Administração de Sistemas (ASIST)

Aula Teórico Prática 04

Gestão da configuração de rede.
Servidores DHCP. Servidores DNS.

Baseado em: A. Moreira, 2018/2019, Aulas teórico-práticas de ASIST

Configuração de interfaces de rede em servidores

A configuração de uma interface de rede pode ser efetuada autonomamente pelo próprio sistema ou recorrer a serviços externos. O serviço externo mais usado pera este efeito é o **DHCP** (*Dynamic Host Configuration Protocol*), no caso da configuração IPv6 também se pode usar **SLAAC** (*Stateless Address Auto Configuration*), eventualmente complementada com **DHCP**.

Como normalmente um servidor presta serviços de rede, deve ter endereços fixos conhecidos dos clientes. Isso exclui a utilização de SLAAC, mas pode ser usado DHCP para atribuir endereços IPv4 e IPv6 fixos.

No entanto, sempre que possível, um servidor deve ser independente de serviços externos. Por essa razão, normalmente opta-se por configurações de rede autónomas em que o sistema possui todos os dados para configurar as suas interfaces de rede. Deste modo, mesmo que os serviços DHCP da rede não estejam operacionais, o servidor pode funcionar normalmente.

Sob o ponto de vista dos protocolos IPv4 e IPv6, para configurar uma interface de rede bastam dois dados: **endereço de nó** e **prefixo de rede** (máscara de rede).

Com estes dois elementos é possível deduzir qual é o **endereço da rede**, e no caso do IPv4, qual é o **endereço de *broadcast***.

Nomes das interfaces de rede

No sistema operativo cada interface de rede (**NIC** – *Network Interface Controller*) é identificada por um nome. Se o servidor possui várias interfaces de rede é fundamental identificar a correspondência entre os nomes e as interfaces físicas, isso poderá não ser trivial, especialmente se o hardware das interfaces é igual.

Linux: tradicionalmente as interfaces de rede Ethernet eram identificadas por **eth** seguido de um sufixo numérico (***eth0***, ***eth1***, ***eth2***, …). Nas atuais versões do *kernel*, as interfaces de rede Ethernet são identificadas por **en** seguido de um sufixo complexo que representa o seu posicionamento sob o ponto de vista do hardware. Por exemplo **ens**… significa que se trata de um interface ligada ao barramento PCI externo, **eno**… significa que se trata de uma interface *on-board*. A restante parte do prefixo também está relacionada com o hardware.

Windows: as interfaces de rede são identificadas com nomes que tentam representar o seu tipo, por exemplo, **Ethernet**. Existindo várias interfaces do mesmo tipo, a primeira mantém o nome e as seguintes recebem um sufixo numérico a partir de 2. Por exemplo num servidor com 3 interfaces Ethernet: **Ethernet**, **Ethernet 2** e **Ethernet 3**. A configuração IPv4 e IPv6 das interfaces de um Windows Server em modo gráfico faz-se facilmente recorrendo ao ***Control Panel***.

Linux - configuração das interfaces de rede

Os comandos **ifconfig** e **ip** podem ser usados para visualizar o estado atual das interfaces de rede e alterar as configurações IP das mesmas. Note-se no entanto que as alterações realizadas por esta via não persistem após um novo ***boot*** da máquina (ao contrário da alterações realizadas no Windows via ***Control Panel***).

As configurações de rede aplicadas durante o ***boot*** da máquina residem em ficheiros de configuração que dependem da distribuição e até da versão; no Ubuntu 16.04 estão no ficheiro **/etc/network/interfaces** e no Ubuntu 18.04 como as interfaces de rede são geridas pelo **Netplan** estão no ficheiro **/etc/netplan/50-cloud-init.yaml**.

O comando **ifconfig** sem argumentos apresenta uma lista das interfaces de rede que estão ativas (*up*) e as respetivas configurações. O comando **ip addr show** apresenta uma lista de todas as interfaces de rede e as respetivas configurações (estejam *up* ou *down*).

O comando **ip** deve ser usado em lugar do comando **ifconfig** que se prevê seja descontinuado no futuro.

Exemplo básico de configuração de uma nova interface com um endereço IPv6 seguido da ativação da interface (admitindo que estava ***down***):

```
ip address add 2001:0db8:85a3::0370:7334/64 dev ens160
ip link set dev ens160 up
```

Default gateway and routing

O endereço de nó e o prefixo da rede são suficientes para conseguir comunicar com outros nós das redes IP a que o servidor está diretamente ligado, no entanto para comunicar com nós de outras redes IP é necessário recorrer a ***routers***.

No cenário mais simples, nas redes IP a que o servidor está ligado existe apenas um *router*, nesse caso basta defini-lo como ***default gateway***. No caso dos sistemas Windows o ***default gateway*** pode ser definido juntamente com as configurações IP. Num sistema Linux terá de ser manualmente adicionado à ***routing table*** do sistema ou definido na configuração da rede.

Como resultado, todos os pacotes destinados a redes IP remotas (a que o servidor não está diretamente ligado) serão transmitidos para o ***default gateway***.

As configurações do ***default gateway*** IPv4 e IPv6 são totalmente independentes, assim se o servidor utiliza IP4 e IPv6 (*dual-stack*) existirá um ***default gateway*** IPv4 e um ***default gateway*** IPv6; podem ser o mesmo, mas os endereços serão certamente diferentes.

Num cenário em que existem vários routers nas redes IP a que o servidor está ligado, a ***routing table*** terá de ser definida em conformidade.

Linux – gestão da *routing table*

O tradicional comando **route** prevê-se que seja descontinuado e deve ser usado o comando **ip [-6] route** em seu lugar. A opção **-6** é usada para gerir a *routing table* IPv6, caso seja omissa atua sobre a *routing table* IPv4.

Podemos visualizar as tabelas através do comando **ip [-6] route show**

Novamente, as alterações de configuração não persistem após um novo ***boot*** da máquina, para isso terão de ser definidas em ficheiros de configuração apropriados dependentes da distribuição.

Exemplos:

- Definição do ***default gateway*** em IPv4 e IPv6 na NIC ens160:

```
ip route add 0.0.0.0/0 via 10.20.0.1 dev ens160
ip -6 route add ::/0 via fd1e:2bae:c6fd:627a::55 dev ens192 protocol static
```

- Adição de uma regra específica em IPv4 e IPv6 na NIC ens160:

```
ip route add 172.16.5.0/24 via 10.20.0.1 dev ens160
ip -6 route add fd1e:2bae::/64 via fd1e:2bae:c6::5 dev ens192 protocol static
```

- Remoção de uma regra em IPv4 e IPv6 na NIC ens160:

```
ip route del 172.16.5.0/24 via 10.20.0.1 dev ens160
ip -6 route del ::/0 via fd1e:2bae:c6fd:627a::55 dev ens192 protocol static
```

Windows – gestão da *routing table*

A gestão da ***routing table*** no que vai além da definição do ***default gateway*** tem de recorrer ao comando **route** na linha de comandos. O comando **route print** apresenta as tabelas atuais (IPv4 e IPv6).

As alterações de configuração são guardadas de forma persistente pelo sistema operativo.

Em função dos endereços fornecidos o comando deduz se se trata de endereços IPv4 ou IPv6, determinando igualmente a interface mais apropriada.

Exemplos (equivalentes aos anteriores em Linux):

- Definição do ***default gateway*** em IPv4 e IPv6:

```
route add 0.0.0.0/0 10.20.0.1
route add ::/0 fd1e:2bae:c6fd:627a::55
```

- Adição de uma regra específica em IPv4 e IPv6:

```
route add 172.16.5.0/24 10.20.0.1
route add fd1e:2bae::/64 fd1e:2bae:c6::5
```

- Remoção de uma regra em IPv4 e IPv6:

```
route delete 172.16.5.0/24
route delete ::/0
```

Caso pretendido pode-se indicar a interface de rede pretendida com a opção **-F**.

Configuração do cliente DNS (resolução de nomes)

Para ser possível a utilização de nomes de máquinas do sistema DNS em lugar de endereços IP é necessário definir quais são **os endereços dos servidores de nomes DNS a utilizar.**

O sistema DNS só resolve nomes qualificados (que incluem o nome de domínio completo), para resolver nomes não qualificados podem ser definidos vários nomes de domínio que serão usados para completar os nomes não qualificados antes de os enviar ao servidor DNS. Num sistema que utiliza DHCP estas informações são normalmente transmitidas por essa via, não sendo usado o DHCP terão de ser definidas manualmente.

Windows: a configuração de cliente DNS é realizada juntamente com as definições dos endereços IP das interfaces de rede. Embora na janela base apenas seja permitida a definição de um ou dois servidores DNS, nas opções avançadas é possível acrescentar mais. Igualmente nas opções avançadas é possível definir um conjunto de nomes de domínio a utilizar para completar nomes não qualificados.

Linux: a configuração do cliente DNS (*resolver*) está tradicionalmente guardada no ficheiro **/etc/resolv.conf**. No entanto na maioria das distribuições atuais este ficheiro é gerado sempre que a máquina arranca, para esse efeito usa-se informação residente em outros ficheiros de configuração (dependem da distribuição). No caso do Ubuntu 16.04 ou 18.04 esta informação deve ser definida no ficheiro de configuração da(s) interface(s) de rede.

Interfaces de rede com vários endereços IP

Uma mesma interface de rede por ter vários endereços IP atribuídos, podendo estes pertencer a redes IP diferentes ou podem pertencer à mesma rede IP.

É possível ter a funcionar sobre uma mesma rede física várias redes IP diferentes. Os nós IP que apesar de estarem ligados à mesma rede física utilizam endereços pertencentes a redes IP diferentes só poderão comunicar entre si através de um *router*. Pode-se evitar a necessidade de recorrer ao *router* se foram atribuídos à interface endereços em cada uma das redes.

Podemos também atribuir a uma mesma interface vários endereços IP pertencentes à mesma rede IP, neste caso o objetivo é normalmente criar vários servidores aparentes (virtuais).

Se o servidor tem vários endereços IP (tipicamente mapeados para nomes DNS distintos), sob o ponto de vista dos clientes vai aparentar tratar-se de um conjunto de vários servidores distintos.

Para que este tipo de configuração se torne interessante tem de ser complementada com uma configuração de serviços de rede que seja dependente do endereço de destino dos pedidos dos clientes. Por exemplo o servidor HTTP Apache permite definir sites virtuais distintos em função desse critério.

Windows: nas opções avançadas de propriedades IPv4 e IPv6 da interface de rede, outros endereços IP podem ser adicionados ao que foi definido na janela base. Neste caso, para endereços pertencentes a diferentes redes não será possível definir um default gateway para cada uma delas, essa configuração terá de ser realizada através do comando **route**.

Linux: através do comando **ip addr add** … podemos ser adicionados vários endereços à interface para além do primeiro (se bem que apenas persistam até ao próximo *boot* do sistema).

Ao adicionar um endereço podemos também definir um nome de interface virtual, por questões de compatibilidade com os *Linux-2.0 net aliases*, o nome deve começar pelo nome da interface física, seguido de dois pontos*, o resto pode ser definido de acordo com as preferências do administrador. Exemplo:

```
root@uvm:~# ip addr add 10.5.10.200/16 broadcast + dev ens160 label ens160:rede1
root@uvm:~# ifconfig ens160:rede1
ens160:rede1 Link encap:Ethernet  Endereço de HW 00:50:56:a7:10:03
      inet end.: 10.5.10.200  Bcast:10.5.255.255  Masc:255.255.0.0
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Métrica:1
```

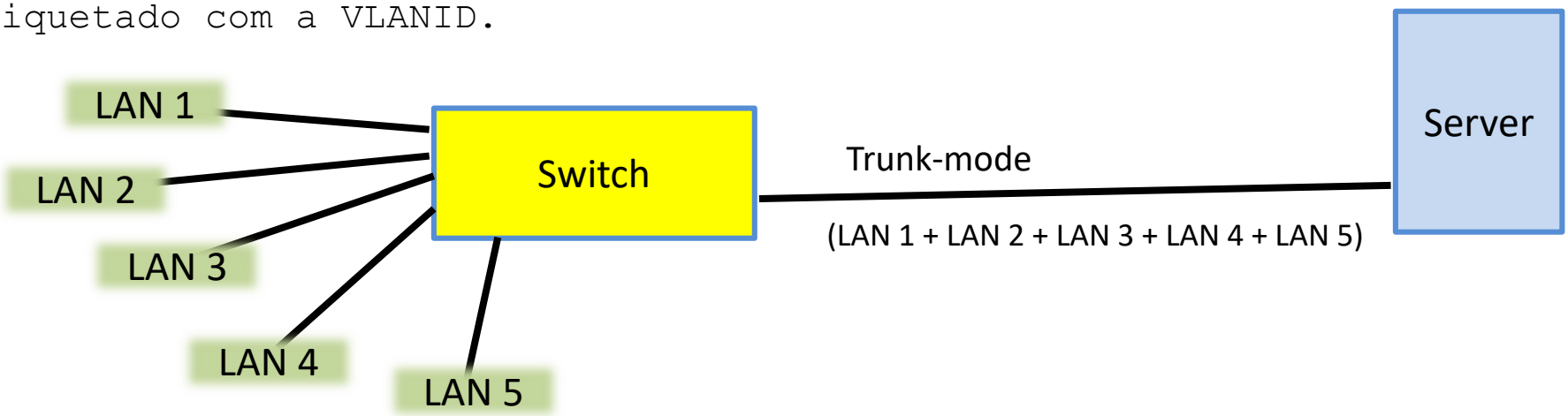
Como se pode observar pelo resultado do comando **ifconfig**, a interface de rede com o nome **ens160:rede1** passou a existir no sistema, esse facto facilita a configuração de serviços de rede pois a maioria deles permite explicitar qual a interface de rede a utilizar.

* Os dois pontos é uma convenção, não uma imposição

Configuração de VLANs

Através da utilização de etiquetas de VLAN IEEE 802.1q (**VLANID**) é possível usar uma única interface de rede para ligar a um conjunto numeroso de redes físicas distintas, esta é a utilização típica de VLANs em servidores.

Para o efeito será necessário um ***switch*** ligado às várias redes físicas e no qual se estabelece um VLANID para cada uma delas. A porta de ligação ao servidor é configurada em ***trunk-mode***, dessa forma o tráfego das várias redes físicas será enviada através dela devidamente etiquetado com a VLANID.



Graças à utilização de etiquetas nas duas extremidades da ligação ao servidor, o tráfego de diferentes redes físicas não se vai misturar. Do lado do servidor, na interface de rede, terá de ser definida uma **interface de VLAN** para cada uma das redes físicas, usando os VLANID estabelecidos no *switch*.

Configuração de VLANs no Windows Server

As versões recentes do Windows Server usam ***NIC Teaming*** para a definição de várias VLANs numa interface de rede usando etiquetas IEEE 802.1q.

A técnica ***NIC Teaming*** é usada para agrupar várias interfaces físicas numa única interface lógica (***NIC Team***), o objetivo é aumentar a performance (***load-balancing***) e garantir a tolerância a falhas (***failover***).

Para definir várias VLANs ligadas à mesma interface física também é usado *NIC Teaming* , mas neste caso o *NIC Team* é constituído por apenas uma interface física.

As interface de VLAN podem depois ser adicionadas ao *NIC Team* definindo o VLANID de cada uma delas e atribuindo nomes apropriados a cada nova interface de VLAN. Note-se que ao adicionar uma interface física a um *NIC Team*, a configuração IP da interface física deixa de ser utilizada, a configuração IP terá de ser definida novamente na interface *NIC Team*, através do ***Control Panel***.

Depois de adicionadas as VLANs ao *NIC Team*, a configuração IP da cada interface de VLAN também é realizada da forma habitual no ***Control Panel***.

A interface *NIC Team* (*primary interface*) não usa etiquetas VLANID (*untagged frames*), as interfaces de VLAN usam etiquetas (*tagged frames*).

Configuração de VLANs em Linux

Para criar interfaces de VLAN sobre uma interface física usa-se o comando **vconfig**:

vconfig add INTERFACE VLANID

O comando cria uma interface de VLAN na interface física INTERFACE onde será usada a etiqueta de VLAN fornecida como segundo argumento (VLANID). A nova interface de VLAN terá o nome **INTERFACE.VLANID** (na realidade a forma de nomear as interfaces de VLAN pode ser estabelecida através do comando **vconfig set_name_type FORMA-DE-NOMEAR** aplicando-se a interfaces adicionadas posteriormente; o valor por omissão da FORMA-DE-NOMEAR é **DEV_PLUS_VID_NO_PAD** que corresponde a **INTERFACE.VLANID**).

Uma vez adicionada a interface de VLAN, as suas configurações IP podem ser definidas através do comando **ip addr add** ….

A interface física (INTERFACE) continua a operar normalmente sem etiquetas VLANID (*untagged frames*) as interfaces de VLAN usam etiquetas (*tagged frames*).

A opção **rem** do **vconfig** remove uma interface de VLAN:

vconfig rem VLAN-INTERFACE

Normalmente: vconfig rem INTERFACE.VLANID

Configuração de VLANs em Linux - exemplo

Neste exemplo criam-se duas interfaces de VLAN sobre a interface física **ens160** com etiquetas **300** e **320,** atribuem-se endereços IPv4 a todas as interfaces e finalmente as interfaces são ativadas:

```
vconfig add ens160 300
vconfig add ens160 320
ip addr add 192.168.5.1/24 broadcast + dev ens160
ip addr add 192.168.6.1/24 broadcast + dev ens160.300
ip addr add 192.168.7.1/24 broadcast + dev ens160.320
ip link set dev ens160 up
ip link set dev ens160.300 up
ip link set dev ens160.320 up
```

Os pacotes emitidos na interface **ens160** não contêm etiquetas de VLAN, os pacotes emitidos na interface **ens160.300** têm a etiqueta de VLAN 300, os pacotes emitidos na interface **ens160.320** têm a etiqueta de VLAN 320.

Na interface **ens160** apenas são recebidos pacotes sem etiqueta, na interface **ens160.300** apenas são recebidos pacotes com etiqueta 300, na interface **ens160.320** apenas são recebidos pacotes com etiqueta 320. Pacotes recebidos com outras etiquetas serão descartados.

VLANs vs. Múltiplos endereços na mesma interface

À primeira vista criar várias interfaces de VLAN sobre a mesma interface física ou atribuir vários endereços IP à mesma interface física podem parecer soluções semelhantes.

Em ambos os casos podemos ter pacotes IP pertencentes a diferentes redes IP a circular sobre a mesma rede física. A diferença está na forma como pacotes de diferentes redes IP são distinguidos uns dos outros.

Não usando VLANs, a distinção é feita através dos endereços IP, portanto no nível 3, isso tem algumas consequências. Por exemplo para pacotes enviados para endereços *multicast* ou *broadcas*t genérico (255.255.255.255) não é possível determinar a que rede pertencem. Isto poderá ter impacto sobre os protocolos que usam este tipo de endereços.

Outro inconveniente é que sem usar VLANs não será possível ligar várias redes a um *switch* porque este opera no nível 2 e não analisa endereços IP.

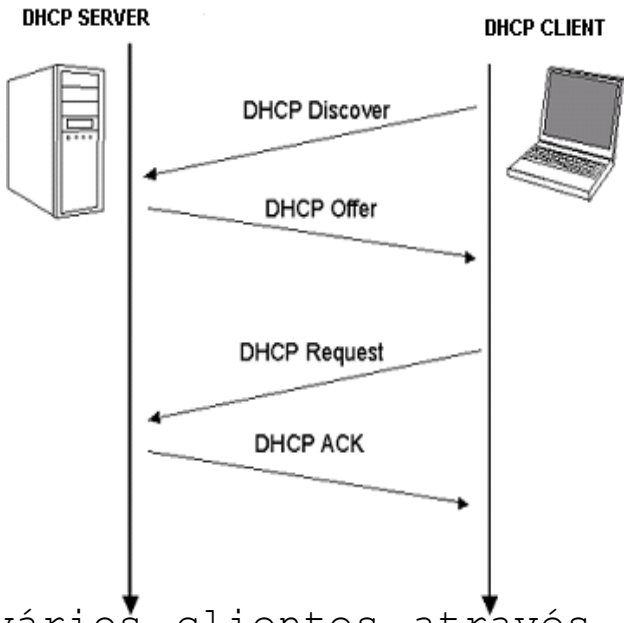
Quando se utilizam VLANs, a distinção é feita através do VLANID transportado nos *frames*, portanto no nível 2. Cada VLAN é para todos os efeitos uma rede física distinta, um *frame* emitido numa VLAN nunca se propaga em circunstancia alguma a uma VLAN diferente.

Nomeadamente, pacotes IP enviados para endereços de *broadcast* e *multicast* nunca passam de uma VLAN para outra VLAN.

Serviço DHCP (DHCP server)

Embora a configuração de interfaces de rede de servidores através de DHCP seja desaconselhada, para postos de trabalho é fundamental. Permite que um utilizador ao ligar o seu posto de trabalho a uma rede que desconhece (cablada ou wireless) obtenha os dados de configuração IP adequados a essa rede. Por exemplo nos sistemas Linux o serviço **ifplugd** deteta quando um cabo de rede é ligado ao equipamento e invoca os serviços necessários à configuração via DHCP da interface de rede.

O serviço DHCP utiliza pacotes UDP, o primeiro pedido do cliente é enviado para o endereço de broadcast genérico **255.255.255.255**, no caso do IPv4, ou para o endereço multicast link local **ff02::1:2**, no caso do IPv6 (**DHCPv6**). A imagem ilustra os vários passos no caso do IPv4, para DHCPv6 o cenário é semelhante embora as mensagens sejam diferentes.



A utilização de broadcast ou multicast link-local implica que na rede física terá de existir pelo menos um servidor DHCP.

Normalmente os servidores DHCP identificam os vários clientes através dos respetivos endereços de nível 2 (MAC *addresses*), a sua principal missão é fornecer a cada cliente endereços IP únicos na rede. Para esse efeito fazem a gestão de gamas de endereços disponíveis na rede.

Além do endereço IP único para cada cliente, o servidor DHCP fornece ou pode fornecer outros parâmetros que são normalmente idênticos para todos os clientes da mesma rede: máscara da rede, *default gateway*, endereços dos servidores DNS e o nome do domínio DNS local (para completar nomes não qualificados). Além destes parâmetros, as respostas DHCP podem ainda conter muitos outros com diversas finalidades.

O serviço DHCP pode ser disponibilizado por diversos tipos de equipamentos de rede como *routers*, *switches Layer 3* e servidores.

Windows Server: o role **DHCP Server** pode ser ativado através da aplicação **Server Manager**, uma vez ativado, a aplicação **DHCP Manager** pode ser usada para o gerir. O serviço deve ser associado às interfaces das redes (**binding**) que se pretende servir. Para cada rede que se pretende servir deve ser adicionado um **Scope** com uma **Pool** de endereços disponíveis e restantes parâmetros a fornecer aos clientes. É também possível definir parâmetros comuns a todos os Scopes. O servidor DHCP dos sistemas Windows atuais suporta tanto DHCPv4 como DHCPv6.

Linux: existem vários servidores DHCP *open source* gratuitos que suportam DHCPv4 e DHCPv6, um dos mais usados é o ***ISC DHCP Server*** disponibilizado pela ISC (Internet Systems Consortium, Inc.). Como é habitual em Linux/Unix, toda a configuração do *ISC DHCP Server* é realizada através de ficheiros de texto.

DHCPv6 (DHCP para clientes IPv6)

No caso do **IPv6** o contexto de aplicação do DHCPv6 é bastante diferente. O IPv6 tem mecanismos próprios para configuração automática dos nós (*Neighbour Discovery Protocol* – **NDP**) que não podem ser contornados. O DHCP pode ser usado ou não dependendo da configuração do *router* IPv6.

Quando um nó IPv6 pretende configurar automaticamente a rede envia um pedido ICMPv6 *Router Solicitation*, para o endereço multicast link-local **ff02::2** (*local routers*). Se existirem routers IPv6 na rede devem responder com a mensagem ICMPv6 ***Router Advertisement***.

O comportamento do cliente é controlado pela resposta do *router*, desde logo a resposta contém o endereço do *router* e também a identificação dos endereços das redes IPv6 que conhece. Com esta informação o cliente fica a conhecer o *router* e a rede a que está ligado, por essa razão os servidores DHCPv6 não necessitam de fornecer aos clientes a identificação da rede nem o *default gateway*.

A resposta contém ainda dois bits que determinam o comportamento do cliente. Se o bit **M** (*Managed address configuration*) tiver o valor 1, então o cliente deve usar DHCPv6 para saber o seu endereço IPv6 e completar a informação que já conhece. Caso contrário o cliente autoatribui-se um endereço sobre a rede que já conhece usando SLAAC.

O bit **O** (*Other configuration*) só é usado para **M=0** (SLAAC). Se **O=1** será usado DHCPv6 para obter parâmetros adicionais necessários (que não incluem o endereço já definido por SLAAC).

IPv6 Router Advertisement Service

A aplicação do DHCPv6 depende das respostas ***Router Advertisement*** do protocolo **NDP** (*Neighbour Discovery Protocol)*, trata-se de um serviço que deve ser implementado e configurado nos *routers* IPv6.

Os servidores Windows e Linux também pode operar como *routers* IPv4/IPv6, nessa caso terão de enviar periodicamente para as redes ***Router Advertisements*** e também quando solicitados pelos clientes.

Windows Server: o serviço está integrado no role designado ***Routing and Remote Access*** (ou ***Remote Access*** apenas). Depois de instalado, a ferramenta administrativa ***Routing and Remote Access*** pode ser usada para ativar o *routing* IPv4/IPv6 e configurar o seu funcionamento.

Linux: o *Router Advertisement Daemon* (**radvd**), *open source*, pode assegurar este serviço nos *routers* Linux. No ficheiro de configuração **/etc/radvd.conf** para cada interface de rede onde o router se anuncia pode ser definida o bit **M** (*AdvManagedFlag*) e o bit **O** (*AdvOtherConfigFlag*).

DHCP Relay – O cliente DHCP só consegue contactar o servidor se ele se encontrar na mesma rede (domínio de broadcast). Quando se pretende utilizar num único servidor DHCP para servir um grande número de redes dispersas, isto pode ser inconveniente. O serviço *DHCP relay pode ser* instalado num dispositivo ou servidor de cada rede, e configurado para retransmitir os pedidos para um servidor DHCP remoto.

ISC DHCP Server (Linux) - Exemplo /etc/dhcpd.conf

```
option domain-name "dei.isep.ipp.pt";
option ip-forwarding off;
option netbios-node-type 8;
option netbios-scope "";
max-lease-time 604800;
default-lease-time 86400;

subnet 100.4.0.0 netmask 255.255.0.0 {
option subnet-mask 255.255.0.0;
option routers 100.4.0.1;
option broadcast-address 100.4.255.255;
option domain-name-servers 192.168.62.15;
option netbios-name-servers 192.168.62.15;
option log-servers 192.168.62.37;
range 100.4.10.0 100.4.30.255;
}
```

As declarações **options** correspondem a parâmetros a fornecer aos clientes via DHCP. Para cada rede a servir deverá haver uma declaração **subnet**. As declarações efetuadas fora de um bloco **subnet** aplicam-se a todas as declarações **subnet**, mas podem ser sobrepostas dentro do bloco.

Fault tolerant DHCP Service

Pelo facto de o primeiro contacto com o servidor ser realizado com recurso a *broadcast* (IPv4) ou *multicast* (IPv6), implementar um serviço DHCP tolerante a falhas é relativamente simples.

Existindo vários servidores DHCP na rede (domínio de *broadcast* ou *multicast link-local*), se um deles não estiver disponível para responder a clientes, outros estarão.

Para evitar conflitos é necessário garantir que cada servidor gere uma gama de endereços IP diferente dos restantes. Os endereços atribuídos pelos vários servidores podem pertencer todos à mesma rede IP, ou pertencer a redes IP diferentes, os restantes parâmetros de configuração fornecidos por cada servidor DHCP têm de ser definidos em conformidade. Esta configuração garante também ***load balancing***, tendencialmente o primeiro servidor a responder será aquele que possui menor carga no momento.

A Microsoft recomenda a existência de dois servidores DHCP, devendo o conjunto de endereços disponíveis ser divido entre um servidor primário com 80% dos endereços e um secundário com 20% dos endereços.

No caso de um domínio Windows, recomenda-se que o role DHCP Server seja instalado em DCs do domínio para permitir uma melhor integração com a base de dados *Active Directory*. Neste caso, os postos de trabalho do domínio (*Domain Members*) devem ser configurados para registarem no DNS do domínio o seu nome obtido por DHCP.

Resolução de nomes através do serviço DNS (DNS servers)

Entre outros equipamentos de rede, servidores Windows e servidores Linux podem exercer a função de servidor DNS.

No caso dos domínios Windows o *role* DNS server faz parte dos vários serviços obrigatórios que têm de estar presentes nos DC. Quando um Windows Server é promovido a *Domain Controller*, o *role* DNS Server é instalado automaticamente. Recorde-se que com o *Active Directory* cada domínio Windows corresponde a um domínio DNS.

Em cada domínio DNS existe obrigatoriamente um servidor DNS ***master*** e deve existir pelo menos mais um servidor DNS ***slave***.

O servidor *master* possui a cópia original da base de dados de registos DNS. Os servidores *slave* também possuem uma cópia local da base de dados, mas obtêm-na do *master*. Ambos se designam ***authoritative name servers*** do domínio porque possuem cópias locais da base de dados de registos do domínio. Todas as modificações à base de dados devem ser feitas no *master*. Os *slaves* comparam a data da última alteração da base da dados que se encontra no master (***serial number*** do registo SOA) com a cópia local; se são diferentes solicitam uma transferência total de zona (***AFXR DNS query***) ou apenas das alterações (incremental) relativamente ao *serial number* que possui localmente (***IFXR DNS query***).

Após algum tempo sem conseguir sincronizar-se com o *master*, um servidor *slave* deixar de ser ***authoritative*** para o domínio.

DNS - Zonas

Zonas são conjuntos de registos DNS que pertencem ao mesmo domínio DNS, um servidor DNS pode servir várias zonas. Cada zona possui obrigatoriamente um registo SOA que contém o nome DNS do servidor master da zona, o endereço de email do administrador da zona (com o símbolo **@** substituído por um **ponto**), o ***serial number*** e parâmetros de configuração relativos à sincronização dos *slaves* da zona.

Cada base de dados de zona deve conter também os registos NS relativos à zona (nomes dos servidores DNS) e registos NS relativos a subdomínios.

Zonas diretas – contêm registos usados na resolução direta, ou seja permitem obter endereços a partir de nomes qualificados. Notoriamente contêm registos do tipo A e AAAA, respetivamente para associar nomes a endereços IPv4 e endereços IPv6.

Zonas inversas – permitem a resolução inversa (***reverse resolution***), ou seja obter o nome DNS através do fornecimento do endereço IP. Estas bases de dados de zona correspondem a subdomínios dos domínios especiais **.in-addr.arpa.** e **ip6.arpa.**, respetivamente para endereços IPv4 e IPv6. Cada octeto em notação decimal (no caso do IPv4) ou símbolo hexadecimal (no caso do IPv6) corresponde a um subdomínio. Em ambos os casos o tipo de registo usado para o efeito é o PTR (*pointer*).

Note-se que não existe qualquer correspondência entre zonas diretas e zonas inversas.

Servidores DNS

Linux – O servidor DNS mais popular é o **bind**, *open source*, disponibilizado gratuitamente pela **ISC** (*Internet Systems Consortium, Inc.*). A sua configuração e gestão das bases de dados das zonas no caso de ser um master é realizada através de ficheiros de configuração em formato de texto. Também suporta alterações através de pedidos de clientes (*Dynamic DNS*).

Windows Server – o role ***DNS Server*** pode ser instalado num servidor *Standalone*, nesse caso a administração deverá ser realizada manualmente através da aplicação ***DNS Manager***.

Quando um Windows Server é promovido a controlador de domínio, é aconselhado que seja permitida a instalação automática do role ***DNS Server***, neste caso a gestão do serviço DNS é automaticamente realizada pelos **AD DS** (*Active Directory Domain Services*) e a gestão manual raramente será necessária.

A replicação de bases de dados DNS entre DCs de um domínio Windows não usa o esquema master/slave. Num domínio Windows, normalmente todos os DC são servidores DNS, a base de dados DNS encontra-se no *Active Directory* e é replicada automaticamente entre os vários DC.

Estando o servidor DNS sob o controlo dos AD DS, qualquer máquina adicionada ao domínio é automaticamente adicionada à base de dados DNS.

DNS Server – Linux – Bind version 9

A configuração do servidor bind assenta no ficheiro **named.conf**, normalmente residente na pasta **/etc/** ou **/etc/bind/**. A partir deste ficheiro são normalmente incluídos outros ficheiros de configuração.

Uma das configurações mais importantes é a definição dos ***forwarders***, trata-se de endereços IP de outros servidores DNS para os quais podem ser retransmitidos pedidos de resolução sem estar a recorrer ao processo de resolução normal do DNS. Exemplo:

```
options {
        forwarders { 192.249.249.1; 192.249.249.3; };
};
```

O bind pode ser *slave* de uma ou várias zonas. Declaração exemplo:

```
zone "dei.isep.ipp.pt" IN {
        type slave;
        file "/var/lib/bind/dei.isep.ipp.pt";
        masters { 193.136.62.15; };
};
```

Sendo um *slave* da zona **dei.isep.ipp.pt**, o ficheiro de zona declarado **/var/lib/bind/dei.isep.ipp.pt** é criado e gerido pelo bind e não deve ser editado. Contém uma cópia da base de dados de zona residente no servidor master da zona (com endereço **193.136.62.15** no exemplo).

DNS Server – Linux – Bind version 9

O bind pode ser *master* de uma ou várias zonas. Declaração exemplo:

```
zone "dei.isep.ipp.pt" IN {
        type master;
        file "/etc/bind/direct";
        allow-query { any; };
        allow-transfer { any; };
};
```

Neste caso trata-se do *master* da zona **dei.isep.ipp.pt**, o ficheiro de zona declarado **/etc/bind/direct** tem de ser criado e gerido manualmente.

Se o bind for configurado para suportar *Dynamic DNS*, vai alterar o conteúdo do ficheiro em conformidade, caso contrário o conteúdo só é alterado manualmente pelo administrador.

O ficheiro de zona, neste exemplo **/etc/bind/direct** contém todos os registos DNS que integram a base de dados da zona.

Nos ficheiros de zona todos os nomes devem ser qualificados (**fqdn**) **qualquer nome que não termine em ponto** é tido como não qualificado e o nome do domínio correspondente à zona é **automaticamente acrescentado.**

DNS Server – Linux – Bind version 9 – ficheiro de zona

O ficheiro de zona é um ficheiro de texto em que cada linha corresponde a um **RR** (*Resource Record*), tipicamente o primeiro registo é o **SOA** *(Start Of Authority)*. O formato geral de cada linha do ficheiro de zona é:

nome-do-registo ttl class tipo-de-registo dados-do-registo

nome-do-registo – nome DNS do registo. Pode ser omitido, nesse caso a linha deverá começar por um espaço e será usado o mesmo valor do registo anterior.

ttl (*Time To Live*) – tempo máximo de vida em segundos. Quando um cliente DNS obtém o registo deve considerar que passado este tempo o registo é obsoleto e deverá obter um novo. Este elemento é opcional, se não for especificado será o valor do registo anterior ou o valor declarado na diretiva **$TTL**.

class – indica o tipo de nome, para nomes da internet o valor é **IN**. Tal como o anterior pode ser omitido sendo usado o valor do registo anterior.

tipo-de-registo – mnemónica que indica de que tipo de registo DNS se trata.

dados-do-registo – conteúdo do registo, a sua forma depende do tipo de registo.

DNS Server – Linux – Bind version 9 – SOA

O registo **SOA** é normalmente o primeiro registo do ficheiro de zona e contém os seguintes elementos:

nome-master email-administrador (serial refresh retry expiry nx)

nome-master – nome DNS do servidor master da zona.

email-administrador – endereço de correio eletrónico do administrador da zona com o símbolo **@** substituído por um **ponto**.

serial – número de série da zona. Regista a última alteração à base de dados, na forma **yyyymmddss**, onde **yyyymmdd** representa o dia em que foi alterada e **ss** o número da alteração nesse dia (começa por ser zero).

refresh – número de segundos entre verificação de atualizações por parte dos servidores *slave*.

retry – número de segundos que os *slaves* devem aguardar em caso de falha no contacto com o master antes de tentarem novamente.

expiry – número de segundos sem conseguirem contactar o master após o qual os servidores *slave* deixam de ser *authoritative* para a zona.

nx – tempo máximo em segundos de *caching* para respostas NXDOMAIN (*non-existent domain*).

Linux – Bind version 9 – outros registos

Além do SOA outros tipos de registos DNS importantes são:

Tipo (mnemónica)	Conteúdo
A	Endereço IPv4
AAAA	Endereço IPv6
NS	Nome de nó servidor de nomes (correspondente a um registo A e/ou AAAA)
CNAME	Nome de nó (correspondente a um registo A e/ou AAAA)
MX	Prioridade e nome de nó servidor SMTP (correspondente a um registo A e/ou AAAA)
PTR	Nome de nó (correspondente a um registo A ou AAAA)
TXT	Um comentário (também usado na implementação de SPF*)

* SPF (*Sender Policy Framework*) identificam que servidores estão autorizados a enviar mails em nome do domínio

Os registos PTR são usados em zonas inversas, ou seja correspondentes a subdomínios de **in-addr.arpa.** e **ip6.arpa.**, respetivamente para endereços IPv4 e IPv6.

Linux – Bind version 9 – exemplo de ficheiro de zona

```
$TTL 1d
$ORIGIN meu.dominio.pt

@    IN    SOA   nserv.meu.dominio.pt.   admin.meu.dominio.pt  ( 2015091502   1d   15M   2W 1h )

            NS          nserv
nserv       A           192.168.70.2
            AAAA        2001:db8:10::2

@           MX          10          mailserv1
            MX          20          mailserv2.meu.dominio.pt.
server1.meu.dominio.pt.             A           192.168.70.90
mailserv1.meu.dominio.pt.           A           192.168.70.23
                                    TXT         “Servidor SMTP principal”
                                    AAAA        2001:db8:10::1
mailserv2                           A           192.168.70.27
mail                    CNAME       mailserv2
www                     CNAME       server1
ftp                     CNAME       server1
```

A declaração **$ORIGIN** define o domínio correspondente à zona, pode depois ser referenciada através do símbolo @. Todos os registos são da classe IN, com TTL de 1 dia (definido na declaração **$TTL**).