Faculdade de Ciências da Universidade de Lisboa
Departamento de Informática

RELATÓRIO DE ESTÁGIO

Anexos

FCCN - Fundação para o Desenvolvimento dos Meios Nacionais de Cálculo Científico

Artur Manuel Pereira Romão

Lisboa, Julho de 1993

Índice

Anexo 2 - Sincronização distribuida

Anexo 1

Designação no Internet

Anexo 1

Designação no Internet

Introdução

A estrutura dos endereços IP (inteiros de 32 bits) não é a mais apropriada para uso humano, mesmo sob a forma "*dot-quad*". Dado o estado actual do Internet, com mais de um milhão de hosts ligados, não é concebível que os utilizadores memorizem os endereços de todas as máquinas com que comunicam no decorrer do seu trabalho. É de todo preferível (leia-se indispensável) utilizar nomes, fáceis de memorizar e utilizar, elaborando-se um esquema de designação que faça corresponder um nome (usado por pessoas) a um endereço (usado pelos protocolos de rede) e vice-versa.

No texto que se segue faz-se uma apresentação do Domain Name System (DNS), a resposta da comunidade Internet a este problema, sendo abordados aspectos teóricos (evolução, especificações e descrição do serviço) e práticos (montagem e administração do sistema), sendo tratado em particular o caso português.

1. O Domain Name System

Hoje em dia milhares de computadores ligados ao Internet são acedidos por utilizadores e aplicações através dos seus nomes, sendo estes traduzidos em endereços IP, de forma transparente, para interacção com os níveis mais baixos da suite de protocolos TCP/IP. O Domain Name System (DNS) é o sistema que possibilita esse serviço, não se limitando, no entanto, a traduzir nomes em endereços, gerindo uma vasta gama de recursos, dos quais depende o correcto funcionamento duma série de aplicações de uso generalizado.

1.1. Evolução dos nomes

Na última década foram implementados no Internet várias políticas de designação, tendo a sintaxe dos nomes evoluído, partindo de uma forma simples, não estruturada, até aos nomes compostos actualmente utilizados, denotando uma estrutura subjacente à sua composição.

1.1.1. *Flat names*

A administração dos nomes começou por ser centralizada, na forma de um ficheiro, contendo uma lista de nomes de redes, *gateways* e *hosts*, com os respectivos endereços. Estes nomes eram identificadores simples de objectos, sem qualquer significado ou estrutura adicional, sendo, por isso,

denominados *flat names*. Aquele ficheiro, chamado HOSTS.TXT, era gerido pelo NIC (Network Information Center), que determinava se um nome era aceitável (eram recusados nomes obscenos ou que entrassem em conflito com outros já existentes). O ficheiro era então copiado pelos administradores dos *sites* que desejassem utilizar o serviço. Por volta de meados de 1986 o ficheiro continha 3100 nomes registados e cerca de 6400 em 1990.

Este esquema tinha várias desvantagens:

- A gestão centralizada do espaço de nomes, cada vez mais complexa à medida que novos *sites* iam aderindo ao Internet;
- Com o crescimento do número de *hosts* ligados a ocorrência de conflitos era cada vez mais frequente;
- Uma vez que novos *hosts* eram constantemente adicionados à lista e outros já existentes mudavam de endereços ou desapareciam, a garantia de se possuir uma cópia actualizada era cada vez menor.

1.1.2. Nomes hierárquicos

Dados os problemas existentes no mecanismo apresentado acima foi necessário encontrar outro tipo de resposta ao problema da designação. A solução encontrada baseava-se na descentralização da gestão, através da delegação de parte do espaço de nomes, distribuindo a responsabilidade pela mapificação de nomes em endereços.

Esta distribuição seria feita duma forma hierárquica, à semelhança da estrutura de uma grande organização, de forma a facilitar a delegação de responsabilidade, estimulando a autonomia de cada nível da hierarquia.

A sintaxe dos nomes neste esquema hierárquico seria do tipo *A.B*, em que *B* é uma das divisões do nível superior do espaço de designação e *A* é uma das divisões de *B*. Por seu turno, *A* também poderia ser dividido, e assim sucessivamente.

1.1.3. Nomes no DNS

O espaço de nomes do DNS é, conceptualmente, organizado numa estrutura em árvore (Fig. 1). Esta tem uma raíz ("*root*"), sem nenhum nível acima, sendo vista como um "pai" para os níveis inferiores. A árvore consiste num conjunto de nós ligados por ramos. A cada nó é associado um *label* que tem de ser distinto de todos os outros situados ao mesmo nível e sob um pai comum.

Cada nó da árvore introduz uma partição (sub-árvore) na mesma, denominada por *domínio*. Cada domínio pode ainda ser dividido em *sub-domínios*.

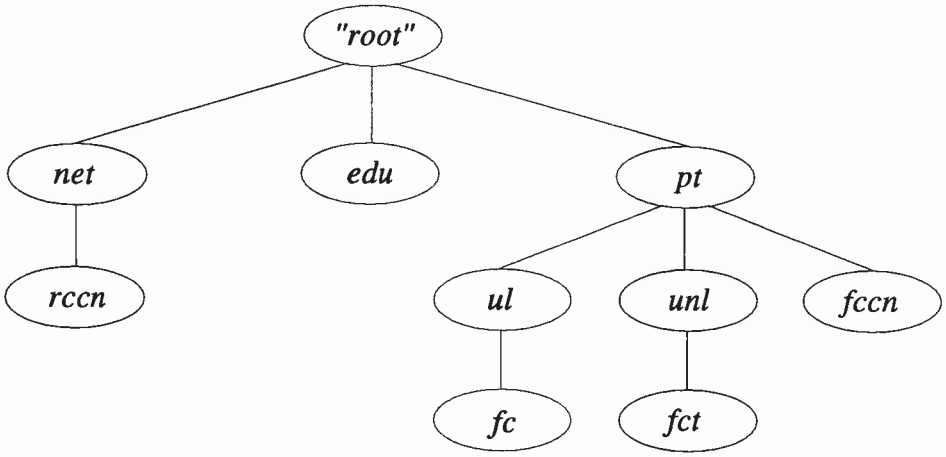

Fig. 1 - O espaço de designação do DNS.

O nome completo (*fully qualified domain name - FQDN*) de um nó é formado pela concatenação dos *labels* que constituem o caminho desde o nó em causa até à raíz, separados por pontos ("."). Assim, o nome completo da máquina *morgaine*, da Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa é *morgaine.fct.unl.pt.*

Pt é o código utilisado para designar Portugal; a designação de cada país é dada por um código de duas letras, definido pelo standard ISO para nomes de países (ISO 3166). Estes nomes são chamados *top-level domains*. Existe, além de nomes de países, mais uma série de *top-level domains*, quase todos utilizados para designar domínios de instituições americanas, a saber:

GOV Instituições goernamentais, por exemplo a NASA (*nasa.gov*).

EDU Instituições de educação, como a Universidade de Berkeley (*berkeley.edu*)

COM Empresas comerciais, como a Sun Microsystems (*sun.com*)

MIL Organizações militares, como a Marinha dos Estados Unidos (*navy.mil*).

INT Organizações internacionais, por exemplo a NATO (*nato.int*).

NET Organizações de *networking*, por exemplo a RCCN (*rccn.net*).

ORG Outro tipo de organizações não comerciais, que não se enquadrem em nenhuma das anteriores, como a Open Software Foundation (*osf.org*).

Assim, através dos *top-level domains* é possível definir conceptualmente duas hierarquias de designação, uma geográfica e outra organizacional.

1.2. Componentes do DNS

Quando um utilizador necessita de efectuar a tradução de um nome num endereço IP recorre a um agente local, providenciando-lhe um conjunto de parâmetros que aquele, denominado *resolver*, organiza segundo um formato bem definido para formular uma pergunta a um outro agente, local ou remoto, chamado *name server*. Este é responsável pela obtenção da resposta, ou indicação de um erro, se for caso disso, que devolve ao *resolver*. É desta forma, bastante simplificada, que funciona a resolução de nomes segundo o DNS, empregando clientes (*resolvers*) e servidores (*name servers*).

1.2.1. *Name servers*

Os programas que armazenam a informação sobre nomes no DNS são chamados *name servers*. São estes que armazenam os dados relativos a partes específicas do espaço de designação, denominados por *zonas*[1].

Estes servidores dizem-se autoritários sobre toda a informação contida nas zonas que servem, sendo aquela armazenada localmente (na máquina em que o *name server* corre) em ficheiros. Para garantir a redundância e acessibilidade dos seus dados, uma zona deve ser servida por mais do que um *name server*, e cada um destes pode gerir mais do que uma zona. Mais adiante, quando nos ocuparmos da configuração de um *name server*, estas questões serão analisadas mais profundamente.

A principal função de um *name server* é responder a perguntas (*queries*) feitas por clientes (*resolvers*) ou por outros servidores. Este serviço é providenciado de uma das seguintes formas:

iterativa O servidor (a) fornece a resposta, (b) indica que houve um erro (o nome perguntado não existe, por exemplo) ou (c) aponta um outro servidor como sendo o mais apropriado para fornecer a resposta. Neste último caso o cliente é responsável por continuar o processo, utilizando o servidor indicado.

recursiva Se o servidor não conhecer a resposta contacta outros *name servers* até a obter. Isto transfere toda a responsabilidade para o servidor, não lhe sendo permitido fornecer uma indicação para outro *name server*. Desta forma o cliente só tem de esperar a resposta, ficando o seu trabalho bastante simplificado.

Como foi dito, os *name servers* armazenam localmente a informação sobre as zonas de que são autoritários, estando assim aptos para, em qualquer momento, responderem a perguntas sobre nomes contidos nessas zonas. Para outros nomes terão de contactar ou re-dirigir as perguntas para outros *servers*. Mas para que não tenham de contactar constantemente outros servidores sobre nomes de que não são autoritários, cada *name server* mantém uma outra forma de armazenamento (em memória) de nomes, denominada por *cache*. Nesta são guardados os dados obtidos através de consultas a outros

[1] A diferença entre um domínio e uma zona é subtil. Considere-se o domínio *pt*, que tem como sub-domínios, entre outros, *unl.pt* e *fccn.pt*. A informação relativa ao domínio *pt* engloba os dados de *unl.pt* e de *fccn.pt*, enquanto que a zona *pt* está confinada ao nó *pt* da árvore DNS.

servers. Após receber uma *query* o servidor procura a resposta na sua *cache* e, caso a encontre aí, devolve imediatamente a resposta ao cliente. Deste forma ganha-se em eficiência na medida em que os tempos de resposta baixam consideravelmente e não existe a necessidade de estar constantemente a contactar outros servidores. Isto só acontecerá se o nome pretendido não estiver na *cache* (passará a estar a seguir ao contacto com o outro *server*).

A cada nome guardado em *cache* é associado um *timer* que quando chega ao fim obriga a que a informação armazenada (para a qual o servidor não é autoritário) seja retirada da *cache*. Desta forma os dados não são mantidos eternamente, o que seria perigoso, dado que a qualquer altura poderão ser modificados nos *name servers* autoritários.

1.2.2. *Resolvers*

Resolvers são os programas que implementam os clientes do DNS. Os utilizadores que necessitem de serviços do DNS contactam os *resolvers* que asseguram a comunicação com os *name servers* enviando-lhes as *queries*, interpretam o resultado e devolvem-no aos utilizadores. Estes utilizadores são, em geral, aplicações que utilizam o DNS, como sejam o *ftp, telnet, sendmail*, etc.

Os *resolvers* são, tipicamente, bibliotecas de rotinas utilizadas no código dos programas referidos, que só sabem fazer perguntas aos servidores e retornar as respostas ao utilizadores. Há, no entanto, casos em que os *resolvers* fazem mais do que isso, sendo capazes de construir a sua própria *cache*, com informação que vão obtendo dos *name servers*.

1.2.3. *Resource records*

Os dados associados aos nomes são armazenados pelos servidores (quer em ficheiros quer nas *caches*), sob a forma de *resource records* (RRs). Uma zona não é mais do que um conjunto de RRs, onde se define informação tão diversa como endereços de *hosts*, permitindo assim a tão desejada tradução de nomes em endereços e vice-versa, os *name servers* para o domínio em questão, as máquinas que fazem de *backup* para a entrega de *mail*, definição de *aliases* para *hosts*, informação àcerca do tipo de máquinas e sistemas operativos utilizados, serviços implementados, etc. A Tabela 1 contém uma lista de todos os RRs em utilização no DNS.

Cada RR é constituído por 6 componentes, a saber:

owner — Contém o nome do nó (da árvore DNS) ao qual o RR pertence.

time to live (TTL) — Define o período durante o qual este RR pode permanecer na *cache* de um *name server* não autoritário.

class — Cada RR pertence a uma classe, segundo o tipo de *software* ou rede que o suportam. Existem três classes: uma baseada nos protocolos Internet, outra na Chaosnet e a última no software Hesiod, sendo a primeira a mais utilizada.

Tipo	Significado	Observações
A	Host address	
NS	Name server	
MD	Mail destination	obsoleto
MF	Mail forwarder	obsoleto
CNAME	Canonical name	
SOA	Start of authority	
MB	Mailbox domain name	
MG	Mailbox member	
MR	Mail rename domain	
NULL	Null RR	
WKS	Well-known services	
PTR	Pointer	
HINFO	Host information	esperimental
MINFO	Mailbox information	
MX	Mail exchanger	
TXT	Text	
RP	Responsible person	experimental
AFSDB	AFS service	experimental
X25	X.25 address	experimental
ISDN	ISDN address	experimental
RT	Route trough	experimental

Tabela 1. Resource Records

type — Tipo de RR (ver Tabela 1).

data — Campo de comprimento variável que contém a descrição do recurso associado ao RR, variando segundo o tipo e classe do *record*.

data length — Comprimento, em bytes, do campo de dados.

Os RRs mais utilizados são os SOA, NS, A, MX, PTR e CNAME. Em menor escala os TXT, HINFO e WKS RRs são também utilizados, dentro da classe Internet.

2. Montagem e administração dum domínio

Nesta secção debruçar-nos-emos sobre o processo de criação de um domínio, o funcionamento do seu serviço de designação e as actividades relacionadas com a manutenção e evolução daquele. O caso particular de Portugal será utilizado como referência, com indicação dos procedimentos concretos a seguir em cada caso. Assim sendo, os administradores dos domínios portugueses, mais concretamente dos sub-domínios de *.PT*, poderão ver neste texto um guia para a gestão dos seus domínios.

2.1. Registo dum domínio

Em Portugal uma instituição com autonomia administrativa e financeira (empresa, universidade, instituto, etc.), pública ou privada, tem o direito de registar um sub-domínio de *.PT* cujo nome corresponda a uma sigla ou abreviatura da designação legal daquela instituição. No caso de conflito de designações terá prioridade a instituição que proceder primeiro ao registo.

É ainda possível a uma instituição registar domínio com o nome de outra, desde que esta última tenha direito à posse do mesmo e exista uma relação contratual entre as duas. Não são aceites registos de nomes por ou de pessoas individuais.

O pedido deve ser feito à FCCN - Fundação Nacional para o Desenvolvimento dos Meios Nacionais de Cálculo Científico, que, por delegação do NIC, gere o domínio *.PT*. Aquele pode ser enviado por carta para a seguinte morada:

```
FCCN - Fundação para o Desenvolvimento dos
Meios Nacionais de Cálculo Científico
(a/c: Administrador do domínio .PT)
Av. do Brasil, 101
1799 LISBOA CODEX
PORTUGAL
```

A carta deverá ser em papel timbrado da instituição que procede ao registo e assinada por um seu representante legal.

No sentido de permitir a aceleração do processo, admite-se que o formulário seja enviado por procuração, através de correio electrónico, para o endereço:

```
request@dns.pt
```

desde que:

1) A FCCN reconheça idoneidade à instituição emissora da mensagem para o efeito de registo de sub-domínios;
2) A instituição emissora tenha capacidade para representar para este efeito a instituição cujo sub-domínio é registado;

3) A instituição emissora possa responder pelas condições técnicas e formais do registo.

É ainda necessário registar os dados relativos ao domínio na base de dados do RIPE - Réseaux IP Européens, de forma a que a informação fique acessível no exterior.

O formulário para pedidos de registo de domínios pode ser solicitado para os endereços acima indicados, e está acessível por FTP anónimo na máquina que gere o servidor primário do domínio *.PT* (*ns.dns.pt*), no ficheiro */pub/dns/rccn/pt-domain-template.txt*.

2.2. Montagem do serviço de designação dum domínio

Após a recepção do pedido de registo, a FCCN certifica-se que está garantido o serviço de designação relativo ao novo domínio, consultando os *name servers* indicados no pedido. Só depois de ter a garantia de que o serviço está a funcionar correctamente é que o registo é efectuado.

Nesta secção procederemos à montagem deste serviço para o domínio *fccn.pt*, pertencente à propria FCCN. Como é obvio, a escolha deste domínio é arbitraria; os dados apresentados não correspondem necessariamente à realidade e servem apenas para ilustrar o processo.

Para montar o serviço vamos usar o software BIND (Berkeley Internet Name Domain), que se constitui na implementação de longe mais utilizada do DNS.

Em seguida discute-se a parametrização do cliente e do servidor, de forma a poder montar um domínio e pôr o DNS a funcionar.

2.2.1. Configuração do servidor

No BIND o servidor é implementado por um *daemon* chamado *named*. A sua configuração, no caso geral, requer a construção do seguinte conjunto de ficheiros[2] :

- *named.boot* Ficheiro de arranque do *named*, indicando em particular a localização dos dados a usar pelo servidor.

- *domínio.db* Ficheiro de descrição da zona, usado normalmente para tradução de nomes em endereços e para indicação dos dados relativos à autoridade sobre o domínio.

- *in-addr.db* Ficheiro utilizado para tradução de endereços IP em nomes de *hosts* (*reverse--mapping*).

- *local.db* Definição do endereço de *loopback*.

- *root.db* Lista dos servidores utilizados para obter informações sobre *root*.

[2] Os nomes apresentados são genéricos. Desde que assinalados devidamente, quaisquer outros nomes podem ser utilizados.

Nesta secção estudaremos os aspectos mais directamente relacionados com a operação do servidor, sendo introduzida a distinção entre os vários tipos de servidores e a parametrização do ficheiro de arranque.

2.2.1.1. Tipos de servidores

Existem três tipos de configurações possíveis para um servidor:

caching-only Um servidor configurado como *caching-only* não dispõe de nenhum ficheiro com informação DNS. Esta é obtida, na sua totalidade, como resultado das respostas que vai obtendo e armazenando na sua *cache*. Todos os servidores armazenam informação na *cache*, no entanto apenas os *caching-only* dependem totalmente desta técnica para obter informação. Um servidor destes não detém autoridade sobre nenhum domínio (à excepção, eventualmente, de *0.0.127.in-addr.arpa*).

primário Um servidor primário de um domínio é a fonte de toda a informação àcerca desse domínio, sendo autoritário sobre o mesmo, sendo aquela obtida a partir dos ficheiros mantidos localmente. De notar que este é o único tipo de servidor que requer um conjunto completo de ficheiros de configuração.

secundário A configuração de um servidor secundário de um domínio apenas requer os ficheiros *named.boot*, *local.db* e *root.db*. A informação propriamente dita sobre o domínio é obtida através daquilo a que se chamam *transferências de zona*, em que os dados mantidos por um servidor são transferidos para outro. O secundário efectua transferências de zona de outros servidores, geralmente de primários, guardando a informação em ficheiros. A partir daqui o servidor pode responder de forma autoritária a perguntas àcerca dos domínios em questão.

Um servidor pode ser configurado para operar numa ou mais de entre estas configurações. No caso geral, um servidor é primário de um ou mais domínios, é secundário de outros e mantém uma *cache* com informação diversa sobre os mais variados domínios que constituem o DNS mundial.

Enquanto a manutenção de um servidor *caching-only* não oferece nenhuma dificuldade especial, já a operação de um primário reveste-se de maior responsabilidade, uma vez que inclui a criação e manutenção do(s) ficheiro(s) de descrição de zonas (tratados mais adiante), modificações e propagação destas pelos secundários, e dum modo geral, a garantia da correcção e disponibilidade de toda a informação relativa ao domínio do qual é primário.

Para efeitos de robustez (tolerância a falhas num servidor), um domínio deve ter mais do que um servidor, no mínimo dois, um primário e um secundário[3]. A operação deste, enquanto tal, implica

[3] Não é necessariamente assim, um domínio pode ter mais do que um primário, mas esta prática é de todo desaconselhada.

apenas o conhecimento de um outro servidor para o domínio, geralmente o primário, do qual se obtém a informação através de transferências de zona.

2.2.1.2. Parametrização do ficheiro de arranque

Para que, após o arranque do sistema ou depois de uma actualização dos ficheiros anteriormente descritos, o servidor saiba onde os encontrar e o que fazer com a informação que cada um contém, é necessário configurá-lo adequadamente. Essa configuração faz-se por meio de um ficheiro, geralmente */etc/named.boot*, que contém as directivas necessárias para o servidor saber o que tem a fazer. O formato deste ficheiro não é definido em nenhum standard, sendo específico do BIND, podendo ainda variar conforme as versões deste software. Apresentam-se a seguir um conjunto de directivas que em geral são implementadas em todas as versões.

Em geral estes ficheiros começam por indicar ao servidor em que directório se encontram os ficheiros de parametrização dos domínios (possivelmente organizados em sub-directórios daquele). O servidor muda para aquele directório antes de ler os ficheiros, pelo que os nomes destes podem ser relativos. A directiva é a seguinte:

```
directory    /usr/lib/namedb
```

Num servidor primário o ficheiro de configuração contém uma linha por cada domínio do qual é autoritário, indicando o ficheiro que contém a informação sobre aquele domínio:

```
primary       fccn.pt              primary/fccn.db
```

Para servidores secundários a linha é semelhante, a menos da primeira palavra. Admitindo que a *ns.fccn.pt* era secundária de *dns.pt* teriamos:

```
secondary     dns.pt               secondary/dns.pt.db
```

Falta apenas indicar ao servidor onde encontrar a informação relativa a *root*:

```
cache                .                    cache/root.db
```

Assim, o ficheiro de configuração da *ns.fccn.pt* teria o seguinte aspecto:

```
; BIND data file to boot a primary name server.
;
; directory where all the data files are stored
;
directory    /etc/namedb
;
;
; type       domain                       source host/file
;
```

```
primary      fccn.pt                        primary/fccn.db
primary      236.26.192.in-addr.arpa        primary/192.26.236.db
primary      239.122.192.in-addr.arpa       primary/192.122.239.db
primary      0.0.127.in-addr.arpa           primary/127.0.0.db
;
secondary    dns.pt                         secondary/dns.pt.db
;
; load the cache data last
cache        .                              cache/root.db
```

O ponto-e-vírgula (";") inicia um comentário, que se estende até ao fim da linha. As linhas *primary* com nomes terminados em *in-addr.arpa* referem-se a *reverse-mapping*. Mais adiante este assunto será discutido em pormenor

Existe mais uma série de directivas, todas elas relacionadas com aspectos particulares do funcionamento do servidor, como por exemplo a restrição de transferências de informação relativa a determinadas zonas e a utilização de outros servidores para obter respostas. Informação sobre estas directivas pode ser encontrada em bibliografia especializada, indicada no fim deste texto..

2.2.2. Compilação da informação DNS

O passo seguinte é a criação dos ficheiros destinados a fazer a tradução de nomes em endereços IP e vice-versa, indicar ao servidor onde deve procurar respostas que não conhece, informação sobre endereços locais, etc. (os ficheiros **.db*, apresentados anteriormente). São estes os ficheiros utilizados pelo servidor para obter a informação necessária sobre o domínio debaixo da sua autoridade, sendo esta transferida para outros servidores que apoiem este domínio.

Apenas um àparte para referir o facto de que a leitura destes ficheiros é facilitada com a introdução de linhas em branco e de comentários. Estes, tal como no *named.boot*, são iniciados por um ponto-e--vírgula e terminam no fim da linha.

2.2.2.1. Descrição da zona

Os dados que descrevem o domínio em termos de autoridade sobre o mesmo, definição dos seus servidores, endereços dos *hosts*, *backup* para *mail*, etc. estão contidos no ficheiro *fccn.db*.

Assim, comecemos por definir a informação relativa à autoridade sobre este domínio. Esta faz-se através do seguinte SOA RR:

```
fccn.pt.     IN   SOA  ns.fccn.pt.   dnsop.fccn.pt. (
                  1993071203     ; serial
                  28800          ; refresh     -  8 horas
                  7200           ; retry       -  2 horas
                  604800         ; expire      -  7 dias
                  86400 )        ; default TTL -  1 dia
```

Este *record* indica e que o domínio *fccn.pt* tem como primário a máquina *ns.fccn.pt* e *dnsop@fccn.pt* é o endereço para onde se deve enviar o *mail* sobre problemas (e não só) relativos ao domínio. A notação deste endereço no RR é ligeiramente diferente da de um endereço de *mail*, e é assim devido a razões historicas. Os servidores não utilizam estes endereços; eles são exculsivamente destinados a uso humano.

O campo IN indica que este RR pertence à classe Internet, como, aliás, todos os *records* que iremos definir ao longo deste texto.

É de notar a presença de um ponto (".") no final de cada nome. Mais adiante nos referiremos à sua função.

O campo 1993071203 define o número de série do ficheiro. De cada vez que se faz uma alteração no ficheiro, por menor que seja, este número tem de ser incrementado para avisar os servidores que houve uma mudança. O esquecimento deste pormenor conduz à não propagação dos novos dados para os outros servidores (secundários). A notação aqui utilizada é do tipo *YYYYMMDDVV*, onde *YYYY* é o ano, *MM* o mês, *DD* o dia e *VV* indica o número da versão no mesmo dia, podendo desta forma fazer-se 100 alterações por dia. Este formato é recomendado devido à facilidade de leitura que introduz, no entanto cada administrador é livre de formar o *serial number* como quiser, desde que não se esqueça de o incrementar de cada vez que fizer alterações.

Os quatro últimos números definem uma série de intervalos (*timers*) em segundos e são destinados essencialmente aos servidores secundários:

refresh — frequência com que o secundário deve consultar o primário para saber se tem a informação actualizada.

retry — se o secundário não conseguir contactar o primário apos o período de *refresh*, deverá começar a tentar periodicamente com uma frequência definida por *retry*.

expire — se entretanto se mantiver a impossibilidade de contacto e passar o tempo definido por *expire*, o secundário deve expirar os dados, i.e., deixa de responder a perguntas sobre o domínio, eliminando os dados da sua *cache*.

default TTL — quando o servidor responde a uma pergunta, associa a essa resposta um TTL aos RRs que envia. Os outros servidores (não autoritários) guardam os dados em *cache* durante este período e depois deitam-nos fora.

Os valores destes *timers* podem revestir-se da maior importância: se forem muito baixos podem saturar os servidores com perguntas, visto que a informação reside muito pouco tempo nas *caches*. Se forem demasiado elevados, as alterações efectuadas levam muito tempo a serem propagadas, devido aos períodos excessivos que os dados antigos permanecem nas *caches*. Embora não haja nenhum conjunto de valores considerados óptimos, os seguintes são recomendados:

Para domínios de nível superior (*.PT*, por exemplo):

```
        86400 ; refresh      24 horas
         7200 ; retry         2 horas
      2592000 ; expire       30 dias
       345600 ; default TTL   4 dias
```

Para outros:

```
        28800 ; refresh      8 horas
         7200 ; retry        2 horas
       604800 ; expire       7 dias
        86400 ; default TTL  1 dia
```

mas neste caso, factores como a frequência das alterações, a velocidade de propagação desejada ou a acessibilidade do primário são determinantes na optimização daqueles valores

Em seguida vamos definir os *name servers* do domínio *fccn.pt*. Temos o primário, *ns.fccn.pt*, e um secundário, *ns.dns.pt* (secundário de todos os sub-domínios de *.PT*). A definição é feita por meio de NS RRs:

```
fccn.pt.      IN   NS   ns.fccn.pt.
fccn.pt.      IN   NS   ns.dns.pt.
```

A seguir definem-se as políticas de *mail routing*, i.e., quais os servidores de *mail* aos quais o correio para máquinas de *fccn.pt* deve ser entregue, recorrendo a MX RRs:

```
fccn.pt.      IN   MX   10   ce.fccn.pt.
fccn.pt.      IN   MX   20   ns.fccn.pt.
fccn.pt.      IN   MX   30   mail.dns.pt.
```

Estes indicam que o *mail* para o domínio *fccn.pt* deve ser entregue a *ce.fccn.pt*. Caso isto não seja possível (a máquina está em baixo, por exemplo), deve-se tentar *ns.fccn.pt* e só no fim *mail.dns.pt*.

Por fim, falta introduzir a informação necessária à obtenção dos endereços IP dos *hosts* do nosso domínio. Para isso definimos os seguintes RRs:

```
localhost.fccn.pt.     IN   A      127.0.0.1

ns.fccn.pt.            IN   A      192.26.236.100
ns.fccn.pt.            IN   A      192.122.239.35

ce.fccn.pt.            IN   A      192.122.239.1

dsa.fccn.pt.           IN   A      192.122.239.2

bind.fccn.pt.          IN   CNAME      ns.fccn.pt.
```

Os A RRs definem os endereços dos *hosts* respectivos. Assim, é possível perguntar pelo endereço de *ce.fccn.pt* obtendo-se como resposta 192.122.239.1. A máquina *ns.fccn.pt* possui dois endereços, pelo que ambos têm de ser indicados através de A RRs.

O CNAME RR apresentado indica que *bind.fccn.pt* é um *alias* do nome *ns.fccn.pt* (*canonical name*). Desta forma também é possível obter o endereço IP de *bind.fccn*.pt. Este é o endereço de *ns.fccn.pt*, 192.26.236.100.

Fica assim completa a informação essencial àcerca do domínio *fccn.pt*. Existem outros RRs que podem ser adicionados, no entanto não são absolutamente indispensáveis, pelo que não são considerados aqui.

2.2.2.2 *Reverse-mapping*

Para que seja possível obter o nome de uma máquina dado o seu endereço é necessário que se disponibilizem os dados utilizados nessa tradução (*reverse-mapping*). Os *hosts* do nosso domínio têm endereços nas redes 192.26.236.0 e 192.122.239.0, pelo que temos de definir as zonas *236.26.192.in-addr.arpa* e *239.122.192.in-addr.arpa*. Além dos SOA RRs têm de ser indicados os PTR RRs que permitem a tradução de endereços em nomes e os NS RRs, visto estarmos a lidar com zonas "normais".

Assim, temos o ficheiro *192.26.236.db*:

```
236.26.192.in-addr.arpa.  IN   SOA  ns.fccn.pt. dnsop.fccn.pt. (
                  1993071200   ; serial
                  28800        ; refresh     -  8 horas
                  7200         ; retry       -  2 horas
                  604800       ; expire      -  7 dias
                  86400 )      ; default TTL -  1 dia

236.26.192.in-addr.arpa.       IN   NS    ns.fccn.pt.
236.26.192.in-addr.arpa.       IN   NS    ns.dns.pt.

100.236.26.192.in-addr.arpa.   IN   PTR   ns.fccn.pt.
```

As perguntas sobre endereços são feitas como se fossem nomes em DNS. O endereço é invertido e o sufixo *.in-addr.arpa* é-lhe acrescentado. Desta forma o servidor só tem de procurar o PTR RR adequado, que aponta para o nome do *host* pretendido.

O ficheiro *192.122.239.0.db* fica:

```
239.122.192.in-addr.arpa. IN  SOA  ns.fccn.pt. dnsop.fccn.pt. (
                  1993071200   ; serial
                  28800        ; refresh     -  8 horas
                  7200         ; retry       -  2 horas
                  604800       ; expire      -  7 dias
```

```
                    86400 )     ; default TTL -  1 dia

239.122.192.in-addr.arpa.      IN    NS     ns.fccn.pt.
239.122.192.in-addr.arpa.      IN    NS     ns.dns.pt.

1.239.122.192.in-addr.arpa.    IN    PTR    ce.fccn.pt.

2.239.122.192.in-addr.arpa.    IN    PTR    dsa.fccn.pt.

35.239.122.192.in-addr.arpa.   IN    PTR    ns.fccn.pt.
```

2.2.2.3. Endereço local

Além do *reverse-mapping* das redes pertencentes ao domínio, há que providenciar informação relativa a uma rede especial: a rede *loopback*, que as máquinas utilizam para re-dirigir o tráfego para elas próprias. No ficheiro *fccn.db* já foi incluído o endereço de um *host* desta rede, chamado *localhost*, também ele usado por qualquer máquina para se designar a ela própria.

Assim, a zona 0.0.127.in-addr.arpa também tem de ser descrita através de um ficheiro do tipo daqueles que construímos até agora a que chamaremos *127.0.0.db*. O seu conteúdo é o seguinte:

```
; BIND data file for local loopback interface.
;
0.0.127.in-addr.arpa.     IN   SOA  ns.fccn.pt. dnsop.fccn.pt. (
             199307120     ; Serial
             86400         ; Refresh     -  1 dia
             7200          ; Retry       -  2 horas
             2592000       ; Expire      - 30 dias
             345600 )      ; Default TTL -  4 dias

0.0.127.in-addr.arpa.      IN       NS       ns.fccn.pt.

1.0.0.127.in-addr.arpa.    IN       PTR      localhost.
```

O SOA record é semelhante aos acima apresentados, tem um *name server* (não precisa de mais!) e existe apenas um *host* nesta rede.

Todos os *sites* que pretendam utilizar o esquema do *loopback address* devem implementar uma configuração deste tipo, de forma a poderem usá-lo. Isto porque, devido às suas características especiais, a rede 127 não está sob a responsabilidade particular de ninguém, mas todos em geral são responsáveis por ela, localmente. É por este motivo que se deve construir o ficheiro *127.0.0.db*, com o tipo de informação apresentada acima. Em geral nunca mais será necessário pensar neste assunto, uma vez que os dados não irão ser alterados.

2.2.2.4. Os servidores de *root*

Para além da informação local, um servidor precisa de saber onde obter infomação àcerca de nomes

exteriores ao seu domínio. No caso de não conseguir obter essa informação por outras vias, ele inicia a sua busca a partir do topo, contactando os servidores do domínio "." (*root*). Para isso precisa de saber em cada momento quais são esses servidores e, uma vez que não pode obter essa informação perguntando "acima" de *root*, a única hipótese e guardá-la em algum sítio da sua base de dados. A solução é ter um ficheiro com a lista dos *root name servers*. Esse ficheiro, no nosso caso chamado *root.db*, tem o seguinte conteúdo:

```
; Initial cache data for root domain servers.
;

.            999999    IN   NS    NS.NIC.DDN.MIL.
.            999999    IN   NS    KAVA.NISC.SRI.COM.
.            999999    IN   NS    AOS.BRL.MIL.
.            999999    IN   NS    C.NYSER.NET.
.            999999    IN   NS    TERP.UMD.EDU.
.            999999    IN   NS    NS.NASA.GOV.
.            999999    IN   NS    NIC.NORDU.NET.
.            999999    IN   NS    NS.INTERNIC.NET.

;
;  Prep the cache (hotwire the addresses)
;  Order does not matter
;

NS.NIC.DDN.MIL.        999999    IN   A    192.112.36.4
KAVA.NISC.SRI.COM.         999999    IN   A    192.33.33.24
AOS.BRL.MIL.           999999    IN   A    128.63.4.82
AOS.BRL.MIL.           999999    IN   A    26.3.0.29
AOS.BRL.MIL.           999999    IN   A    192.5.25.82
C.NYSER.NET.           999999    IN   A    192.33.4.12
TERP.UMD.EDU.              999999    IN   A    128.8.10.90
NS.NASA.GOV.           999999    IN   A    192.52.195.10
NS.NASA.GOV.           999999    IN   A    128.102.16.10
NIC.NORDU.NET.         999999    IN   A    192.36.148.17
NS.INTERNIC.NET.       999999    IN   A    198.41.0.4
```

Esta informação é tida simplesmente como uma lista de "pistas" para obter a informação sobre *root*. Assim, a lista de *root name servers* pode ser qualquer, desde que através dela se obtenha a informação desejada. Muitos *sites* sem acesso (pelo menos completo) ao Internet utilizam um ficheiro *root.db* contendo uma lista de *hosts* que lhes providenciam a informação desejada, se bem que indirectamente. Desde que se garanta que aquela lista não é propagada para o exterior, esta situação é perfeitamente aceitável.

O ficheiro apresentado acima contém todos os *root name servers* actualmente em operação. Esta informação é muito estável por isso os servidores armazenam-na à parte na sua *cache* e não a deitam fora quando os TTLs chegam a zero (o número 999999 significa apenas "muito tempo" e não é

tratado da forma habitual). No entanto, de tempos a tempos, surgem alterações a esta lista, pelo que é conveniente que os administradores que desejem manter nos seus ficheiros *root.db* a informação completa e actualizada àcerca dos verdadeiros *root name servers* devem procurar formas de terem sempre a última versão daquela lista. Uma dessas formas é consultar o arquivo da *ns.dns.pt*, que contém no ficheiro */pub/dns/root-dns-servers.txt* a versão actualizada.

2.2.2.5. Abreviaturas

Até aqui apresentámos os ficheiros necessários para a operação do primário de *fccn.pt*. No entanto existe uma série de abreviaturas que permitem uma maior legibilidade dos ficheiros, evitando a escrita de informação repetida. Estas incluem:

i) A omissão da origem, no fim de cada nome. A cada nome incluído num ficheiro é associada uma origem. Esta consiste num nome que é acrescentado a todos os nomes que não terminem com um ponto final ("."). Este nome é, por defeito, o do segundo campo da directiva *primary* ou *secondary*, no ficheiro *named.boot*, podendo, no entanto, ser modificada pela directiva *$ORIGIN*. Assim, no ficheiro *fccn.db*, no qual a origem é *fccn.pt*, em vez de

```
ns.fccn.pt.   IN    A    192.26.236.100
```

poderíamos ter escrito

```
ns            IN    A    192.26.236.100
```

No entanto esta facilidade introduz um problema encontrado demasiadas vezes, o chamado "*missing trailing dot problem*". Se tivessémos escrito

```
fccn.pt.      IN    NS   ns.dns.pt
```

quando alguém perguntasse pelos *name servers* de *fccn.pt* obteria, entre outros, *ns.dns.pt.fccn.pt*. Uma vez que não se colocou um ponto no fim do nome *ns.dns.pt* este é transformado com a adição da origem. Por esta razão há que ter muito cuidado quando se escrevem nomes que não pertençam à origem corrente.

ii) Se o nome do RR for o mesmo da origem, aquele pode ser substituído por "@". Esta notação é encontrada frequentemente em SOA records. Por exemplo:

```
@   IN    SOA   ns.fccn.pt.    dnsop.fccn.pt. (
                199307120      ; Serial
                28800          ; Refresh     -  8 horas
                7200           ; Retry       -  2 horas
                604800         ; Expire      -  7 dias
                86400 )        ; Default TTL -  1 dia
```

iii) Se o nome de um RR (o que começa na primeira coluna) for o mesmo do anterior, então pode-se

substituí-lo por um espaco ou *TAB*. Assim,

```
fccn.pt.      IN   NS   ns.fccn.pt.
fccn.pt.      IN   NS   ns.dns.pt.
```

é equivalente a

```
fccn.pt.      IN   NS   ns.fccn.pt.
              IN   NS   ns.dns.pt.
```

Assim, os ficheiros acima apresentados poderiam ter sido escritos de forma abreviada, aumentando a sua legibilidade. Por exemplo, o ficheiro *fccn.db* ficaria:

```
@  IN   SOA  ns.fccn.pt.   dnsop.fccn.pt. (
        199307120      ; Serial
        28800          ; Refresh     -  8 horas
        7200           ; Retry       -  2 horas
        604800         ; Expire      -  7 dias
        86400 )        ; Default TTL -  1 dia

        IN   NS   ns.fccn.pt.
        IN   NS   ns.dns.pt.

        IN   MX   10   ce.fccn.pt.
        IN   MX   20   ns.fccn.pt.
        IN   MX   30   mail.dns.pt.

localhost     IN   A    127.0.0.1

ns            IN   A    192.26.236.100
              IN   A    192.122.239.35

ce            IN   A    192.122.239.1

dsa           IN   A    192.122.239.2

bind          IN   CNAME    ns
```

2.2.2.6. Notificação do servidor

Para que o servidor tome em conta a configuração acabada de estabelecer é necessário "avisá-lo". Isto faz-se por meio de um sinal de *hangup* (*kill -HUP*, no sistema UNIX) ao *named*, o que leva o servidor a reler o ficheiro *named.boot* e todos os ficheiros que neste são indicados, para ver se houveram algumas alterações aos números de série dos SOA RRs.

Depois disto a parametrização do domínio está completa e, desde que esteja correcta, aquele fica imediatamente operacional.

2.2.3. Configuração do *resolver*

Como já foi referido, o *resolver* é a parte do cliente do BIND. Aquele não é um processo separado, mas sim uma biblioteca de funções utilizada por outros processos que necessitem de aceder ao DNS.

Existem duas alternativas para configurar o *resolver*: ou se usa a configuração por defeito ou cria-se um ficheiro para o fazer. De seguida são estudadas as duas abordagens.

2.2.3.1. Configuração por defeito

Este tipo de configuração destina-se a sistemas que operem um servidor (*named*) e funciona da seguinte forma:

- o *host* local é o *name server* a quem recorrer em todas as situações;
- o nome do domínio local é determinado pelo resultado do comando *hostname*, retirando a parte relativa ao nome do *host*.

Este segundo aspecto implica que o nome da máquina esteja convenientemente configurado, i.e., *hostname* deve mostrar um FQDN. Para que isto aconteça basta executar o seguinte comando (em modo *superuser*):

```
# hostname ns.fccn.pt
```

Desta forma o *resolver* sabe que o domínio local se chama *fccn.pt*.

Se *hostname* retornar apenas o nome da máquina então tem de se recorrer a um ficheiro de configuração para que o resolver possa funcionar. Há, no entanto, um caso em que isto não se verifica: em sistemas que utilizem o Network Information System (NIS), *hostname* é utilizado para definir o nome da máquina e *domainname* o do domínio NIS. Uma vez que este é usado como o domínio DNS por defeito, é aconselhável que ambos os sistemas usem o mesmo nome.

2.2.3.2. Ficheiro de configuração

O recurso ao ficheiro de configuração (tipicamente */etc/resolv.conf*) faz-se em sistemas que não estejam a correr o *named* ou quando *hostname* não retorna um FQDN. Aquele é um ficheiro com uma sintaxe bastante simples e, embora existindo algumas variantes, tem geralmente dois tipos de entradas:

domain nome

Define o nome do domínio a utilizar por defeito. A qualquer nome de *host* que não contenha nenhum ponto o *resolver* acrescenta o nome deste domínio.

nameserver endereço

Identifica, pelo endereço IP, um servidor a que o *resolver* pode recorrer. Geralmente podem haver até três *nameservers* indicados, sendo questionados segundo a ordem pela qual aparecem no ficheiro. Se um não responder recorre-se ao seguinte na lista. Se existir um servidor localmente este só é questionado se houver uma entrada que explicitamente o mencione.

Assim, o aspecto mais comum do ficheiro *resolv.conf* é o seguinte:

```
domain fccn.pt
nameserver 127.0.0.1
nameserver 192.122.238.22
nameserver 192.84.62.1
```

De notar que esta configuração é equivalente à configuração por defeito, desde que se verifiquem as condições referidas em 2.2.3.1. Caso não esteja a correr nenhum servidor localmente o que haveria a fazer era retirar a linha correspondente ao endereço 127.0.0.1.

2.3. Manutenção do domínio

Agora que o nosso domínio já existe e está a funcionar o serviço de designação para o mesmo, surge uma série de tarefas relacionadas com a manutenção do bom funcionamento daquele serviço, bem como com a gestão do domínio, nomeadamente alterações na sua configuração.

2.3.1. Actividade e configuração do servidor

É conveniente que a actividade do servidor seja regularmente acompanhada de forma a poder detectar anomalias. Em geral o *named* deixa mensagens em ficheiros de *log* (via *syslogd(8)*) ou na consola do sistema. A sua consulta é essencial para um acompanhamento adequado da operação do servidor.

Por outro lado é também necessário ter a certeza de que as configurações do servidor e dos domínios estão correctas, caso contrário uma série de problemas podem surgir. O recurso a ferramentas de *debug* como o *nslookup(1)*, *dig(1)* ou *ddt(1)* é um bom auxiliar neste sentido. Uma vez que os manuais *on-line* daqueles programas são suficientemente esclarecedores, uma descrição dos mesmos não é aqui incluída. Recomenda-se vivamente a consulta daqueles manuais e, sobretudo, a utilização dos programas.

2.3.2 Alterações nas zonas

Com o decorrer do tempo vão sendo necessárias algumas alterações aos ficheiros de descrição dos domínios. Algumas destas alterações são simples e, desde que não sejam de grande volume, não implicam nenhuma acção especial (por exemplo, a adição de novas máquinas). A única precaução a ter sempre em mente é o incremento do número de série no SOA.

Mas por vezes é necessário proceder a alterações de fundo (mudanças de endereços de rede de todos os *hosts*, ou mesmo que seja só do endereço do primário, por exemplo). Esse processo deve ser efectuado obedecendo a um conjunto de regras, dada a importancia da informação envolvida, para garantir que a alteração é feita sem provocar grandes problemas a nível do DNS, tanto nacional como internacional. Essas regras, a respeitar pelo administrador do primário, são as seguintes:

1 - Avisar com antecedência todos os gestores dos secundários e obter deles a confirmação de disponibilidade para executar a mudança num determinado dia e a uma determinada hora.

2 - Não fazer a mudança nos últimos dias da semana.

3 - Dois ou três dias antes da mudança, descer os parâmetros:

```
Refresh para 2 horas      (7200)
Retry para 1 hora         (3600)
Expire para 10 horas     (36000)
Default TTL para 1 hora   (3600)
```

no SOA do domínio, sem esquecer de incrementar o número de série, e esperar que os secundários tenham esta nova versão.

Esta medida destina-se a acelarar o processo de expurgar as incoerências do DNS mundial e a evitar o impacto de erros.

4 - Subir aqueles parâmetros para os valores normais logo que haja a garantia que tudo está normal (2 ou 3 dias depois).

2.3.3. Criação de sub-domínios

Quando um domínio ganha uma certa dimensão, a quantidade de trabalho inerente à sua administração leva o administrador a considerar a hipótese de distribuir ou delegar aquelas tarefas noutras pessoas. Noutras situações, a instituição a quem pertence o domínio está organizada de tal forma que é conveniente distinguir a quem, dentro da organização, pertencem as várias máquinas (veja-se o caso duma universidade e das suas faculdades e departamentes, por exemplo).

Nestes casos o que o administrador tem a fazer é criar sub-domínios, delegando neles a autoridade sobre os recursos que lhes pertencem e a responsabilidade da sua gestão.

No entanto um administrador só deve delegar a autoridade se tiver a garantia de que os futuros responsáveis pelo sub-domínio têm condições de o administrar correctamente.

Do lado do administrador do domínio o que há a fazer é incluir no ficheiro de configuração deste a informação sobre os servidores dos sub-domínios. Imaginemos que se criava o domínio *sccn.fccn.pt*, cujo primário seria *dns.sccn.fccn.pt* e *dsa.fccn.pt* seria secundário. Em *fccn.db* teriam de ser acrescentados os seguintes RRs:

```
sccn     IN   NS   dns.sccn.fccn.pt.
         IN   NS   dsa.fccn.pt.
```

Assim, se um cliente perguntar à *ns.fccn.pt* pelo domínio *sccn.fccn.pt*, aquela responde simplesmente (desde que não tenha informação para dar a resposta imediatamente) que os servidores desse domínio são *dns.sccn.fccn.pt* e *dsa.fccn.pt*. De seguida o cliente dirige-se a um daqueles dois *hosts* para obter a informação que deseja, visto ambos serem autoritários para aquele domínio.

Quando a *ns.fccn.pt* indicou os dois servidores de *sccn.fccn.pt*, de certeza que adicionou a essa resposta o endereço de *dsa.fccn.pt*, uma vez que o conhecia por estar dentro da sua jurisdição. Desta forma o servidor de *dsa.fccn.pt* poderia ter sido contactado directamente. Mas *dns.sccn.fccn.pt* não está sob a autoridade de *ns.fccn.pt*, visto esta ter sido delegada. Assim, a única maneira da *ns.fccn.pt* poder saber daquele endereço é tê-lo explicitamente indicado no ficheiro *fccn.db*, por meio de um A *record*:

```
dns.sccn      IN   A    192.26.236.200
```

A este tipo de RR, relativo a uma máquina que não pertence ao domínio em questão, chama-se *glue--record*. Neste caso era necessária a inclusão do *glue-record*, visto que para se chegar a *sccn.fccn.pt* é necessário passar primeiro por *fccn.pt*. Mas no caso da *ns.dns.pt* o mesmo não foi feito, visto o domínio *dns.pt* ser completamente independente de *fccn.pt*. Essa informação pode ser obtida exteriormente a este último domínio, pelo que não se incluíu o endereço da *ns.dns.pt* em *fccn.db*.

A razão disto prende-se com o facto de que o administrador de *fccn.pt* não pode ser responsável pela precisão dos dados relativos a domínios exteriores ao seu. Se um dia se mudasse o endereço da *ns.dns.pt*, essa mudança seria transparente para ele.

No entanto, para as máquinas servidoras de domínios delegados esta é a única solução, pelo que os *glue-records* são um mal necessário, que só devem ser usados nestes casos.

Neste caso a *ns.fccn.pt* não se constituíu em secundária de *sccn.fccn.pt*, pelo que nada mais há a fazer. Caso contrário, seria necessário acrescentar uma cláusula *secondary* no ficheiro *named.boot*, a indicar a nova situação.

Assim, depois de incrementar o número de série do SOA de *fccn.pt* e de ter avisado o *name server* àcerca da mudança, o domínio *sccn.fccn.pt* passa a estar operacional.

Bibliografia

Albitz, P. e *C. Liu*, DNS and BIND, O'Reilly & Associates Inc., Out. 1992

Black, U., TCP/IP and Related Protocols, McGraw-Hill, 1992

Comer, D. E., Internetworking with TCP/IP, Vol. I; Principles, Protocols and Architecture, 2nd Edition, Prentice-Hall, 1991

Dunlap, K., M. Karels e *P. Vixie*, Name Server Operations Guide for BIND - Release 4.9, 1992

Hunt, C., TCP/IP Network Administration, O'Reilly & Associates Inc., Ago. 1992

Lottor, M., Domain Administrators Operations Guide, RFC 1033, SRI International, Nov. 1987

Mockapetris, P., Domain Names - Concepts and Facilities, RFC 1034, Information Sciences Institute, Nov. 1987

Mockapetris, P., Domain Names - Implementation and Specification, RFC 1035, Information Sciences Institute, Nov. 1987

Stahl, M., Domain Administrators Guide, RFC 1032, SRI International, Nov. 1987

Anexo 2

Sincronização distribuida

Anexo 2

Sincronização distribuida

Introdução

A noção de tempo tem sido, desde sempre, fundamental para a existência humana. A prová-lo estão os diversos mecanismos desenvolvidos ao longo da História para medir o tempo. Começando por registar intervalos entre acontecimentos, duma forma mais ou menos subjectiva, o Homem foi desenvolvendo processos objectivos, cada vez mais apurados, para medir a progressão do tempo, desde os primeiros relógios de pêndulo no séc. XVII até aos relógios atómicos dos nossos dias [KOP92].

A obtenção de medições temporais o mais exactas possíveis é fundamental em aplicações científicas, em transacções legais e financeiras, em sistemas de transportes e distribuição, duma forma geral, em aplicações envolvendo recursos distribuidos. Da mesma importancia é a noção que duas ou mais entidades têm do tempo, i.e., como é que elas sabem que horas são? Ou, em termos mais abstractos, como saber se um dado acontecimento ocorreu antes de outro, como ordená-los em termos temporais [LAM78]? Quando este tipo de questão é posta em termos de eventos ocorridos numa dada máquina, a resposta é simples. No entanto, se cada evento ocorre em máquinas diferentes, é absolutamente necessário que os relógios de cada uma estejam de acordo em relação a hora que marcam.

Neste documento apresentam-se mecanismos destinados a permitir a obtenção deste tipo de acordo (a que chamaremos sincronização), em particular o protocolo usado no Internet para sincronizar os relógios de milhares de computadores distribuidos pelo mundo inteiro. Introduzem-se conceitos sobre o que e um serviço de sincronização (time service) e protocolos associados, mais numa perspectiva de "o que é feito" do que "como é feito". É, portanto, um texto de divulgação e não de especificação. Para conhecimento dos detalhes dos protocolos apresentados, bases matemáticas e tecnologia associada sugere-se a consulta da bibliografia apresentada no fim deste documento.

1. A medição do tempo

Para se conseguir uma sincronização efectiva entre os diversos relógios, ou seja, fazer com que todos marquem a mesma hora, primeiro é preciso saber que hora é essa. Não faz sentido que todos os computadores de um país estejam de acordo que em dado momento sejam 17h30m58s e os de outro (que, para simplificar, se situa no mesmo fuso horário) digam que são 17h31m01s. Para, em cada caso, poderem marcar determinada hora, os relógios tem de a obter a partir de alguma fonte, à qual

reconhecem a autoridade necessária para lhes dar essa informação como correcta. Na situação atrás referida essa autoridade teria um âmbito nacional. É então evidente que este conjunto de entidades também têm de estar sincronizadas para que todo o esquema faça sentido. Assim, torna-se necessária uma referência a nível global, e é em relação a esta que todos os relógios têm de se sincronizar. Se continua ou não a haver a necessidade de existirem autoridades de âmbito nacional ou regional é uma questão diferente e que discutiremos adiante.

1.1. Um pouco de História

Os primeiros calendários, enquanto sistemas de medição sistemática do tempo, surgiram na Antiguidade, formulados a partir de observações dos movimentos da Terra em volta do Sol e da Lua em volta da Terra. Há quatro mil anos foram feitas observações astronómicas de que resultariam o estabelecimento dos solstícios de Verão e de Inverno e cálculos da duração do ano (baseado no Sol) e do mês (baseado na Lua) bastante aproximados aquelas que hoje são universalmente aceites [MIL91]. Desde então estes sistemas têm conhecido sucessivas modificações, para as quais contribuiram povos como os Antigos Chineses, Egípcios, Maias e Gregos. Os próprios Romanos debruçaram-se seriamente sobre o tema, tendo sido responsáveis pela criação do chamado calendário Juliano, em uso em alguns países do Ocidente até ao princípio deste século.

O calendário Juliano já previa a introdução de um dia suplementar na duração do ano em anos divisíveis por 4 (bissextos). No entanto sofria de algumas deficiências, a principal das quais resultava numa discrepância entre a duração do ano, relativa ao ano solar, que em 1545 se cifrava já numa diferença de dez dias. Para resolver o problema, em Fevereiro de 1582 o Papa Gregório XII emitiu uma bula decretando que o dia seguinte a 4 de Outubro desse ano no calendário Juliano, seria 15 de Outubro, segundo o novo calendário, designado por Gregoriano. Além disso actualizava a duração do ano para 365,2425 dias, obrigando a que apenas os anos seculares divisíveis por 400 fossem anos bissextos. Presentemente o calendário Gregoriano é adoptado praticamente em todo o mundo.

1.2. Os standards do tempo

Para se poder medir grandezas temporais como a idade do Universo ou o decaimento de um protão é necessário ter uma referência para que as medições tenham sentido globalmente. Nesse sentido, a União Astronómica Internacional adoptou standards para a duração do segundo (9.192.631.770 períodos da radiação correspondente a transição entre dois níveis hiperfinos do átomo de césio-113), do dia (86.400 segundos) e do ano (365,25 dias) [MIL91].

Desde 1972 a frequência standard e baseada no Tempo Atómico Internacional (TAI), mantida com o auxílio e vários relógios de cesio, enquanto que do ponto de referência para o tempo civil e baseado no Tempo Universal Coordenado (UTC), determinado pelo Sistema Internacional de Pesos e Medidas, com o auxílio de observações astronómicas feitas pelo Observatório Naval dos Estados Unidos.

O UTC atrasa-se em relação ao TAI na ordem de uma fracção de segundo em cada ano. Quando a

magnitude da diferença se aproxima dos 0,7 segundos é inserido ou eliminado um segundo (*leap second*) da escala TAI. Estas alterações são feitas nos últimos dias dos meses de Junho ou Dezembro, não podendo ser previstas antecipadamente, sendo, como é óbvio, afectada a duração dos anos. Desde o início do estabelecimento do UTC (0h00 de 1 de Janeiro de 1972) já se inseriram 18 *leap seconds* no TAI, tendo a última vez ocorrido em 30 de Junho de 1993.

2. Serviços de sincronização

Um serviço de sincronização é um sistema que providencia informação de caracter temporal, destinada a ser usada pelos clientes para poderem acertar os seus relógios, relativamente a standards com autoridade reconhecida em termos de informação horária. Um serviço de sincronização distribuido, a funcionar no Internet, distribuindo o tempo exacto por vários milhares de computadores em todo o mundo é um serviço que necessita de especial cuidado na sua idealização e configuração, devido às características próprias do meio em que opera. Nesta secção introduziremos alguns conceitos relacionados com um serviço deste tipo e discutiremos os requisitos a que deve obedecer para que seja efectivo.

2.1. Terminologia

Neste documento seguiremos a nomenclatura e definições adoptadas em [MIL89], traduzidas para português sempre que a equivalência seja possível e faça sentido.

Assim, por *estabilidade* dum relógio entende-se a capacidade que este tem de manter constante a sua frequência, a *exactidão* é o grau de aproximação da sua frequência e hora as do standard nacional (ou da autoridade sob a qual opera, tal como apresentada em 1) e a *precisão* relaciona-se com a facilidade de manutenção destas quantidades num determinado sistema. O *offset* de dois relógios é a diferença horária entre eles. A *fiabilidade* dum relógio é o período em que este consegue operar num contexto distribuido, i.e., ligado em rede. Os relógios são mantidos em *servidores de tempo*, pertencentes a uma *rede de sincronização*, na qual cada servidor mede o *offset* entre o seu relógio e os dos seus vizinhos, designados por *pares*[1], na rede. *Sincronizar a frequência* significa ajustar os relógios da rede de maneira a progredirem segundo uma mesma frequência, *sincronizar o tempo* é pô-los de acordo com o UTC, tal como é definido pelos standards nacionais e *sincronizar os relógios* envolve a sincronização tanto a nível de frequências como de tempo.

Por fim, define-se *stratum* de um servidor de tempo como a distância (*hop count*) deste a uma fonte de sincronização de alta precisão (relógio atómico, por exemplo), sendo que um servidor directamente ligado a esta (*servidor primário*) tem *stratum* 1.

[1] Ao longo deste texto a palavra *par* aparece sempre neste contexto e nunca para designar uma quantidade de 2.

2.2. Caracterização do serviço

Para se conseguir uma sincronização efectiva e estável das frequências é necessário o acesso a relógios de alta precisão e estabilidade, assim como a possibilidade de poder comparar valores necessários a sincronização com outros sistemas, ao longo de períodos relativamente alargados. Por outro lado uma sincronização fiável do tempo requer um cuidadoso desenho dos algoritmos de selecção (ver adiante), e a utilização de fontes de sincronização redundantes e de caminhos diversos para lhes aceder. Tendo em conta o meio heterogéneo que é o Internet, com grandes variações a nível de tempos de trânsito e fiabilidade, devido essencialmente à qualidade e carga das linhas, um serviço de sincronização baseado nesta infra-estrutura tem de obedecer a um conjunto de requisitos bem definidos e relativamente rígidos. Assim, podemos identificar os seguintes:

1 - A existência de fontes de sincronização primárias, sincronizadas com os standards nacionais, por meio de linhas, rádio ou relógios atómicos. Os servidores de tempo devem distribuir informação temporal baseada no UTC, de uma forma contínua.

2 - Os servidores devem ser capazes de fornecer informação temporal precisa, mesmo na presença de grandes variações na performance dos meios de transmissão.

3 - A rede de sincronização deve ser fiável e tolerante a falhas devido a instabilidade das linhas ou mesmo nos servidores, mesmo quando a conectividade se perder durante períodos relativamente longos. Em particular, a sua arquitectura deve ser reconfigurável dinâmicamente.

4 - O protocolo de sincronização deve operar continuamente e providenciar informação a um ritmo suficiente de modo a poder compensar as deficiências próprias dos relógios usados nos computadores. Deve ser eficiente mesmo em presença de um número elevado de servidores, organizados segundo diferentes configurações.

5 - O sistema deve ser capaz de funcionar num conjunto alargado de plataformas, desde estações de trabalho pessoais a super-computadores, mas deve ser pouco exigente a nível de recursos de sistema operativo.

Além do que foi enunciado é conveniente ainda ter em atenção aspectos de segurança, tal como em qualquer sistema a funcionar no Internet.

Assim, uma configuração típica será constituída por um conjunto de servidores primários, sincronizados directamente com fontes externas de alta precisão e servidores secundários que se sincronizam com os primários e entre si.

Até ao momento vários protocolos têm sido propostos com vista a implementar este tipo de serviço, nem todos obedecendo à totalidade dos requisitos acima enunciados. Dentre eles podemos distinguir o *Daytime Protocol* [POS83a], *Time Protocol* [POS83b], *ICMP Timestamp Message* [POS81], *ICMP Timestamp Option* [SU81] e o *DCE Time Service* [ROS92]. Há ainda a considerar o *time server* do

UNIX 4.3BSD (*timed*), utilizado em redes locais. Por fim, temos o protocolo correntemente em uso no Internet como standard para serviços de sincronização, o *Network Time Protocol* [MIL92], sobre o qual incidirá o nosso estudo daqui por diante.

3. Sincronização no Internet: Network Time Protocol

O Network Time Protocol (NTP) é o standard Internet utilizado para organizar e manter um conjunto de servidores de tempo e as ligações entre eles de forma a obter uma rede de sincronização. O NTP tem sido desenvolvido ao longo de mais de uma década, com base em diversas experiências, iniciadas ainda nos primeiros anos do Internet [MIL81], [MIL85a], [MIL85b]. Presentemente o protocolo vai na terceira versão [MIL92], existindo ainda suporte para as duas anteriores.

O NTP é implementado sobre o Internet Protocol (IP) [DAR81] e o User Datagram Protocol (UDP) [POS80]. A sua evolução partiu do Time Protocol e da Timestamp Message do ICMP, tendo sido idealizado para manter a exactidão e a fiabilidade dos servidores, mesmo sobre ligações de má qualidade.

A operação do NTP é baseada na obtenção de três quantidades, todas elas relativas a um relógio de referência: *offset* entre dois relógios, que representa a quantidade a ajustar no relógio local para que fique de acordo com a referência, tempo em trânsito de uma mensagem enviada ao relógio de referência e dispersão, que representa o erro máximo entre os dois relógios.

Pretende-se que cada servidor de tempo se sincronize com um dos seus pares, pelo que cada uma das quantidades atrás referidas é constituida por duas componentes, uma determinada pelo par em relação ao relógio de referência e a outra medida pelo servidor local em relação ao par. Cada uma destas componentes é mantida a parte pelo protocolo para efeitos de controlo de erros e de manutenção da rede de sincronização.

3.1. Determinação dos tempos

O protocolo baseia-se na troca de *timestamps* entre servidores, que são usados para estimar tempos de trânsito e *offsets* entre cada dois pares. Com base nesta informação cada servidor pode calcular aqueles valores de uma forma contínua e independente do outro par. Este esquema tem, entre outras, a vantagem de não requerer meios muito fiáveis para transmissão, se bem que, como é natural, as precisões atingidas dependem de alguma forma das propriedades daqueles meios [MIL85a] e [MIL85b]. É de destacar ainda o facto de que os tempos de transmissão e a ordem de chegada das mensagens são factores que não têm a mesma importância que em outros protocolos usados no Internet.

3.2. Modos de operação

Como já foi referido, o NTP funciona com base em associações entre servidores, constituindo no seu conjunto uma rede de sincronização. É de notar, no entanto, que o protocolo não providencia meios

para descoberta de pares na rede, nem manutenção de circuitos virtuais. A configuração da rede é dinâmica, i.e., associações mantidas entre servidores podem ser quebradas (por exemplo, devido a falhas nas ligações), e novas associações podem ser criadas com outros servidores. Não há controlo de fluxo nem retransmissões, sendo a integridade dos dados providenciada pelos mecanismos do IP e do UDP.

Geralmente, quando dois servidores trocam mensagens pela primeira vez é criada uma ligação (fraca) chamada *associação.* Cada um cria uma instanciação do protocolo, incluindo informação de estado persistente acerca da associação. A associação pode funcionar num dos seguintes modos: *simétrico activo, simétrico passivo, cliente, servidor* ou *broadcast.*

As ligações em *modo simétrico* (*passivo ou activo*) são usadas na generalidade dos casos onde o NTP é utilizado, envolvendo a participação de diversos servidores de tempo, organizados numa configuração distribuida, dinâmica e hierarquizada. Ao operar num destes modos os pares dispõem-se a sincronizar e serem sincronizados por outros. O *modo activo* é adequado para servidores de *stratum* mais elevado, com endereços de pares (em número reduzido) conhecidos à partida, por exemplo, por intermédio de ficheiros de configuração. O servidor envia mensagens periodicamente, qualquer que seja o estado da associação ou o *stratum* do seu par. O *modo passivo* é usado por servidores situados mais perto da referência primária (menor *stratum*), com um número relativamente elevado e variável de pares. A associação é criada após a chegada de uma mensagem de um par operando em modo simétrico activo e dura enquanto aquele estiver acessível e tiver um *stratum* menor ou igual ao seu.

Em *modo cliente*, um par anuncia a sua intenção de ser sincronizado por outros pares, mas nunca de lhes fornecer informação para os sincronizar. Em *modo servidor*, o par aceita sincronizar outros, mas nunca deixa que eles o sincronizem.

O *modo broadcast* é usado em redes locais de alta velocidade, com um número considerável de estações de trabalho e onde não são requeridas altas precisões. Neste caso, tipicamente existe um servidor que envia mensagens NTP em *broadcast*, constituindo as restantes máquinas os clientes que se sincronizam assumindo um tempo de trânsito da ordem de alguns milisegundos.

Os modos simétrico activo, cliente e *broadcast* são chamados activos enquanto que simétrico passivo e servidor são modos passivos. Normalmente um par opera num modo passivo e outro num modo activo, resultando uma condição de erro sempre que dois pares operem no mesmo modo, a não ser que este seja simétrico activo. Neste caso, os pares ignoram as mensagens um do outro.

3.3. Formato dos dados

Todas as operações matemáticas que o protocolo envolve são efectuadas em complemento para dois, com operandos inteiros ou de vírgula fixa, com os bits numerados segundo a forma *big-endian.*

Sendo os *timestamps* a informação mais preciosa do NTP, um formato especial foi criado para os

representar. Assim, um *timestamp* NTP é um número de vírgula fixa, sem sinal, com 64 bits. A parte inteira é formada pelos primeiros 32 bits e a fraccionária pelos últimos 32. A interpretação de um *timestamp* é feita em segundos standard, relativos às 0 horas do dia 1 de Janeiro de 1900. Desta forma, aquando do início do UTC, o relógio do NTP marcava 2.272.060.800, assumindo que não tinha ocorrido nenhum *leap second* [MIL89].

Uma mensagem NTP (versão 3) contém uma área de dados, que segue imediatamente o cabeçalho UDP, cujos campos se descrevem brevemente a seguir. Para uma descrição mais completa sugere-se a consulta de [MIL92].

Leap Indicator	Aviso de que no fim do corrente dia vai ser inserido ou eliminado um *leap second* da escala UTC.
Version Number	Identifica o número de versão NTP da origem da mensagem.
Mode	Modo de operação do servidor.
Stratum	*Stratum* do servidor, em relação à referência primária.
Poll Interval	Intervalo entre o envio de mensagens NTP. Cada servidor utiliza o mínimo entre o seu intervalo e o do seu par.
Precision	Precisão do relógio local.
Root Delay	Tempo de trânsito até à referência primária.
Root Dispersion	Dispersão em relação à referência primária.
Reference Identifier	Tipo de relógio local.
Reference Timestamp	Hora da última alteração no relógio local.
Originate Timestamp	Hora a que foi enviada a última mensagem originária do par.
Receive Timestamp	Hora a que foi recebida a última mensagem originária do par.
Transmit Timestamp	Hora local a que esta mensagem foi enviada.
Authenticator (opcional)	Se estiver implementado o mecanismo de autenticação do NTP, contém o identificador e *checksum* encriptados do conteúdo da mensagem.

Para além das quantidades descritas acima, o protocolo mantém um conjunto de variáveis de estado, relativas a cada par. Assim, temos o endereço e porto do servidor local e do par, o que serve para identificar a associação; um contador associado a cada par, usado para controlar o intervalo entre envios de mensagens; um registo usado para determinar o estado de acessibilidade do par; um registo

usado pelo algoritmo de filtragem de dados (descrito adiante); o tempo de trânsito, *offset* e dispersão correntes; fonte de sincronização, i.e., o par usado para sincronizar o relógio local; a hora actual, tal como é indicada pelo relógio local.

3.4. Processamento de eventos

No NTP os eventos significativos podem ocorrer quando o contador de um par atinge o valor zero ou quando chega uma mensagem. Outros eventos detectados pelo protocolo podem ser um comando de um operador ou a detecção de uma falha no sistema.

Quando o contador associado a um par atinge zero é invocado um procedimento de transmissão, que envia uma nova mensagem ao par, incluindo os endereços, variáveis e *timestamps* requeridos. O contador é então actualizado.

Após a chegada de uma mensagem NTP é invocado um procedimento de recepção que identifica a associação a que pertence, com base nos endereços e portos nela indicados. Caso tal associação ainda não exista, é automaticamente estabelecida uma nova. Seguidamente são feitos alguns testes de validação e calculados o tempo de trânsito, *offset* e dispersão, tal como descrito previamente. O valores finais do tempo de trânsito, *offset* e dispersão (a que chamaremos estimadores) são calculados com base no conjunto de valores disponíveis para esta associação, dados pelas mensagens mais recentes recebidas do par em questão. Sempre que daqui resulte um novo conjunto de estimadores é invocado um procedimento de actualização que determina o melhor par, podendo este vir a tornar-se na fonte de sincronização. No caso de o par resultante já ser a fonte de sincronização, o *offset* estimado é usado para actualizar o relógio local.

3.5. Procedimentos de filtragem e selecção

Como foi dito atrás, de cada vez que há uma actualização no conjunto de estimadores (um triplo constituído pelo tempo de trânsito, *offset* e dispersão) é necessário verificar se se mantêm as condições relativas ao par a utilizar como fonte de sincronização. O cálculo dos novos estimadores faz-se tendo em conta um conjunto de dados obtidos aquando da recepção das oito últimas mensagens originárias de determinado par e considerando que o melhor conjunto de estimadores é aquele em que o tempo de trânsito é menor [MIL89].

Assim, para cada par é estabelecido um triplo de estimadores que vai ser usado por um algoritmo de selecção, cujo resultado é um par, usado a partir daí como fonte para sincronização.

Este algoritmo começa por construir uma lista de candidatos, ordenada por *stratum* e dispersão, não ultrapassando um limite máximo tanto em tamanho como em *stratum*, sendo eliminados à partida pares que demonstrem comportamentos anómalos. A lista resultante é então re-ordenada, desta vez por *stratum* e distância de sincronização (*Root Delay*). A partir daqui, inicia-se um ciclo de filtragem de candidatos, calculando-se para cada um uma quantidade denominada *dispersão de selecção*, sendo os candidatos eliminados por ordem decrescente desta. O procedimento termina quando as dispersões

de selecção de todos os candidatos forem menores que o mínimo da dispersões de todos eles, ou quando restar apenas um candidato.

Este algoritmo favorece candidatos com baixo *stratum* e pequeno tempo de trânsito. Se a fonte de sincronização corrente for um dos candidatos sobreviventes e não houver nenhum de menor *stratum* então o procedimento termina sem que nada seja alterado. Caso contrário, a fonte de sincronização passa a ser o primeiro sobrevivente da lista.

Este procedimento actualiza ainda o *stratum* do servidor local (fica igual ao da fonte mais um), a distância de sincronização e a dispersão de sincronização (*Root Dispersion*). Todos estes valores serão subsequentemente incluídos nas mensagens NTP geradas localmente.

3.6. Aspectos de segurança

Num meio aberto como é o Internet é necessário ter em atenção os aspectos que se prendem com protecções e autenticação de forma a tornar segura a operação dos vários sistemas. No caso do NTP, a especificação standard prevê a opção de inclusão de esquemas de autenticação baseados em encriptação e mecanismos de controlo de acessos por meio de filtragem de endereços [MIL89].

Num serviço deste tipo existem cinco formas de pôr em causa a correcta operação dos servidores (que passaremos a designar por ataques) [BIS90]: a operação de servidores falsos, a modificação de mensagens enviadas por um servidor, o re-envio de mensagens a servidores, intersecção e eliminação de mensagens e o atraso da entrega de mensagens.

O objectivo de operar servidores falsos é persuadir um servidor que está a dialogar com um dos seus pares autorizados a sincronizá-lo e desta forma fornecer-lhe informação que o leva a alterar o seu relógio, desviando-o significativamente do standard. No entanto os mecanismos de autenticação acima referidos, se implementados, serão suficientes para proteger o servidor deste tipo de ataques.

Ao alterar uma mensagem, um atacante pode causar a incorrecta sincronização de um servidor ou levar a que uma associação seja quebrada. Uma vez que ambas as situações são geridas com base em campos de uma mensagem NTP, basta que se alterem os campos certos para conseguir o objectivo. A autenticação, baseada na encriptação de um *checksum* do conteúdo da mensagem é um meio eficaz de precaver ataques desta natureza.

Os efeitos do re-envio de mensagens são muito semelhantes ao da alteração das mesmas. Desta vez, se o servidor não detectar o duplicado pode usar a informação contida na mensagem para se sincronizar (incorrectamente). Por outro lado, o re-envio de mensagens pode esconder alterações significativas na rede de sincronização, como por exemplo a mudança de *stratum* do par donde partiu a mensagem original, dado este que é importante na escolha da fonte de sincronização. De notar que o NTP, devido aos pormenores da sua especificação, detecta este tipo de ataque baseado na hora de recepção da última mensagem do par. Assim, a precaução contra este tipo de ataques envolve a configuração adequada dos intervalos de envio de mensagens, de forma a diminuir a possibilidade de