DNS

O texto introduz o **DNS (Domain Name System)** como o serviço de directório da Internet, responsável por traduzir nomes de host (hostnames) em endereços IP. Assim como as pessoas podem ser identificadas por diferentes tipos de identificadores (nomes, números de segurança social, números de carta de condução), os equipamentos na Internet podem ter identificadores em diferentes formatos.

Hostnames vs. Endereços IP:

- **Hostnames**: São mais fáceis de lembrar por serem mnemonicamente semelhantes a palavras (por ex. www.google.com). Contudo, não fornecem indicações sobre a localização do host na rede e não são convenientes para os routers, que preferem identificadores numéricos estruturados.
- **Endereços IP**: São compostos por quatro bytes (cada um variando entre 0 e 255) e têm uma estrutura hierárquica. Por exemplo, um endereço como 121.7.106.83 pode ser entendido como pertencendo a uma determinada rede e sub-rede, facilitando a tarefa de encaminhamento no interior da Internet.

O DNS surge para conciliar estas duas preferências: **os utilizadores e as aplicações preferem usar hostnames, enquanto a infraestrutura da rede necessita de endereços IP para chegar ao destino**.

O que é o DNS:

- É um serviço de directório distribuído, implementado sobre uma hierarquia de servidores DNS espalhados globalmente.
- É um protocolo de aplicação, que funciona sobre UDP e utiliza a porta 53.
- As suas funções vão muito além da simples tradução de nomes para IPs. Também fornece outros serviços importantes.

Funcionalidade Básica do DNS:

Quando um utilizador digita um URL no browser (por exemplo, www.someschool.edu/index.html), o browser extrai o hostname (www.someschool.edu) e entra em contacto com o cliente DNS (o resolver) na máquina local. Este faz uma consulta (query) a um servidor DNS para obter o endereço IP correspondente. Uma vez obtido o IP, o browser pode então estabelecer uma conexão TCP com o servidor web na porta 80, enviar o pedido HTTP e receber a página solicitada. Este processo introduz um pequeno atraso adicional, mas o DNS adopta mecanismos de cache para acelerar a resolução de nomes com frequência consultados.

Outros Serviços do DNS:
Além de mapear nomes em endereços IP, o DNS oferece:

1. **Host aliasing**: Permite que um hostname complexo ou "canónico" tenha um ou mais nomes alternativos (alias). Por exemplo, o hostname canonical relay1.west-coast.enterprise.com pode ter um alias mais simples como enterprise.com. Através do DNS, as aplicações podem obter o nome canónico e o IP associado a um alias.
2. **Mail server aliasing**: Da mesma forma, endereços de e-mail podem ser mais fáceis de memorizar (por exemplo, bob@yahoo.com) do que o hostname real do servidor de correio. O DNS ajuda a obter o hostname canónico e o endereço IP do servidor de e-mail para o qual aquela conta aponta.
3. **Load distribution (distribuição de carga)**: Grandes sites costumam ter múltiplos servidores (replicação), cada um com um endereço IP diferente. O DNS pode associar um único hostname a um conjunto de endereços IP e, a cada consulta, entregar a lista desses endereços com uma ordem rotativa (DNS rotation), balanceando a carga entre os servidores. Este mecanismo também é utilizado para servidores de e-mail, e até companhias como a Akamai utilizam DNS em combinações sofisticadas para fornecer redes de distribuição de conteúdo (CDNs).

Normalização e Referências:
O DNS é definido nos RFCs 1034 e 1035, com actualizações em documentos adicionais. É um sistema complexo e possui vários pormenores não abordados em profundidade no trecho apresentado. Existem referências a obras e artigos que descrevem a história, o porquê e o funcionamento do DNS.

As imagens e o texto associados a esta parte do capítulo descrevem em profundidade o funcionamento do DNS (Domain Name System), focando-se na sua natureza distribuída, hierárquica e nos diferentes tipos de servidores envolvidos no processo de resolução de nomes de host em endereços IP.

Porque não um único servidor DNS?
No início desta secção, é destacado que um design centralizado de DNS (um único servidor global) seria completamente inviável. Seria um:

1. **Ponto único de falha**: Se o servidor avariasse, toda a Internet ficaria sem capacidade de resolução de nomes.
2. **Sobrecarregado por tráfego**: Milhões (ou biliões) de pedidos diários tornariam o servidor lento e ineficaz.
3. **Distante de muitos utilizadores**: Ter um servidor numa única localização geográfica significaria atrasos elevados para utilizadores noutras partes do mundo.
4. **Manutenção difícil**: Actualizar e manter uma base de dados centralizada, com biliões de hosts a surgir e desaparecer todos os dias, seria impraticável.

Desta forma, o DNS é distribuído e desenhado para ser escalável, resiliente e eficiente.

Base de dados Distribuída e Hierárquica

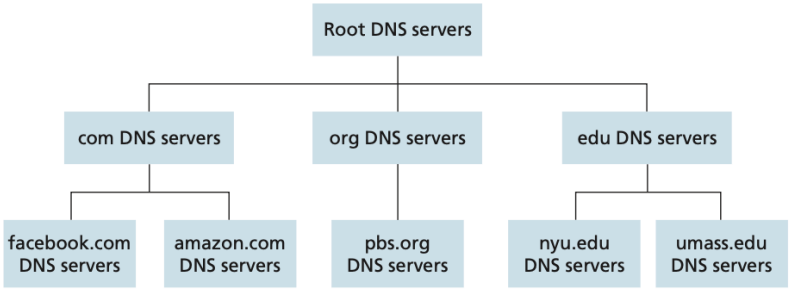


Figure 2.17 ♦ Portion of the hierarchy of DNS servers

Para resolver os problemas de escala, o DNS recorre a uma hierarquia de servidores distribuídos globalmente. A organização básica, conforme ilustrado na figura (Figura 2.17), segue um modelo hierárquico, com três níveis principais:

1. **Root DNS servers (Servidores de raiz)**:

- Existem mais de mil instâncias de servidores de raiz em todo o mundo (a Figura 2.18 mostra a distribuição geográfica).
- Estes não armazenam informações sobre hosts específicos, mas sabem quais os servidores de topo (TLD) a contactar para cada domínio de topo.
- Respondem tipicamente com informações sobre qual TLD server contactar a seguir.

Estes root servers são geridos por várias organizações e coordenados pela IANA (Internet Assigned Numbers Authority). Cada root server tem múltiplas cópias, distribuídas geograficamente para maior resiliência e melhor latência.

2. **Top-level domain (TLD) servers (Servidores de Domínio de Topo)**:

- Há um conjunto de TLD servers para cada domínio de topo (como .com, .org, .net, .edu, bem como domínios de código de país como .uk, .fr, .br).
- Estes servidores contêm registos que apontam para os servidores autoritativos dos domínios sob a sua responsabilidade. Por exemplo, os TLD servers do .com sabem quais os servidores autoritativos para amazon.com, google.com, etc.
- Grandes empresas, como a Verisign, gerem os TLD servers de domínios populares como .com, enquanto instituições como a Educause gerem o .edu.

3. **Authoritative DNS servers (Servidores Autoritativos)**:

- Estes servidores encontram-se na base da hierarquia e são mantidos pelas organizações que possuem o nome de domínio. Por exemplo, a própria Amazon terá servidores autoritativos para o domínio amazon.com.
- São estes que contêm os registos DNS com a correspondência exacta de hostnames (como www.amazon.com) para endereços IP específicos.
- Uma empresa ou universidade pode manter os seus próprios servidores autoritativos, ou contratar um fornecedor de serviços que hospede estes registos.

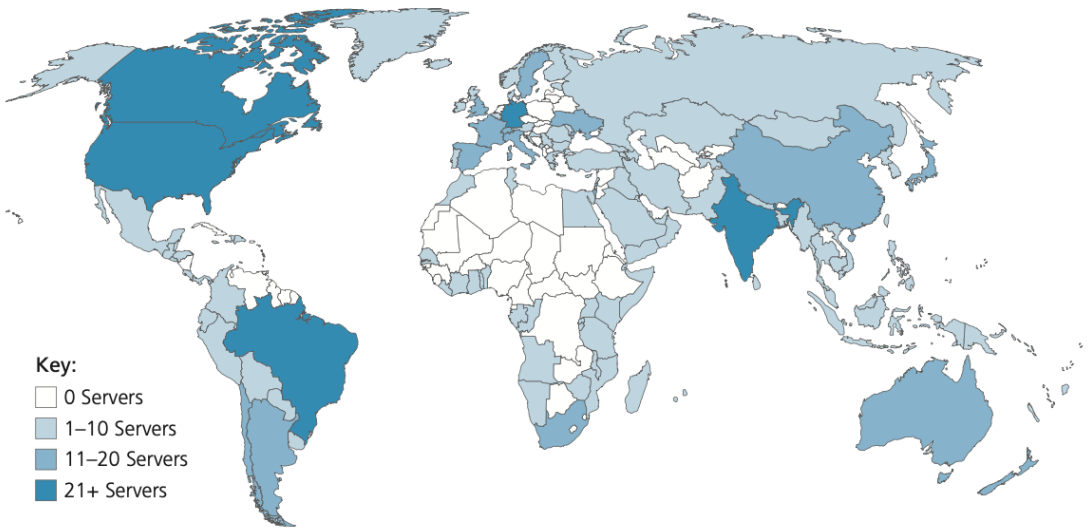


Figure 2.18 ♦ DNS root servers in 2020

Funcionamento da Resolução Hierárquica
Quando um cliente (por exemplo, o browser do utilizador) quer resolver www.amazon.com, o processo pode ocorrer de forma iterativa (aproximação simplificada):

1. O cliente contacta um servidor DNS de resolução local (um resolver), que pode já ter a informação em cache. Caso não tenha, envia uma query a um root DNS server.
2. O root DNS server responde com o endereço do TLD server responsável pelo domínio de topo: neste caso, o .com.
3. O cliente (via o seu DNS resolver) contacta então o TLD server (.com) e recebe a informação sobre o servidor autoritativo da Amazon.
4. Finalmente, contacta-se o servidor autoritativo da Amazon, obtendo-se o endereço IP de www.amazon.com.

Este processo acontece rapidamente graças ao paralelismo, caching e elevada distribuição dos servidores DNS. Além disso, tipicamente o utilizador não vê estes passos; o DNS resolver local (geralmente fornecido pelo ISP ou pela rede empresarial) abstrai o processo ao utilizador final, muitas vezes obtendo informação em cache e reduzindo a necessidade de consultar repetidamente os servidores de raiz ou TLD.

Caching e Desempenho
Para tornar o DNS mais eficiente, as respostas são armazenadas em cache pelos resolvers e mesmo pelos navegadores. Assim, se outro utilizador, ou o mesmo utilizador em outra sessão, pedir novamente o endereço IP de www.amazon.com, é provável que esse endereço já esteja disponível em cache no DNS local, evitando novas consultas à hierarquia.

Nesta parte do texto e imagens são introduzidos conceitos práticos do funcionamento do DNS no dia-a-dia, com destaque para o papel dos **Local DNS**

servers, o mecanismo de **resolução recursiva e iterativa**, e a importância do **DNS caching**.

Local DNS Server (Servidor DNS Local):
Além da hierarquia principal do DNS (composta por servidores root, TLD e autoritativos), cada ISP ou rede institucional disponibiliza pelo menos um servidor DNS local para os seus utilizadores. Este local DNS server funciona como um ponto de entrada no sistema DNS para o utilizador:

- Quando um utilizador (por exemplo, no host cse.nyu.edu) quer resolver o nome de um host (como gaia.cs.umass.edu), o seu pedido DNS vai primeiro para o servidor DNS local do ISP ou da instituição (neste caso dns.nyu.edu).
- O local DNS server actua como um "proxy" que encaminha a consulta para os servidores do sistema DNS (root, TLD, autoritativos) necessários para encontrar o endereço IP correspondente ao hostname solicitado.
- Ao estar geograficamente próximo do utilizador, o local DNS server reduz a latência inicial da requisição e pode armazenar em cache resultados para futuras consultas, melhorando o desempenho.

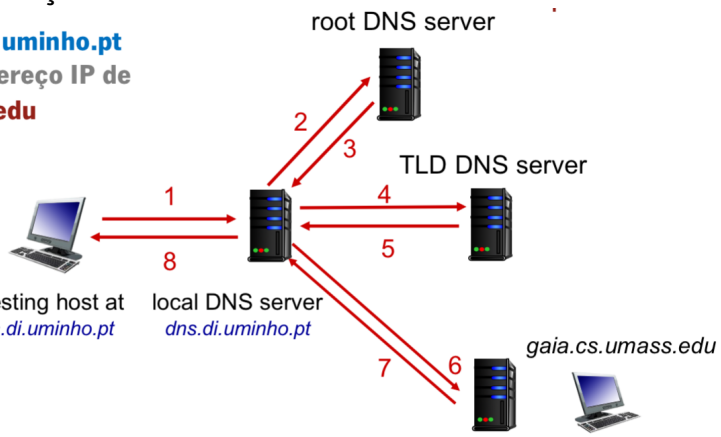
Resolução Recursiva vs. Iterativa:
Na resolução de um hostname em endereço IP, podem ocorrer dois tipos de consultas DNS:

- **Consulta Recursiva**: O host requer que o seu servidor DNS local obtenha a resposta completa pelo utilizador. Neste caso, o pedido do host ao seu local DNS server é geralmente recursivo, ou seja, o host diz: "Encontra o endereço IP para mim". O servidor local, então, contacta outros servidores DNS ao longo da hierarquia para obter a resposta final.
- **Consulta Iterativa**: Uma vez que o local DNS server obtém informações parciais (por exemplo, o endereço de um TLD server a partir de um root server), as consultas subsequentes a outros servidores DNS para refinar a pesquisa (ex: do TLD server ao autoritativo) tendem a ser iterativas. Nestes casos, o local DNS server recebe a resposta parcial ("não tenho o IP, mas tenta neste outro servidor") e faz ele próprio a próxima consulta, sem pedir aos servidores contactados que façam o trabalho completo.

O resultado é um processo no qual o utilizador faz um pedido recursivo para o seu local DNS server, e este depois faz uma sequência de consultas iterativas à hierarquia DNS até encontrar o IP desejado.

Exemplo de Resolução de um nome

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

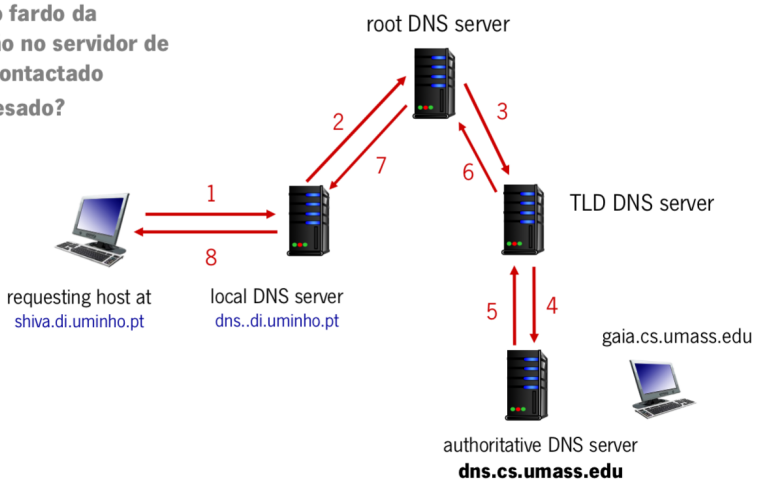


Modo iterativo:

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

Modo recursivo:

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



DNS Caching (Memorização de Resultados):
Um aspecto fundamental para a eficiência do DNS é o caching:

- Sempre que um servidor DNS (incluindo o local DNS server) recebe uma resposta com o mapeamento de um hostname para um endereço IP, pode armazenar essa informação em cache durante um período de tempo (tipicamente algumas horas ou dias).
- Se outro utilizador (ou o mesmo) pedir o mesmo hostname dentro do prazo de validade desse registo em cache, o servidor DNS pode devolver imediatamente a informação, evitando contactar novamente os root, TLD ou servidores autoritativos.
- Este mecanismo reduz o número de mensagens DNS circulando na Internet, diminui a latência percebida pelos utilizadores e diminui a carga nos servidores DNS mais altos na hierarquia. Como consequência, a maioria dos pedidos para hostnames comuns acaba por ser respondida sem precisar voltar a contactar os root servers, pois o local DNS server já tem a informação em cache ou rapidamente obtém os IPs dos TLD servers a partir do seu próprio cache.

Exemplo Ilustrado (Figura 2.19 e Figura 2.20):

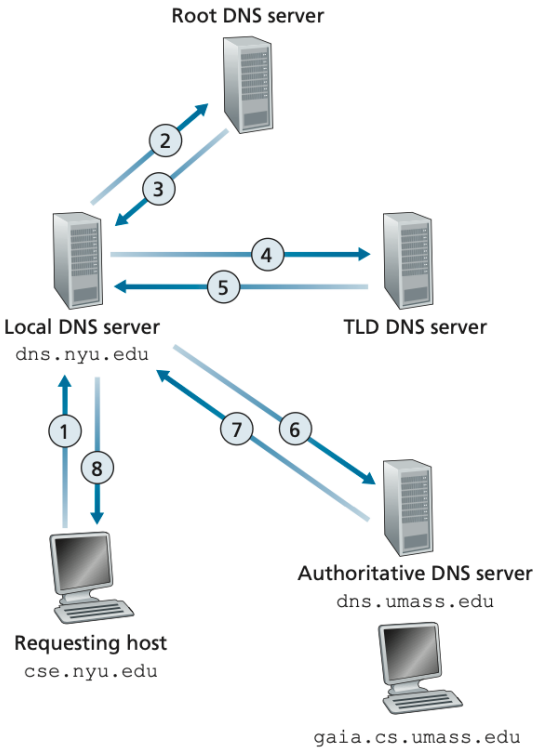


Figure 2.19 ♦ Interaction of the various DNS servers

A figura 2.19 mostra um caso concreto em que o host cse.nyu.edu solicita o IP de gaia.cs.umass.edu. O pedido vai para dns.nyu.edu (local DNS server), que contacta um root DNS server, recebe as referências para TLD servers, contacta um TLD server, recebe referências para o servidor autoritativo dns.umass.edu, e finalmente obtém o IP do host desejado. O processo envolve múltiplas mensagens (oito no exemplo), mas graças ao caching, consultas futuras ao mesmo hostname podem ser resolvidas muito mais rapidamente.

A figura 2.20 mostra um cenário de consultas todas recursivas, mas na prática a mistura entre recursivas (do host para o local DNS server) e iterativas (do local DNS server para a hierarquia) é a mais comum.

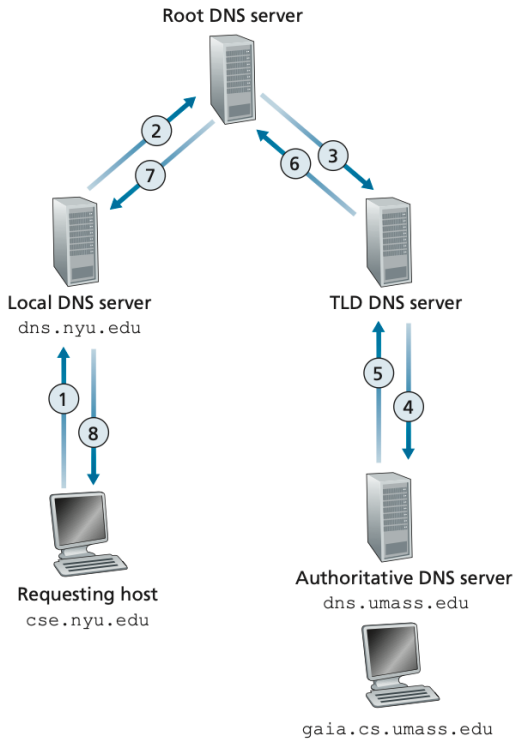


Figure 2.20 ♦ Recursive queries in DNS

Nesta parte final sobre DNS, o texto e as imagens abordam os **registos de recursos (Resource Records, RRs)**, a estrutura das **mensagens DNS** (tanto de consulta como de resposta), e o processo pelo qual novos registos são inseridos na base de dados DNS ao registar um domínio.

Resource Records (RRs):
Um registo DNS é um **quádruplo** da forma: (Name, Value, Type, TTL), onde:

- **Name**: Normalmente um hostname.
- **Value**: Depende do tipo de registo.
- **Type**: Indica o tipo de informação armazenada (A, NS, CNAME, MX, etc.).
- **TTL (Time to Live)**: Determina por quanto tempo o registo pode permanecer em cache antes de ser considerado obsoleto e precisar ser atualizado.

Principais tipos de RR:

- **Tipo A**: Fornece o mapeamento padrão de um hostname para o seu endereço IP. Por exemplo, (relay1.bar.foo.com, 145.37.93.126, A).
- **Tipo NS**: Associa um domínio a um servidor DNS autoritativo para aquele domínio. Por exemplo, (foo.com, dns.foo.com, NS).

- **Tipo CNAME**: Fornece um nome canónico para um alias. Por exemplo, (foo.com, relay1.bar.foo.com, CNAME) significa que foo.com é um alias para relay1.bar.foo.com.
- **Tipo MX**: Similar ao CNAME, mas especificamente para servidores de mail, mapeando um domínio a um hostname canónico de um servidor de correio. Por exemplo, (foo.com, mail.bar.foo.com, MX).

Mensagens DNS:
As mensagens DNS (tanto pedidos como respostas) têm o mesmo formato geral. Incluem:

- **Cabeçalho (header)**: Com 12 bytes, contém um identificador de 16 bits (para combinar respostas com pedidos), flags (indicando se é um pedido ou resposta, se o servidor é autoritativo, se é desejada recursão, etc.), e quatro campos numéricos que indicam quantos registos de cada tipo seguem.
- **Secção de Pergunta (question)**: Contém o nome e o tipo do registo que o cliente procura.
- **Secção de Resposta (answer)**: Contém um ou mais RRs que respondem à pergunta feita.
- **Secção de Autoridade (authority)**: Contém RRs apontando para servidores DNS autoritativos adicionais.
- **Secção de Informação Adicional (additional)**: Pode conter RRs úteis, por exemplo, para fornecer endereços IP adicionais relacionados com a resposta (como o A record de um mail server fornecido por um MX record na secção de resposta).

A Figura 2.21 ilustra o formato de uma mensagem DNS, mostrando o header, as secções question, answer, authority e additional, bem como a ordem em que aparecem.

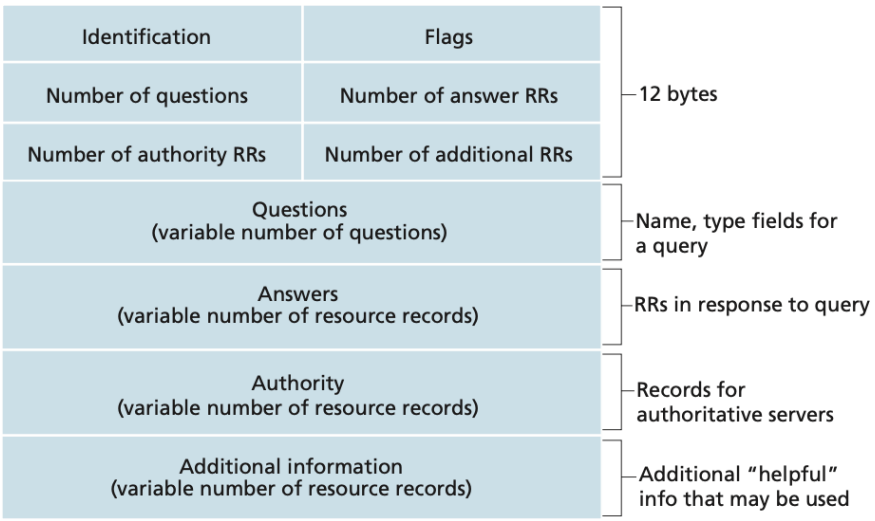


Figure 2.21 ♦ DNS message format

Inserir Registos no DNS:
O texto descreve o processo de registar um novo domínio, por exemplo, networkutopia.com:

1. Escolhe-se um **registar** (entidade credenciada pela ICANN) para verificar a unicidade do nome de domínio e proceder ao seu registo.
2. Ao registar o domínio, o proprietário informa o registar dos servidores DNS autoritativos e os seus endereços IP correspondentes.
3. O registar insere os registos do tipo NS e A apropriados nos servidores TLD do domínio (por exemplo, no .com).
4. Para que o site seja acessível, é necessário ainda inserir, nos servidores autoritativos da empresa, registos tipo A para o hostname principal (por ex. www.networkutopia.com) e tipo MX para o servidor de mail (por ex. mail.networkutopia.com).
5. Uma vez concluído este processo, as pessoas podem aceder ao site e enviar e-mails, com o DNS garantindo que consultas ao domínio networkutopia.com resultam no encaminhamento para os servidores correctos.

Este processo mostra como novos domínios são adicionados à base de dados distribuída do DNS e se tornam globalmente acessíveis.

Nesta secção, o texto e as imagens analisam o **desempenho e escalabilidade de arquitecturas de distribuição de ficheiros**, comparando o tradicional modelo **cliente-servidor** com o modelo **peer-to-peer (P2P)**, e introduzindo o protocolo **BitTorrent** como um exemplo destacado de distribuição P2P.

Contexto:
Suponha que existe um ficheiro grande (por exemplo, uma nova versão de um sistema operativo ou um ficheiro de vídeo de alta qualidade) num servidor, e que um número crescente de utilizadores (“peers”) pretende obtê-lo. No modelo cliente-servidor, cada peer descarrega o ficheiro directamente do servidor de origem, o que coloca toda a pressão de largura de banda e capacidade de upload no servidor. Já no modelo P2P, os próprios peers que recebem partes do ficheiro tornam-se capazes de o redistribuir, partilhando a carga do envio entre todos os participantes.

Arquitectura Cliente-Servidor e Tempo de Distribuição:

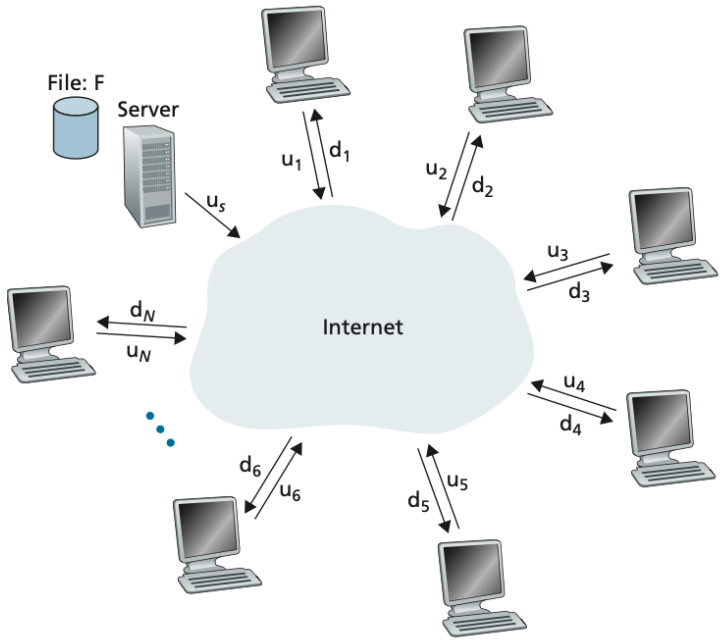


Figure 2.22 ♦ An illustrative file distribution problem

A figura e o texto descrevem uma análise quantitativa simples. Admitindo:

- **F**: O tamanho do ficheiro em bits.
- **N**: O número de peers que pretendem o ficheiro.
- **u_s**: A taxa de upload do servidor (bits/segundo).
- **d_i**: A taxa de download do i-ésimo peer. Definimos d_min como a menor taxa de download entre todos os peers (d_min = min{d_1, d_2, ..., d_N}).

No modelo cliente-servidor, o servidor tem de enviar o ficheiro completo F a cada um dos N peers, transferindo assim um total de N × F bits. Como o servidor só pode enviar com taxa u_s, o tempo mínimo para enviar todos os ficheiros é pelo menos (N × F)/u_s. Além disso, o peer mais lento não poderá terminar o download antes de F/d_min segundos. Combinando ambos os factores, o tempo mínimo D_cs (para cliente-servidor) é:

$$D_{cs} \geq \max\left\{\frac{NF}{u_s}, \frac{F}{d_{\min}}\right\}.$$

Isto mostra que, à medida que N cresce, o termo (N × F)/u_s domina. Portanto, no modelo cliente-servidor, o tempo de distribuição aumenta linearmente com o número de peers. Se o número de peers passar, por exemplo, de mil para um milhão (um aumento por um factor de mil), o tempo total aumenta também por um factor de mil.

Arquitectura P2P e Tempo de Distribuição:
No modelo P2P, a situação é diferente. Uma vez que cada peer não só descarrega dados como também pode fazer upload de partes do ficheiro que já obteve, a capacidade total de upload do sistema é aumentada. Neste caso, consideramos:

- Todos os peers (incluindo o servidor) contribuem com a sua taxa de upload.
- A soma total de todas as taxas de upload no sistema é $u_s + \sum u_i$, onde u_i é a taxa de upload do i-ésimo peer.

As observações principais para o P2P são:

1. O servidor tem de enviar pelo menos cada bit do ficheiro uma vez, logo, não pode ser mais rápido que F/u_s (semelhante ao caso cliente-servidor inicialmente).
2. O peer mais lento não pode terminar antes de F/d_{min} (igual ao caso anterior).
3. Agora existe capacidade total de upload agregada: $N \times F$ bits têm de ser distribuídos e a capacidade total de upload é $u_s + \sum u_i$. Assim, o tempo não pode ser menor que $(N \times F)/(u_s + \sum u_i)$.

Combinando estes factores, o tempo mínimo D_{P2P} (para peer-to-peer) é:

$$D_{P2P} \ge \max\left\{\frac{F}{u_s}, \frac{F}{d_{min}}, \frac{NF}{u_s + \sum_{i=1}^{N} u_i}\right\}.$$

Ao contrário do modelo cliente-servidor, aqui o termo $(N \times F)/(u_s + \sum u_i)$ pode crescer muito mais lentamente com N, pois cada novo peer introduz tipicamente alguma capacidade de upload adicional. Na prática, se todos os peers tiverem capacidades de upload não negligenciáveis, o tempo de distribuição cresce muito mais lentamente com N, podendo até manter-se praticamente constante, independentemente do número de peers. Isto significa que o P2P é **auto-escalável**.

Gráfico de Comparação (Figura 2.23):

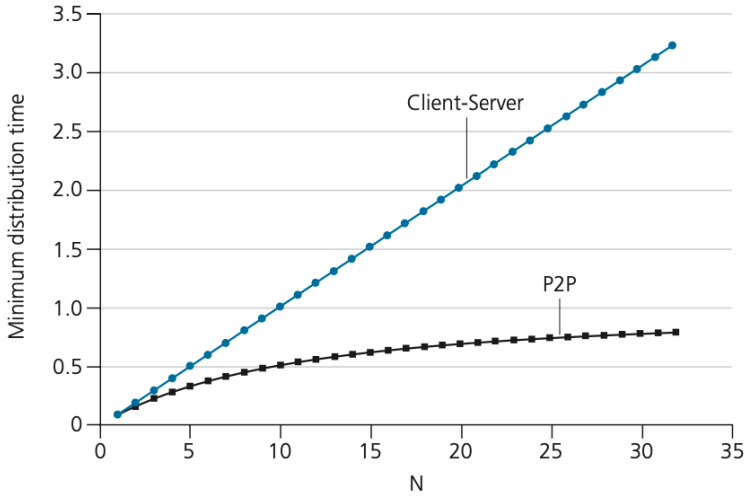


Figure 2.23 ♦ Distribution time for P2P and client-server architectures

A figura mostra um exemplo numérico em que a capacidade do servidor é fixada, e cada peer tem a mesma taxa de upload. A análise revela que:

- No modelo cliente-servidor, o tempo de distribuição cresce linearmente com N.
- No modelo P2P, o tempo de distribuição mantém-se baixo, praticamente inalterado mesmo quando o número de peers aumenta. No exemplo do texto, o tempo fica sempre abaixo de uma hora, mesmo para um número muito grande de peers. Isto é notável e reforça o conceito de auto-escalabilidade do P2P.

Esta eficiência vem do facto de cada peer contribuir para a redistribuição dos dados, tornando o sistema mais robusto e mais rápido à medida que a comunidade de utilizadores cresce.

DHT (Distributed Hash Table):
O texto termina referindo a existência de DHTs (tabelas de hash distribuídas) como outra aplicação interessante do P2P. Uma DHT distribui registos de uma base de dados por vários peers, permitindo buscas eficientes e descentralizadas sem depender de um servidor central. O BitTorrent e outros sistemas P2P podem usar DHTs para localizar peers sem a necessidade de um tracker central.

- **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.
...
```

- 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)

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!

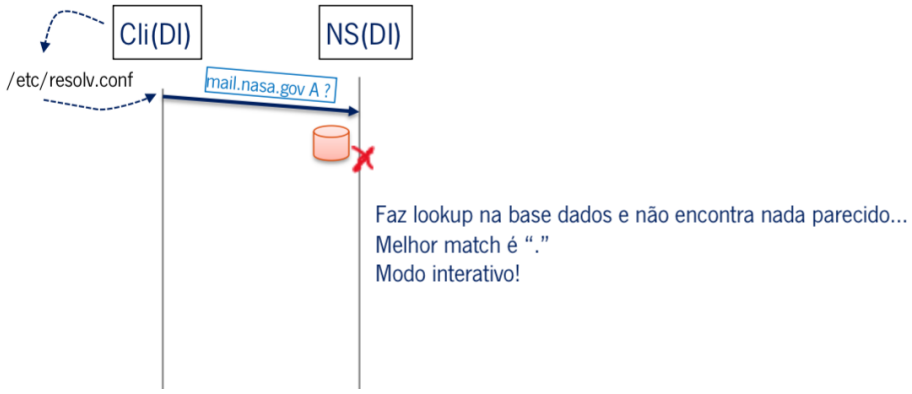
Tem de ser **muito** eficiente!

- **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**

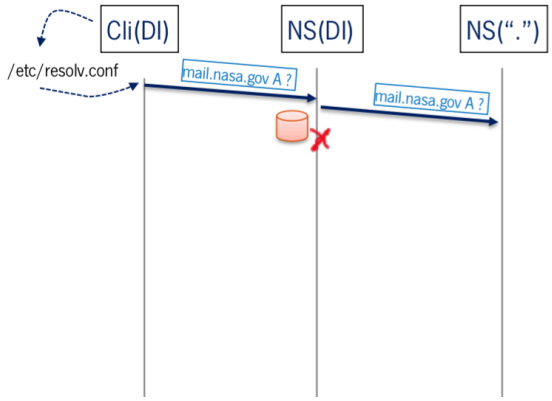
Exercício

- **Para todos os clientes da rede fixa do DI, designados em abstrato por Cli(di), o servidor local do DNS é, também em abstrato, 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.

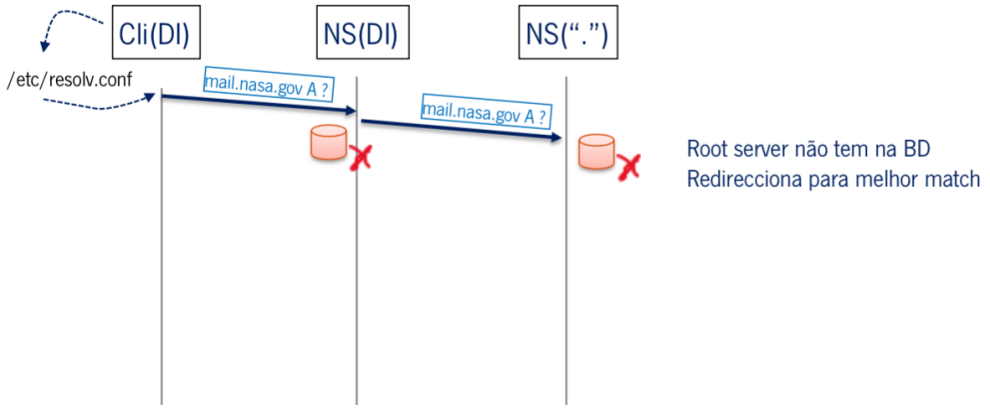
1ªIteração



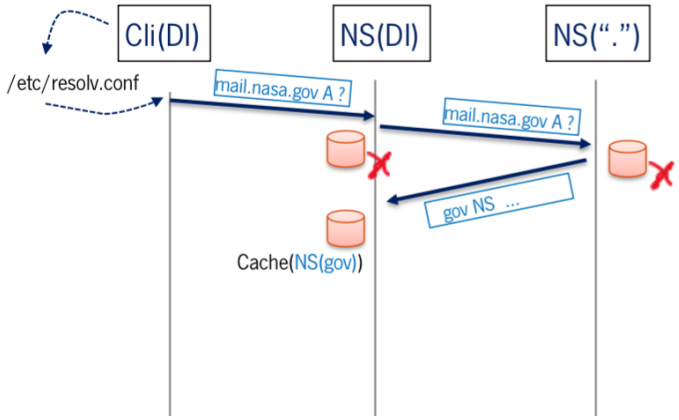
2ªIteração



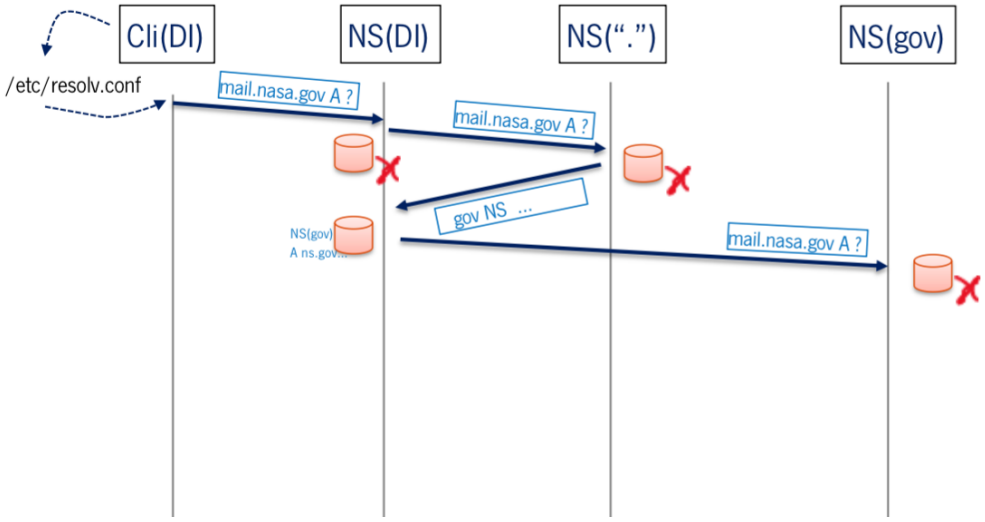
3ªIteração



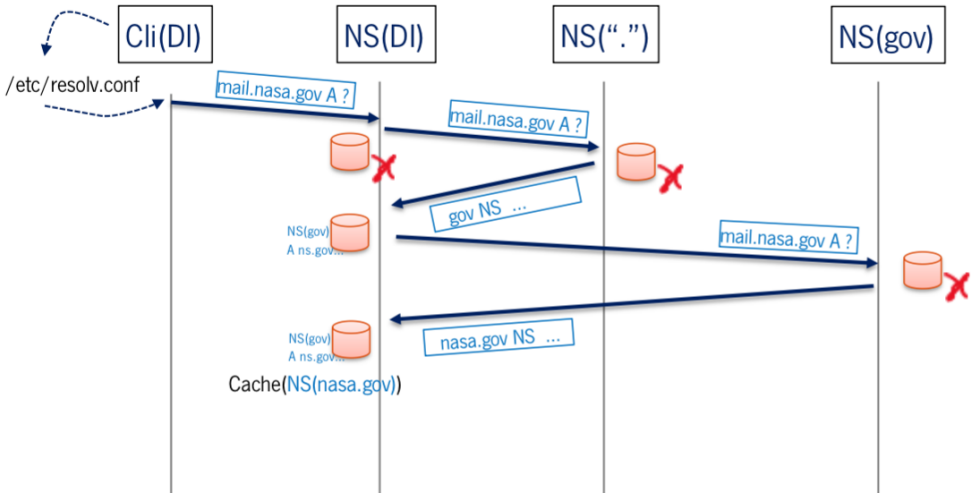
4ªIteração



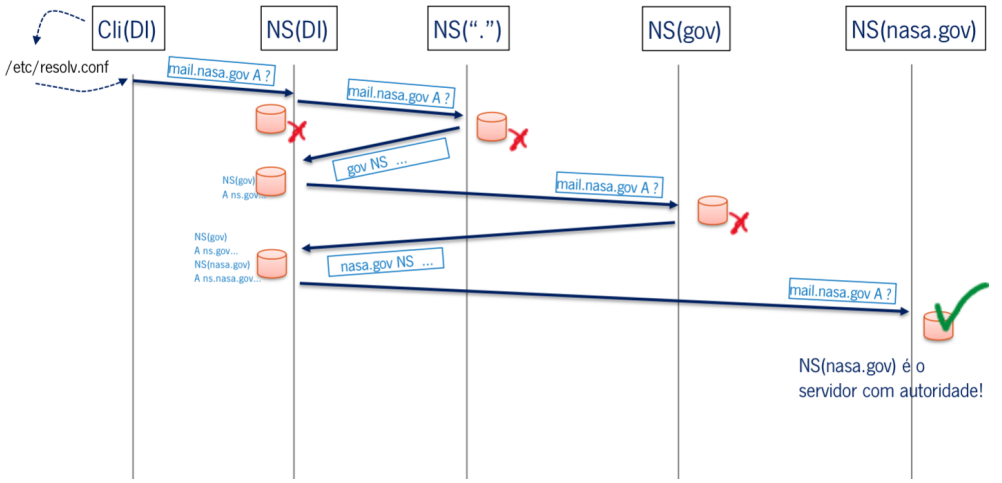
5ªIteração



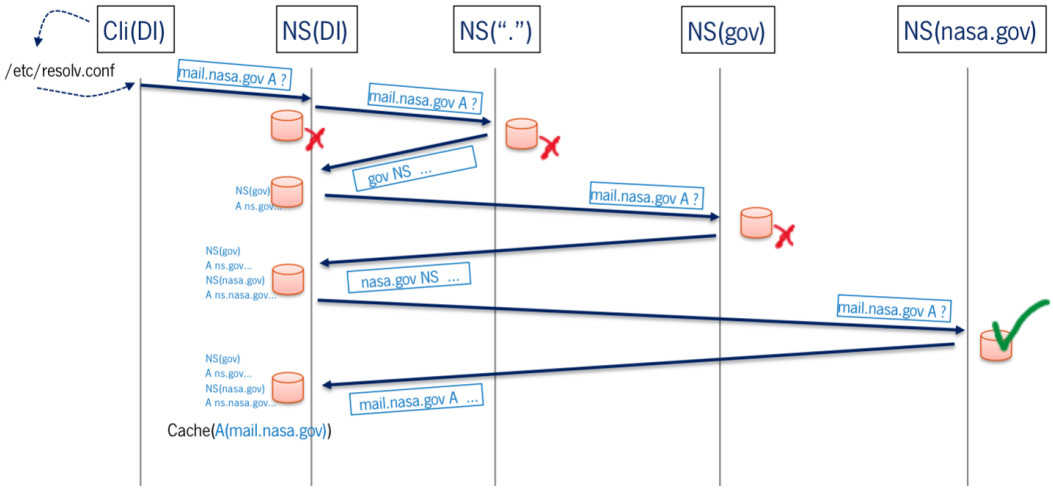
6ªIteração



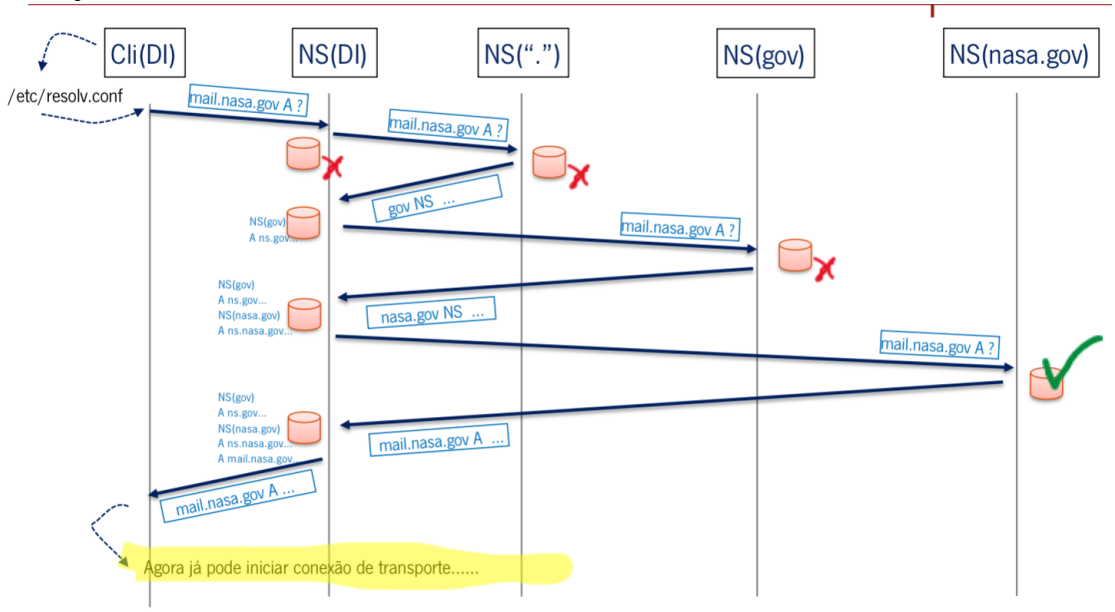
7ªIteração



8ªIteração



9ªIteração



TEMPO TOTAL DNS: 1 RTT local + 3 RTT externos