Administração de Redes 2019/20

Domain Name System (DNS)

Motivação

- Máquinas trabalham bem com endereços IP
- Pessoas trabalham melhor com nomes
  - Ninguém quer ter que saber que o servidor web da UP é o 193.137.55.13 e o do jornal Público o 195.23.42.21
  - Mas é fácil saber que o primeiro é *www.up.pt* e o segundo é *publico.pt*
- Necessária infraestrutura para tradução entre nomes e endereços IP
  - Originalmente mantinham-se as traduções no ficheiro `hosts`
  - Na Internet actual, tal seria impensável
    - Embora o ficheiro `hosts` continue a existir e a ser consultado
- Domain Name System (DNS)
  - Serviço de tradução de nomes
  - Base de dados distribuída, hierárquica, frouxamente coerente, escalável, fiável e dinâmica

Objectivos do DNS

- Espaço de nomes global, escalável e consistente
- Controlo local sobre recursos locais
  - E.g., deve ser possível gerir nomes no DCC sem envolver a Reitoria (UP)
- Sistema distribuído para evitar pontos singulares de falha ou de estrangulamento
- Universalidade de aplicação
  - Além da tradução de nomes, descoberta de servidores de email, etc.
- Suporte para múltiplos protocolos
  - Permite traduzir nomes para outros endereços além dos IP
- Suporte por qualquer hardware
  - Tanto supercomputadores como pequenos sistemas embutidos podem usar o DNS

Pressupostos no *design* do DNS

- Crescimento rápido da base de dados
- Taxa de modificação variável
  - Algumas zonas são muito estáveis enquanto outras estão constantemente a ser actualizadas
- Importância do acesso à informação de nomes
  - É preferível ter informação ligeiramente desactualizada a não ter nenhuma informação
- Responsabilidade organizacional delegável
- Tratamento de pedidos para os quais não há informação local
  - Recorrer a outros servidores para obter a resposta e devolvê-la
  - Devolver referência a outro servidor de nomes
  - Erro…
- Uso intensivo de *caching* para desempenho e escalabilidade

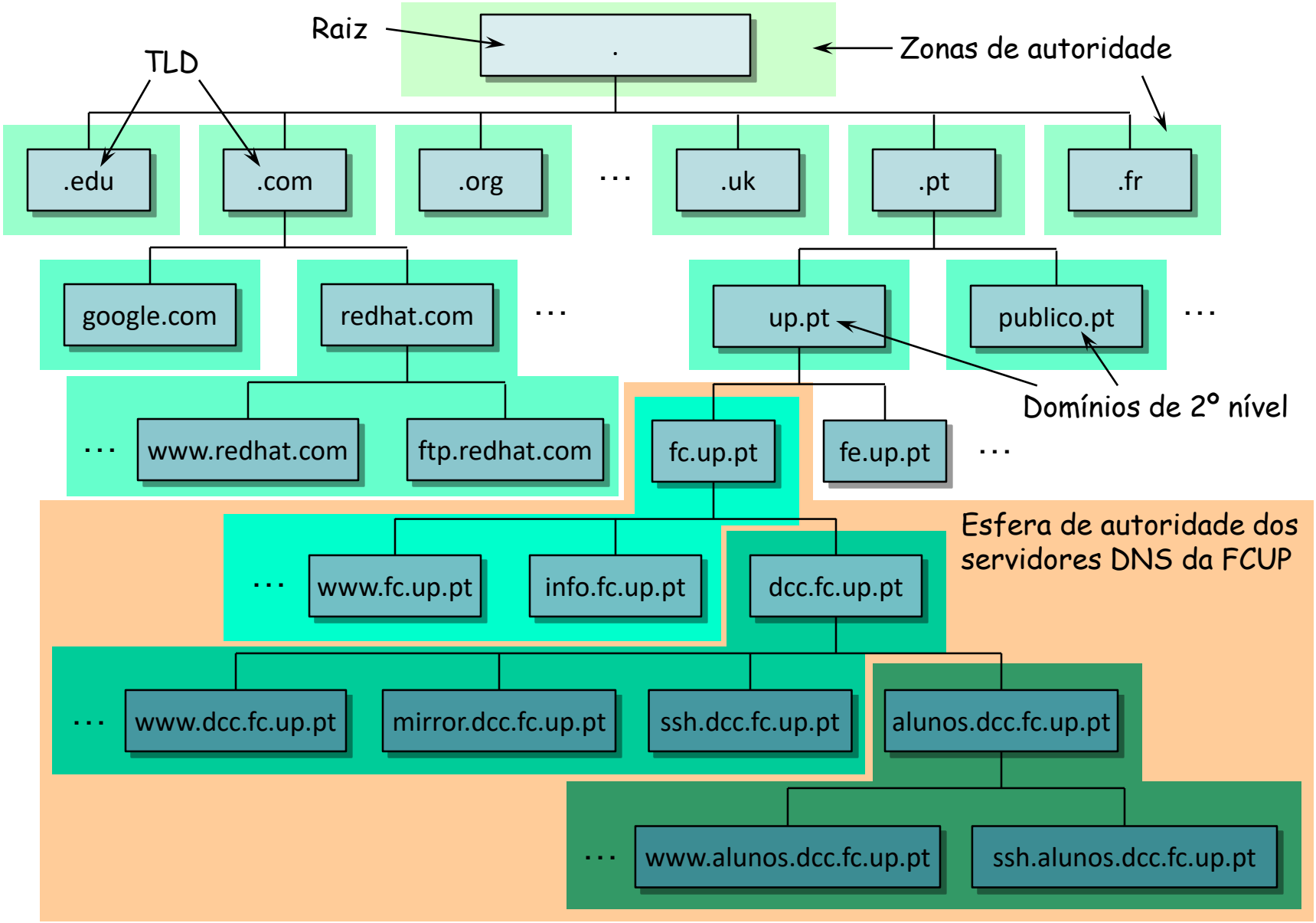
Alguma terminologia

- Zona de autoridade
  - Secção da base de dados correspondente a um subespaço do espaço de nomes do DNS mantida por uma determinada entidade
- Servidor autoritário
  - Servidor responsável por uma determinada zona de autoridade, com informação fidedigna sobre essa zona
- Delegação
  - Passagem de autoridade sobre um subespaço de nomes (zona) a outro(s) servidor(es)
- Servidor de raiz
  - Servidor autoritário para a zona "." (raiz da árvore hierarquica de nomes do DNS)
- Servidor de topo (TLD — *Top Level Domain*)
  - Servidor autoritário para um domínio com um só identificador (e.g., ".com"), imediatamente abaixo dos servidores de raiz
  - Domínios de topo podem ser organizacionais (e.g., ".com") ou geopolíticos (e.g., ".pt")

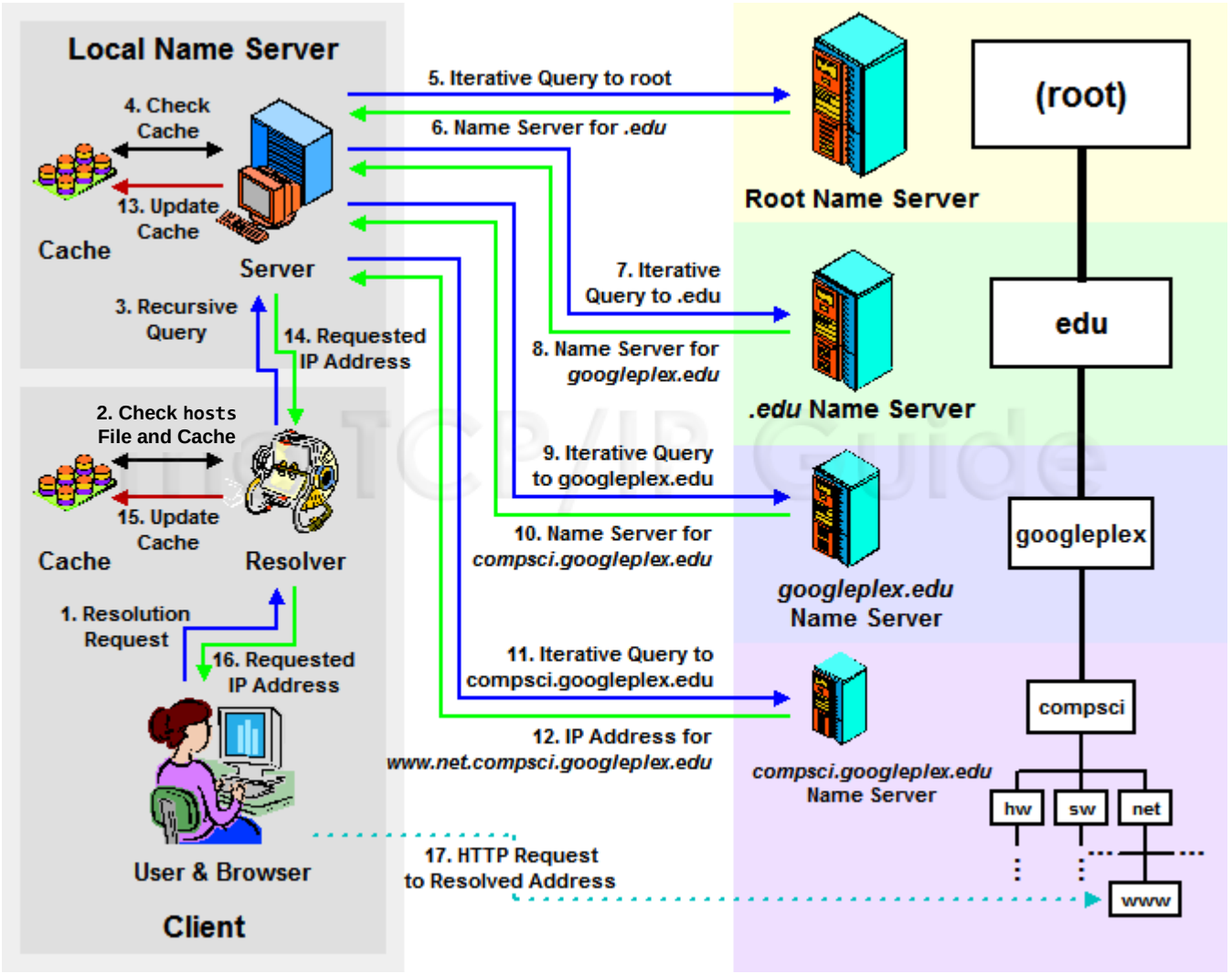
Alguma terminologia

- Nome de domínio completamente qualificado (FQDN)
  - Nome especificando o caminho completo até à raiz da hierarquia de nomes (e.g., "www.dcc.fc.up.pt."; frequentemente omite-se o último ponto)
- Nome parcialmente qualificado
  - Nome relativo a uma dada zona (e.g., "www.dcc")
- *Resolver*
  - Serviço ou biblioteca capaz de usar o DNS para traduzir nomes ("cliente" DNS)
- Servidor de nomes local
  - Servidor DNS indicado pelo servidor DHCP (ou configurado estaticamente) que actua como proxy para a resolução de nomes (i.e., aceita pedidos recursivos)
- Pedido recursivo
  - Pedido cuja resposta tem que ser a resolução pedida ou um erro
- Pedido não-recursivo
  - Pedido que pode também ter como resposta a referência a outro servidor DNS (para resolução iterativa)

Hierarquia de nomes e zonas



Exemplo de resolução de nome



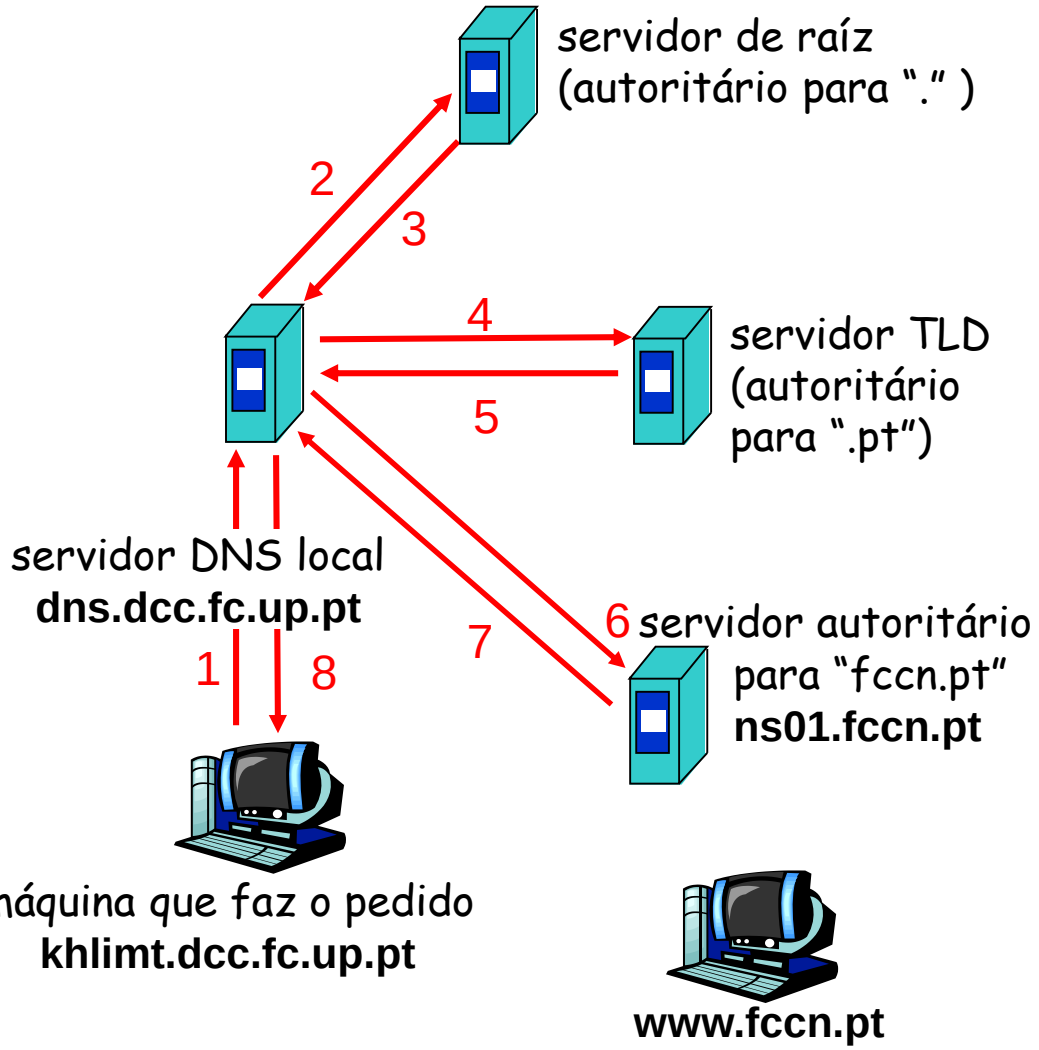
Exemplo: resolução do nome *www.net.compsci.googleplex.edu*

Resolução iterativa e recursiva

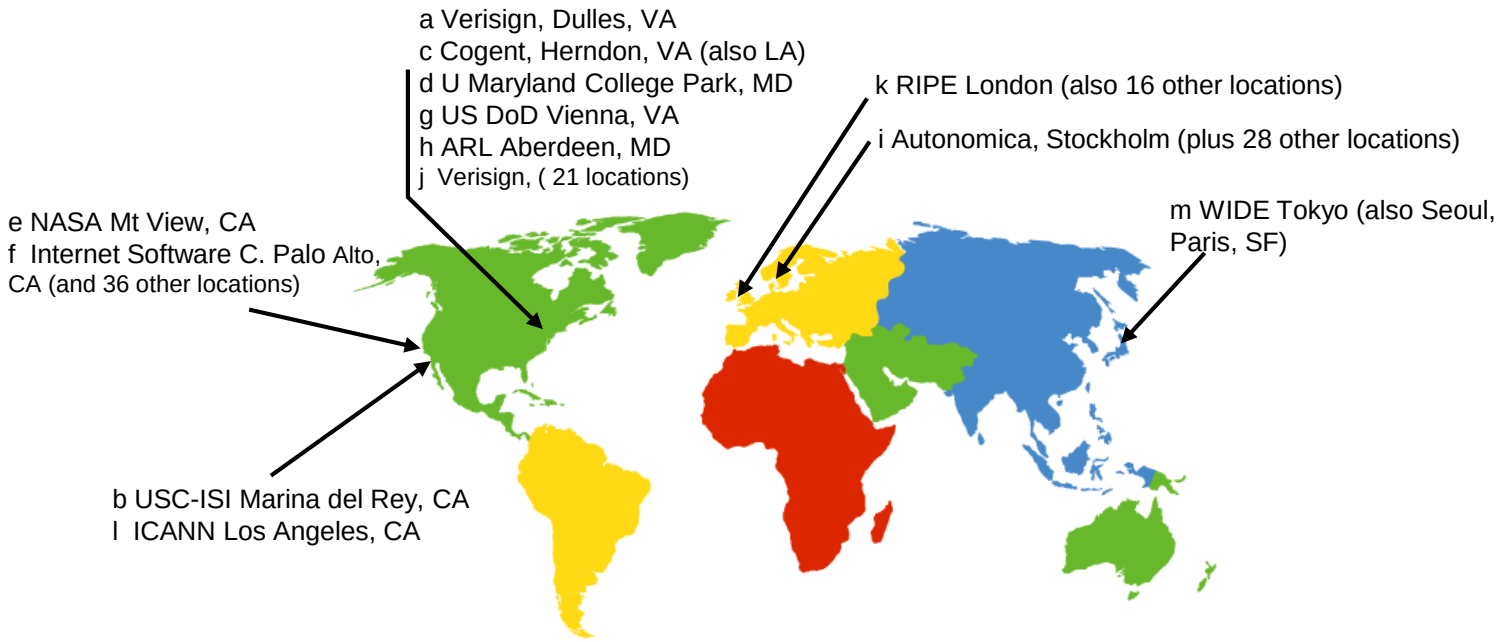
- Resolução iterativa
  - Cliente faz pedido não-recursivo para resolução de um nome
  - Se o servidor tiver a resposta, devolve-a
  - Se não tiver mas for relativa a um subespaço do espaço de nomes para o qual é autoritário, devolve uma referência
    - Indicação de outro servidor DNS em quem foi delegada autoridade sobre um subespaço
    - Cliente consulta esse servidor e repete procedimento até obter resolução (ou erro)
  - Caso contrário, devolve um erro
- Resolução recursiva
  - Cliente faz pedido recursivo e servidor devolve resolução ou erro
    - Eventualmente consultando outros servidores
  - Consome recursos do servidor
    - Pedidos recursivos normalmente autorizados apenas a máquinas locais

Caso normal de resolução de nomes

- Terminal faz pedido recursivo ao servidor de DNS local (1 e 8)
- Servidor de DNS local resolve iterativamente o nome (2 a 7)
- Normalmente, um servidor DNS só aceita pedidos recursivos feitos por máquinas locais



Servidores de raiz



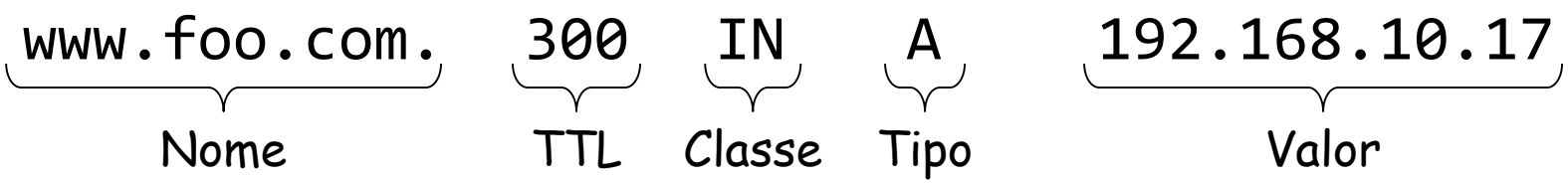
- 13 servidores de raiz espalhados pelo mundo
- Alguns deles replicados em diversas localizações geográficas (*anycast*)
- Devolvem referências a servidores TLD
  - Quase sempre em cache nos servidores de nomes locais ➔ não são necessários muitos servidores de raiz para toda a Internet

Tipos de servidor

- Master
  - Servidor autoritário para uma ou mais zonas
  - Base de dados dessa(s) zona(s) configuradas em ficheiros locais
- Slave
  - Servidor autoritário para uma ou mais zonas
  - Base de dados dessa(s) zona(s) obtida do *master* por transferência de zona (AXFR)
- Caching
  - Aceita pedidos recursivos e faz resolução iterativa para os completar
  - Faz *caching* dos resultados obtidos para acelerar consultas futuras
- Forwarding
  - Semelhante ao *caching*, mas em vez de resolução iterativa faz pedidos recursivos a outro(s) servidor(es) configurado(s)

Registos de recurso (RR)

- Os RR são a unidade básica de dados no DNS
  - O DNS é uma base de dados de registos de recurso
- Associam um nome a um valor de um determinado tipo
  - Valor (também designado RDATA) pode ser composto
- Têm ainda
  - Um tempo de vida (TTL) que indica o prazo de validade em cache
  - Uma classe, quase sempre `IN` (significando Internet)
- Exemplo:



Address (A)

- Define o endereço IP para a máquina com um dado nome

- Exemplo:

```
www.example.com.          A        172.17.2.12
```

  - Neste exemplo omite-se
    - A classe (assume-se IN)
    - O TTL (assume-se o valor definido na directiva $TTL)

Pointer (PTR)

- Usado para resolução inversa†
- Para obter o nome da máquina com um dado endereço IP (e.g., 172.17.2.12) faz-se o seguinte:
  1. Invertem-se os octetos do endereço ➔ 12.2.17.172
  2. Pede-se o registo PTR para o endereço invertido acrescido de `.in-addr.arpa.`

```
12.2.17.172.in-addr.arpa.          PTR      www.example.com.
```

†*Reverse mapping* (também existiram *inverse queries*, mas são obsoletas)

Name Server (NS)

- Identifica um servidor DNS autoritário para o domínio
  - Múltiplos registos NS se houver mais do que um
- Necessário para delegar um subespaço de nomes noutro servidor
- Exemplos (na zona example.com.):

```
@     NS    dns1.example.com.     ; example.com.  NS  dns1.example.com.
      NS    dns2                  ; example.com.  NS  dns2.example.com.
; ...
sub   NS    dns.sub.example.com.  ; Delega a zona sub.example.com. no
                                  ; servidor dns.sub.example.com.
; ...
```

Mail Exchange (MX)

- Permite descobrir o servidor de email para um dado domínio
  - Usado pelos servidores SMTP no envio de email
- O valor consiste em
  - Uma preferência (mais baixo = preferido)
  - O nome do servidor
- Exemplo:

```
example.com.    MX  10  smtp.example.com. ; Servidor principal
                MX  20  mail.other.net.   ; Servidor de backup
```

Canonical Name (CNAME)

- Define um pseudónimo para uma máquina
- O nome real (canónico) pode estar fora da zona
- Exemplos:

```
ftp.example.com.    CNAME    www.example.com. ; Servidor ftp e web
foo.example.com.    CNAME    abc.other.net.   ; Fora da zona
```

- Outro exemplo de como configurar o DNS para aceder indistintamente ao web site com http://example.com/ ou http://www.example.com/

```
example.com.        A        172.18.75.11  ; www.example.com. é um pseudó-
www.example.com.    CNAME    example.com.  ; nimo para example.com.
```

Start Of Authority (SOA)

- Define parâmetros gerais para a zona
  - Servidor DNS principal *(primary master)*
  - Endereço de email do administrador da zona
    - Substituindo o "@" por "." para ser um nome DNS válido
  - Número de série
    - Normalmente no formato *AAAAMMDDVV*
      - Ano, mês, dia, número da alteração nesse dia
  - Período de refrescamento (transferência de zona para os *slaves*)
  - Período para nova tentativa se falhar a transferência de domínio
  - Período de expiração da zona, após o qual um *slave* deixa de ser autoritário para esta zona (se não a conseguir refrescar)
  - Período para *caching* negativo
    - I.e., da indicação de que um dado nome não existe nesta zona
- É sempre o primeiro registo de qualquer zona

Start Of Authority (SOA)

- Exemplo:

Servidor de DNS principal (primary master)

O email do administrador desta zona é hostmaster@example.com

```
example.com.    IN    SOA   ns.example.com. hostmaster.example.com. (
                              2010080800 ; sn = número de série
                              172800     ; ref = refrescamento = 2d
                              900        ; ret = nova tentativa = 15m
                              1209600    ; ex = expiração = 2w
                              3600       ; nx = nxdomain ttl = 1h
                             )
```

Zona

- Quando passa o período de refrescamento, o *slave* pede novamente o SOA ao *master*
  - Se o número de série for o mesmo, não houve alterações à zona
  - Se for diferente, é necessário transferir novamente a zona
    - Pedido AXFR (zona completa) ou IXFR (transferência incremental)

Outros tipos de RR

- Além dos referidos, existem outros tipos de RR
  - AAAA — Semelhante ao A mas para endereços IPv6
  - HINFO — Informação sobre uma máquina (*hardware* e S.O.)
  - SRV — Permite descobrir o servidor responsável por um dado serviço
    - Pode ser visto como uma generalização do MX
  - NAPTR — Usado para descoberta de serviços
    - E.g., em telefonia SIP *(Session Initiation Protocol)*, juntamente com o SRV
  - TXT — Informação textual genérica associada a um dado nome
    - Usado, e.g., para autenticação de email (SPF ou DKIM)
  - DNAME — Semelhante ao CNAME, mas para redireccionar todos os nomes de um domínio
    - Directa ou indirectamente abaixo dele
    - Gera também registos CNAME implícitos para nomes pedidos
  - Etc.

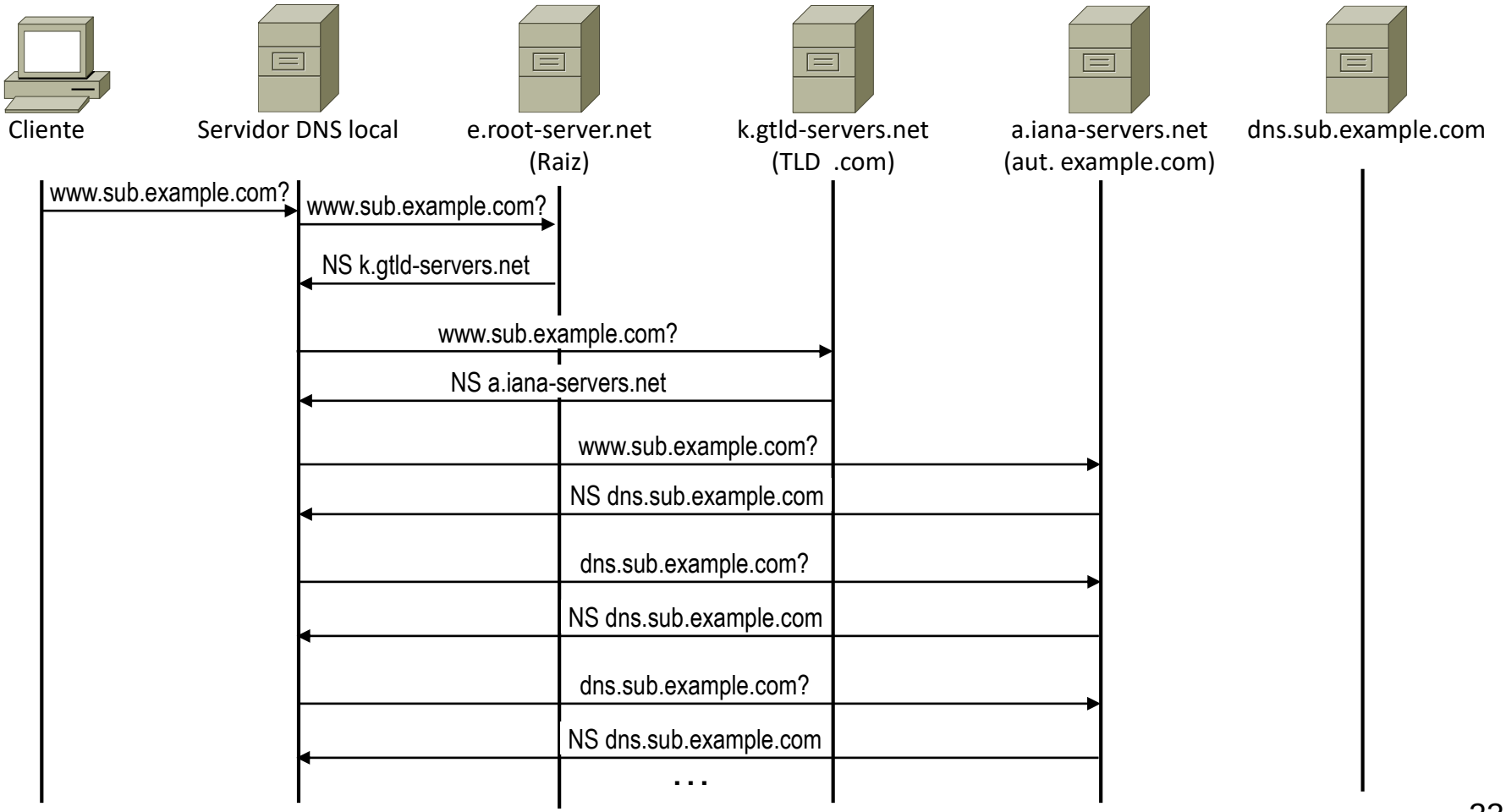
Delegação

- Delegação é a passagem de autoridade sobre um subespaço de nomes a outro(s) servidor(es) DNS
  - Mecanismo fundamental para descentralizar a gestão dos domínios
- Faz-se criando registos NS
  - Por exemplo, para delegar o subdomínio *sub.example.com* no servidor *dns.other.net* cria-se o registo

    ```
    sub.example.com.    NS    dns.other.net.
    ```
- Uma delegação cria uma nova zona
  - Mesmo que seja para o próprio servidor DNS
- Esfera de autoridade *(bailiwick)* de um servidor DNS
  - É o subespaço de nomes oficialmente delegado nesse servidor
    - Domínio oficialmente delegado (incluindo todos os subdomínios)
  - Obtém-se descendo a hierarquia desde a raiz no âmbito da resolução de um nome

Delegação num servidor pertencente ao subespaço delegado

Exemplo: delegação de *sub.example.com* no *dns.sub.example.com*



Registo-cola (*glue record*)

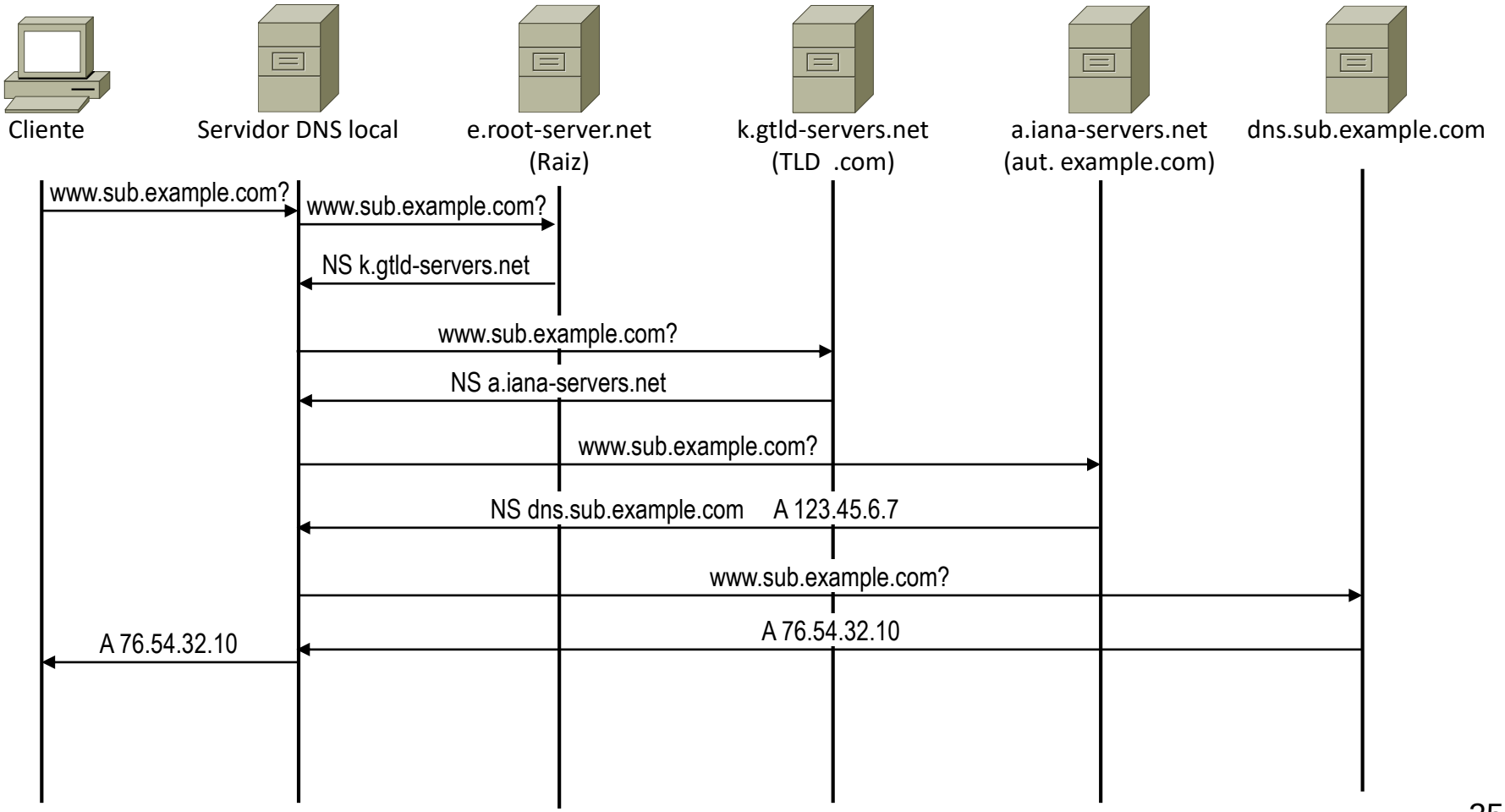
- Problema
  - Para resolver *www.sub.example.com* é preciso consultar *dns.sub.example.com*
  - Mas para isso é preciso resolver *dns.sub.example.com*
  - O que implica consultar *dns.sub.example.com*
  - Dependência cíclica!!!
- Solução
  - Juntamente com o registo NS que remete a pergunta sobre *www.sub.example.com* para *dns.sub.example.com*, enviar um registo A com o endereço IP desse servidor ➔ Registo-cola

```
sub.example.com.        NS    dns.sub.example.com.
dns.sub.example.com.    A     147.24.112.3
```

  - Registo A está fora da zona (mas dentro da esfera de autoridade) de quem delega
  - Desnecessário para delegar num servidor cujo nome não está debaixo do subdomínio delegado
    - O seu endereço IP pode obter-se normalmente, sem dependência cíclica

Delegação num servidor pertencente ao subespaço delegado

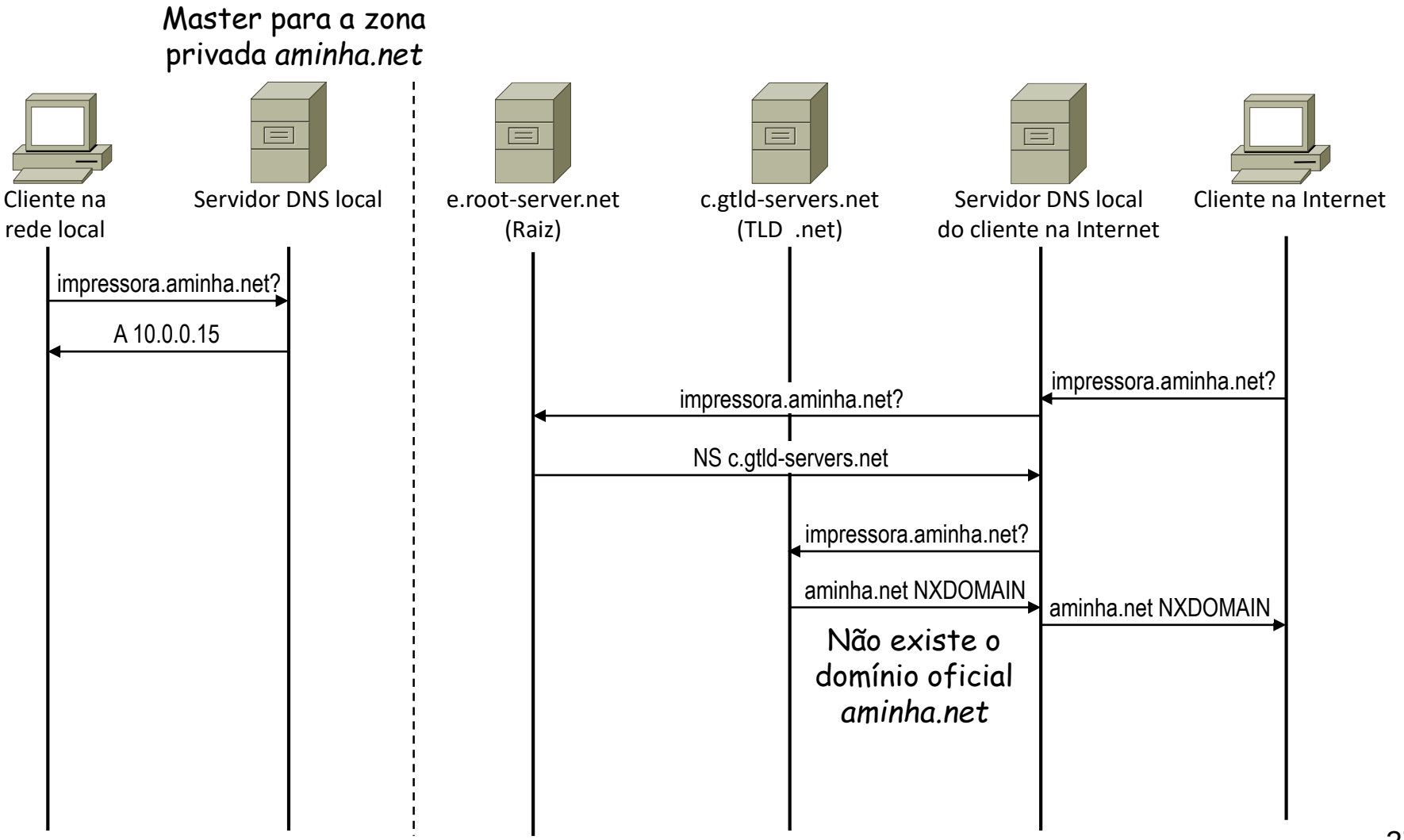
O mesmo exemplo, mas com utilização de registo-cola



Zonas privadas

- É possível configurar num servidor DNS zonas que não lhe foram (oficialmente) delegadas
  - Não fazem parte da esfera de autoridade desse servidor
- São visíveis para quem consultar esse servidor para resolver nomes dessa zona
  - E.g., se além de *master* for também servidor de nomes local
- Não são visíveis para a Internet em geral
- Se a zona existir na hierarquia oficial (na esfera de autoridade de outro servidor) há conflito
  - Máquinas na rede local não conseguem resolver nomes da zona oficial

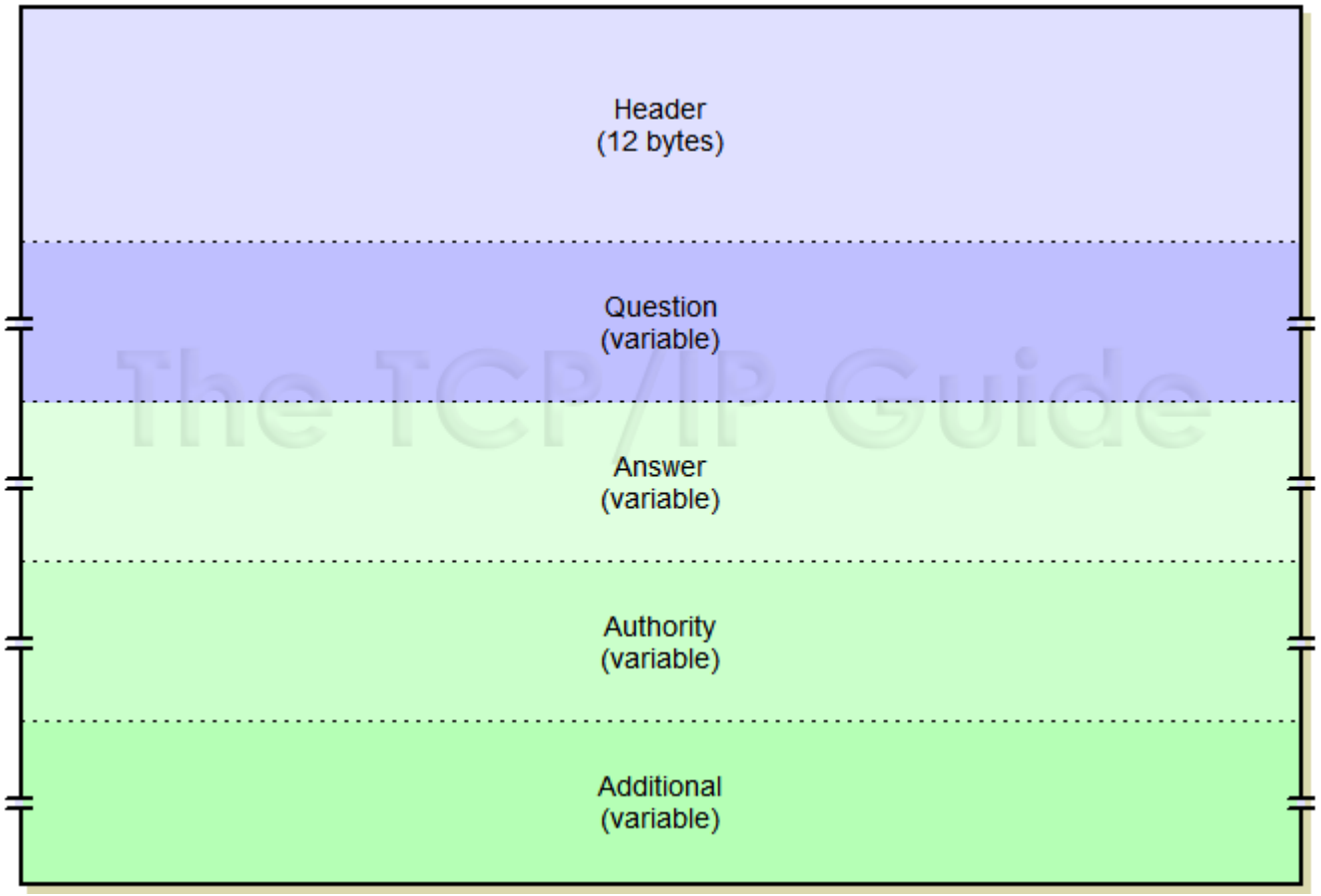
Zonas privadas



Protocolo

- Mensagens DNS normalmente encapsuladas em UDP
  - Normalmente pedidos e respostas curtas, cabem num único pacote
  - Estabelecimento de conexão TCP seria um desperdício
- Mensagens limitadas a 512 bytes
  - Se uma resposta for mais longa, é truncada
    - Indicação do facto numa *flag*
  - O cliente pode optar por repetir o pedido sobre TCP
- Transferências de zona (entre *master* e *slave*) sempre sobre TCP
  - Potencialmente grande volume de informação
  - Necessária fiabilidade
  - Mensagens precedidas por um número de 16 bits indicando o tamanho das mesmas (TCP não faz delineação de mensagens)
- Servidor DNS na porta 53, tanto UDP como TCP

Formato das mensagens



Formato das mensagens

Question section

- Contém pedido para um ou mais registos (normalmente apenas um)

Answer section

- Contém os registos pedidos (resposta ao que foi pedido)

Authority section

- Contém registos NS indicando servidores autoritários sobre as zonas dos registos pedidos
  - Indicação ou delegações de autoridade

Additional section

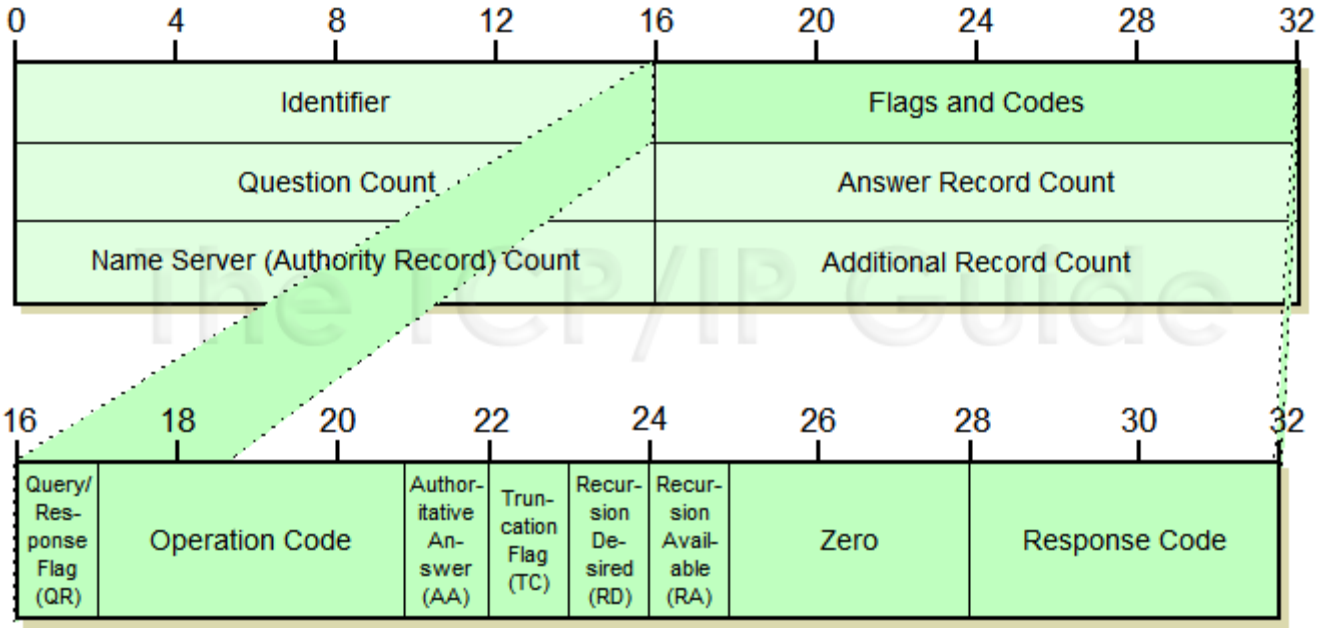
- Contém registos não pedidos mas que podem ser úteis
  - E.g., registos A e/ou AAAA para os servidores da *authority section* (registos-cola) ou para o nome correspondente a um MX pedido

Exemplo de resposta DNS

```
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 7908
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 3, ADDITIONAL: 6
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;www.up.pt.                     IN      A

;; ANSWER SECTION:
www.up.pt.              86400   IN      A       193.137.55.13

;; AUTHORITY SECTION:
up.pt.                  86400   IN      NS      dns3.up.pt.
up.pt.                  86400   IN      NS      dns4.up.pt.
up.pt.                  86400   IN      NS      dns1.up.pt.

;; ADDITIONAL SECTION:
dns1.up.pt.             86400   IN      A       193.137.55.20
dns1.up.pt.             86400   IN      AAAA    2001:690:2200:a10::20
dns3.up.pt.             86400   IN      A       193.137.35.100
dns3.up.pt.             86400   IN      AAAA    2001:690:2200:b10::100
dns4.up.pt.             86400   IN      A       193.136.37.10
dns4.up.pt.             86400   IN      AAAA    2001:690:2200:910::10
```

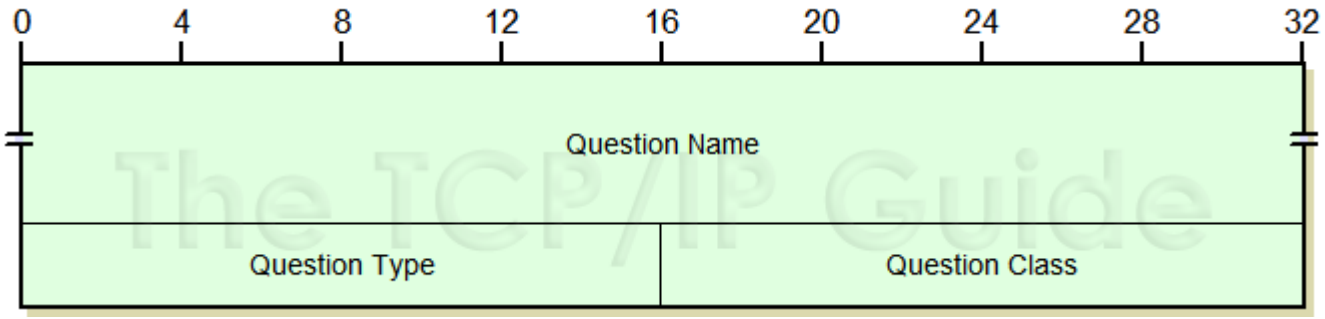
- Representação textual (saída do comando `dig`)
  - A mensagem propriamente dita usa formato binário

Cabeçalho das mensagens



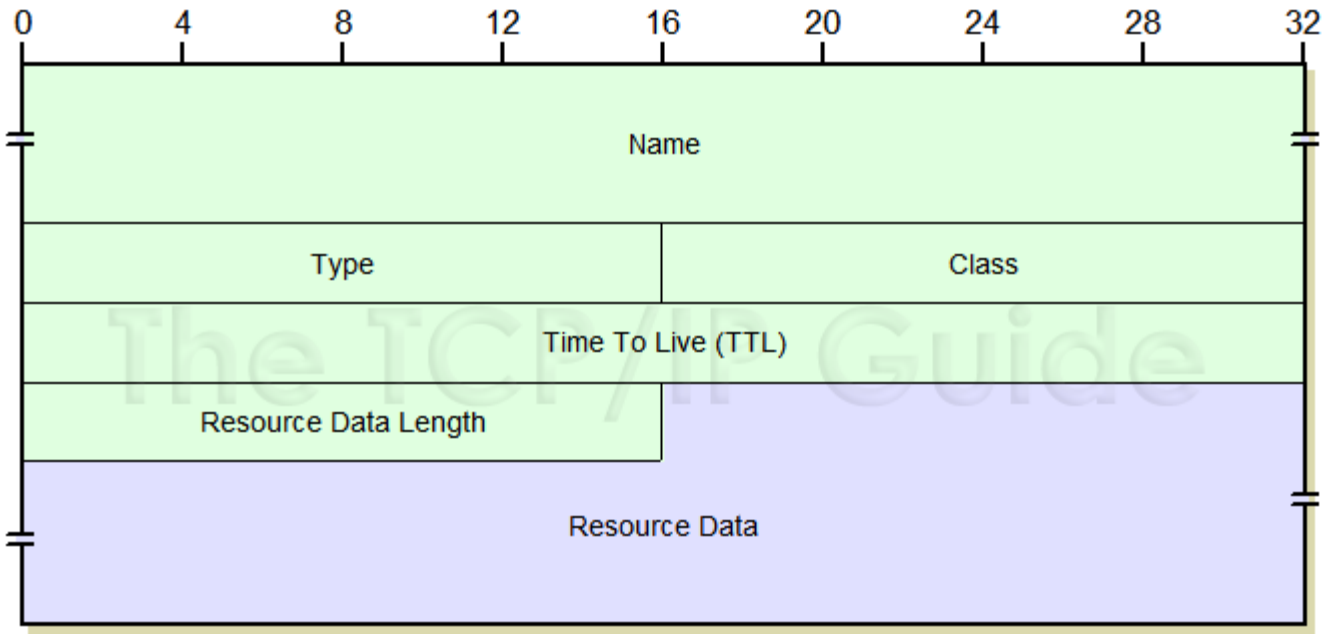
- *Identifier* permite associar respostas a pedidos
  - Necessário porque o UDP é *connectionless*
- *Operation code* indica tipo de operação, e.g., QUERY (0)
- *Response code* indica sucesso (0) ou erro (>0)
  - E.g., nome inexistente (3), recusado (5), sem autoridade (9), …

Formato das questões (*query section*)

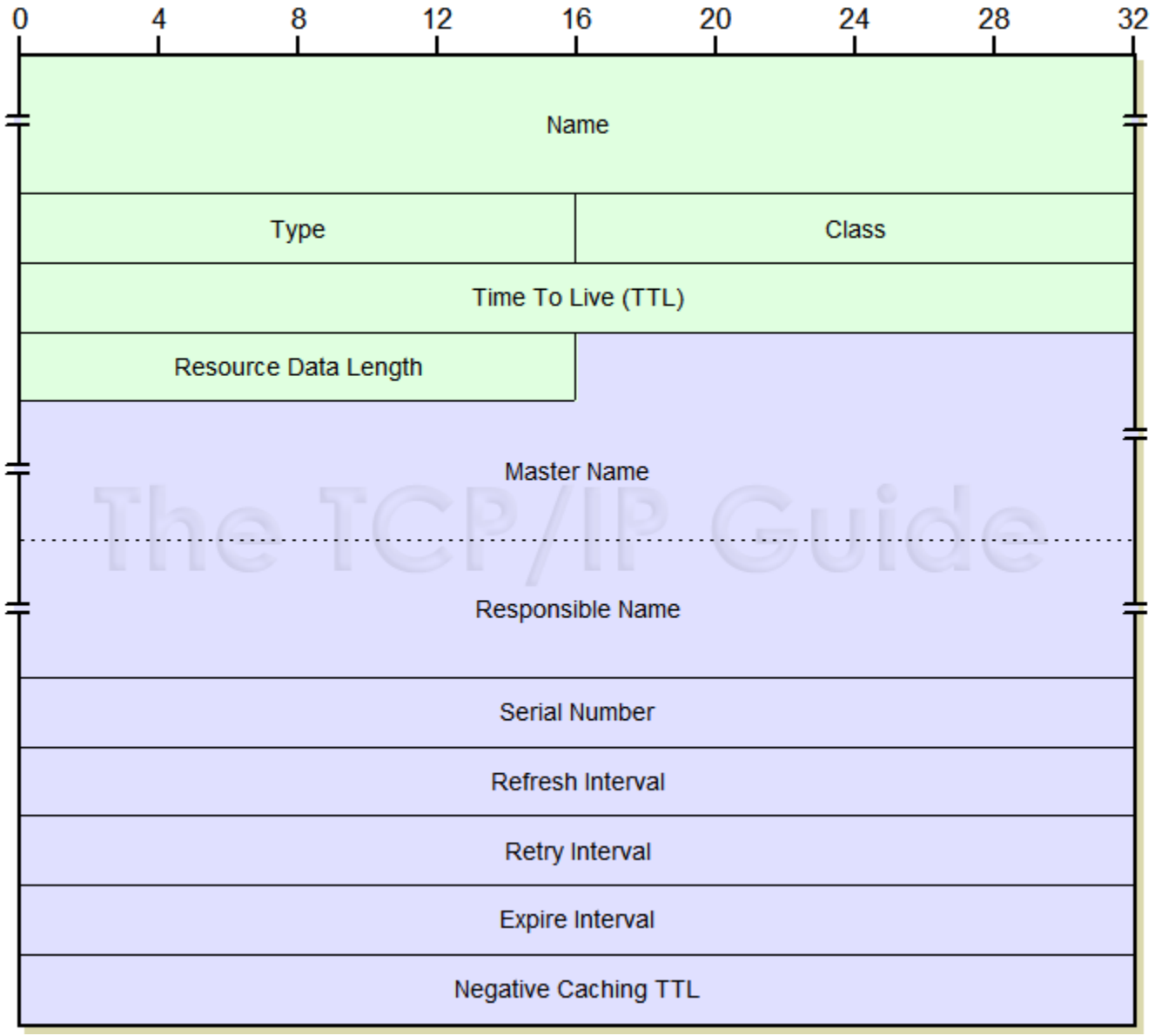


- Codificação dos nomes
  - Cada um dos identificadores que compõem um nome é codificado como comprimento (binário, 1 byte) + valor
  - Exemplo: *www.example.com* é codificado como [3]www[7]example[3]com[0]
  - É possível incluir apontadores como forma de poupar espaço na representação de domínios que aparecem repetidos (compressão)
- Question type indica o tipo de RR pedido
  - Ou então AXFR, IXFR, ou * (para pedir todos os registos com um dado nome)

Formato genérico dos registos (outras secções)



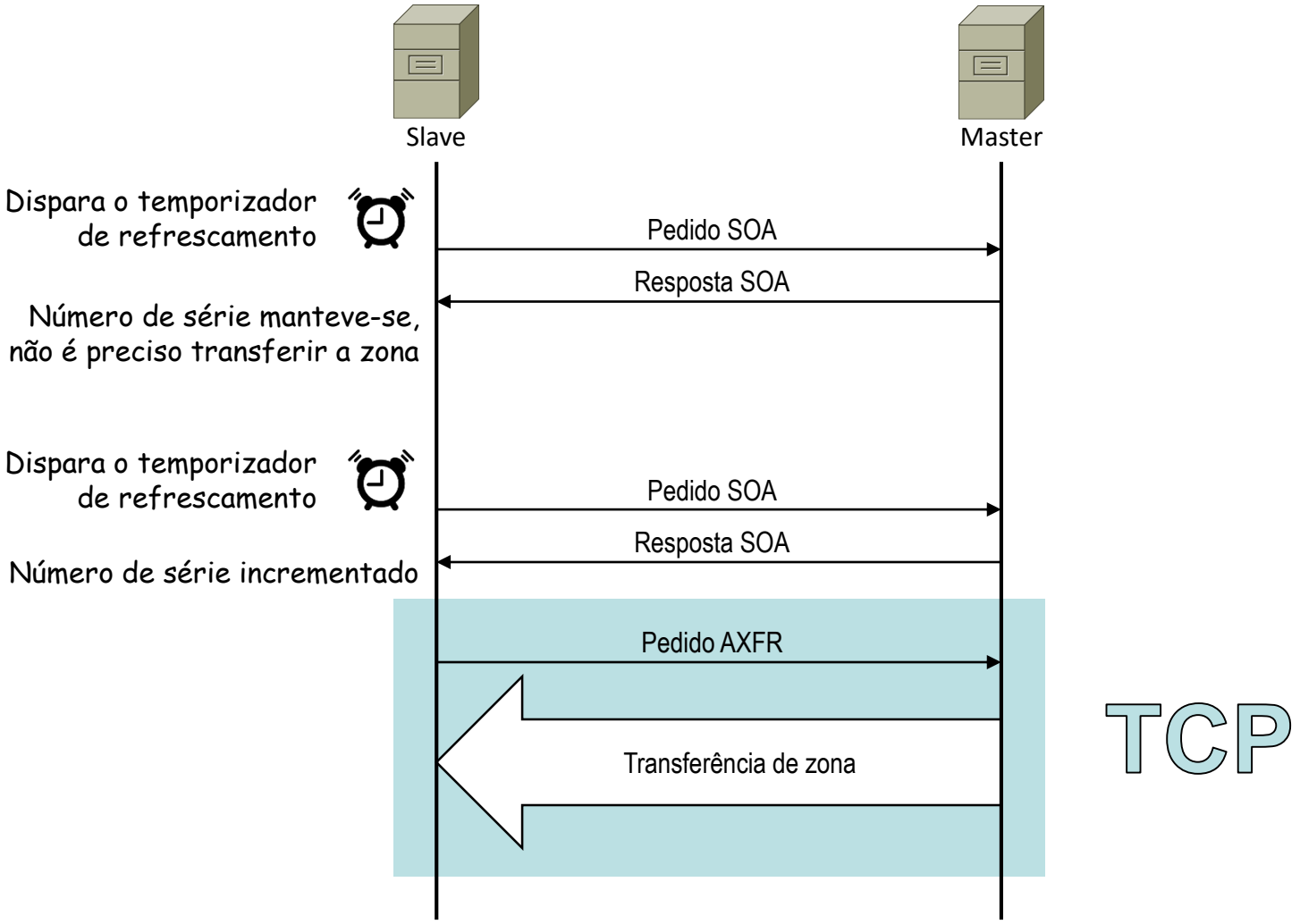
Exemplo do formato de um registo (SOA)



Transferências de zona

- Os servidores *slave* duma zona obtêm a informação dos *master* através da transferência de zona
  - AXFR (*Authority Transfer*)
- Quando passa o período de refrescamento, o *slave* pede novamente o *SOA* ao *master*
  - Normalmente, começa por pedir o *SOA* da zona e comparar o número de série com o que tem actualmente
  - Se o número de série for o mesmo, não houve alterações à zona
  - Se for diferente, é necessário transferir novamente a zona
- A transferência de domínio propriamente dita é feita sobre TCP
  - Potencialmente grande volume de informação, necessária fiabilidade
  - Resposta pode consistir em mais do que uma mensagem DNS
- Se este procedimento falhar, o *slave* volta a tentar mais tarde *(retry)*
- Uma alternativa ao AXFR é ter todos os servidores como *master* e transferir os ficheiros de zona por meios externos ao DNS
  - E.g., rsync

Transferência de zona



Transferência de zona: exemplo

Pedido

Header	OPCODE=QUERY
Question	QNAME=example.com., QCLASS=IN, QTYPE=AXFR
Answer	–
Authority	–
Additional	–

Resposta

Header	OPCODE=QUERY
Question	QNAME=example.com., QCLASS=IN, QTYPE=AXFR
Answer	example.com. IN SOA serial=3 ... example.com. IN NS ns.example.com. ns.example.com. IN A 10.0.0.1 www.example.com. IN A 10.0.3.1 www.example.com. IN A 10.0.2.1 example.com. IN SOA serial=3 ...
Authority	–
Additional	–

Transferências incrementais

- Qualquer alteração a uma zona obriga a actualizar o nº de série
  - Mesmo que afecte um único registo
- Um *slave* que veja o novo nº de série faz nova transferência
- Muito ineficiente se a zona tiver muitos registos e poucos deles tiverem sido alterados
- Solução: transferências incrementais (IXFR)
  - *Slave* faz pedido IXFR contendo o SOA que tem (na secção *authority)*
  - O servidor responde com alterações incrementais
    - Começa com o SOA da versão actual
    - Depois o SOA de cada versão antiga envolvida, seguido dos registos alterados
    - Termina com o SOA da versão actual
  - É possível condensar alterações sucessivas
  - Se o *master* não conseguir determinar as alterações, responde com a zona completa

Transferência incremental: exemplo

Pedido

Header	OPCODE=QUERY
Question	QNAME=example.com., QCLASS=IN, QTYPE=IXFR
Answer	–
Authority	example.domain. IN SOA serial=1
Additional	–

Resposta incremental

Header	OPCODE=QUERY
Question	QNAME=example.com., QCLASS=IN, QTYPE=IXFR
Answer	example.com. IN SOA serial=3 ... example.com. IN SOA serial=1 ... ftp.example.com. IN A 10.0.1.1 example.com. IN SOA serial=2 ... **www.example.com. IN A 10.0.1.2** **www.example.com. IN A 10.0.2.1** example.com. IN SOA serial=2 ... www.example.com. IN A 10.0.1.2 example.com. IN SOA serial=3 ... **www.example.com. IN A 10.0.3.1** example.com. IN SOA serial=3 ...
Authority	–
Additional	–

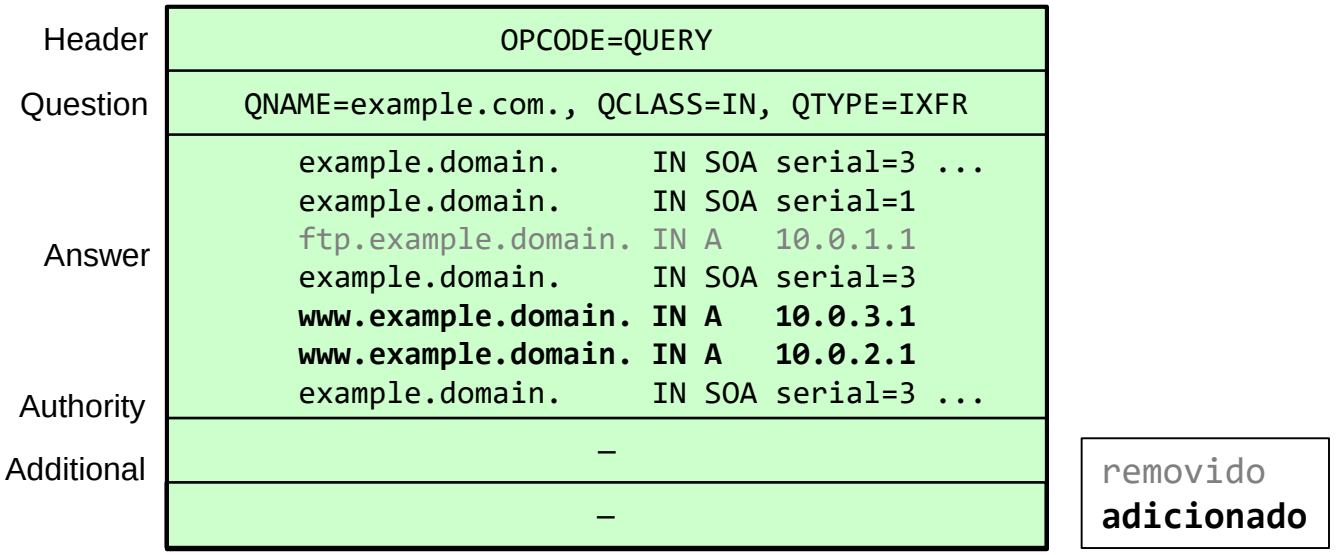
Alterações da v1 para a v2

Alterações da v2 para a v3

removido
adicionado

Transferência incremental: exemplo

Resposta incremental condensada



- Resposta com zona completa seria igual à do AXFR
  - Excepto campo QTYPE

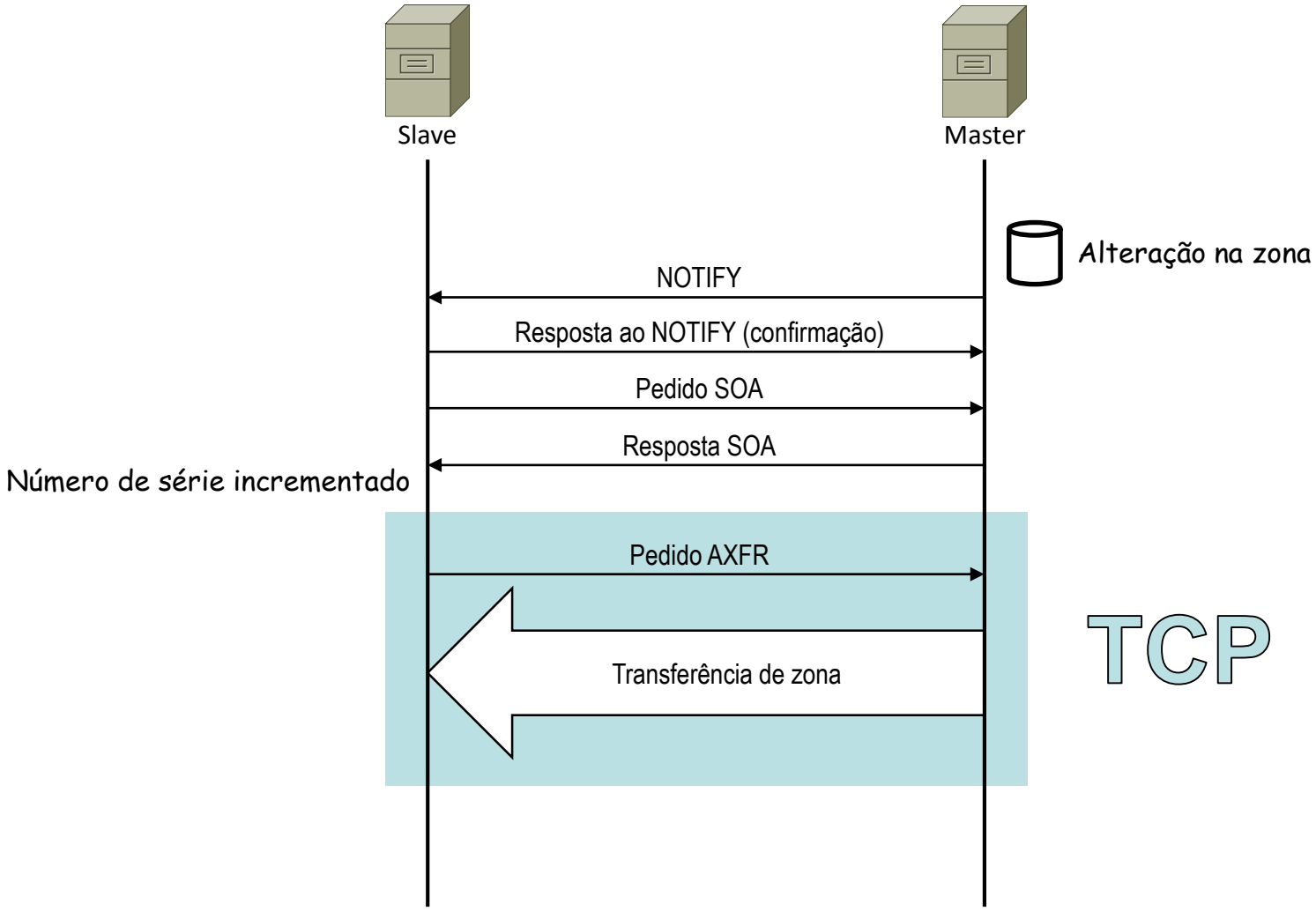
Notificações

- Originalmente, os *slaves* verificavam alterações à zona no *master* fazendo *polling*
- Este processo pode tornar lenta a propagação de alterações
- Mecanismo de notificação acelera o processo
- Quando a zona muda, o *master* envia pedido NOTIFY aos *slaves*
  - Indica SOA e zona na secção *questions*
  - Pode também incluir o registo SOA actualizado na secção *answers*
- Slaves (potenciais) podem obter-se
  - Dos registos NS da zona
    - Exceptuando o próprio servidor e o *primary master*
  - Por configuração explícita
- A recepção de um pedido NOTIFY é confirmada
  - Até lá, o *master* pode retransmitir o pedido

Notificações

- Se o pedido NOTIFY incluir o registo SOA
  - *Slave* verifica número de série
  - Se for diferente do que tem, pede transferência de domínio
    - Completa ou incremental
- Se não incluir
  - *Slave* procede como se o período de refrescamento tivesse terminado
    - Pede SOA
    - Compara número de série
    - Se necessário, pede transferência de domínio

Notificações



Actualizações dinâmicas

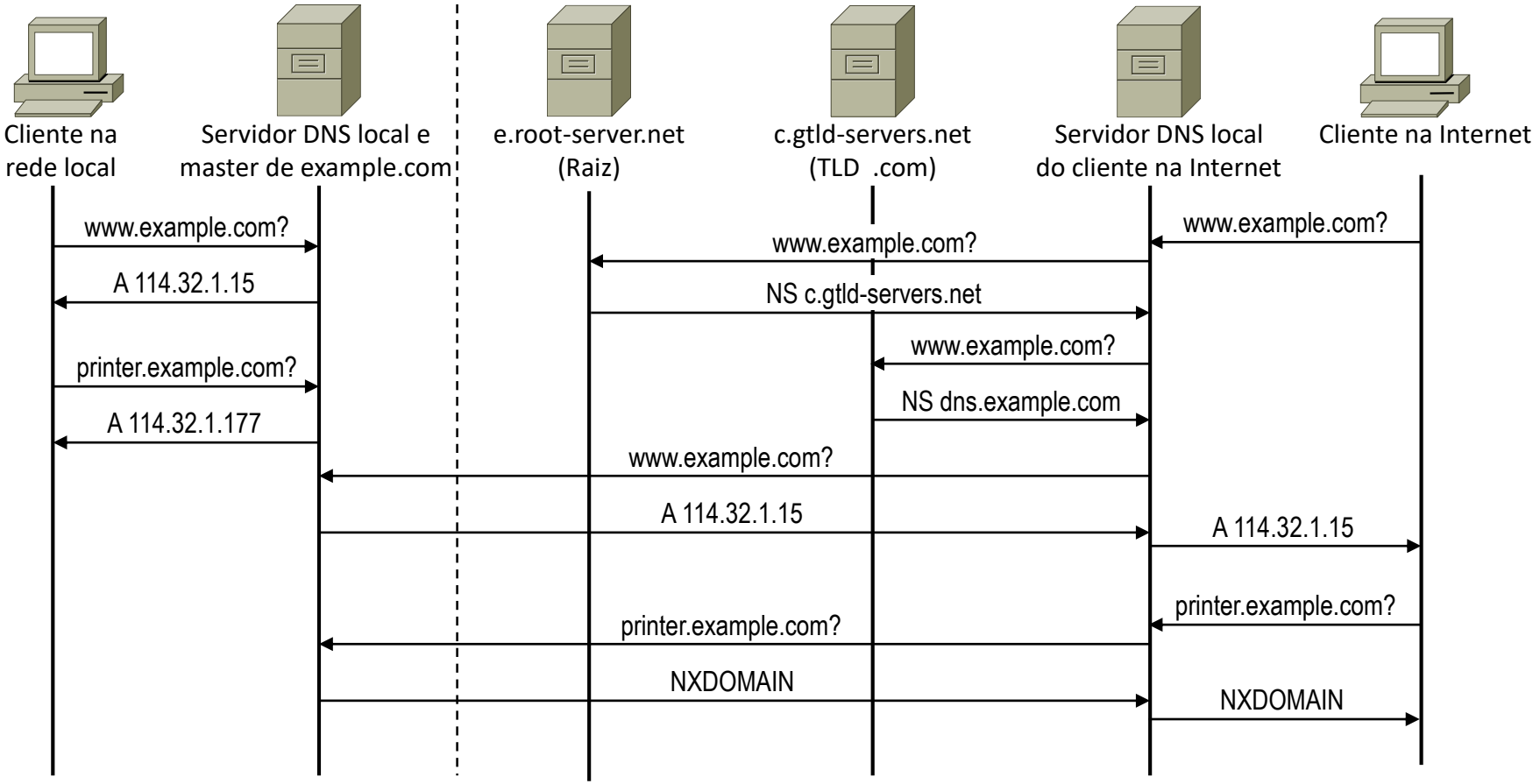
- Tradicionalmente as alterações a uma zona faziam-se editando o respectivo ficheiro no *master*
  - Pouco prático se as alterações forem muito frequentes
- Duas soluções possíveis
  - Integração com uma base de dados
  - Actualizações dinâmicas com pedidos UPDATE (RFC-2136)
- Alterações dinâmicas devem fazer-se apenas no *Primary Master*
- As actualizações dinâmicas são potêncialmente perigosas
  - Normalmente desactivadas, têm que ser explicitamente activadas
  - Normalmente associadas a mecanismos de segurança TSIG/TKEY
  - E/ou permitidas apenas a partir do *localhost*

Vistas

- Em determinadas situações é útil ter definições das zonas diferentes consoante o sítio de onde vem o pedido
  - Ter nomes de servidores e workstations para a rede interna, mas só de servidores para o exterior
  - Rede interna por trás dum NAT com mapeamento estático
- Tradicionalmente instalavam-se servidores *master* diferentes
- Uma alternativa melhor é a definição de vistas num único *master*
- Vistas permitem ter configurações diferentes consoante o endereço IP de quem faz o pedido de resolução
  - Incluindo as zonas e o seu conteúdo

Exemplo de utilização de vistas

www visível de dentro e de fora, mas *printer* visível apenas de dentro



Configuração

- A configuração faz-se no ficheiro `/etc/named.conf`
- Este ficheiro inclui
  - Cláusulas (partes principais da configuração)
    - Opções, zonas, vistas, inclusão de ficheiro externos, etc.
  - Declarações (itens individuais dentro das cláusulas)
- Exemplo:

```
options {
  directory "/var/named";
  version "Querias";
  allow-transfer {"none";};
  allow-recursion {192.168.3.0/24;};
};

zone "." {
  type hint;
  file "root.servers";
};
```

```
zone "example.com" IN {
  type master;
  file "master/master.example.com";
  allow-transfer {192.168.23.1;
                  192.168.23.2;};
};
```

Ficheiro /etc/named.conf pré-existente

- O ficheiro `/etc/named.conf` pré-existente no Fedora tem algumas configurações que é preciso alterar
- Põe o servidor à escuta apenas no `localhost` (IPv4 e IPv6). Remover estas linhas para escutar em todas as interfaces:

```
listen-on port 53 { 127.0.0.1; };
listen-on-v6 port 53 { ::1; };
```

- Permite apenas pedidos vindos do `localhost`. Remover também para permitir pedidos vindos de qualquer IP (necessário para master ou slave)

```
allow-query     { localhost; };
```

- Aceita pedidos recursivos independentemente de onde venham. Remover a linha

```
recursion yes;
```

e incluir outra com a declaração `allow-recursion {…};` para limitar os pedidos recursivos aos endereços IP da rede local

- Em geral, devem rever-se todas as configurações pré-existentes

Configuração — Master

- Um servidor *master* para uma ou mais zonas inclui
  - Definição no `/etc/named.conf` das zonas do tipo *master*
  - Respectivos ficheiros de zona

Configuração — Master

- Exemplo de `/etc/named.conf` para um *master*:

```
options {
  ...
  recursion no; // Só se deve activar nos servidores que fazem caching
};

...

// Zona para resolução directa
zone "example.com" IN {
  type master;
  file "master/example.com.zone";
};

// Zona para resolução inversa
zone "0.168.192.in-addr.arpa" IN {
  type master;
  file "reverse/192.168.0.zone";
};
```

Ficheiros de zona

- Existem nos servidores Master
- Contêm
  - Directivas
    - `$ORIGIN` — domínio acrescentado aos nomes parcialmente qualificados (i.e., que não terminam com "`.`")
      - O seu valor pode ser referido explicitamente usando "@"
      - Se omitida, assume-se o nome da zona indicado no `named.conf`
    - `$INCLUDE` — permite incluir ficheiros com informação adicional
      - Útil e.g. para evitar duplicação de registos quando são usadas vistas
    - `$TTL` — define um valor-padrão para o TTL dos RR definidos na zona
  - Registos de recurso da zona
    - I.e., com nomes pertencentes à zona*
  - Podem ainda conter comentários, iniciados por "`;`" e até ao fim da linha

*Os registos-cola podem ter nomes fora da zona (mas dentro da esfera de autoridade)

Exemplo: zona para resolução directa

```
$ORIGIN example.com.
$TTL    86400 ; valor de TTL para todos os registos que não o definam
@  1D  SOA ns rprior.dcc.fc.up.pt. (
                    2015042601 ; número de série
                    3h         ; período de refrescamento da zona
                    15         ; período para nova tentativa de refresc.
                    1w         ; período para expiração da zona
                    3h         ; período para caching negativo
          ) ; parêntesis permitem usar várias linhas para definir um RR
       NS     ns                    ; servidor de DNS dentro do domínio
       NS     dns.other.net.        ; outro servidor fora do domínio
       MX  10 mail.another.com.     ; servidor de email (externo)
; ### Máquinas ###
ns     A      192.168.0.1           ; servidor DNS
www    A      192.168.0.2           ; servidor web
ftp    CNAME  www                   ; servidor ftp é o mesmo que o web
; ### Delegação do subdomínio sub.example.com ###
sub      NS   dns.sub          ; interno ao subdomíno, precisa de ...
dns.sub  A    172.16.0.2       ; ... registo-cola
sub      NS   dns.other.net.   ; externo ao subdomínio, não precisa
```

Exemplo: zona para resolução directa

- O ficheiro de zona do slide anterior é interpretado pelo *Bind* como se tivesse o seguinte conteúdo

```
example.com. 86400 IN SOA ns.example.com. rprior.dcc.fc.up.pt. 2015042601 10800 15 604800 10800
example.com.          86400 IN NS           ns.example.com.
example.com.          86400 IN NS           dns.other.net.
example.com.          86400 IN MX           10 mail.another.com.
ftp.example.com.      86400 IN CNAME        www.example.com.
ns.example.com.       86400 IN A 192.168.0.1
sub.example.com.      86400 IN NS           dns.sub.example.com.
sub.example.com.      86400 IN NS           dns.other.net.
dns.sub.example.com.  86400 IN A 172.16.0.2
www.example.com.      86400 IN A 192.168.0.2
```

- Este formato do ficheiro de zona designa-se formato canónico
  - Obtém-se do ficheiro de zona original usando `named-checkzone -D`
- Formato canónico torna mais evidente que um ficheiro de zona é apenas uma lista de registos de recurso

Exemplo: zona para resolução inversa

```
$ORIGIN 0.168.192.in-addr.arpa.
$TTL    86400 ; valor de TTL para todos os registos que não o definam

@  1D  SOA ns.example.com. rprior.dcc.fc.up.pt. (
                    2015042601 ; número de série
                    3h         ; período de refrescamento da zona
                    15         ; período para nova tentativa de refresc.
                    1w         ; período para expiração da zona
                    3h         ; período para caching negativo
           )

       NS ns.example.com.
       NS dns.other.net.

; ### Máquinas (resolução inversa) ###
1      PTR    ns.example.com.
2      PTR    www.example.com.
; Opcionalmente, também se poderia acrescentar a seguinte entrada:
; 2      PTR    ftp.example.com.
```

Configuração — Slave

- Define uma ou mais zonas do tipo *slave*
- Para cada uma delas deve indicar o(s) respectivo(s) *master(s)*

- Adicionar um *slave* implica alterações noutros servidores
  - No *master* de cada uma das zonas para as quais é *slave*
    - Adicionar ao ficheiro de zona uma entrada NS apontando para o novo *slave*
    - Autorizar a transferência de zona para o novo *slave* (`allow-transfer`)
  - No *master* que delegou cada uma das zonas para as quais é *slave*
    - Adicionar um registo NS apontando para o novo *slave* (que também é autoritário para a zona delegada)
    - Um registo-cola para o novo *slave* (só se estiver dentro da própria zona para a qual é *slave*)

Configuração — Slave

- Exemplo de `/etc/named.conf` para um *slave*:

```
options {
  ...
  recursion no; // Só se deve activar nos servidores que fazem caching
};
...
// Zona para resolução directa
zone "example.com" IN {
  type slave;
  masters { 192.168.0.1; };
  file "slave/example.com.zone"; // Com permissão de escrita para named
};

// Zona para resolução inversa
zone "0.168.192.in-addr.arpa" IN {
  type slave;
  masters { 192.168.0.1; };
  file "slave/192.168.0.zone";   // Com permissão de escrita para named
};
```

Configuração — Caching

- Tem que aceitar pedidos recursivos
  - Mas convém restringi-los às máquinas da(s) rede(s) local(is)
- Precisa de ser capaz de resolver qualquer nome
  - Zona "." do tipo *hint*
  - Ficheiro com lista de servidores de raiz e respectivos endereços IP (e IPv6)

Configuração — Caching

- Exemplo de `/etc/named.conf` para um *caching nameserver*:

```
options {
  ...
  // Só aceita pedidos das máquinas locais
  allow-query {192.168.0.0/24;};
  // NOTA: a recursão é permitida por predefinição
};

...

zone "." IN {
  type hint;
  file "named.ca";
};
```

Configuração — Caching

- Exemplo do ficheiro para a zona *hint* (named.ca):

```
.                       3600000      NS    a.root-servers.net.
a.root-servers.net.     3600000      A     198.41.0.4
a.root-servers.net.     3600000      AAAA  2001:503:ba3e::2:30

.                       3600000      NS    b.root-servers.net.
b.root-servers.net.     3600000      A     192.228.79.201
b.root-servers.net.     3600000      AAAA  2001:500:84::b

.                       3600000      NS    c.root-servers.net.
c.root-servers.net.     3600000      A     192.33.4.12
c.root-servers.net.     3600000      AAAA  2001:500:2::c

.                       3600000      NS    d.root-servers.net.
d.root-servers.net.     3600000      A     199.7.91.13
d.root-servers.net.     3600000      AAAA  2001:500:2d::d

...
```

Configuração — Forwarding

- Semelhante ao caching, mas em vez de resolver iterativamente os pedidos, reenvia-os para outro servidor
- Precisa apenas da indicação desse(s) servidor(es)
  - Na cláusula `options` se for *forwarding* para todas as zonas

    ou
  - Em zonas específicas do tipo `forward` se for *forwarding* apenas para essas zonas

Configuração — Forwarding

- Exemplo de `/etc/named.conf` para um *forwarding nameserver*:

```
options {
  ...
  // Identificação dos servidores para quem reenvia os pedidos
  forwarders { 8.8.8.8; 8.8.4.4; };
  // Só aceita pedidos das máquinas locais
  allow-query { 192.168.0.0/24; };
  // NOTA: a recursão é permitida por predefinição
};
```

Configuração — Vistas

- Zonas têm que ser definidas dentro das vistas

- Exemplo com duas vistas
  - Uma para a rede interna com todas as máquinas
  - Outra para o exterior apenas com os servidores públicos

Configuração — Vistas

- Ficheiro /etc/named.conf

```
...
view "dentro" {
  match-clients { 114.32.1.0/24; };  // clientes da rede interna

  zone "example.com" IN {
    type master;
    file "master/example.com-all.zone";
  };
}

view "fora" {
  match-clients { any; };  // todos os que chegarem aqui (i.e., que não
                           // foram apanhados na vista "dentro")
  zone "example.com" IN {
    type master;
    file "master/example.com-public.zone";
  };
}
```

Configuração — Vistas

- Ficheiro example.com-public.zone

```
$ORIGIN example.com.
$TTL    86400
@  1D  SOA ns.example.com. rprior.dcc.fc.up.pt. 2015042601 3h 15 1w 3h
       NS     ns.example.com.

; ### Servidores públicos ###
ns     A      114.32.1.2
www    A      114.32.1.15
```

- Ficheiro example.com-all.zone

```
$ORIGIN example.com.
$TTL    86400
$INCLUDE master/example.com-public.zone ; importa todos os registos pú-
                                        ; blicos, incluindo SOA e NS

; ### Máquinas privadas ###
printer    A      114.32.1.177
```

Utilitários

- `host`
  - Usado para consultar o DNS a partir da linha de comando
  - Alternativa ao tradicional `nslookup`, considerado obsoleto
- `dig`
  - Ferramenta "expert" para consulta e depuração do DNS
- `rndc`
  - Usado para controlar o servidor DNS (*e.g.*, *reload*, *flush*, etc.)
- `nsupdate`
  - Usado para actualizar dinamicamente zonas no *primary master*
- `named-checkconf`
  - Usado para detectar erros no named.conf
- `named-checkzone`
  - Usado para validar ficheiros de zona