

Neste trabalho pretende-se emular no CORE, cinco redes locais e interligá-las entre si. A interface gráfica do *Core* será usada para desenhar as topologias de rede e configurar as ligações 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.

### **1. Emulação de LANs Ethernet**

Neste primeiro exercício pretende-se emular no CORE uma rede local. 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.

**1.a)** Construa (em modo de edição) uma topologia de rede local em árvore usando HUBs e SWITCHs.

**1.b)** Efetue testes de conectividade entre os sistemas terminais da topologia de rede criada usando o comando *ping* (em modo execução).

**1.c)** Planeie um teste que lhe permite perceber o detalhe de funcionamento dos HUBs e dos SWITCHs. Através de capturas de tráfego, usando o Wireshark, comprove aquilo que aprendeu nas aulas acerca do funcionamento destes dispositivos de interligação de redes Ethernet, bem como dos protocolos envolvidos (ARP e ICMP). Anote as suas conclusões.

### **2. Interligação de redes**

Para interligar redes locais distintas é necessário utilizar routers capazes de encaminhar o tráfego IP de umas redes para as outras. No CORE, os routers estão preparados para executar o Quagga. O objetivo deste exercício consiste em construir uma rede de interligação que permita interligar redes locais.

**2.a)** Construa uma topologia em malha parcialmente conectada utilizando os routers do CORE. A topologia deverá ter no mínimo seis routers e deverão existir caminhos alternativos entre eles, ou seja, a topologia não deverá ser em linha, nem em árvore. Associados a 5 routers desta topologia deverão estar associadas 5 redes locais (rede A, rede B, rede C, rede D e rede E), semelhantes às que foram construídas no ponto anterior (ou mais simples, em estrela).

**2.b)** Estabeleça um esquema de endereçamento de acordo com o seguinte critério:

- Os vários *routers* deverão estar interligados por sub-redes com máscaras de 30 *bits*, da rede **192.168.100.0/24**.
- As redes locais (A, B, C, D e E) devem estar na gama de endereços **172.16.160.0/20**, podendo conter cada uma delas no máximo: 2000 computadores (rede A), 1000 computadores (rede B), 500 computadores (rede C), 300 computadores (rede D) e 150 computadores (rede E).
- Em todas as redes IP, o endereço mais alto da rede deverá ser o endereço atribuído à interface do *router*.

**2.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 estratégia coerente em todos os routers.

**2.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.

**2.e)** Teste a conectividade entre todas as redes, com os comandos *pinge* *traceroute*. Anote as conclusões a que chegou no final do ponto 2 deste trabalho.

### **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 objetivo 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 isc-dhcp-server**. Deverá depois ativar e configurar o serviço, já no CORE.

**3.b)** Volte à topologia criada no ponto 2 e altere a configuração da rede E para que a mesma passe a configurar-se de forma automática e dinâmica, utilizando o DHCP. Para isso, deverá, por um lado, ativar e configurar o serviço numa das máquinas (servidor de DHCP) e, por outro, definir a configuração de rede 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 da configuração de rede. Anote as conclusões a que chegou.

### **4. 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.

**4.a)** Active um servidor HTTP num PC da rede local D, e teste-o usando os clientes a partir de redes locais distintas.

**4.b)** Capture pacotes e analise os protocolos envolvidos das diferentes camadas da pilha TCP/IP, de acordo com os conhecimentos já adquiridos. No relatório, inclua imagens das capturas de tráfego que ilustrem o estabelecimento de uma conexão TCP e outras que ilustrem um pedido HTTP e a correspondente resposta.

## **5. Interligação via NAT (Network Address Translator)**

O NAT (Network Address Translation) é um mecanismo que consiste em reescrever o endereço IP e porta de origem de um pacote que passa por um router com o objetivo de permitir que um computador de uma rede interna (privada) tenha acesso ao exterior.

**5.a)** Acrescente uma rede privada (local) à sua topologia, rede essa que deve usar endereços privados da gama **192.168.200.0/24**. Essa rede, para ter conectividade, terá que estar ligada à rede local associada, 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.

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

**5.c)** Crie um servidor HTTP e um servidor FTP na rede privada NAT e configure o NAT de forma a que os mesmos estejam acessíveis a partir da rede externa. Recorrendo ao Wireshark verifique o correto o funcionamento da configuração do serviço NAT.

## **Instruções para a realização e entrega do trabalho**

A versão do CORE a utilizar deverá ser obrigatoriamente a versão 9.

O trabalho deve ser realizado em grupo (de três elementos). A constituição de cada um dos grupos é da responsabilidade dos alunos, mas deve ser enviada por email para a docente da UC (joao@dsi.uminho.pt) até **2 de Dezembro de 2024**.

Apesar de termos algumas aulas TP dedicadas à realização do trabalho, este deverá ser realizado maioritariamente fora do horário das aulas.

No final do trabalho, cada grupo deverá realizar um pequeno relatório (20 páginas no máximo), que descreva o trabalho realizado e apresente as principais conclusões a que chegaram. Os ficheiros xml e respetivo relatório em pdf devem ser submetidos, na plataforma de elearning, até ao final do dia **17 de dezembro de 2024**. Na aula teórico-prática do dia **19/12/2024** cada grupo terá de demonstrar o trabalho realizado.