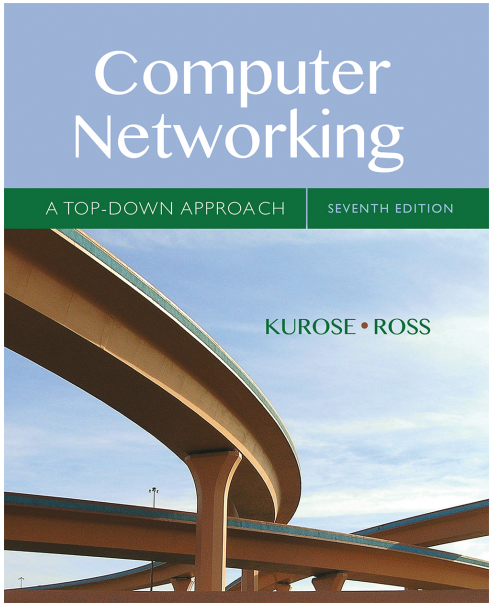
DNS

Comunicações por Computador

Mestrado Integrado em Engenharia Informática

3º ano/2º semestre

2018/2019



***Computer Networking: A Top Down Approach*, Capítulo 2**

DNS

- E-Mail precisa entregar uma mensagem dirigida a a10000@alunos.uminho.pt, costa@di.uminho.pt e belem@presidencia.pt ... onde entregar? Como fazer?
- Servidor de mail recebe pedido de entrega suspeita de 222.122.229.55... deve aceitar a conexão ou não? Como decidir?
- Browser precisa de iniciar conexão TCP com www.google.com ... para onde mandar o pacote de SYN? Para qual dos googles mandar?
- Tenho uma chamada VoIP para o número de telefone +351 253 604442? O que faço com ela?
- Vou ligar-me a www.cgd.pt... Mas será mesmo a Caixa Geral de Depósitos??

DNS

- **Ferramentas cliente: host, nslookup, dig**

  $ host www.google.pt

  $ host 193.136.19.20

  $ nslookup www.google.pt

  $ dig www.google.pt

DNS: Domain Name System

Pessoas: muitos identificadores:

- Segurança Social, Contribuinte, Nome, BI, Nº Passaporte...

Internet: hosts e routers:

- Endereços IP (32 ou 128 bit) - usados para endereçar datagramas
- "nome", ex: www.google.com – usado pelos humanos

Q: Como mapear os endereços IP nos nomes ?

Uma App? A fazer coisas da rede? Não devia estar no layer 3?

Domain Name System:

- ***Base de dados distribuída*** **implementada numa hierarquia de** ***servidores de nomes***
- ***Protocolo da camada de aplicação*** **hosts, routers, servidores de nomes comunicam parar** ***resolver*** **nomes (tradução endereço/nome)**
  - nota: uma função nuclear da Internet implementada como protocolo de aplicação
  - Complexidade na periferia ("*edge*") da rede!

DNS

Serviços DNS

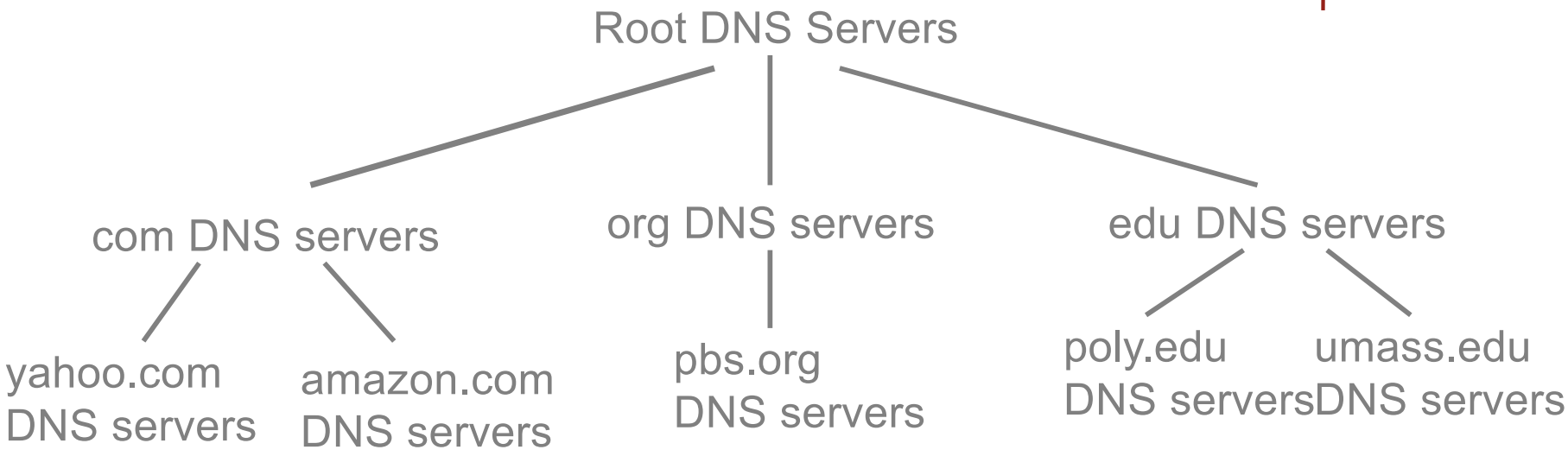
- **Tradução dos nomes dos hosts para endereços IP**
- **Aliases dos hosts (nomes alternativos)**
  - Nome principal, aliases
- **Definição do servidor de mail**
- **Distribuição de carga**
  - Servidores Web replicados: um conjunto de endereços IP associados a um único nome

Porque não centralizar o DNS?

- **ponto de falha único**
- **Volume de tráfego**
- **Base de dados centralizada distante**
- **manutenção**

É claro que não é *escalável!*

Base de dados distribuída e hierárquica

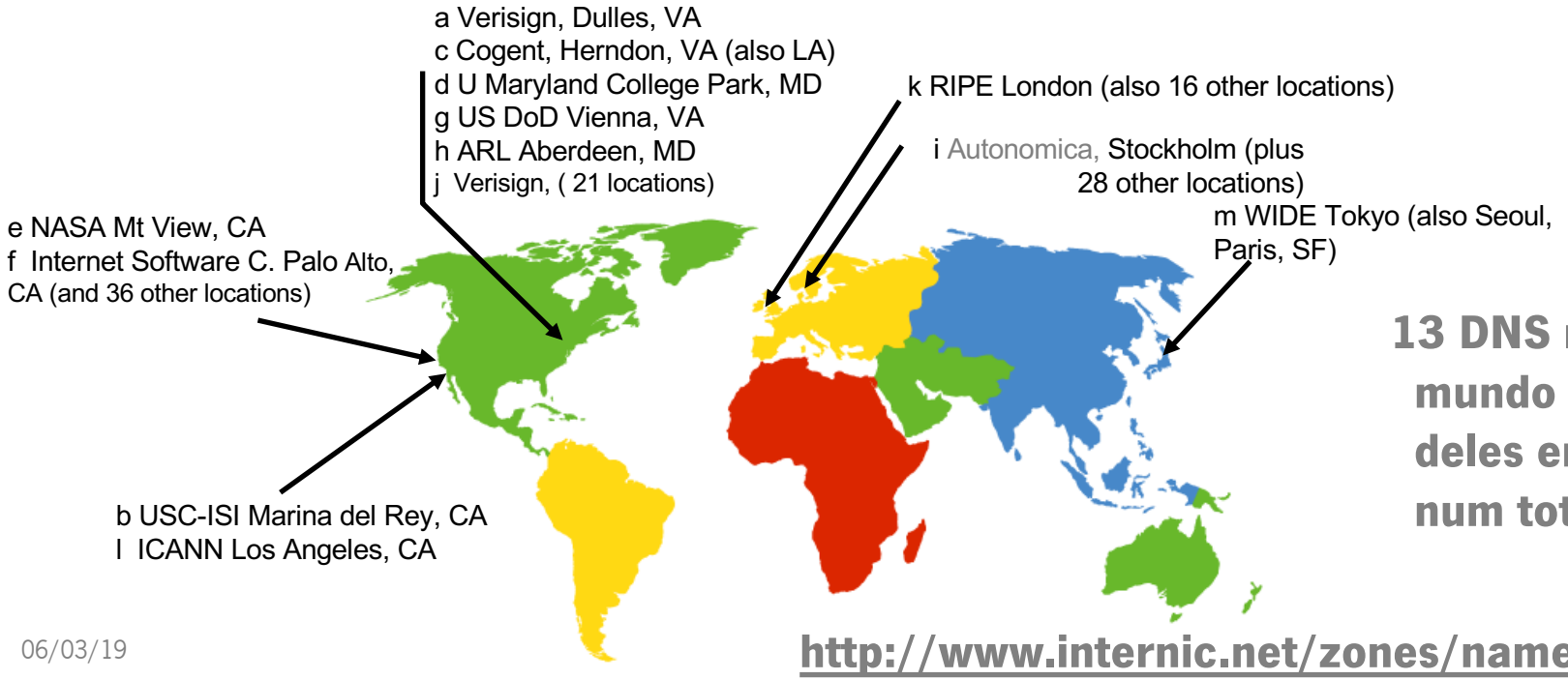


Cliente pretende o IP de www.amazon.com; 1ª aproximação (funciona?):

- **Cliente interroga um *root server* para descobrir servidores de DNS para o domínio de topo *com***
- **Cliente interroga servidor DNS de *com* para obter o servidor DNS de *amazon.com***
- **Cliente interroga servidor DNS de *amazon.com* para obter o endereço IP de *www.amazon.com***
- **Cliente guarda toda a informação obtida nesta interacção em cache (servidores DNS, endereços IP, etc.)**

DNS: Root servers

- **São contactados pelos servidores de nomes locais que não conseguem resolver um nome**
- **O que pode/poderia fazer o *root server*:**
  - Contactar servidor DNS autoritativo se o mapeamento do nome não é conhecido
  - Obtém mapeamento (nome, servidor DNS)
  - Retorna esse mapeamento ao servidor de nomes local



13 DNS root servers no mundo inteiro, mas 9 deles em AnyCast, num total de 239!

http://www.internic.net/zones/named.root
http://www.iana.org/domains/root/files

TLD e Servidores Autoritativos

- **Domínios de topo (TLD: Top-Level Domain Servers)**
  - responsáveis por *com, org, net, edu,* etc, e todos os domínios de topo dos países *pt, uk, fr, ca, jp*, etc.
  - *Network Solutions* administra os servidores TLD para o domínio *com*
  - *Educause* gere o TLD *edu*
  - TLD para Portugal: Associação DNS Portugal
- **Servidores DNS autoritativos:**
  - Servidores DNS das organizações, com autoridade sobre um domínio de nomes local e sobre os mapeamentos nome/endereço dessa organização
  - Pode ser gerido pela própria organização ou pelo seu ISP

Servidor de Nomes local

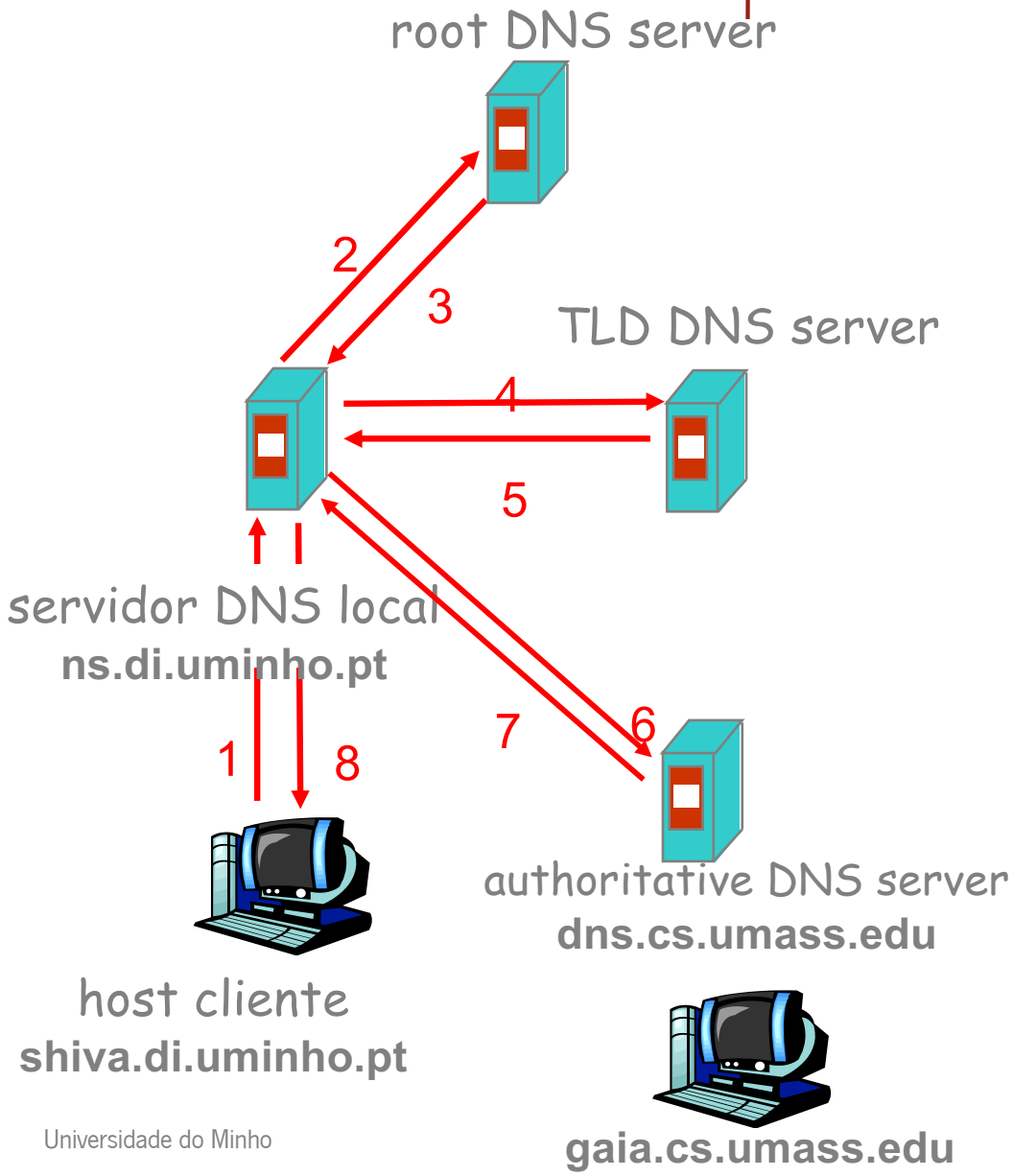
- **Pode pertencer à hierarquia**
- **Cada ISP (ISP residencial, empresas, universidades) tem um.**
  - Também designado por "*default name server*"
- **Quando um host formula uma interrogação DNS ela é sempre dirigida ao seu Servidor DNS local**
  - Funciona como um proxy, redireccionando a query para a hierarquia quando necessário; designa-se por ***forwarder***
  - Faz caching
  - O papel de ***proxy/caching*** justifica só por si a existência do servidor local; pode ainda acumular funções de ***servidor autoritativo***;

DNS: exemplo da resolução dum nome

- O Host *shiva.di.uminho.pt* pretende o endereço IP de *gaia.cs.umass.edu*

Modo interactivo:

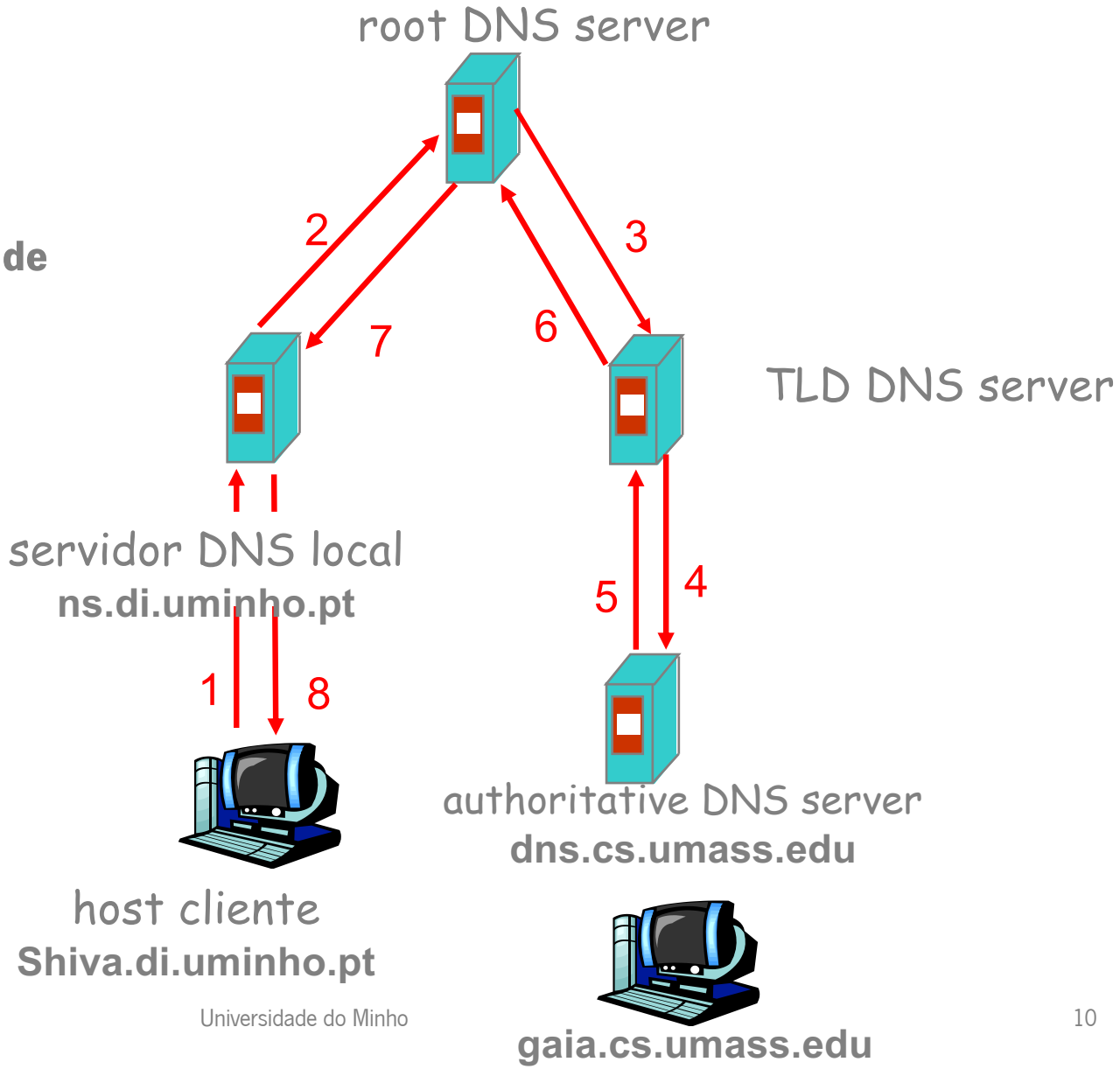
- Servidor contactado responde com o nome do servidor a contactar
- "Eu não conheço esse nome, mas pergunte a este servidor"



DNS: exemplo da resolução dum nome

Modo recursivo:

- **Coloca o fardo da resolução no servidor de nomes contactado**
- ***Fardo* pesado?**



DNS: modo de operação

- **Todas as aplicações consultam o DNS!!**
  - Enviar uma mensagem de e-mail pode implicar 2 ou três consultas!
  - Aceder a uma página WWW, implica pelo menos 1 consulta!

Tem de ser **muito** eficiente!

- **Funciona sobre UDP:**
  - basta um único datagrama (512 bytes) por cada pedido e por cada resposta
- **Existem múltiplos servidores por cada domínio:**
  - Um servidor **primário** e um ou mais **secundários**
  - Os servidores **secundários** mantém, de forma automática, réplicas dos primários
- **Os servidores e os clientes armazenam as respostas obtidas durante um certo tempo (TTL) para não andarem sempre a perguntar a mesma coisa...**
  - **caching**

DNS: actualização de dados

- **sempre que um (qualquer) servidor de nomes aprender um mapeamento, guarda-o de imediato em *cache***
  - as entradas na cache expiram (*timeout*) e desaparecem após algum tempo
  - Os servidores de nomes TLD guardam normalmente em cache os servidores DNS locais
    - Portanto os *root name servers* acabam por não ser assim tão visitados...
- **Mecanismos de actualização dinâmica e notificação (update/notify) já definidos pelo IETF**
  - RFC 2136
  - http://www.ietf.org/html.charters/dnsind-charter.html

DNS resource records (RR)

DNS: BD distribuída que armazena *resource records* **(RR)**

Formato RR: `(name, value, type, ttl)`

- **Type=A**
  - **`name`** é o nome de um *host*
  - **`value`** é o *endereço IP*
- **Type=NS**
  - **`name`** é um nome de um *domínio* (ex:. uminho.pt)
  - **`value`** é o nome do host do servidor DNS autoritativo para o domínio
- **Type=CNAME**
  - **`name`** é um *alias* (nome alternativo) para outro nome "canónico" (o real!)

    `www.dn.pt` é na realidade `dn.sapo.pt`
  - **`value`** é o nome canónico (real)
- **Type=MX**
  - **`name`** é um nome de domínio
  - **`value`** é o nome do servidor de e-mail associado ao nome **`name`**

DNS resource records (RR)

http://en.wikipedia.org/wiki/List_of_DNS_record_types

SOA	(Start Of Authority)	Define o início de uma zona e todos os seus parâmetros...
NS	(Name Server)	Define o(s) servidor(es) que detém autoridade numa zona
MX	(Mail Exchanger)	Define o(s) servidor(es) de mail para o domínio...
A	(Address)	Endereço IP v4.
AAAA	(IPv6 Address)	Endereço IP v6.
HINFO	(Hardware Info)	Define o CPU e o SO de um sistema...
WKS	(Well Known Services)	Define serviços (portas) disponíveis num sistema
PTR	(Pointer)	Apontador para o nome... usado no reverse-mapping
SRV	(Service Locator)	Generalização do MX para localizar outros serviços…
CNAME	(Canonical name)	Nome alternativo...
KEY	(Public key)	Chave pública
SIG	(Signature)	Assinatura digital

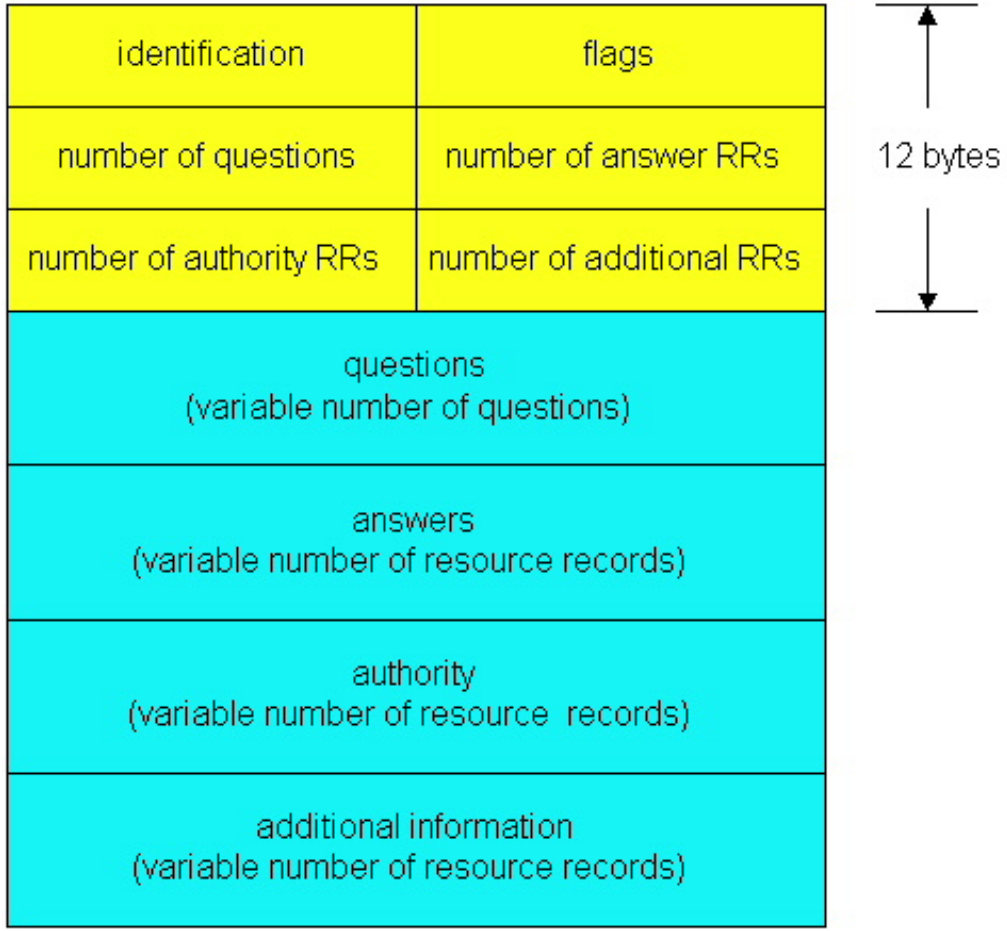
Segurança (Veremos mais tarde)
Só em 2010 nos root servers!!

DNS protocolo, mensagens protocolares

Protocolo DNS: duas mensagens (*query* e *reply)*, exactamente com o mesmo *formato*

cabeçalho da mensagem

- **identification:** quantidade de 16 bits; uma resposta usa sempre o mesmo valor da interrogação
- **flags:**
  - query or reply
  - recursion desired
  - recursion available
  - reply is authoritative



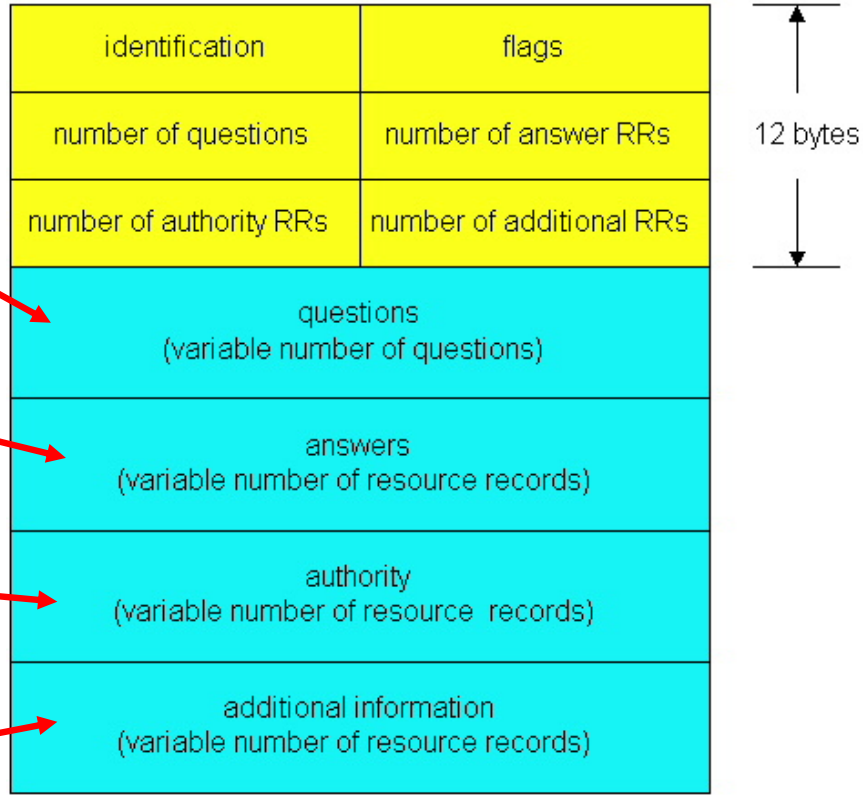
DNS protocolo, mensagens protocolares

Name e ***Type*** da interrogação

RRs de resposta à interrogação

records com nomes de servidores autoritativos

informação adicional, não requerida, mas que pode ser muito útil



Formato binário (não é texto!) compactado.
Nunca se repete duas vezes a mesma *label*.
Usam-se apontadores sempre que possível.

Resolução de Nomes (DNS)

- **O programa** nslookup **permite consultar o DNS directamente:**
  - Obter um endereço IP, dado um nome

```
C:> nslookup xpto.com
Server: dns.xyz.pt
Address: 194.67.3.20

Name: xpto.com
Address: 198.41.0.6
```

```
C:> nslookup
>  set type=A
>  xpto.com
...
```

  - Obter um nome, dado um endereço IP

```
C:> nslookup
> set type=PTR
> 193.136.9.240
...
```

```
C:> nslookup
>  set type=PTR
>  240.9.136.193.in-addr.arpa.
...
```

Resolução de Nomes (DNS)

- **O programa** nslookup **permite consultar o DNS directamente:**
  - Obter os servidores de E-Mail de um domínio (Mail eXchangers)

```
C:> nslookup
>  set type=MX
>  uminho.pt
...
```

  - Obter os servidores de DNS de um domínio (Name Servers)

```
C:> nslookup
>  set type=NS
>  uminho.pt
...
```

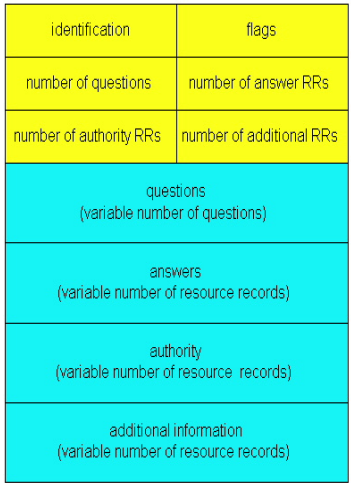
  - Saber quem é o responsável por um domínio (type = SOA)

Resolução de Nomes (DNS)

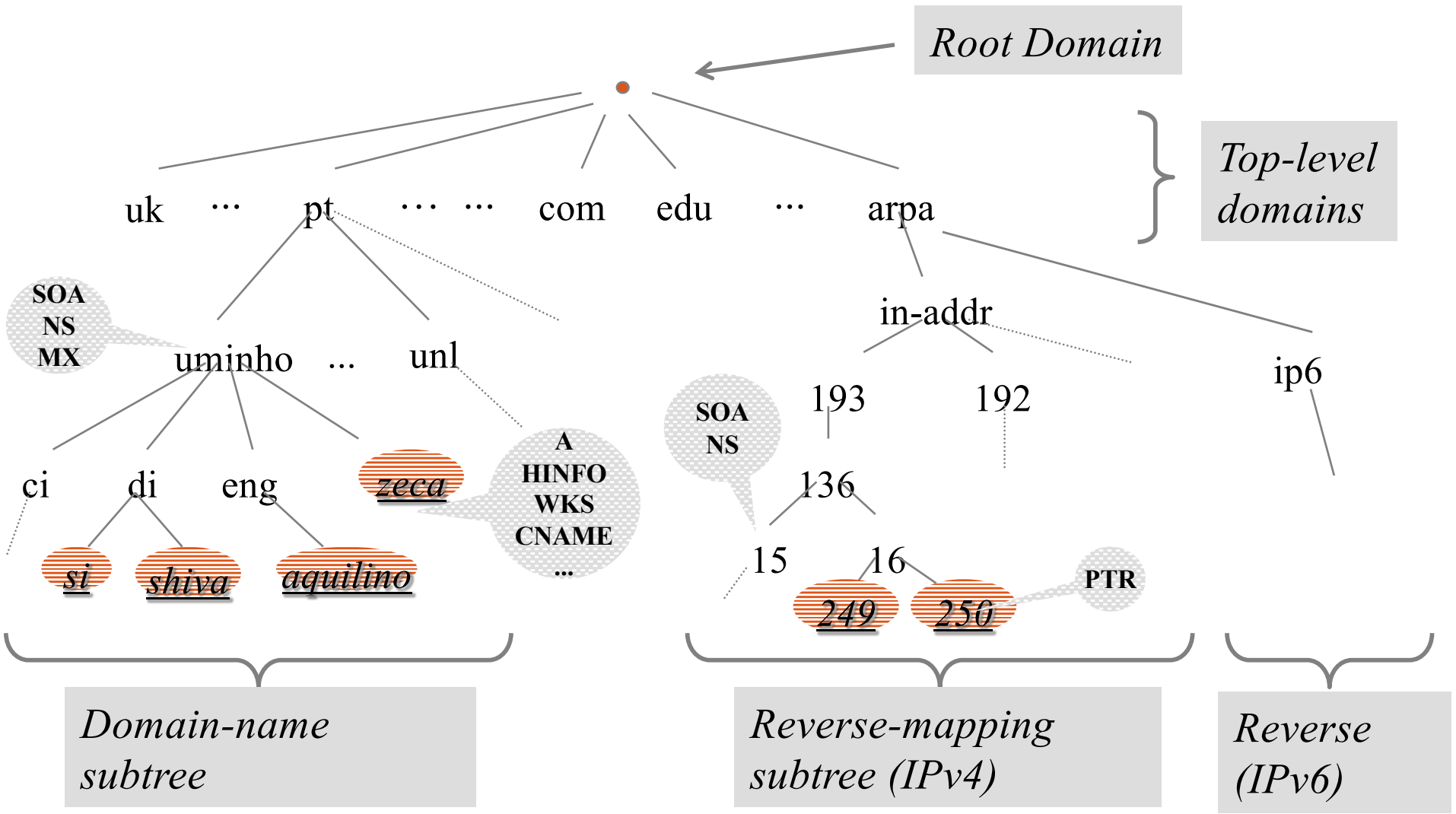
```
> set debug
> set ty=mx
> alunos.uminho.pt.
Non-authoritative answer:
Server:  marco.uminho.pt
Address:  193.136.9.240

------------
Got answer:
    HEADER:
        opcode = QUERY, id = 2, rcode = NOERROR
        header flags:  response, want recursion, recursion avail.
        questions = 1,  answers = 1,  authority records = 3,  additional = 7

    QUESTIONS:
        alunos.uminho.pt, type = MX, class = IN
    ANSWERS:
    ->  alunos.uminho.pt
        MX preference = 5, mail exchanger = mx.uminho.pt
        ttl = 3600 (1 hour)
    AUTHORITY RECORDS:
    ->  alunos.uminho.pt
        nameserver = dns3.uminho.pt
        ttl = 46640 (12 hours 57 mins 20 secs)
    ->  alunos.uminho.pt
        nameserver = dns2.uminho.pt
        ttl = 46640 (12 hours 57 mins 20 secs)
    ->  alunos.uminho.pt
        nameserver = dns.uminho.pt
        ttl = 46640 (12 hours 57 mins 20 secs)
    ADDITIONAL RECORDS:
    ->  mx.uminho.pt
        internet address = 193.137.9.142
        ttl = 150 (2 mins 30 secs)
    ->  dns3.uminho.pt
        internet address = 193.137.16.65
        ttl = 150 (2 mins 30 secs)
    ->  dns3.uminho.pt
        AAAA IPv6 address = 2001:690:2280:1::65
        ttl = 150 (2 mins 30 secs)
    ->  dns.uminho.pt
        internet address = 193.137.16.75
        ttl = 150 (2 mins 30 secs)
    ->  dns.uminho.pt
        AAAA IPv6 address = 2001:690:2280:1::75
        ttl = 150 (2 mins 30 secs)
    ->  dns2.uminho.pt
        internet address = 193.137.16.145
        ttl = 150 (2 mins 30 secs)
    ->  dns2.uminho.pt
        AAAA IPv6 address = 2001:690:2280:801::145
        ttl = 150 (2 mins 30 secs)
```



Resolução de Nomes (DNS)



Resolução de Nomes (DNS)

```
> set ty=PTR
> 1.19.136.193.in-addr.arpa.
Non-authoritative answer:
Server:  marco.uminho.pt
Address:  193.136.9.240

------------
Got answer:
    HEADER:
        opcode = QUERY, id = 21, rcode = NOERROR
        header flags:  response, want recursion, recursion avail.
        questions = 1,  answers = 1,  authority records = 6,  additional = 10

    QUESTIONS:
        1.19.136.193.in-addr.arpa, type = PTR, class = IN
    ANSWERS:
    ->  1.19.136.193.in-addr.arpa
        name = dns.di.uminho.pt
        ttl = 251 (4 mins 11 secs)
    AUTHORITY RECORDS:
    ->  19.136.193.in-addr.arpa
        nameserver = alfa.di.uminho.pt
        ttl = 251 (4 mins 11 secs)
    ->  19.136.193.in-addr.arpa
```

Exercício

- Quais os servidores TLD para pt?
- Qual deles é o servidor primário?
- Qual o(s) servidor(es) de correio electrónico do DI?
- Qual o endereço IP do Google?
- Qual a máquina que usa o IP 209.197.89.22?
- É possível fazer balanceamento de carga usando o DNS?

Exercício

- **Para todos os clientes da rede fixa do DI, designados em abstracto por Cli(di), o servidor local do DNS é, também em abstracto, NS(di). Essa informação é colocada manualmente no ficheiro /etc/resolv.conf, ou obtida automaticamente por DHCP. Suponha agora que um Cli(di) quer resolver o nome mail.nasa.gov. Explique como se processa a resolução nas seguintes circunstâncias:**
  - Todos os servidores de nomes da hierarquia aceitam responder em modo recursivo (altamente improvável!)
  - Só o servidor local admite responder em modo recursivo ao cliente;
  - Nenhum servidor admite o modo recursivo;

Implementação de um domínio

- **O MIEI quer estabelecer-se em Portugal e montar o site WEB www.miei.pt;**

Passo 1: Instalar e configurar **ns.miei.pt** como servidor primário

Incluir ficheiro com todos os ROOT servers;

Passo 2: Instalar um servidor secundário noutro domínio;

(Atualização automática; pedir a outra organização)

Passo 3: Pedir ao servidor primário de PT (ver qual é) que adicione dois RR debaixo de pt para ***"delegar autoridade"*** em ns.lei.pt para lei.pt:

miei.pt	**IN**	**NS**	**ns.miei.pt.**
ns.miei.pt.	**IN**	**A**	**193.136.130.1**

Implementação de um domínio

- **Ficheiro** *named.conf*

```
options {
        directory "/var/named";
        allow-transfer "193.168.100.1"; // secundário
  };

  // type    domain     source          file
   zone "." {
        type hint;
        file "named.root";
   };
   zone "localhost" {
        type master;
        file "named.local";
   };
   zone "127.in-addr.arpa" {
        type master;
        file "named.rev-local";
   };
```

```
zone “miei.pt" {
        type master;
        file “miei.db";
};

zone “130.136.193.in-addr.arpa" {
        type master;
        file “miei-rev.db";
};
```

Exemplo de um ficheiro de dados

- **Ficheiro** *miei.db*

```
$ORIGIN pt.
Miei        86400   IN      SOA     ns.miei.pt.     admin.miei.pt. (
            2017031501
            86400
            7200
            604800
            86400 )
        IN      NS      ns.miei.pt.
        IN      NS      ns.dns.pt.
        IN      MX      10      mail.miei.pt.
$ORIGIN miei.pt.
            IN      A       193.136.130.80
mail        IN      A       193.136.130.25
www         IN      A       193.136.130.80
ns          IN      A       193.136.130.1
```

Exemplo de um ficheiro de dados

- **Ficheiro** *miei.rev*

```
130.136.193.in-addr.arpa. IN    SOA    ns.miei.pt.   admin.miei.pt. (
                2017031501      ; Serial
                28800           ; Refresh - 24 hours
                7200            ; Retry - 2 hours
                604800          ; Expire - 7 days
                86400 )         ; Minimum TTL - 1 days
      IN     NS     ns.miei.pt.
      IN     NS     ns.dns.pt.

$ORIGIN 130.136.193.in-addr.arpa.
1           IN     PTR    ns.miei.pt.
25          IN     PTR    mail.miei.pt.
80          IN     PTR     www.miei.pt.
```

Programação – Consulta simples

- **Em Java**

java.lang.Object

java.net.InetAddress

```
static InetAddress      getByName(String host)
static InetAddress[]    getAllByName(String host)
```

java.lang.Object

java.net.InetAddress

java.net.Inet6Address

- **Em C**

```
#include <netdb.h>
struct hostent *    gethostbyname(const char *name);
struct hostent *    gethostbyaddr(const void *addr, socklen_t len, int type);
```

Programação – Todas as queries

- **Em Java**

  Package javax.naming

  Provides the classes and interfaces for accessing naming services.

- **Em C**

  Usar as funções da biblioteca "***Resolver***"

```
#include <sys/types.h>
#include <netinet/in.h>
#include <arpa/nameser.h>
#include <resolv.h>

  res_mkquery(...)
  res_send(...)
```