

Redes de Computadores

Nomes e endereços

Departamento de Informática da
FCT/UNL

Nomes

- Só se pode manipular, aceder, discutir, etc. algo (uma entidade qualquer) que sabemos como nomear
- Nomes são **sequências de símbolos**, válidos num dado contexto, que designam entidades. Para além das propriedades de designação, essas sequências (*strings*) de símbolos podem também codificar atributos das entidades a que estão associadas
- Na prática, os nomes são de vários tipos, sendo os mais importantes os nomes propriamente ditos, os nomes do tipo identificador e os os nomes do tipo endereço
- Em geral, os endereços estão associados ao acesso à entidade e podem mudar dinamicamente se a entidade pode mudar de forma ou local de acesso ou é móvel por exemplo

Exemplos de Nomes

João das Neves e Sousa	Nome de uma pessoa (só é válido num certo contexto de interpretação)
Presidente	Nome de uma função (indireção)
Abóborinha	Alcunha (alias) para o nome de uma pessoa na agenda de outra
www.fct.unl.pt	Nome hierárquico de um servidor que pode ser interpretado no DNS
IBM	Símbolo na bolsa de uma empresa (só pode ser corretamente interpretado nesse contexto)

A estrutura de um nome está relacionada com o seu contexto de interpretação e condiciona o que se pode fazer com ele

Nomes e Atributos

- Alguns nomes codificam as propriedades da entidade que designam
 - www.fct.unl.pt (o servidor WWW da FCT/UNL)
 - João Pacheco, o filho de Pacheco
- A estrutura de um nome está relacionada com o contexto de interpretação e condiciona o que se pode fazer com ele
- Os nomes mais comuns são diretamente interpretáveis pelos humanos e estão associados a propriedades da entidade que designam
 - www.fct.unl.pt

Endereços

- Exemplos de endereços
 - Um endereço IP: 193.136.120.10 (hierárquicos)
 - Uma porta IP: porta 80 (*flat*)
 - Um endereço de memória (0x040004400)
 - Coordenadas: X=32, Y=100, Z=88 (por tuplos)
 - Coordenadas: Latitude=25.561W, Longitude=57.678N
 - Quinta do Monte, Edifício II, Gabinete P3/1, Monte da Caparica (hierárquicos)
- Os endereços são nomes que codificam atributos de localização da entidade a que estão associados

Nomes do tipo Identificador

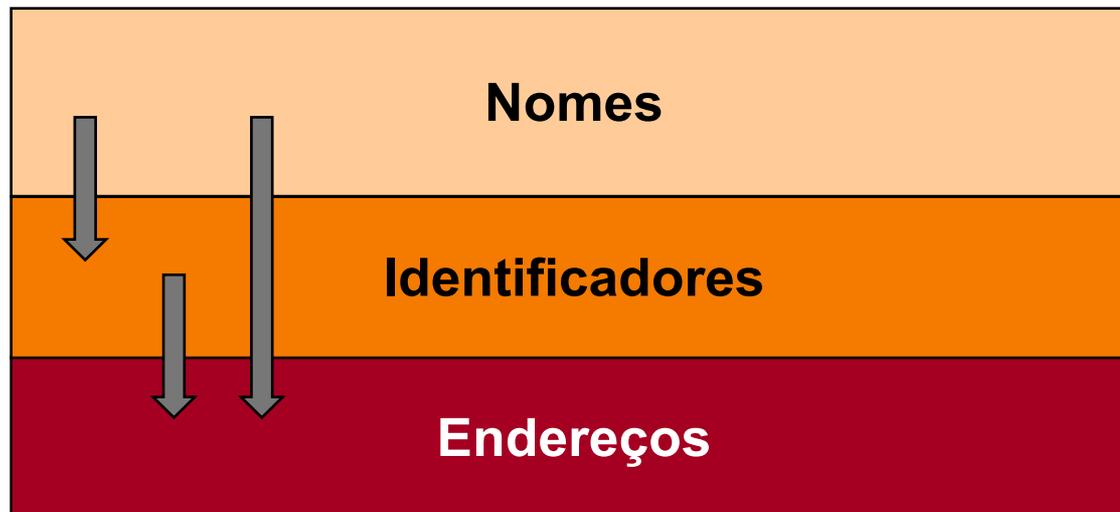
- Este tipo de nomes são geralmente unívocos
 - Não há duas entidades distintas com o mesmo NIF (número de contribuinte)
- Identificadores únicos não reutilizáveis (UUID)
 - São identificadores que estão associados para todos o sempre à mesma entidade, mesmo que esta mude de localização
 - Estas propriedades facilitam a sua gestão
 - A geração de UUIDs aleatórios é a mais simples
- Estrutura
 - Podem ser hierárquicos ou *flat*
 - Quando são *flat* não codificam propriedades das entidades
- O método de geração às vezes codifica atributos
 - Exemplo: NIF começado por $n \geq 5$ codifica uma entidade coletiva

Exemplos de Identificadores

- Exemplos de identificadores
 - O número de identificação fiscal (identificadores únicos)
 - O número do bilhete de identidade (identificadores únicos)
 - Um número de telefone móvel (identificador reutilizável)
 - Outros não são geralmente visíveis mas um exemplo é uma chave pública (um identificador com muita semântica)
 - O endereço hardware de uma placa Ethernet (admitindo que não é alterado)
 - Um identificador diz-se global se a sua interpretação é independente da localização (isto é, não se altera se a entidade muda de localização)
 - Um identificador diz-se único se a sua interpretação é independente do tempo (isto é, não será reutilizado)
- Os identificadores geralmente têm uma estrutura condicionada à sua geração e aos algoritmos que os usam

Nomes, identificadores e endereços

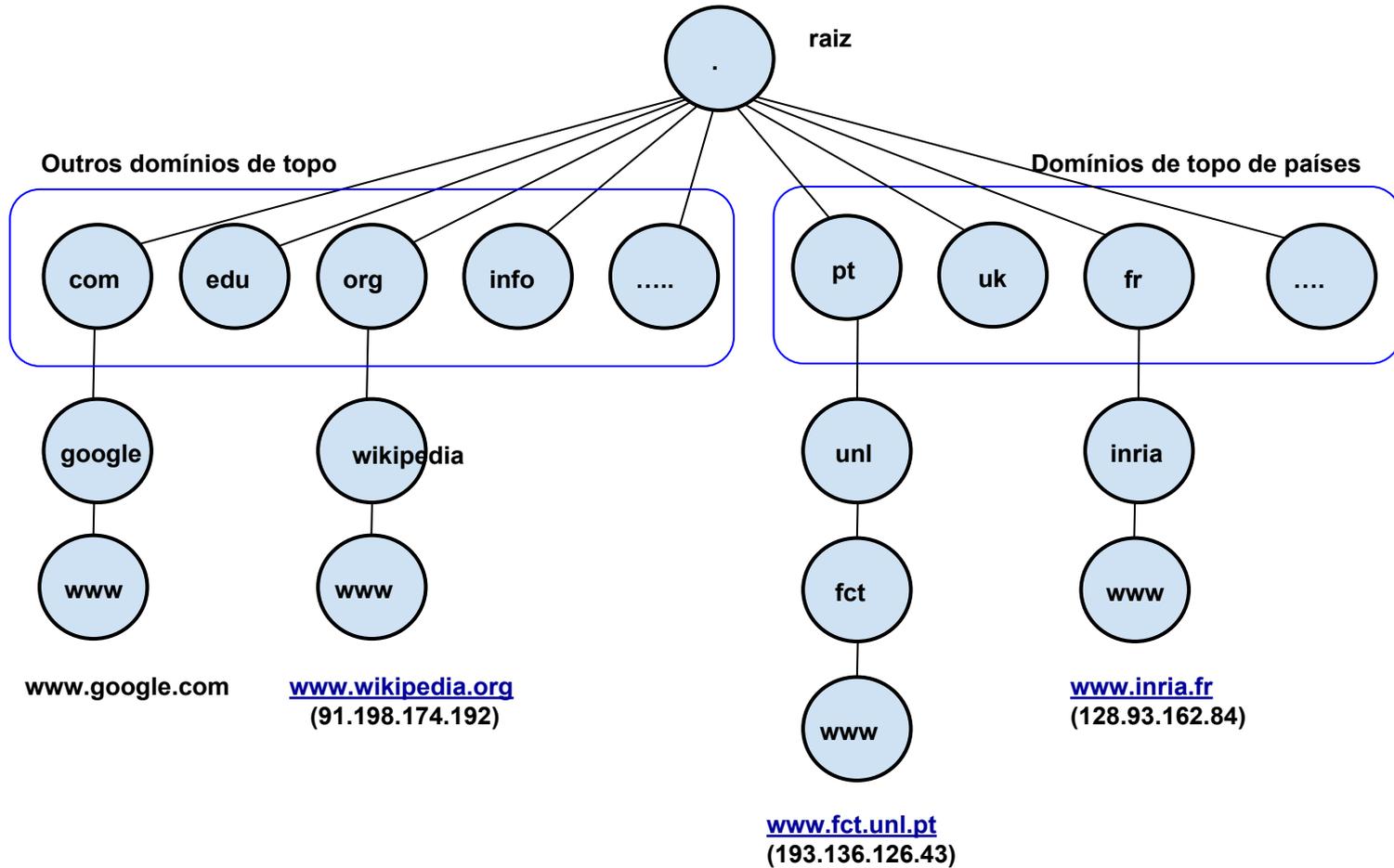
- Os diferentes tipos de nomes estão organizados em hierarquia
- Os nomes propriamente ditos estão no topo da hierarquia e são geralmente acessíveis aos utilizadores finais
- Os identificadores são nomes intermédios manipuláveis por algoritmos
- Os endereços são usados pelo nível rede
- Existem sistemas de tradução entre níveis de designação



Diretórios / catálogos de nomes

- As aplicações e os sistemas necessitam de traduzir entre camadas de designação
 - Por exemplo, traduzir um nome de um servidor no seu endereço IP
 - Traduzir o identificador de um recurso no endereço de um servidor que dá acesso a uma sua réplica (uma cópia de um livro identificado pelo seu ISBN)
 - Traduzir um endereço IP num endereço de placa Ethernet (localização)
 - Traduzir um número de telefone móvel (número do CHIP) no endereço da célula / canal onde o mesmo está
- Para esse efeito usam-se diretórios (*directories*) ou catálogos

Domain Name System



Sintaxe dos nomes DNS

- Construídos hierarquicamente e de baixo para cima (www.fct.unl.pt)
- As componentes do nome são separadas por "."
- Um *fully qualified name* termina com "." que é o nome da raiz (www.fct.unl.pt == www.fct.unl.pt.)
- São *case insensitive* (www.fct.unl.pt == WWW.FCT.UNL.PT)
- Cada domínio pode ter até 63 caracteres e um nome pode ter até 255 caracteres

Visão simplificada do DNS como Base de Dados

Nome	TTL	Tipo	Valor
.	254996	NS	a.root-servers.net.
a.root-servers.net.	254785	A	198.41.0.4
a.root-servers.net.	257226	AAAA	2001:503:ba3e::2:30
publico.pt.	3349	A	195.23.42.21
publico.pt.	1744	NS	ns1.novis.pt.
publico.pt.	1744	NS	dns.publico.pt.
dns.publico.pt.	864	A	193.126.13.202
www.unl.pt.	12215	A	193.137.110.30
ns.unl.pt.	86400	A	193.137.110.15
ns.unl.pt.	86400	A	193.137.110.9
fct.unl.pt.	9308	NS	dns1.fct.unl.pt.
fct.unl.pt.	9308	NS	ns3.unl.pt.
www.fct.unl.pt.	27041	A	193.136.126.43
fct.unl.pt.	10800	MX	30 ASPMX2.GOOGLEMAIL.COM.
fct.unl.pt.	10800	MX	30 ASPMX3.GOOGLEMAIL.COM.

DNS Records

- O DNS é uma base de dados distribuída de nomes e seus atributos
- Cada atributo de um nome é dado por um RR (Resource Record)
 - Formato dos RRs: (nome, valor, tipo, TTL)
- Tipos de RRs mais comuns
 - A (host name, A, IP address, TTL)
 - NS (domain name, NS, name server name, TTL)
 - CNAME (name, CNAME, canonical name, TTL)
 - MX (domain name, MX, priority and mail server name, TTL)
 - TXT (name, TXT, comment, TTL)

Interface Resolver: Exemplo em Java

```
import java.net.InetAddress;

public class FindAddr
{
    public static void main(String[] argv) throws Exception
    {
        InetAddress addr = InetAddress.getByName(argv[0]);
        System.out.println(addr.getHostAddress());
    }
}
```

```
import java.net.InetAddress;

public class GetHostnameByAddr
{
    public static void main(String[] argv) throws Exception
    {
        InetAddress addr = InetAddress.getByName(argv[0]);

        System.out.println("Host name: "+addr.getHostName());
        System.out.println("Ip address: "+ addr.getHostAddress());
    }
}
```

Exemplo de Interrogação do DNS

```
$ dig www.fct.unl.pt a
```

```
; <<>> DiG 9.8.3-P1 <<>> www.fct.unl.pt a
```

```
;; global options: +cmd
```

```
;; Got answer:
```

```
;; ->HEADER<<- opcode: QUERY, status: NOERROR, id: 25128
```

```
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 2
```

```
;; QUESTION SECTION:
```

```
;www.fct.unl.pt.                IN      A
```

```
;; ANSWER SECTION:
```

```
www.fct.unl.pt.                914     IN      A      193.136.126.43
```

```
;; AUTHORITY SECTION:
```

```
fct.unl.pt.                    8846    IN      NS      dns2.fct.unl.pt.
```

```
fct.unl.pt.                    8846    IN      NS      dns1.fct.unl.pt.
```

```
.....
```

```
;; Query time: 22 msec
```

```
;; SERVER: 192.168.1.254#53(192.168.1.254)
```

```
;; WHEN: Sun Oct 30 12:08:28
```

A performance do DNS baseia-se em *caching*

- Todos os servidores localizam os servidores de *root* e fazem *caching* dos seus endereços
- Todos os servidores quando obtêm uma resposta, mantêm-na em cache e dessa forma respondem imediatamente se aparecer um pedido semelhante
- Uma entrada é mantida na cache até um limite de tempo controlado pelo administrador do servidor responsável pelo nome *cached* através do atributo TTL
- As entradas *cached* podem estar desatualizadas!

Estratégia de *caching* e consistência

- Sem *caching*, a resolução de nomes acrescentaria facilmente 1 segundo até que fosse possível começar um *download*
- Assim os *local name servers* e os *browsers* fazem *caching* intensivo dos endereços dos servidores mais populares
- Como garantir que as *caches* estão consistentes e atualizadas?
- Notificações explícitas seriam demasiado dispendiosas por isso a consistência baseia-se em envelhecimento (TTL)
- O TTL está incluído em todas as respostas e quando é ultrapassado a informação é suprimida

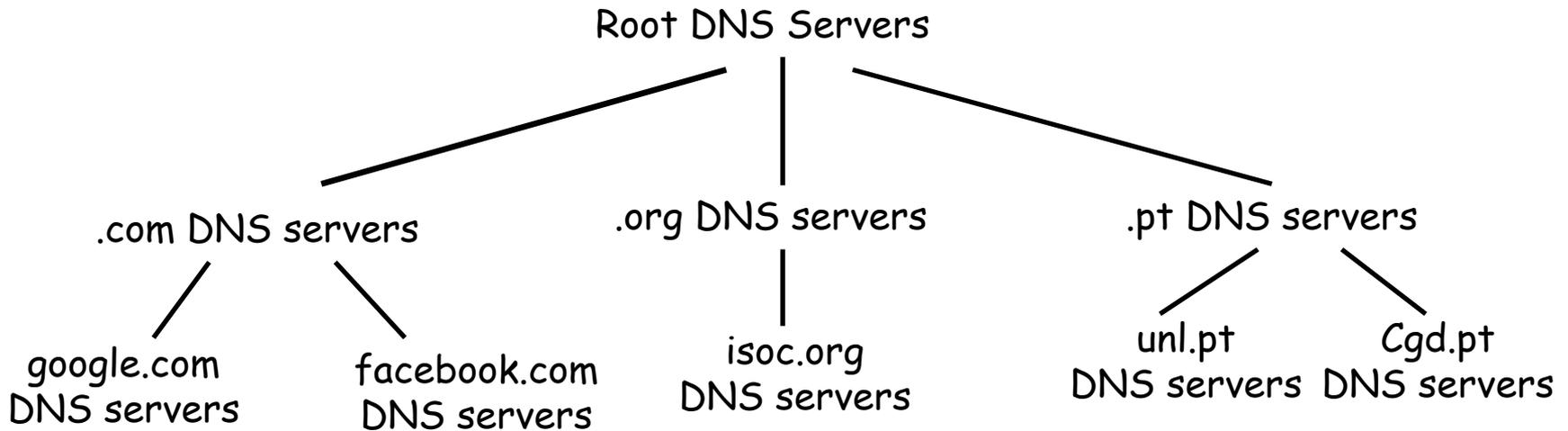
Servidores DNS

- A cada domínio estão associados pelo menos dois ou mais servidores de DNS
- Um conjunto especial de servidores permitem chegar aos outros, são os *root name servers*
- Esses servidores garantem a unicidade da árvore do DNS caso estejam sincronizados entre si
- A gestão desses servidores é assegurada pela ICANN

Root Name Servers

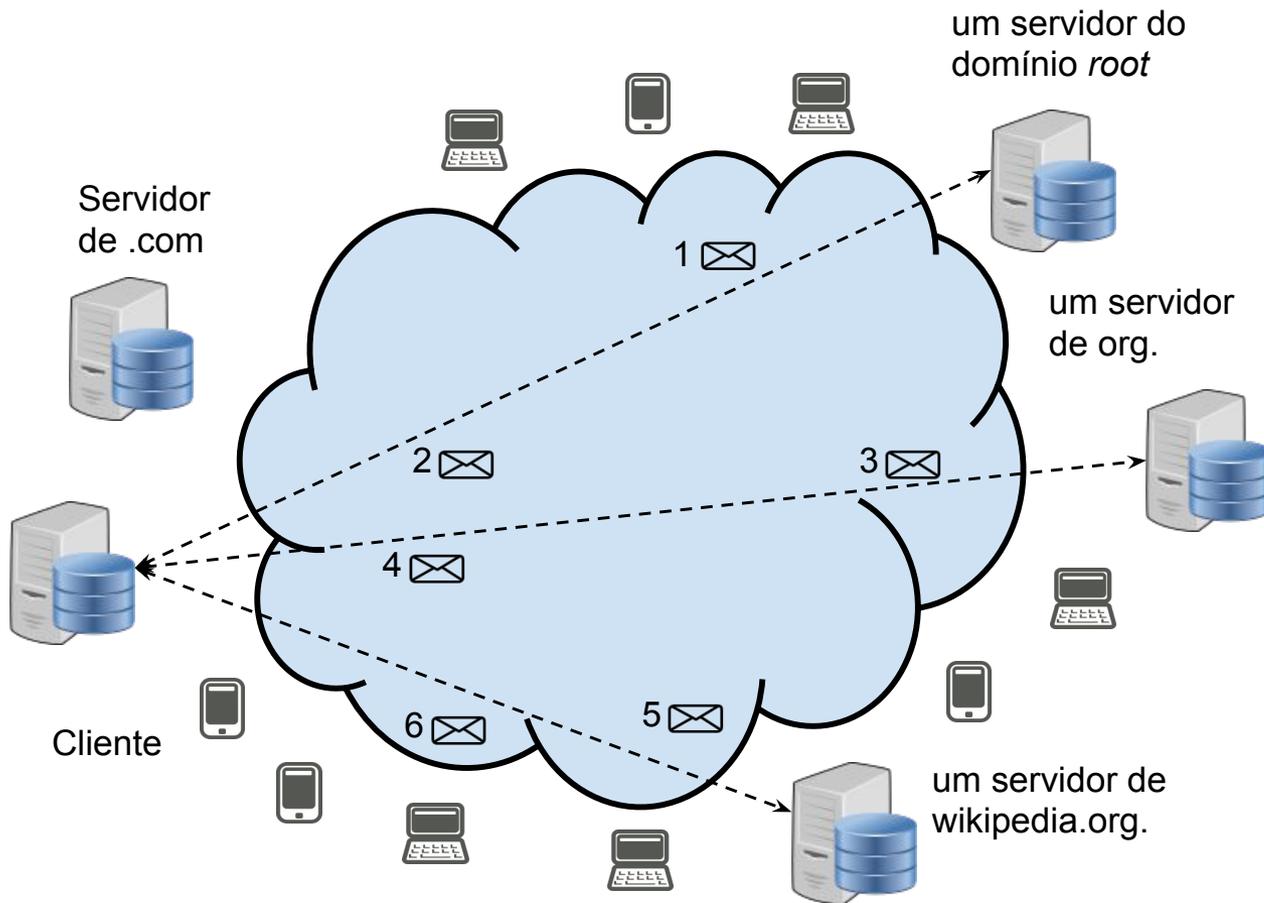


Servidores dos Domínios e Resolução das Queries DNS



- Se o cliente quer conhecer o IP de um nome (e.g. isoc.org)
 - Começa por contactar um dos *root name server* que lhe indicará o endereço IP de servidores de .org
 - O cliente contacta um desses servidores e obterá o endereço IP dos servidores de isoc.org
 - O servidor contactará finalmente um desses servidores servidores para obter a resposta pretendida

Resolução de um nome DNS



- O protocolo utiliza pedidos e respostas colocadas dentro de *datagramas* UDP
- Em caso de falha, os clientes são responsáveis pela re emissão dos pedidos e por contactar servidores alternativos

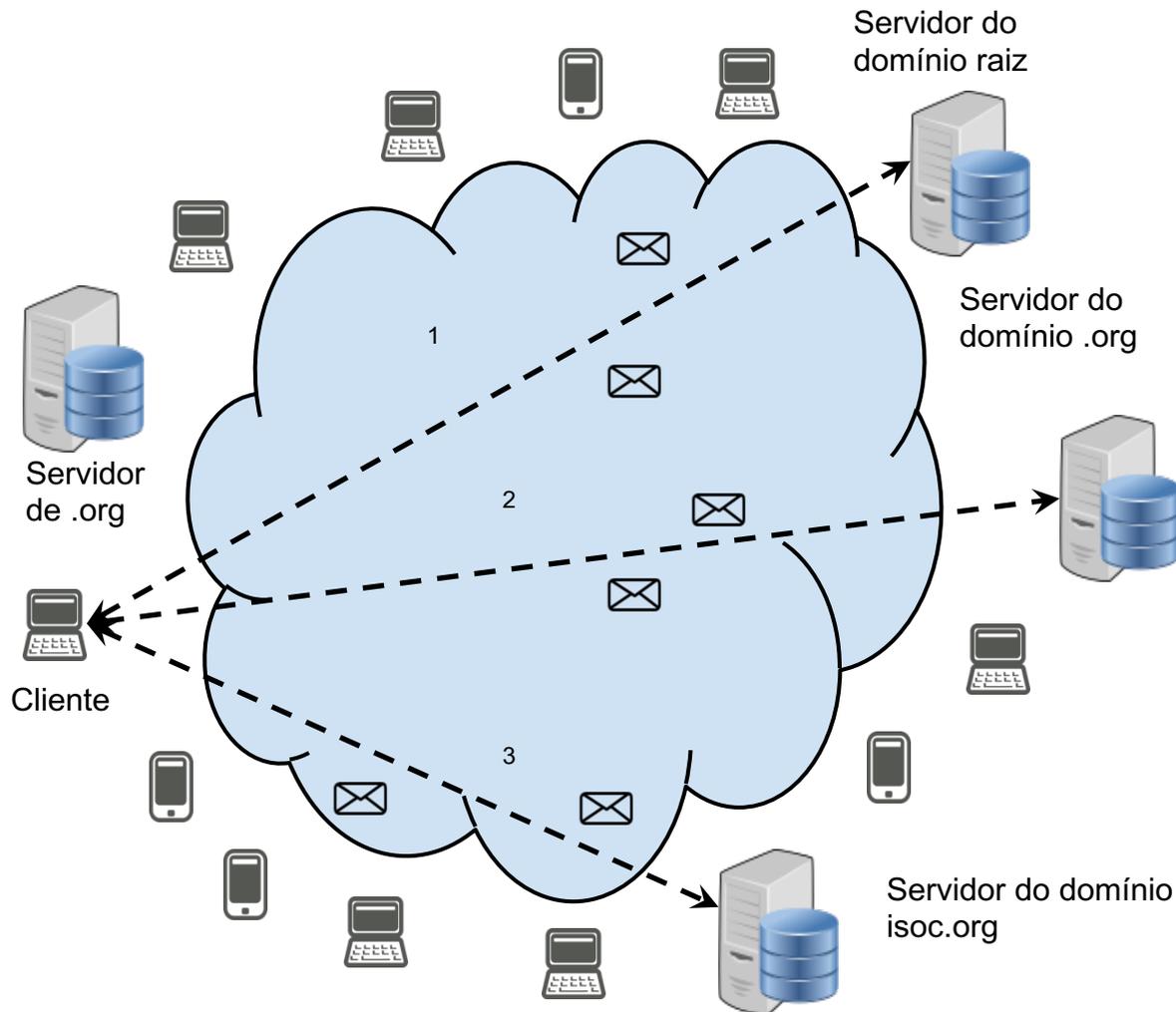
Exemplo do progresso da Resolução da Query

	.	3596604	IN	NS	g.root-servers.net.		
	.	3596604	IN	NS	a.root-servers.net.		
						
zona root	g.root-servers.net.	3596468	IN	A	192.112.36.4		
	a.root-servers.net.	3596555	IN	A	198.41.0.4		
						
	org.	81806	IN	NS	a0.org.afili-as-nst.info.		
	a0.org.afili-as-nst.info.	84913	IN	A	199.19.56.1		

zona org.	org.	81806	IN	NS	a0.org.afili-as-nst.info.		
	org.	81806	IN	NS	c0.org.afili-as-nst.info.		
						
	a0.org.afili-as-nst.info.	84913	IN	A	199.19.56.1		
	c0.org.afili-as-nst.info.	81747	IN	A	199.19.53.1		
						
zona wikipedia.org.	wikipedia.org.	74189	IN	NS	ns0.wikimedia.org.		
	ns0.wikimedia.org.	75786	IN	A	208.80.154.238		

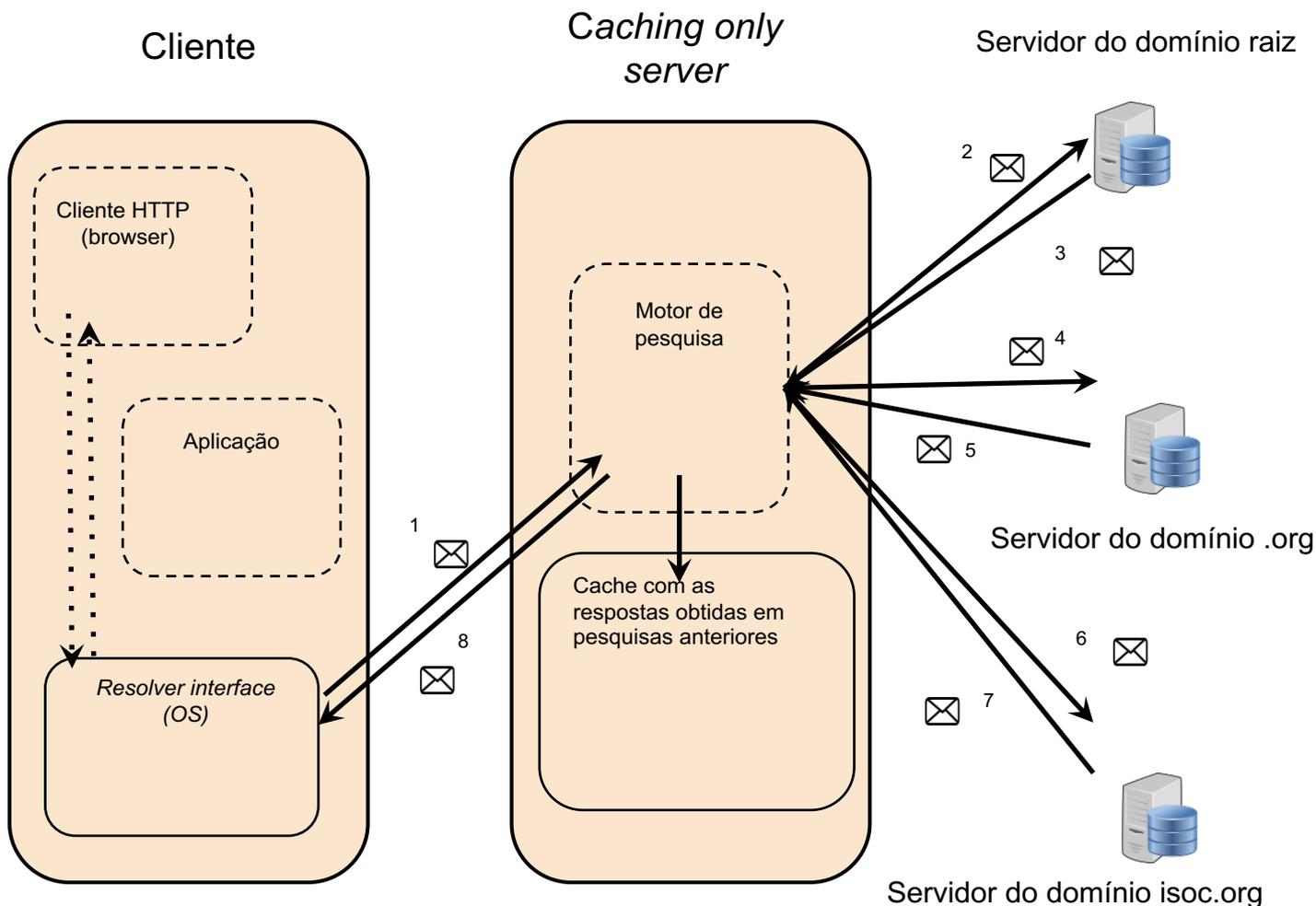
	wikipedia.org.	74189	IN	NS	ns1.wikimedia.org.		
wikipedia.org.	74189	IN	NS	ns0.wikimedia.org.			
wikipedia.org.	74189	IN	NS	ns2.wikimedia.org.			
.....							
ns1.wikimedia.org.	75807	IN	A	208.80.153.231			
ns0.wikimedia.org.	75786	IN	A	208.80.154.238			
ns2.wikimedia.org.	79032	IN	A	91.198.174.239			
.....							
	www.wikimedia.org.	600	IN	A	91.198.174.192		

Outro exemplo



- O protocolo utiliza pedidos e respostas colocadas dentro de *datagramas* UDP
- Em caso de falha, os clientes são responsáveis pela re emissão dos pedidos e por contactar servidores alternativos

DNS Resolver e Local DNS Server



O Caching é baseado num *time-to-live (TTL)* definido pelo servidor responsável pelo nome. Permite evitar estar sempre a contactar os servidores remotos.

Estratégia de *caching* e consistência

- Sem *caching*, a resolução de nomes acrescentaria facilmente 1 segundo até que fosse possível começar um *download*
- Assim os *local name servers* e os *browsers* fazem *caching* intensivo dos endereços dos servidores mais populares
- Como garantir que as *caches* estão consistentes e atualizadas?
- Notificações explícitas seriam demasiado dispendiosas por isso a consistência baseia-se em envelhecimento (TTL)
- O TTL está incluído em todas as respostas e quando é ultrapassado a informação é suprimida

Caching, consistência e soft state

- Notificações explícitas seriam demasiado dispendiosas
 - Os servidores teriam de memorizar que caches tinham nomes seus memorizados para poderem notificá-las das alterações
 - Que aconteceria a essas caches se o servidor tivesse um problema?
 - Este sistema não escalaria
- A consistência baseia-se em envelhecimento (com TTLs) pois
 - A informação evolui muito lentamente (pelo menos na maioria dos casos)
 - Quando se usa informação inconsistente isso pode ser detectado
 - O custo da inconsistência é baixo
- O TTL de um dado é fixado pelo seu responsável original e está incluído em todas as respostas
- Quando o TTL expira a informação associada é suprimida das caches
- Este tipo de gestão de dados replicados designa-se por **Soft State**

Que valor de TTL usar?

- Trade-offs

- Pequeno TTL: atualização rápida quando há alterações
- Grande TTL: maior *hit-ratio* na cache

- Seguir a hierarquia

- TTL no topo da hierarquia: dias, semanas, meses
- TTL no fundo da hierarquia: segundos, minutos, horas

- Problemas

- As CDN usam TTLs de alguns segundos para fazerem distribuição de carga
- Os browsers fazem *caching* durante alguns minutos

Negative Caching

- Quando se dá um nome errado, o mesmo não pode ser resolvido na *cache*
- E portanto levam muito tempo a resolver
 - Exemplos www.cnn.com e www.cnnn.com
 - Podem levar segundos para dar como resposta "nome desconhecido"
- Portanto convém memorizar os nomes inexistentes
 - ... de forma a que no futuro o erro leve menos tempo a detectar
 - Mas é melhor não memorizar por muito tempo e usar um TTL para expirar também os nomes não existentes

Distribuição de Carga com o DNS

```
$ dig fct.unl.pt MX
```

```
; <<>> DiG 9.8.3-P1 <<>> fct.unl.pt MX
```

```
;; global options: +cmd
```

```
;; Got answer:
```

```
;; ->HEADER<<- opcode: QUERY, status: NOERROR, id: 57538
```

```
;; flags: qr rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 2, ADDITIONAL: 11
```

```
;; QUESTION SECTION:
```

```
;fct.unl.pt.                IN          MX
```

```
;; ANSWER SECTION:
```

```
fct.unl.pt.                4963        IN          MX          10 ASPMX.L.GOOGLE.COM.  
fct.unl.pt.                4963        IN          MX          20 ALT1.ASPMX.L.GOOGLE.COM.  
fct.unl.pt.                4963        IN          MX          20 ALT2.ASPMX.L.GOOGLE.COM.  
fct.unl.pt.                4963        IN          MX          30 ASPMX2.GOOGLEMAIL.COM.  
fct.unl.pt.                4963        IN          MX          30 ASPMX3.GOOGLEMAIL.COM.
```

```
;; AUTHORITY SECTION:
```

```
fct.unl.pt.                3185        IN          NS          dns1.fct.unl.pt.  
fct.unl.pt.                3185        IN          NS          dns2.fct.unl.pt.
```

Continuação

;; ADDITIONAL SECTION:

ASPMX.L.GOOGLE.COM.	285	IN	A	64.233.166.27
ALT1.ASPMX.L.GOOGLE.COM.	164	IN	A	74.125.131.26
ALT1.ASPMX.L.GOOGLE.COM.	249	IN	AAAA	2a00:1450:4010:c0b::1b
ALT2.ASPMX.L.GOOGLE.COM.	105	IN	A	74.125.200.27
ALT2.ASPMX.L.GOOGLE.COM.	69	IN	AAAA	2404:6800:4003:c00::1a
ASPMX2.GOOGLEMAIL.COM.	189	IN	A	173.194.222.27
ASPMX2.GOOGLEMAIL.COM.	164	IN	AAAA	2a00:1450:4010:c0b::1b
ASPMX3.GOOGLEMAIL.COM.	10	IN	A	74.125.200.26
ASPMX3.GOOGLEMAIL.COM.	164	IN	AAAA	2404:6800:4003:c00::1a
dns1.fct.unl.pt.	9891	IN	A	193.136.126.101
dns2.fct.unl.pt.	3185	IN	A	193.136.126.102

;; Query time: 26 msec

;; SERVER: 192.168.1.254#53(192.168.1.254)

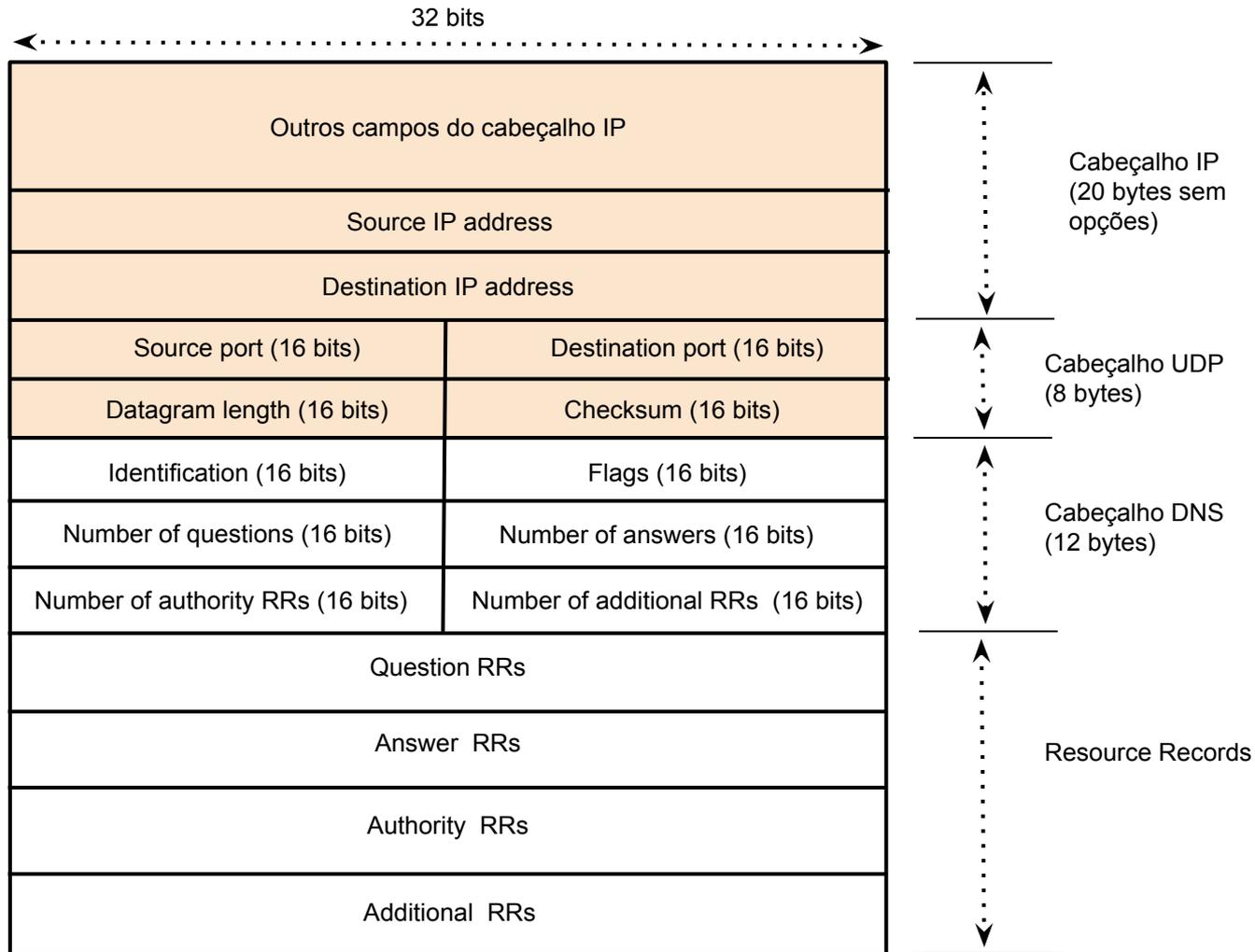
;; WHEN: Sun Oct 30 12:30:19 2016

;; MSG SIZE rcvd: 423

Mensagens do protocolo

- É um protocolo cliente / servidor
- Utiliza UDP
- É pressuposto o cliente repetir o pedido se não obtém resposta
- Após um certo número de tentativas falhadas dirige o pedido para um servidor alternativo (por isso uma resposta pode levar um tempo significativo quando a rede está sobrecarregada)
- As mensagens de pedido e resposta têm o mesmo formato

Formato das Mensagens DNS



Formato das mensagens

- **Identification**: colocada pelo cliente para permitir relacionar pedidos com respostas
- **Flags**:
 - Query or reply
 - Reply is authoritative
 - Recursion desired
- **questions** = query
- **answers** = replies
- **authority** = RRs de servidores com autoridade sobre as respostas
- **additional info** = informações suplementares que podem ser úteis ao cliente

Exemplo de criação de um domínio

- A existência do novo domínio é contratado com um registrar (e.g. shoppingheaven.com com a verisign.com)
- Para esse efeito são montados dois novos servidores e os seus RRs são colocados no domínio .com
 - (shoppingheaven.com, NS, dns1.shoppingheaven.com)
 - (dns1.shoppingheaven.com, A, 212.212.212.1)
 - (shoppingheaven.com, NS, dns2.shoppingheaven.com)
 - (dns2.shoppingheaven.com, A, 211.211.211.2)
- Na tabela de servidor primário de utopia.com são inseridos os RRs para os nomes do domínio
 - (mail.shoppingheaven.com, A, 212.212.212.2)
 - (shoppingheaven.com, MX, 10 mail.shoppingheaven.com)

Para interrogar o DNS

- Usar o comando DIG

```
dig [@server name] name [RR type or ANY]
```

- Usar o comando nslookup

```
nslookup [ name ]
```

- Ver o ficheiro /etc/resolv.conf em Unix*

Conclusões

- Para designar, localizar, recursos numa rede ou num sistema distribuído usam-se nomes, identificadores e endereços
- Nomes, identificadores e endereços desempenham papéis diferentes e têm características distintas
- Por vezes é necessário realizar a tradução de nomes em endereços, de nomes em identificadores, de endereços noutros endereços, etc.
- Para esse efeito usam-se catálogos, *directories or registries*
- O DNS é um exemplo de *directory ou registry* distribuída