MIETI
Redes de Computadores I
Trabalho prático: LANs Ethernet e redes TCP/IP usando o CORE

1. Introdução

Neste trabalho pretende-se emular no CORE vários tipos de redes e interligá-las entre si. A interface gráfica *core* será usada para desenhar as topologias de rede e configurar os links e os endereços. A configuração deverá depois ser feita em modo de execução, equipamento a equipamento, imitando tanto quanto possível a rede real. Os exercícios terminam com o diagnóstico de conectividade e a análise de capturas de tráfego que deverão ser efetuadas usando o Wireshark.

Comece por descarregar e instalar o CORE no seu computador pessoal. Utilize o Anexo I se necessitar de apoio para esta primeira tarefa.

2. Emulação de LANs Ethernet

Neste primeiro exercício pretende-se emular no CORE pequenas redes locais. A interface gráfica do CORE poderá ser usada para desenhar a topologia e configurar as ligações, os endereços e os serviços que se vão executar em cada máquina. O exercício termina com o diagnóstico de conectividade, capturas e análise do tráfego.

2.a) Construa (em modo de edição) uma topologia de rede local em estrela usando um HUB. A esta topologia vamos chamar TOPOLOGIA A.

2.b) Efetue testes de conectividade entre os sistemas terminais da sua topologia usando o comando *ping* (em modo execução). Efetue capturas de tráfego para perceber ao detalhe o funcionamento do HUB bem como dos protocolos envolvidos (ARP e ICMP).

2.c) Crie uma segunda topologia (TOPOLOGIA B), substituindo o HUB por um SWITCH. Repita a captura de tráfego e verifique as diferenças ao nível do funcionamento de um e outro equipamento. O que concluí?

2.d) Por último, crie uma topologia em árvore usando HUBs e SWITCHs (TOPOLOGIA C) e verifique o seu funcionamento.

3. DHCP

Em vez de configurar os elementos de uma rede manualmente, é possível recorrer ao DHCP para o fazer de forma automática e dinâmica. O objectivo deste exercício é a configuração de uma rede local recorrendo a este protocolo.

3.a) Faça *download* e instale no *host* (máquina Linux) o servidor de DHCP. Para isso abra um terminal e digite **`sudo apt-get install dhcp3-server`**. Deverá depois ativar e configurar o serviço, já no CORE.

3.b) Crie no CORE uma quarta topologia (TOPOLOGIA D), de uma rede local (em estrela ou árvore, usando hubs ou switches) onde o endereçamento é feito de forma dinâmica, utilizando o DHCP. Para isso, deverá, por um lado, ativar o serviço numa das máquinas (servidor) e, por outro, definir o endereço das outras máquinas como automático, ou seja obtido via DHCP (clientes).

3.c) Faça a captura de pacotes de forma a identificar a sequência de interações entre um cliente DHCP e o respetivo servidor, tendo em vista a obtenção de um endereço IP.

4. Interligação de redes

Para interligar redes IP distintas é necessário um router capaz de encaminhar o tráfego IP de umas redes para as outras. No CORE, os routers estão preparados para executar o Quagga. O objectivo deste exercício consiste em construir uma rede de interligação que permita interligar as redes locais criadas e configuradas nos exercícios anteriores.

4.a) Faça *download* e instale no *host* (máquina Linux) o software de *routing* Quagga. Para isso abra um terminal e digite **`sudo apt-get install quagga`** (se estiver a utilizar a máquina virtual do CORE não necessita de efetuar este passo, uma vez que o Quagga já está pré-instalado).

4.b) Crie uma topologia em que junte as quatro topologias criadas anteriormente (A, B, C e D). A, B, C e D devem estar ligadas a um *router* cada, e estes ligados entre si com caminhos alternativos. As redes das TOPOLOGIAS A, B, e C e D devem estar na gama de endereços **`10.0.0.0/24.`** Os vários *routers* deverão estar interligados por uma sub-rede com máscara de 30 *bits* da rede **`192.168.0.0/24.`**

4.c) Conceba um esquema de encaminhamento que faça sentido para a topologia de rede criada na alínea anterior. **Não se esqueça que deverá evitar os ciclos de encaminhamento**, para isso deve usar uma política coerente em todos os routers.

4.d) Desative o encaminhamento dinâmico em todos os routers e adicione manualmente as rotas (encaminhamento estático) necessárias para garantir a conectividade IPv4 entre todas as redes de acordo com o esquema de encaminhamento concebido na alínea anterior.

4.e) Teste a conectividade entre todas as redes, com os comandos *ping* e *traceroute*.

5. Uso das camadas de rede e transporte por parte das aplicações

Depois de estabelecida, a rede emulada está pronta para suportar serviços e executar aplicações de rede. É possível instalar, na máquina Linux que está a executar o CORE, serviços de rede bem conhecidos, como por exemplo, um servidor HTTP, e usá-lo depois num host da rede emulada no CORE.

5.a) Use aplicações de rede bem conhecidas na rede implementada. No mínimo ative um servidor HTTP e um servidor FTP e teste-os usando os clientes a partir de redes locais distintas.

5.b) Capture pacotes e analise os protocolos envolvidos das diferentes camadas da pilha TCP/IP, de acordo com os conhecimentos já adquiridos.

6. Interligação via NAT (Network Address Translator)

6.a) Acrescente uma rede privada (local) à sua topologia, rede essa que deve usar endereços privados da gama **`192.168.1.0/24`**. Essa rede, para ter conectividade, terá que estar ligada a uma das redes LAN que já configurou, através de um *router* NAT.

Para configurar o NAT num *router*, o Emulador CORE não instala automaticamente nenhum *software* específico para esse efeito. Porém, como o Emulador CORE usa máquinas virtuais Linux para virtualizar os diversos componentes das redes virtuais, podemos perfeitamente usar as soluções Linux para configurar o NAT. A solução utilizada para configurar NATs no Linux é o programa `iptables`, que vem instalado em quase todas as distribuições de Linux. O `iptables` é normalmente utilizado para configurar *firewalls*, por forma a aplicar políticas de restrição de acesso e transmissão de dados entre duas redes.

6.b) Configure o router NAT de forma a dar conectividade à rede privada que acabou de criar. Teste a conetividade entre a rede privada e a rede externa e, recorrendo ao Wireshark verifique o funcionamento do serviço NAT.

6.c) Crie um servidor HTTP ou FTP na rede privada NAT e configure o NAT de forma a que o mesmo esteja acessível a partir da rede externa. Recorrendo ao Wireshark verifique o correto o funcionamento desta configuração do serviço NAT.

7. Entrega do trabalho

Elabore um pequeno relatório (20 páginas no máximo), que descreva o trabalho realizado e apresente as principais conclusões a que chegou.

O trabalho deve ser realizado em grupo (de dois ou três elementos) e demonstrado na semana de 11 a 15 de Janeiro. Os ficheiros imn e respetivo relatório em pdf devem ser submetidos, na plataforma de elearning, até ao dia 11 de Janeiro.

ANEXO I -

O CORE funciona **apenas** em FreeBSD e Linux (Ubuntu) com um kernel alterado para ter mais do que uma stack TCP/IP. Uma alternativa à instalação nativa desta ferramenta consiste na utilização de uma máquina virtual. Veja em http://cs.itd.nrl.navy.mil/work/core/ como obter um ambiente CORE adequado. Se optar por usar o CORE numa máquina virtual, deve começar por instalar o VMWare Player ou a VirtualBox para depois descarregar e descompactar a respetiva imagem do CORE no disco local. A imagem depois de descompactada ocupa aproximadamente 2GB.

Interface gráfico do CORE

Para invocar a interface gráfico do CORE, digite na linha de comando:

core-gui (sem argumentos)	– executa o ambiente de desenvolvimento gráfico do CORE
core-gui <file.imn>	– carrega o ficheiro <file.imn> no ambiente gráfico CORE

Se ficar sem rede, depois de executar o core pode tentar recuperá-la com os seguintes comandos:

/sbin/dhclient –r	– liberta explicitamente o endereço anteriormente obtido por DHCP
/sbin/dhclient	– pede um novo endereço por DHCP
/sbin/ifconfig –a	– verifica se o interface ficou devidamente configurado

Criação da topologia no CORE (Modo de Edição)

O CORE, no modo de edição, permite desenhar topologias. Existem vários objetos que podem ser usados: *ligações*, *hub*, *switch*, *router*, *pc*, *host*, tomada de saída, etc.

Tanto o *pc* como o *host* representam sistemas terminais, do tipo computador pessoal ou servidor. A única diferença entre ambos é que o *pc* não executa nenhum processo no arranque enquanto o *host* arranca logo com alguns servidores. Além dos servidores que arrancam por defeito, é possível ativar outros, como por exemplo, um servidor HTTP. Para isso é necessário verificar se o software está instalado na máquina onde está a executar o CORE (máquina nativa ou máquina virtual). Só depois do software estar devidamente instalado é que é possível usá-lo no CORE. Para instalar software adicional no Ubuntu utilize o comando: sudo apt-get install. Depois ainda pode ser necessário configurar a script de arranque dos respetivos serviços.

Para interligar as máquinas em rede, existem três equipamentos de interligação, que operam a diferentes níveis da pilha protocolar: o *router* (nível 3), o Switch (nível 2) e o Hub (nível 1). Naturalmente que só o *router* usa endereços IP e faz encaminhamento.

Há apenas um objeto para representar ligações, mas que pode ser parametrizado para diferentes cenários de utilização. Os parâmetros configuráveis são por exemplo a largura de banda disponível e a taxa de erros. É ainda possível ligar a topologia virtual com a topologia real, usando o objeto *tomada de saída.* Este objeto, com a forma de uma ficha RJ45, pode ser associado a um interface físico do computador.

Preservação da topologia criada e das configurações estabelecidas para utilização futura

Para preservar a topologia de rede concebida, bem como as configurações efetuadas em modo de Edição, é necessário gravar antes de passar ao modo de Execução ou abandonar a interface gráfica. Para isso escolha a Opção "File->Save", que irá gravar tudo o que foi feito num ficheiro com a extensão imn. Este ficheiro é um ficheiro de texto que contém tudo o que é necessário para editar de novo a topologia no Core.

Testes de conectividade e captura de tráfego (Modo de Execução)

Uma vez terminada a topologia, passa-se ao modo de execução escolhendo a opção "Start the session" (barra do lado esquerdo).

Esta opção faz despoletar a criação de múltiplas imagens (stacks virtuais), uma por cada objeto representado. É preciso algum cuidado com os erros nesta fase, porque podem implicar ter de sair da aplicação para limpeza das stacks virtuais.

No modo de execução, estão disponíveis novas ações associadas a alguns objetos, nomeadamente ao *pc, host,e router.* Selecionando um desses equipamentos com o botão do rato do lado direito obtém-se um menu com as opções disponíveis. No caso dos *pc* e *host* é possível abrir uma shell (bash ou sh) no nó respetivo para neles executar comandos. Nestes dispositivos é também possível capturar o tráfego usando um analisador de protocolos: o Wireshark. Para usar o Wireshark deverá instalar primeiro esta ferramenta no seu computador (se tiver optado por instalar o CORE numa máquina nativa), ou na máquina virtual onde está a executar o CORE. Para isso basta introduzir o seguinte comando no terminal:

```
sudo apt-get install wireshark
```

E, logo em seguida retificar as permissões para que seja possível capturar tráfego a partir do CORE.

```
sudo setcap 'CAP_NET_RAW+eip CAP_NET_ADMIN+eip' /usr/bin/dumpcap
```

Depois disto, basta selecionar o equipamento a partir do qual quer efetuar a captura de tráfego, com o botão do rato do lado direito, escolher a opção "wireshark" e dar início à captura na interface de rede que desejar.