agent:

- permite composição, envio e recebimento de mensagens
- permite manipular caixas postais

➔ MTA - Message Transfer Agent:

- encaminha mensagem até o destinatário com base nas informações do envelope usando o protocolo SMTP

Identificação do Destinatário

➔ **Destino da mensagem é identificado por:**

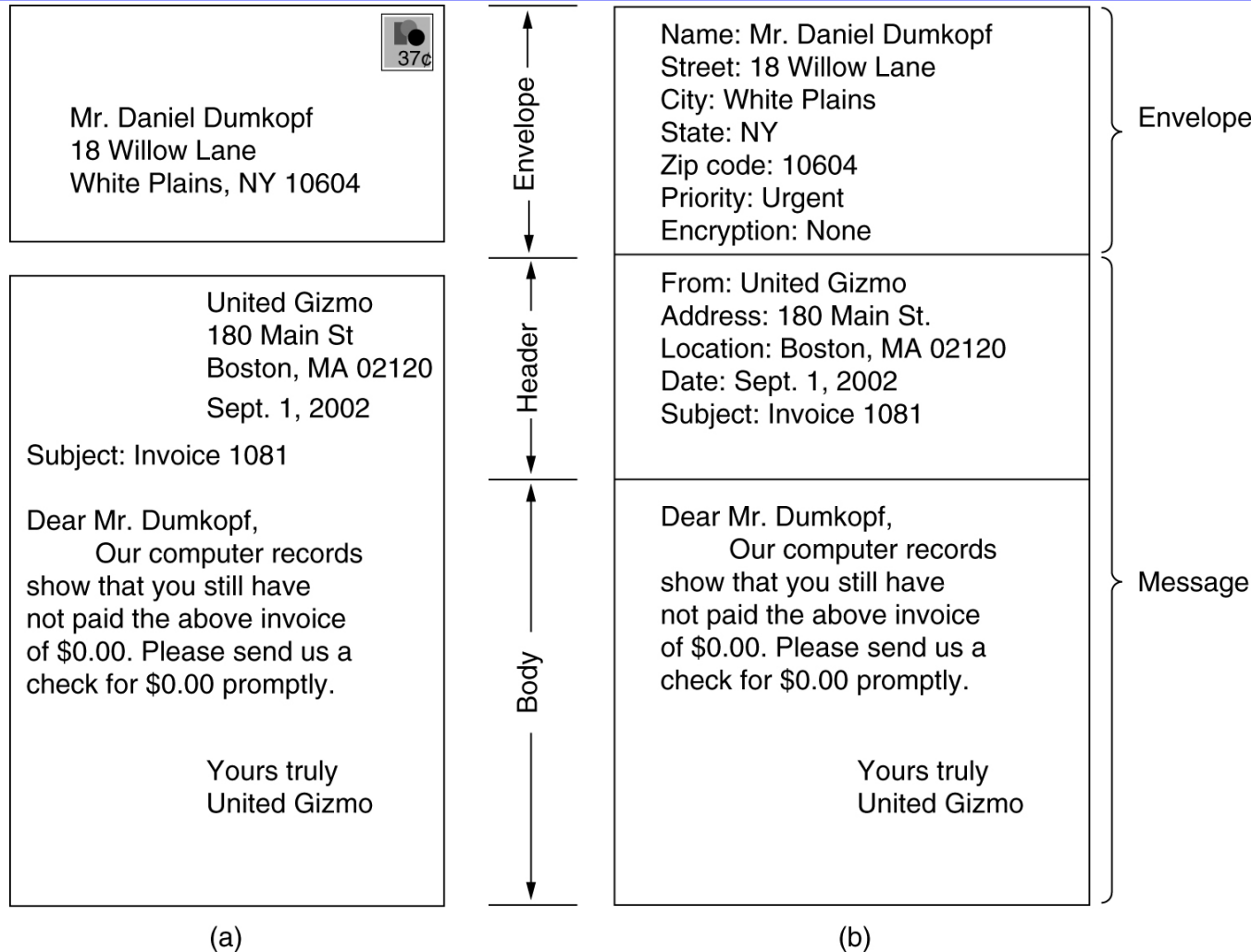
- o nome da máquina de destino (*domain-name*) e
- a caixa postal do usuário de destino (*local-part*)

local-part@domain-name

- Exemplo: joao@acme.com.br

➔ **Existe um registro especial no servidor DNS (MX) para possibilitar a resolução do nome da máquina de destino (servidor de mails) em endereço IP**

The User Agent



Envelopes and messages.

(a) Paper mail. (b) Electronic mail.

Formato de Mensagem: RFC 822

Redes de Computadores

➔ Cabeçalho seguido de linha em branco e corpo da msg

To:	Destinatários principais
Cc:	Destinatários em cópia
Bcc:	Destinatários em cópia invisível
From:	Criador da mensagem
Received:	Adicionado por MTAs no caminho
Date:	Data e hora do envio da mensagem
Reply-To:	E-mail para enviar respostas
References:	Outros identificadores relevantes
Subject:	Descrição do assunto da mensagem

➔ Exemplo de msg

```
Date: Wed, 29 Nov 1999 18:49:35 -200 (EDT)
From: Joao da Silva <joao@acme.com.br>
To: Antonio Tadeu Azevedo Gomes <atagomes@telemidia.puc-rio.br>
Cc: l-tcpip@engenhocom.br
Subject: Teste de e-mail
```

Testando e-mail

Formato de Mensagem: RFC 822

Redes de Computadores

➔ Envelope: linhas inseridas no cabeçalho pelos MTAs

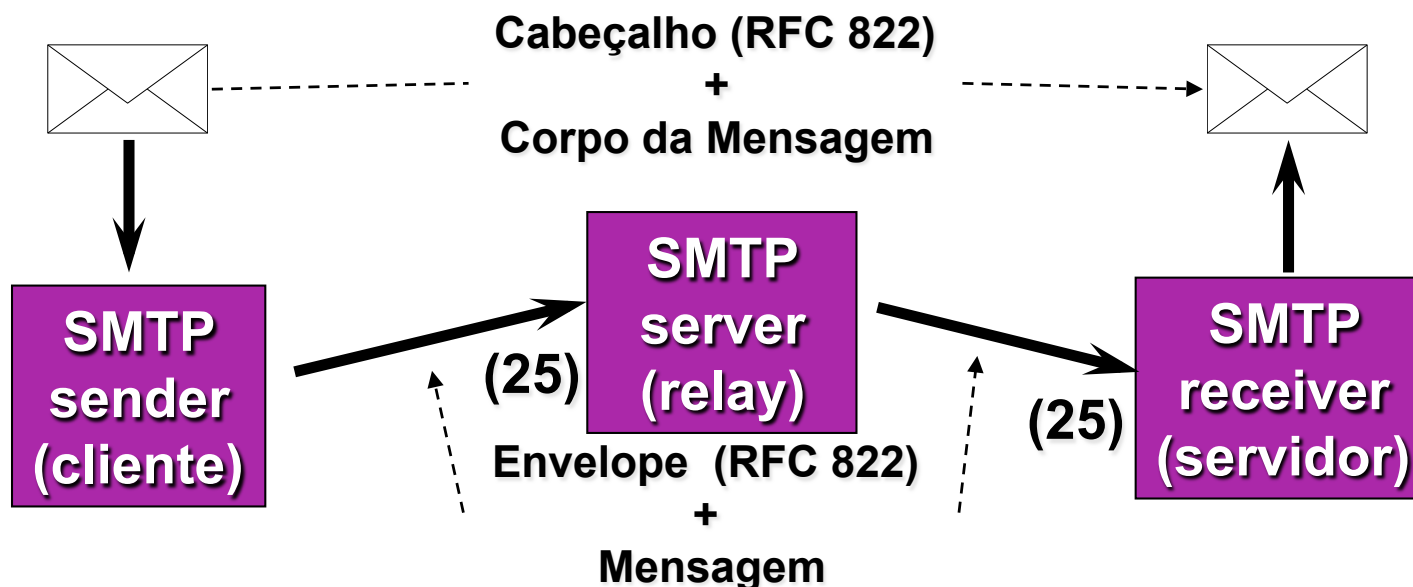
- Exemplo de msg “envelopada”:

```
Received: from acme.com.br (mail.acme.com.br [200.250.30.3])
  by telemidia.puc-rio.br (8.8.8+Sun/8.8.8)
  with ESMTP id TAA16860
  for <atagomes@telemidia.puc-rio.br>;
  Wed, 29 Nov 1999 19:32:57 -200 (EDT)
Received: from modem1.acme.com.br [200.250.28.1]
  by acme.com.br with SMTP (Eudora Internet Mail Server 1.2)
  id LAA04471; Wed, 29 Nov 1999 18:50:01 - 200 (EDT)
Date: Wed, 29 Nov 1999 18:49:35 -200 (EDT)
From: Joao da Silva <joao@acme.com.br>
To: Antonio Tadeu Azevedo Gomes <atagomes@telemidia.puc-rio.br>
Cc: l-tcpip@engenhocom.br
Subject: Teste de e-mail
```

Testando e-mail

Protocolo SMTP

- ➔ Protocolo textual para comunicação entre MTAs
 - Utiliza padrão ASCII de 7 bits
- ➔ servidor SMTP usa porta TCP 25



Sistemas de correio que usam entrega fim-a-fim garantem que a mensagem permanece armazenada na máquina transmissora até que ela seja copiada com sucesso para a máquina receptora.

Principais Comandos SMTP

Redes de Computadores

- **HELO <domain>** - identifica o cliente SMTP
- **MAIL FROM: <remetente@domínio>** - Identifica o originador
- **RCPT TO: <destino@domínio>** - Identifica o destinatário
- **DATA** - Início dos dados, terminados por uma linha com um “.”
- **VRFY <nome>** - Confirma a existência do usuário <nome>
- **TURN** - Reverte papéis e permite ao destino enviar mensagens
- **QUIT** - Termina a conexão SMTP

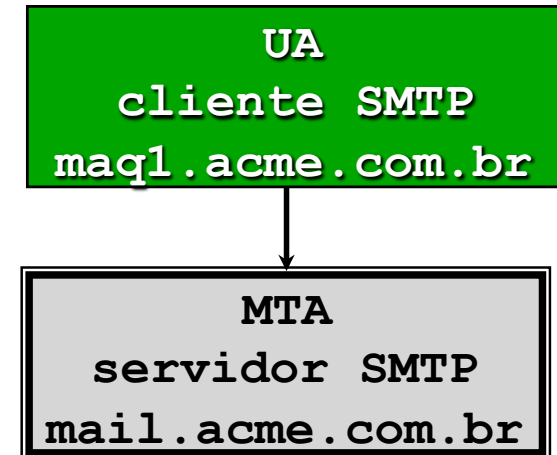
Principais Respostas SMTP

Redes de Computadores

- ➔ **220 - Serviço pronto**
- ➔ **221 - Encerrando conexão**
- ➔ **250 - Ação completada corretamente**
- ➔ **354 - Inicie entrada da mensagem**
- ➔ **500, 501 - Erro de sintaxe**
- ➔ **502 - Comando não implementado**

Protocolo SMTP

... (abertura de conexão TCP com servidor mail.acme.com.br)
220 mail.acme.com.br Eudora Internet Mail Server 1.2 ready
>>> HELO maq1.acme.com.br
250 mail.acme.com.br Hello maq1.acme.com.br, pleased to meet you
>>> MAIL From: joao@acme.com.br
250 <joao@acme.com.br> Sender OK
>>> RCPT To: atagomes@telemidia.puc-rio.br
250 <atagomes@telemidia.puc-rio.br> Recipient OK
>>> RCPT To: l-tcpip@engenho.com.br
250 <l-tcpip@engenho.com.br> Recipient OK
>>> DATA
354 Enter mail, end with "." on a line by itself
>>> [cabeçalho + conteúdo]
>>> .
250 Mail accepted
>>> QUIT
221 mail.acme.com.br closing connection
... (encerramento de conexão TCP)



```
From: joao@acme.com.br
To: atagomes@telemidia.puc-rio.br
Cc: l-tcpip@engenho.com.br
Subject: Teste de e-mail
Testando e-mail
```

Protocolo SMTP

... (abertura de conexão TCP com
servidor *telemidia.puc-rio.br*)
220 *telemidia.puc-rio.br* Sendmail 8.8.5 ready
>>> *HELO mail.acme.com.br*
250 *telemidia.puc-rio.br* Hello *mail.acme.com.br*, pleased to meet you
>>> *MAIL From: joao@acme.com.br*
250 <*joao@acme.com.br*> Sender OK
>>> *RCPT To: atagomes@telemidia.puc-rio.br*
250 <*atagomes@telemidia.puc-rio.br*> Recipient OK
>>> *DATA*
354 Enter mail, end with "." on a line by itself
>>> [*cabeçalho + conteúdo*]
>>> .
250 Mail accepted
>>> *QUIT*
221 *telemidia.puc-rio.br* closing connection
... (encerramento de conexão TCP)

MTA
cliente SMTP
mail.acme.com.br

MTA
servidor SMTP
telemidia.puc-rio.br

From: *joao@acme.com.br*
To: *atagomes@telemidia.puc-rio.br*
Cc: *l-tcpip@engenho.com.br*
Subject: *Teste de e-mail*
Testando e-mail

Características do SMTP

- ➔ **RFC 821 define que SMTP só suporta caracteres ASCII de 7 bits**
 - Não suporta caracteres acentuados, arquivos binários, ...

Como transportar outros formatos?

MIME

(Multi-Purpose Internet Mail Extensions) - RFC 1521

Redes de Computadores

➔ MIME estende SMTP especificando:

- Novos campos para o cabeçalho RFC 822
- Novos formatos de conteúdo
- Técnicas de codificação desses formatos em ASCII 7 bits

➔ Principais campos MIME

MIME-Version:	Identifica a versão MIME
Content-Description:	Descreve o conteúdo da mensagem
Content-Id:	Identificador único
Content-Transfer-Encoding:	Como o corpo é codificado para TX
Content-Type:	Tipo da mensagem

➔ Funcionamento básico:

- UA remetente acrescenta campos MIME ao cabeçalho da msg, identificando mídia e codificação usada
- UA destino usa campos para recuperar conteúdo (decodificação)
- SMTP não toma conhecimento do uso de MIME

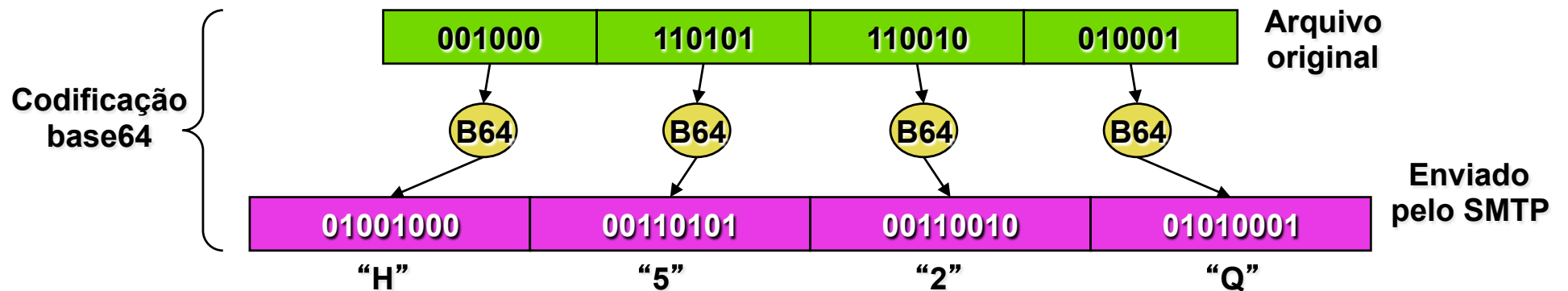
MIME

➔ Alguns tipos de mensagem:

- text/plain: texto sem formatação (ASCII ou ISO 8859)
- image/gif, image/jpeg, video/mpeg
- multipart/mixed: permite envio de submensagens
- application/octet-stream: arquivos genéricos

➔ Algumas codificações:

- Texto: 7-bit (ASCII), 8-bit, quoted-printable
- Binário: base64, user-defined



Submensagens MIME

```
Date: Wed, 29 Nov 1999 18:49:35 -200 (EDT)
From: Joao da Silva <joao@acme.com.br>
To: Antonio Tadeu Azevedo Gomes <atagomes@telemidia.puc-rio.br>
MIME-Version: 1.0
Content-Type: multipart/mixed; Boundary=0011separator
Subject: Teste de e-mail
```

```
--0011separator
Content-Type: text/plain; charset=iso-8859-1
Content-Transfer-Encoding: quoted-printable
```

```
S=F3 testando e-mail com acentua=E7=A5o.
```

} **Mensagem**
“Só testando e-mail
com acentuação.”
codificado em quoted-printable

```
--0011separator
Content-Type: application/octet-stream; name="relat.doc"
Content-Transfer-Encoding: base64
```

```
OM8R4KGxGuEAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAPgADAP7
```

...

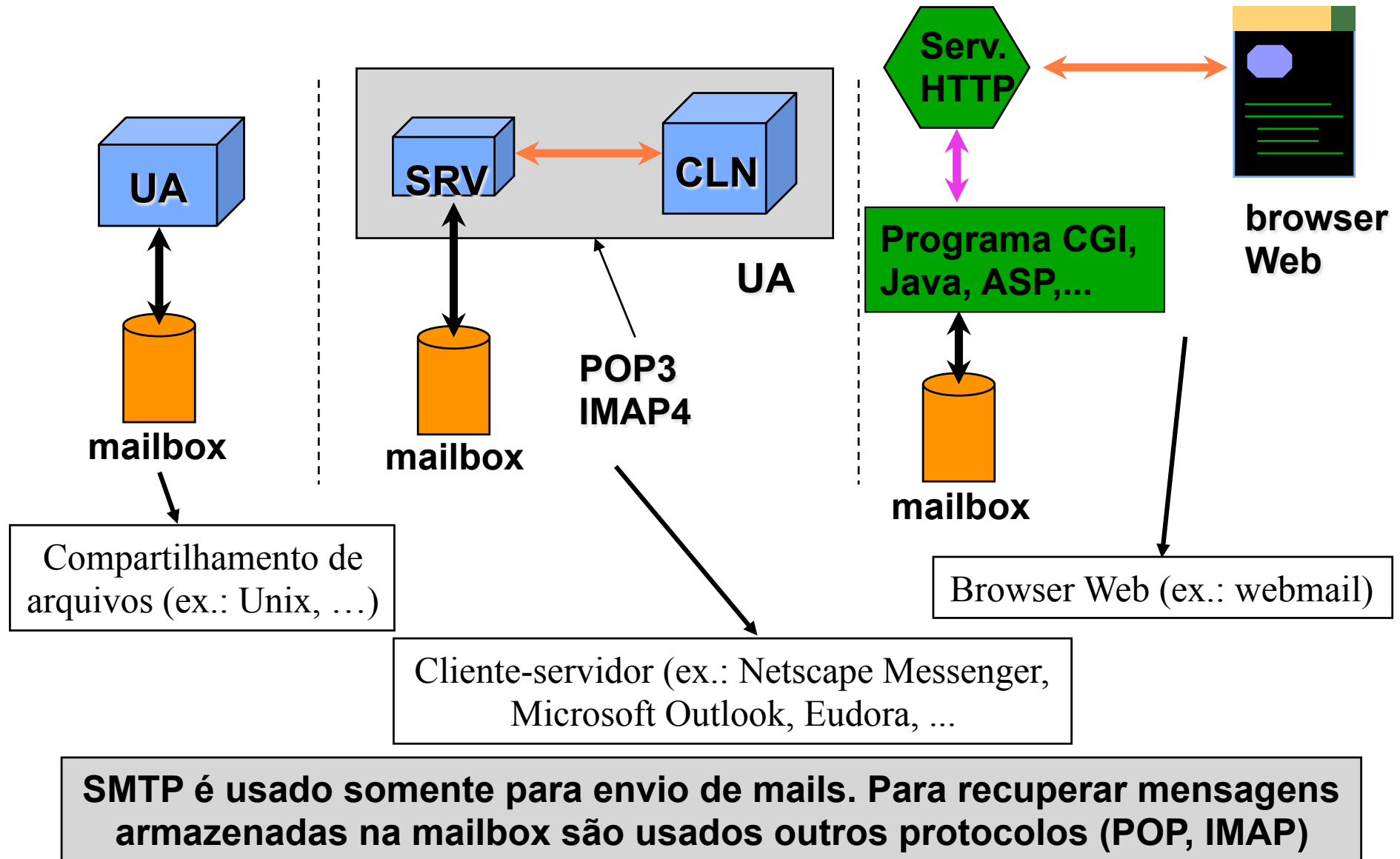
```
//////////AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

```
--0011separator
```

} **Arquivo**
“relat.doc”
codificado em base 64

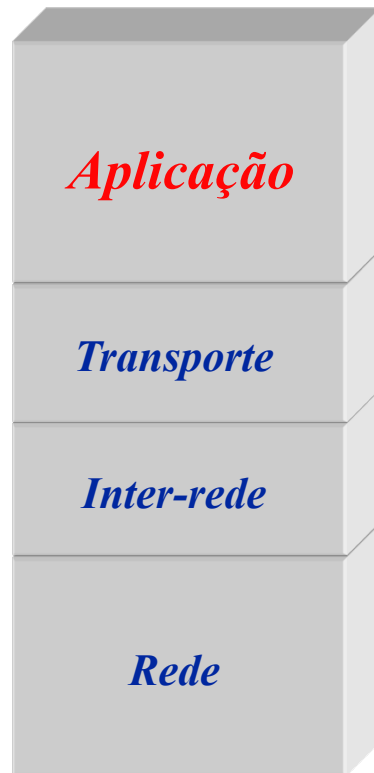
Arquitetura do UA em TCP/IP

Redes de Computadores



WWW
World Wide Web

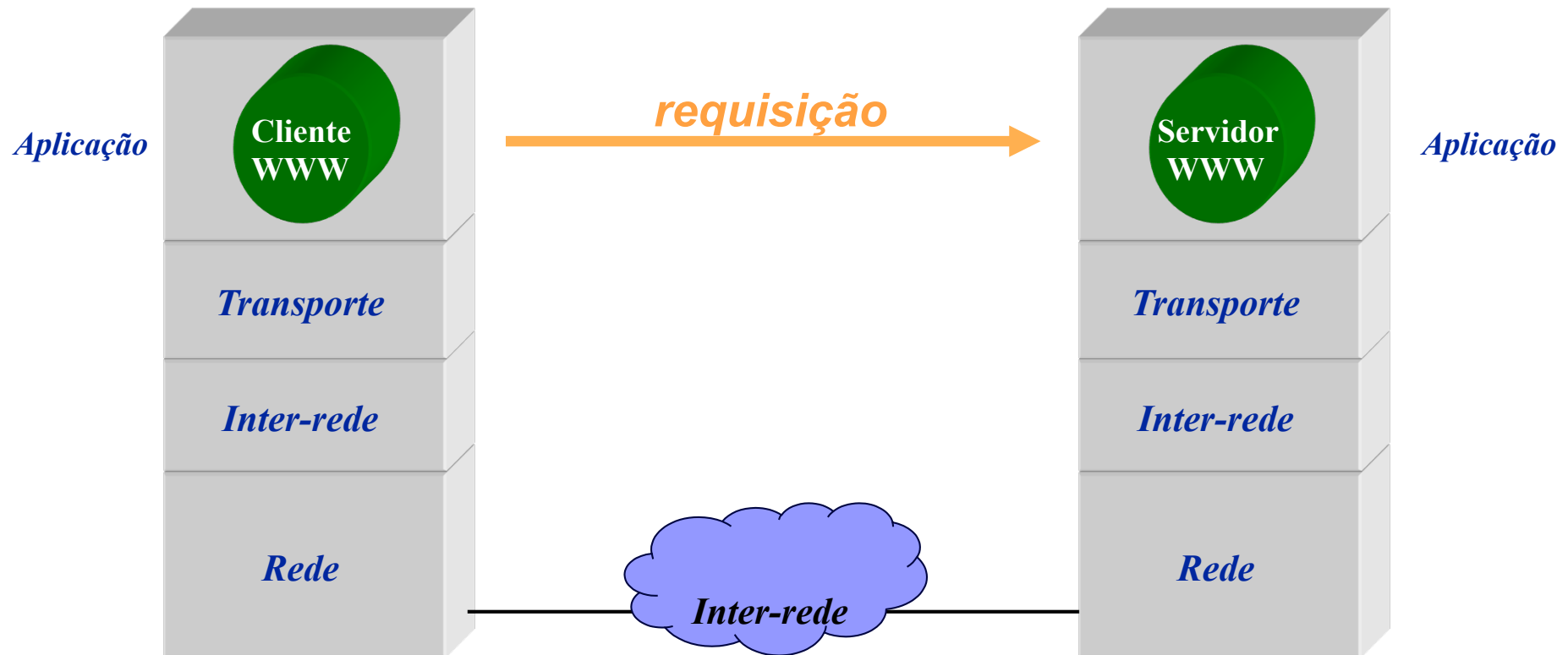
WWW



- ➔ exemplo de aplicação TCP/IP
- ➔ utiliza o serviço confiável da camada de transporte (TCP)
- ➔ baseada no paradigma cliente/servidor

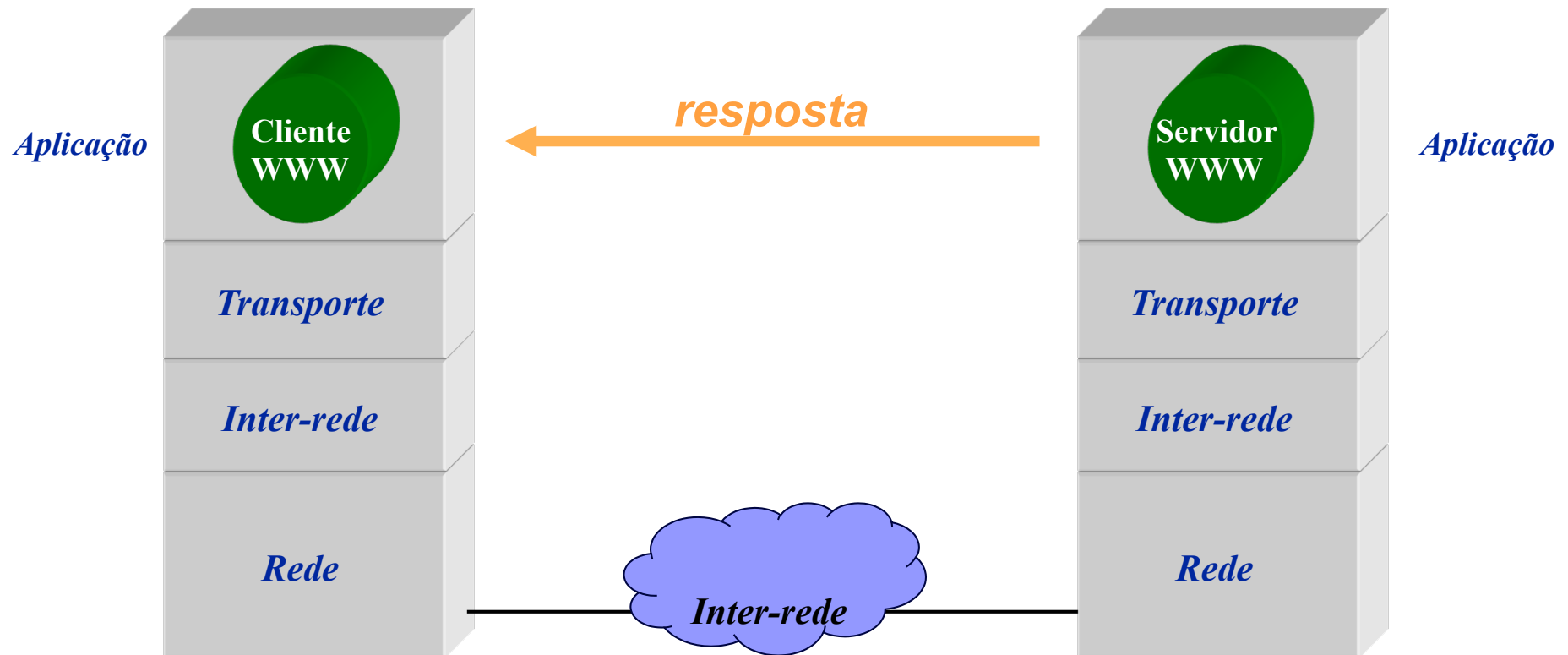
Arquitetura WWW

Redes de Computadores



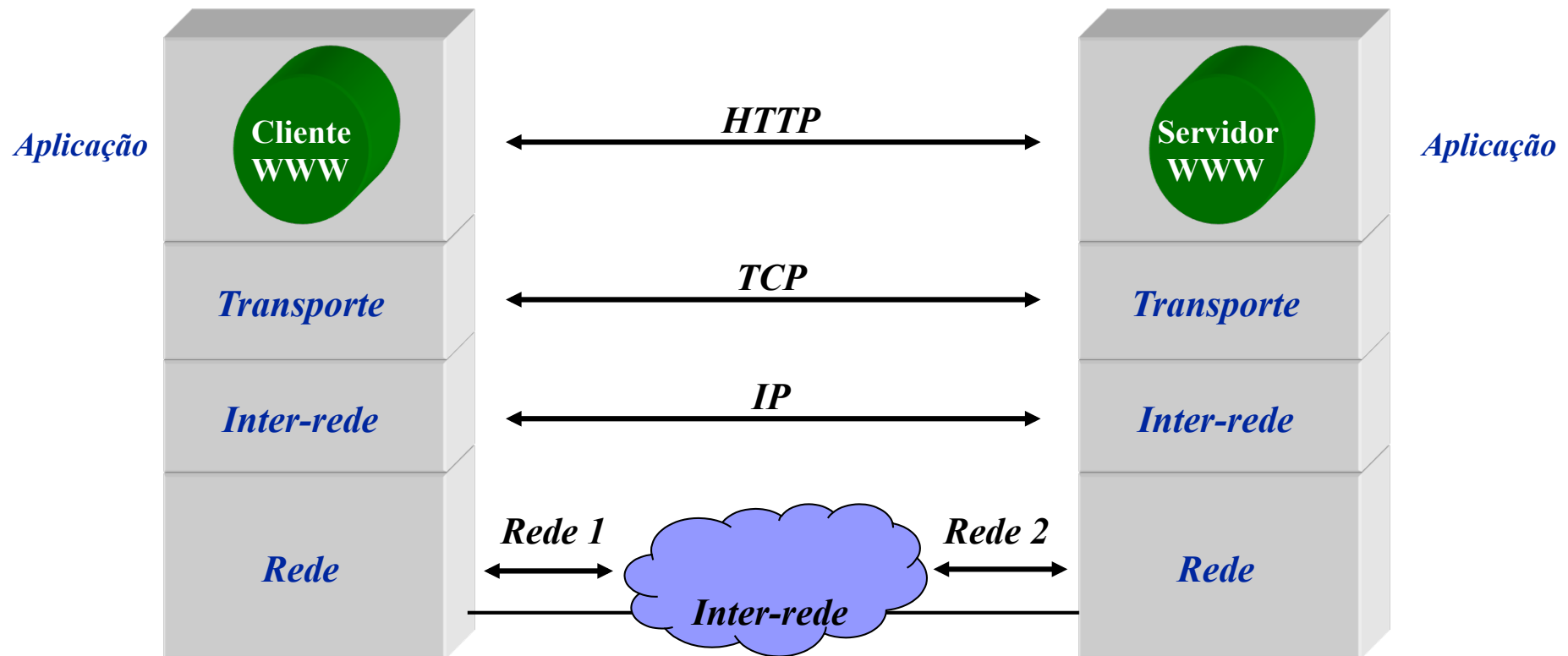
Arquitetura WWW

Redes de Computadores



Arquitetura WWW

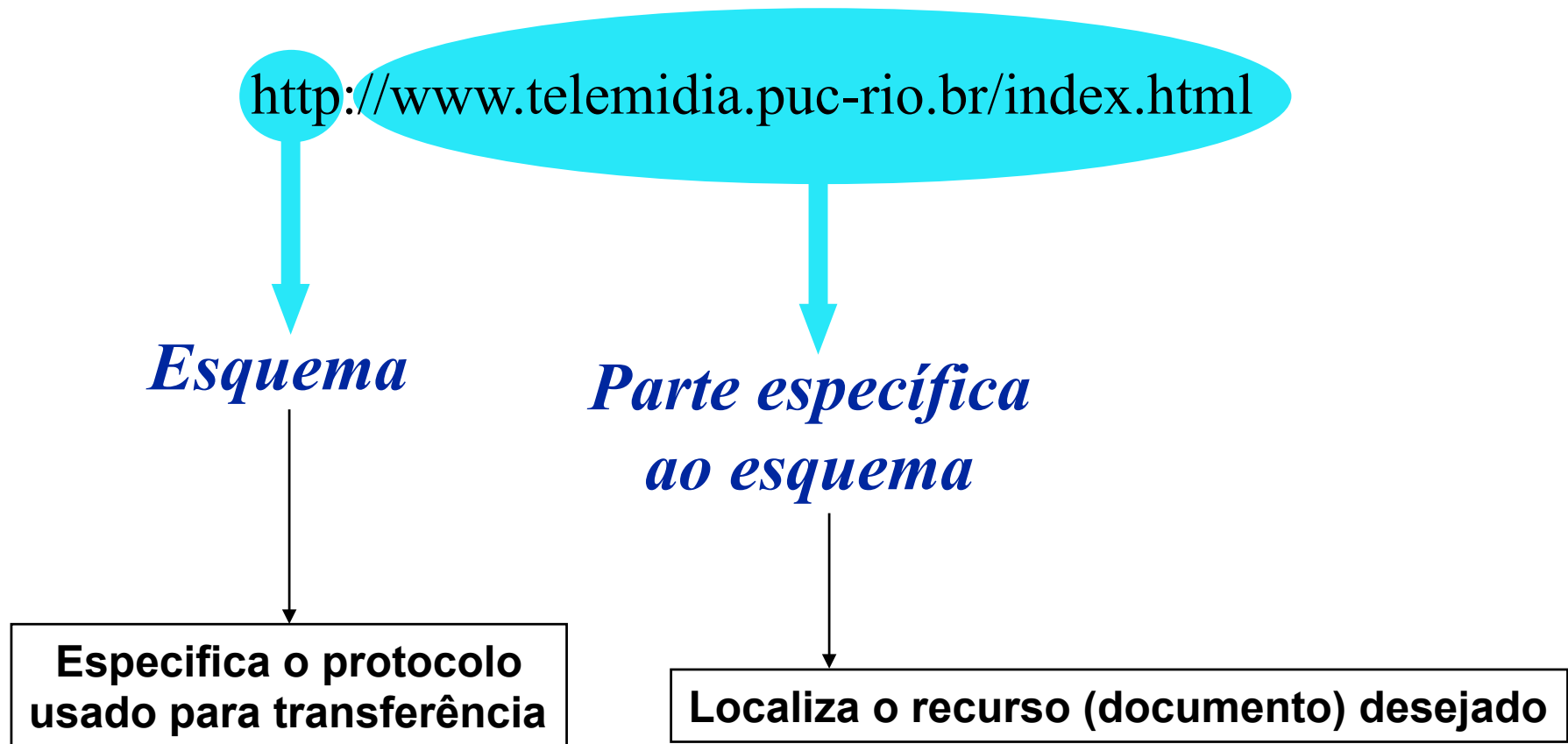
Redes de Computadores



Conceitos Básicos na Web

- ➔ **Como identificar os recursos (documentos)?**
 - URL (Uniform Resource Locator)
- ➔ **Como recuperar um documento?**
 - HTTP (HyperText Transfer Protocol)
- ➔ **Como definir o formato do conteúdo dos documentos?**
 - HTML (HyperText Markup Language)

Exemplo de URL



URL para esquema HTTP

Redes de Computadores

➔ Sintaxe

```
"http://" host [ ":"port ] "/" [path [ "?" query ] ]
```

➔ Exemplos de URL (esquema HTTP)

```
http://www.dimap.ufrn.br:80/~sbmidia2000/
```

```
http://www.telemidia.puc-rio.br/index.html
```

```
http://www.altavista.com/cgi-bin/query?q=client  
%2Fserver
```

```
http://139.82.95.14/index.html
```

HTTP

HyperText Transfer Protocol

HTTP

➔ **Objetivo original**

- capacidade de recuperar de um servidor, documentos simples baseados na mídia texto
- protocolo textual leve e rápido

➔ **Utiliza um serviço de transporte confiável, orientado a conexão (TCP), onde o servidor HTTP usa a porta TCP 80**

HTTP

- **Baseado em um modelo simples de arquitetura cliente-servidor**
 - requisição/resposta
- **Protocolo sem estado: o servidor não mantém registro de requisições e respostas anteriores**
- **Permite transferências bidirecionais**
- **Permite negociações entre cliente e servidor (representação do documento e codificação usadas na transferência) - cabeçalho HTTP usa formato MIME para representar tipos dos dados**
- **suporte para *caching* no cliente**
- **suporte para intermediários na comunicação (*proxy server*)**

HTTP

➔ Versões anteriores: HTTP/0.9, HTTP/1.0

- obrigam o uso de uma nova conexão TCP para cada requisição/resposta

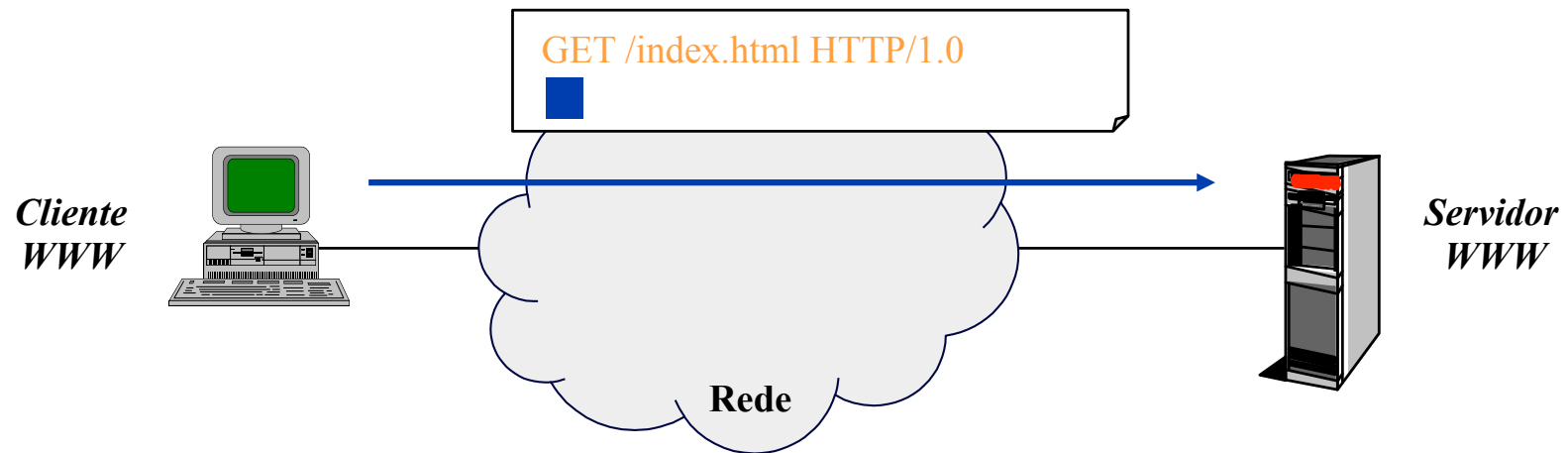
➔ versão HTTP/1.1 - janeiro de 1997 (RFC 2068, RFC 2616)

➔ Principais modificações em relação a versão 1.0:

- Melhora o modelo de uma conexão por requisição/resposta
 - HTTP persistente (P-HTTP): mantém uma conexão aberta durante várias requisições para um mesmo servidor
- campo Host no cabeçalho
 - permite múltiplos hosts em um mesmo endereço IP (*virtual hosts*)
 - obrigatório (código de erro na sua ausência)

Mensagens HTTP/1.0

Redes de Computadores

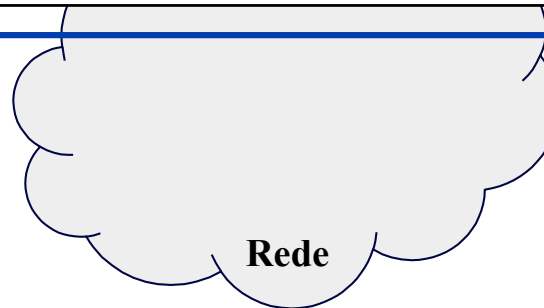
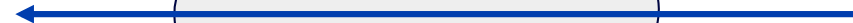
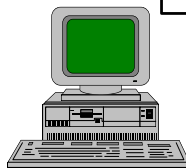


Mensagens HTTP/1.0

```
HTTP/1.0 200 OK
Server: Microsoft-IIS/3.0
Date: Thu, 01 Jun 2000 18:29:26 GMT
Content-Type: text/html
Accept-Ranges: bytes
Last-Modified: Mon, 10 Aug 1998 21:03:04 GMT
Content-Length: 646
```

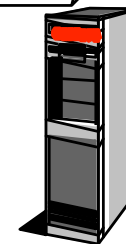
```
■
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">
<HTML>
...
</HTML>
```

*Cliente
WWW*



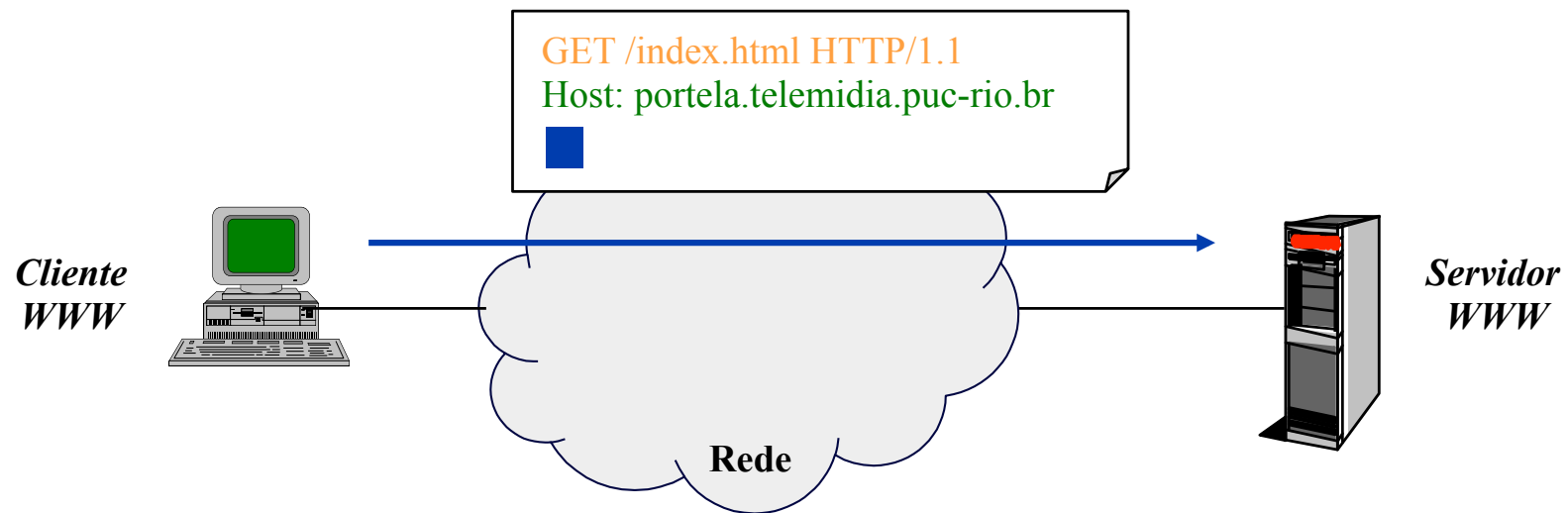
Rede

*Servidor
WWW*



Mensagens HTTP/1.1

Redes de Computadores



Mensagens HTTP/1.1

HTTP/1.1 200 OK

Date: Thu, 18 May 2000 18:41:46 GMT

Server: Apache/1.3.9 (Unix) (Red Hat/Linux)

Last-Modified: Tue, 21 Sep 1999 14:46:36 GMT

Etag: "31282-799-37e79a4c"

Accept-Ranges: bytes

Content-Length: 1945

Content-Type: text/html



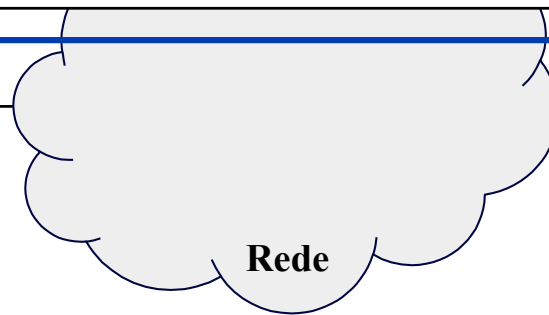
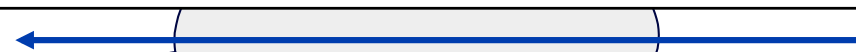
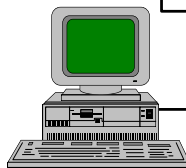
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//  
EN">
```

```
<HTML>
```

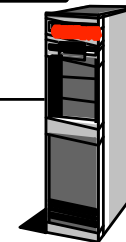
...

```
</HTML>
```

*Cliente
WWW*



*Servidor
WWW*



Mensagens HTTP

```
request =
    request-line
    *(general-header | request-header | entity-header)
    CRLF
    [message-body]
request-line =
    method SP request-URI SP HTTP-version CRLF
```

```
response =
    status-line
    *(general-header | response-header | entity-header)
    CRLF
    [message-body]
status-line =
    HTTP-version SP status-code SP reason-phrase CRLF
```

Mensagens HTTP/1.1

➔ Métodos de requisição

- GET, HEAD, POST, PUT, DELETE, TRACE, OPTIONS e CONNECT

➔ Códigos de retorno (3 dígitos)

- divididos em categorias em função do primeiro dígito
 - 1xx (informativo)
 - 2xx (sucesso)
 - 3xx (redireção)
 - 4xx (erro do cliente)
 - 5xx (erro do servidor)
- podem ser estendidos
- normalmente mensagens de erro são enviadas no formato HTML

Críticas ao HTTP

➔ Sem estado

- requisições em paralelo numa mesma conexão precisam ser enfileiradas

➔ Implementação integral complexa

➔ Fundamentado no TCP como protocolo de transporte

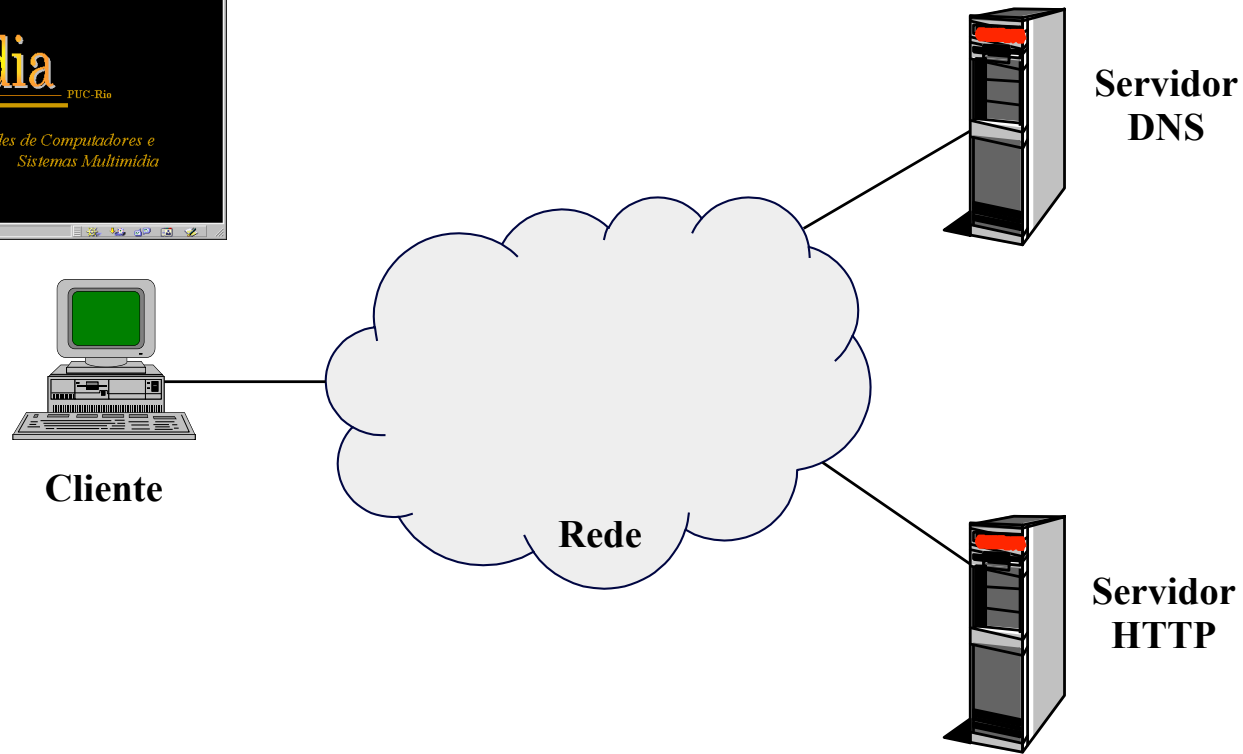
➔ Requisições em um único sentido

➔ Ausência de um padrão para definição de extensões

WWW



imperatriz 139.82.95.14

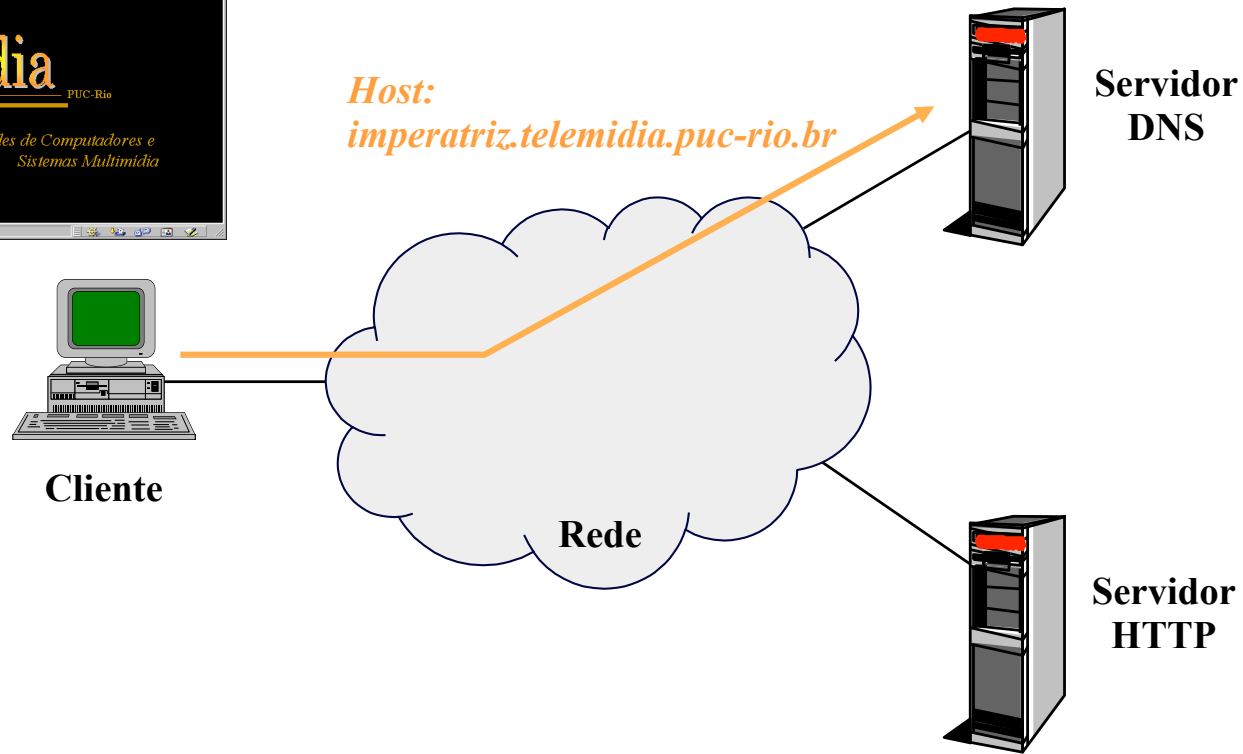


`http://imperatriz.telemidia.puc-rio.br/index.html`

WWW



imperatriz 139.82.95.14 *res*

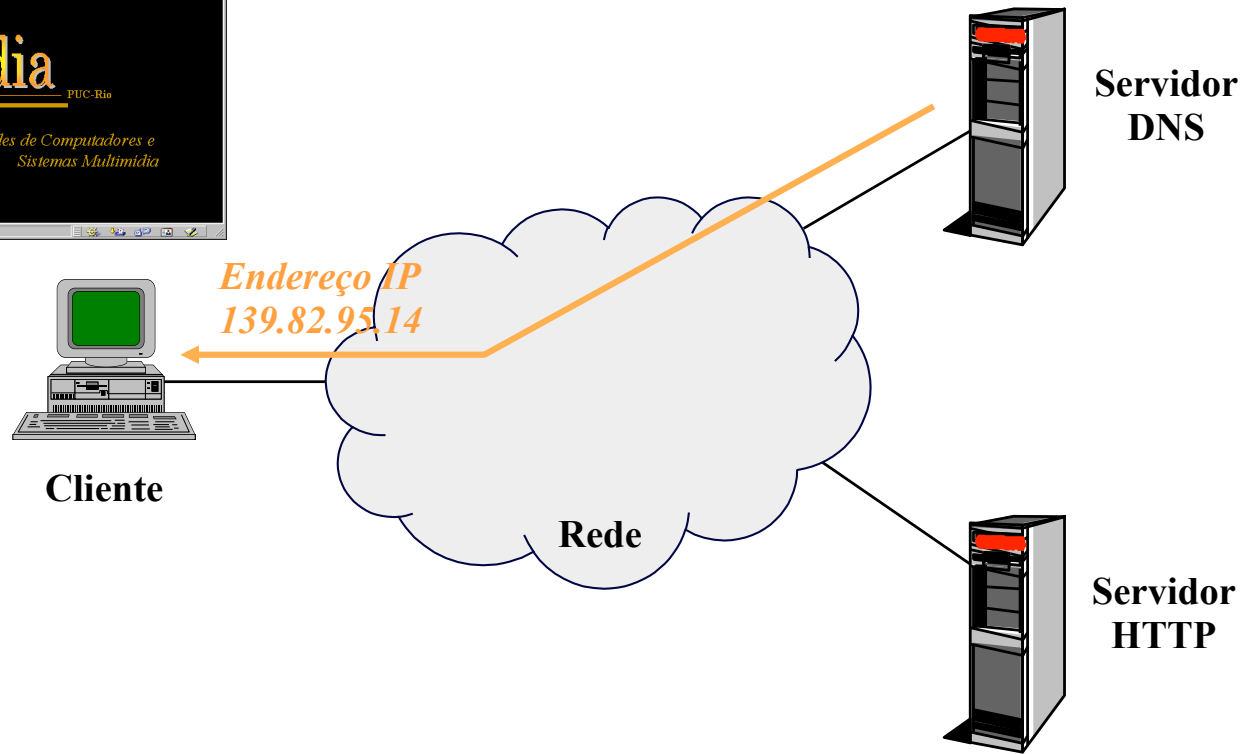


`http://imperatriz.telemidia.puc-rio.br/index.html`

WWW



imperatriz 139.82.95.14

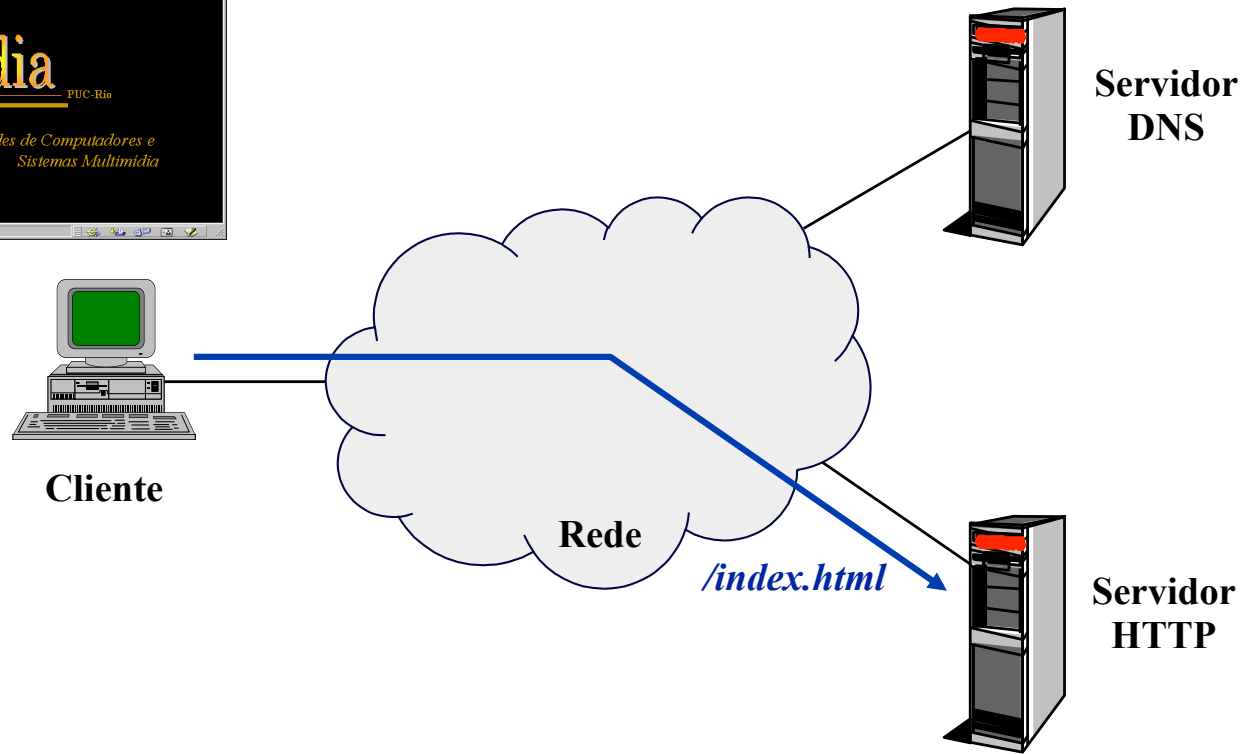


`http://imperatriz.telemidia.puc-rio.br/index.html`

WWW



imperatriz 139.82.95.14

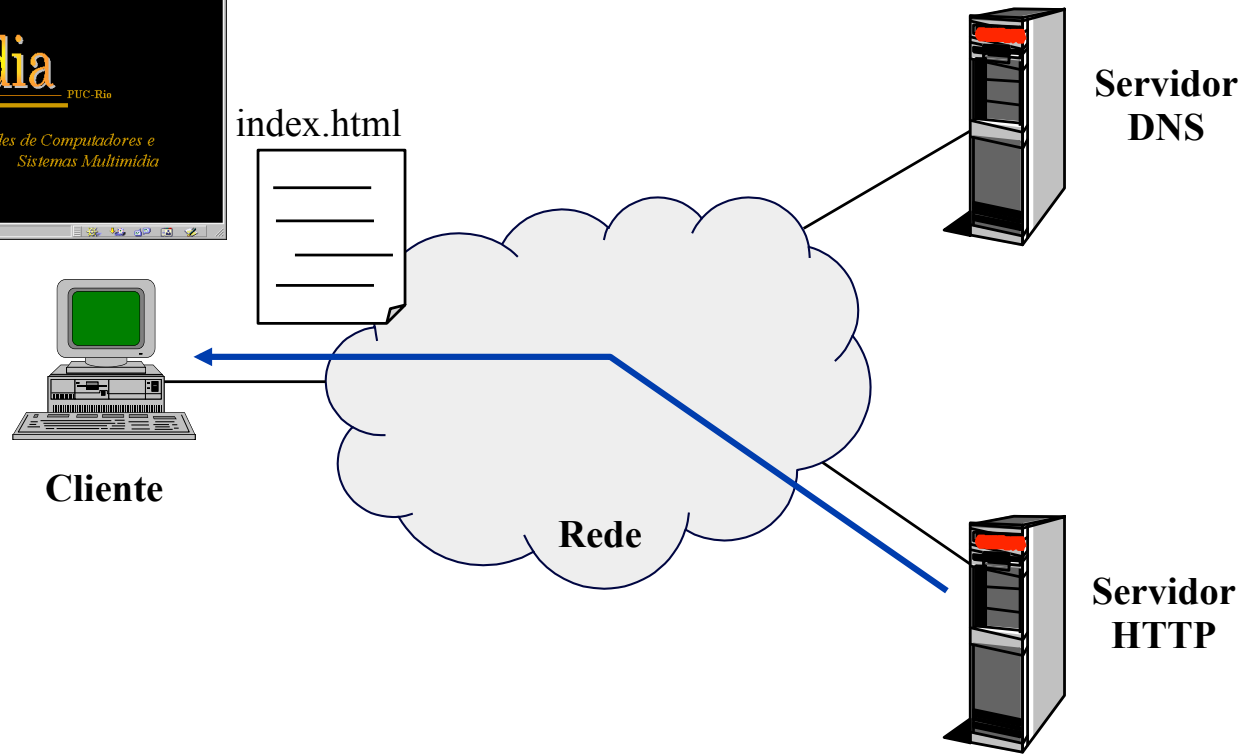


`http://imperatriz.telemidia.puc-rio.br/index.html`

WWW

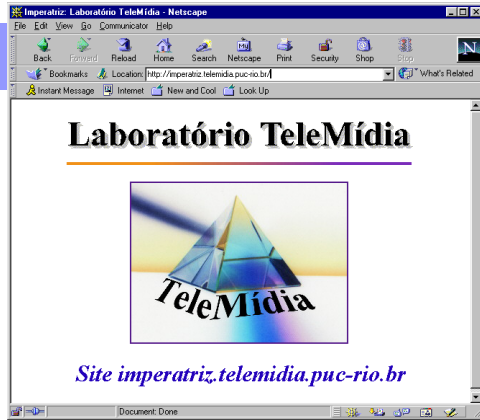


imperatriz 139.82.95.14 *res*



`http://imperatriz.telemidia.puc-rio.br/index.html`

WWW



imperatriz 139.82.95.14 *res*

