Universidade do Minho
Escola de Engenharia

Mestrado Integrado em Engenharia de Telecomunicações e Informática

Redes de Computadores II

Relatório I

David Faria

João Silva

Jorge Bastos

Índice

1. Introdução

O *Common Open Research Emulator* é uma ferramenta desenvolvida para a construção de redes virtuais. Na qualidade de emulador, o CORE constrói uma representação de uma rede que corre em tempo real e se comporta como se fosse real, ao contrário das ferramentas de simulação, onde modelos abstratos são usados para a representação das redes.

A simulação em tempo real pode ser conectada a uma rede física e a *routers* físicos, fornecendo um ambiente que nos permite correr aplicações reais e protocolos dentro da emulação, aproveitando o facto de que os sistemas operativos Linux e FreeBSD permitem essa virtualização.

O CORE é tipicamente usado na área das redes e dos protocolos de rede, com o intuito da pesquisa, demonstração, aplicação ou teste de plataformas, avaliação de cenários de rede, realização de estudos de segurança ou aumento do tamanho de uma rede física.

Wireshark é um software de análise de protocolos de rede. Permite ao utilizador que capture e interactivamente pesquise o tráfego ocorrendo numa rede, ou seja, permite controlar e saber tudo o que entra e sai da rede à qual estamos ligados, sendo também possível controlar o tráfego num determinado dispositivo ou máquina virtual. Corre em todo o tipo de sistemas operativos e tem um conjunto de ferramentas que lhe permitem ser o software mais popular do género no mundo.

Neste primeiro exercício pretende-se emular no CORE vários tipos de redes e interliga-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. Terminámos com o diagnóstico de conectividade e a análise de capturas de tráfego que são efetuadas usando o Wireshark.

2. Implementação do Problema

2.1. Conceitos Teóricos

2.1.1. DHCP (*Dynamic Host Control Protocol*) [1]

Fornece aos administradores um mecanismo de alocação dinâmica de endereços IP, ao invés da alocação manual que é praticamente impraticável, dada uma rede com um número elevado de *hosts*. Por norma, um servidor DHCP liberta uma gama de endereços IP disponíveis para os clientes DHCP, durante um período específico de tempo. Os clientes que queiram conectar-se à rede simplesmente terão de descobrir o servidor, enviar um pedido ao servidor e esperar pela aceitação do mesmo para entrarem em ação.

2.1.2. Encaminhamento Estático versus Encaminhamento Dinâmico [2]

Encaminhamento Estático é um método adequado a redes de pequena dimensão, onde o cenário da rede é pouco complexo e onde existem muito poucas alterações. Neste caso, o administrador da rede é que introduz as rotas manualmente na tabela de encaminhamento dos *routers*, tendo em conta o conhecimento de toda a infraestrutura de rede. Caso existam alterações na rede, como as rotas são estáticas, as rotas não se vão conseguir adaptar a essas mudanças, dando origem a problemas de comunicação dentro da infraestrutura de rede.

Encaminhamento Dinâmico é adequado ao uso em redes mais complexas, com redundância de caminhos e cuja topologia pode sofrer alterações com frequência. Neste caso, os caminhos são calculados recorrendo a protocolos de encaminhamento dinâmico que conseguem adaptar-se às alterações na infraestrutura da rede. A grande desvantagem deste processo, resulta numa sobrecarga maior dos *routers* e obriga a conhecimentos do protocolo de encaminhamento a ser usado para a configuração dos mesmos. Entre os protocolos mais conhecidos, encontram-se o RIP, o OSPF, o BGP, etc..

2.1.3. RIP (Routing *Information Protocol*) [3]

É um dos primeiros protocolos de Vetor de Distâncias, desenhado para o uso em redes de pequena dimensão. Algumas das suas características são: a atualização periódica das tabelas de encaminhamento, o envio das ditas tabelas apenas para a vizinhança, o uso da métrica número de saltos e o uso do algoritmo *Bellman-Ford* para determinar o melhor "caminho" para um destino em particular e

o número máximo de saltos ser igual a 16. Qualquer rede que esteja a 16 ou mais saltos de distância é considerada inalcançável pelo RIP, tendo a designação de caminho envenenado e a métrica de infinito.

2.1.4. OSPF (*Open Shortest Path First*) [4]

É um protocolo de roteamento de Estado de Ligação, desenhado à escala para eficientemente suportar redes de dimensões mais elevadas. Contem algumas características, tais como: áreas para empregar o uso da hierarquia, relações com a vizinhança através de routers adjacentes à mesma área, utilização de LSA (*Link-State Adevertisements*), atualizações só ocorrem se uma ligação sofrer alterações, tráfego é "multicasted" ou para todos os routers OSPF ou para routers designados, utilização do algoritmo de Dijkstra para determinar o caminho mais curto e a métrica custo, que tem como base a largura de banda de uma ligação. A construção de tabelas é mais complexa no OSPF, pelo que no processo de construção são mantidas as seguintes tabelas: tabela da vizinhança, tabela da topologia, com todas as rotas possíveis, e a tabela de encaminhamento do router.

2.1.5. NAT (*Network Address Translation*) [5]

Criado em resposta ao crescimento rápido da internet e à escassez de endereços IPv4 disponíveis. Para tal efeito, um subconjunto da gama de endereços referida foi considerado privado, de modo a aliviar o problema. Um endereço privado é usado, como o próprio nome indica, para uso privado e não pode ser roteado (ou achado) na internet. Aliás, os *routers* da internet estão configurados para largar qualquer tipo de tráfego com endereços privados, pelo que torna-se necessária a criação de um meio de contacto (ou tradução) entre endereços públicos e privados, de modo a ser possível o acesso destes últimos à internet. É aí que entra o NAT.

2.2. Conceção

2.2.1. Desenho da topologia de REDE e endereçamento

Neste projeto, reconstruiu-se a topologia usada no exercício teórico-prático nº1, adicionando-lhe sub-redes em todos os nós. No nosso caso, a topologia foi adaptada de acordo com a resolução do exercício feita nas aulas, tal como a Figura 1 demonstra.

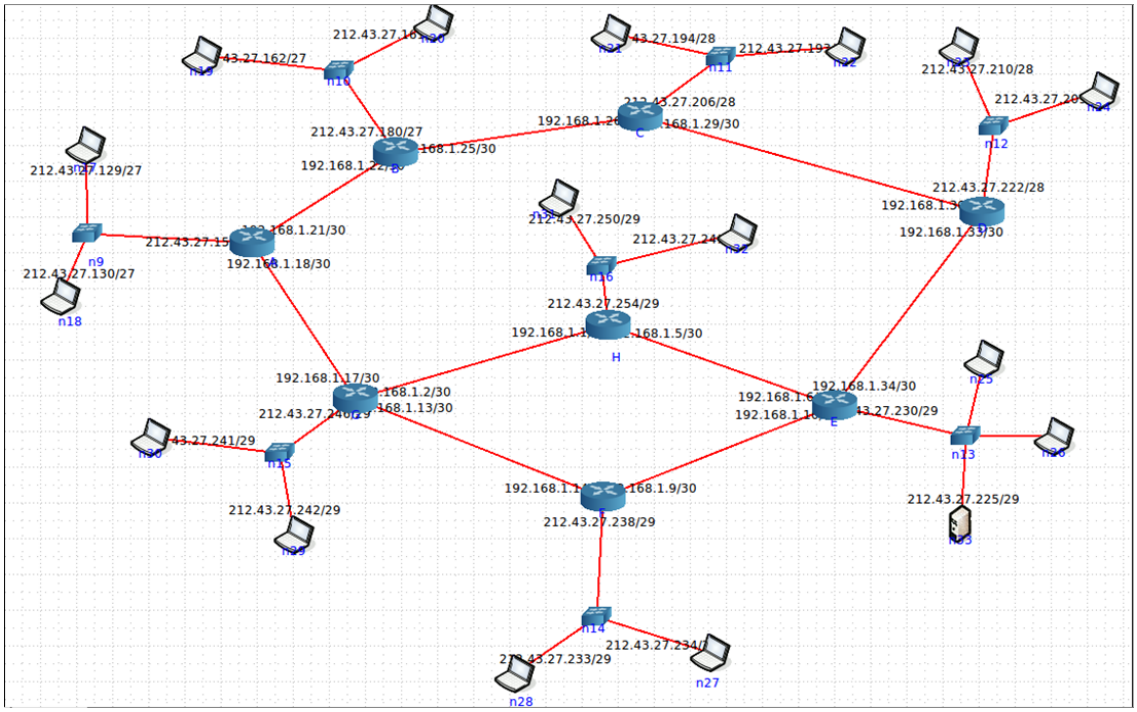

Figura 1 – Topologia da infraestrutura de rede idêntica à do exercício teórico-prático 1

No caso do servidor DHCP, adicionámo-lo a uma das nossas LAN, ligado a um *switch*, que por sua vez está ligado ao *router* E. Configurámo-lo da forma vista na Tabela 1:

Tabela 1 - Serviços e comandos ativos no servidor DHCP

Serviços ativos	Comando (s) inserido (s)
DefaultRoute	ip route add default via 212.43.27.230
SSH	“ Por defeito “

DHCP	subnet 212.43.27.224 netmask 255.255.255.248 {
	pool {
	range 212.43.27.225 212.43.27.229;
	default-lease-time 600;
	option routers 212.43.27.230; }}

2.2.2. Encaminhamento Estático

Em primeiro lugar é necessária a instalação do *software* de roteamento *Quagga*. Para tal, digitamos no terminal da máquina nativa: sudo apt-get install quagga. Depois, recorremos ao esquema de encaminhamento usado para resolver o exercício 1 nas aulas, tendo em atenção que temos que desativar todos os protocolos de encaminhamento dinâmico que vêm por defeito nos *routers*, tais como o OSPF e o RIP. Em seguida, temos de configurar no serviço zebra o encaminhamento estático do *router*

através do seguinte comando: ip route network gateway. No campo "*network*" endereçamos a rede de destino e no campo "*gateway*" a interface de sáida do router para esse destino.

2.2.3. Encaminhamento Dinâmico

Para conseguirmos estabelecer o protocolo de encaminhamento dinâmico RIP, tivemos que ativar vários serviços e posteriormente, configurá-los. Para a configuração, corremos o CORE e depois, na consola *vtysh* de um *router* qualquer (por exemplo o A) digitamos os seguintes comandos:

```
A# configure terminal
A(config)# router rip
A(config-router)# network 192.168.1.20/30
A(config-router)# network 192.168.1.16/30
A(config-router)# network 212.43.27.128/27
A(config-router)# end
```

No comando network, temos de inserir o endereço de rede e a própria máscara. Neste caso, como estamos a utilizar o router A, na ligação entre o A e B, o endereço de rede é 192.168.1.20, mas do A para o G, o endereço de rede já é 192.168.1.16. Do router A para a respetiva sub-rede, o endereço de rede é 212.43.27.128. Depois de compreendida a configuração do RIP, podemos configurar o router diretamente nos seus serviços, como podemos ver na Tabela 2:

Tabela 2 - Serviços e comandos ativos no Router para fazer o RIP

Serviços ativos	Comando (s) inserido (s)
Zebra	router rip network 192.168.1.20/30 network 192.168.1.16/30 network 212.43.27.128/27
RIP	" Por defeito "
Vtysh	" Por defeito "
IPForward	" Por defeito "

Para conseguirmos estabelecer o protocolo de encaminhamento dinâmico OSPF, tivemos que

ativar vários serviços e posteriormente, configurá-los. Para a configuração, digitamos os seguintes comandos:

A# configure terminal
A(config)# router ospf
A(config-router)# network 192.168.1.20/30 area 0.0.0.0
A(config-router)# network 192.168.1.16/30 area 0.0.0.0
A(config-router)# network 212.43.27.128/27 area 0.0.0.0
A(config-router)# end

No comando network, temos de inserir o endereço de rede e a própria máscara. Neste caso, como estamos a utilizar o *router* A, na ligação entre o A e B, o endereço de rede é 192.168.1.20, mas do A para o G, o endereço de rede já é 192.168.1.16. Do router A para a respetiva sub-rede, o endereço de rede é 212.43.27.128. Depois de compreendida a configuração do OSPF, podemos configurar o router diretamente nos seus serviços, como podemos ver na Tabela 3.

Tabela 3 - Serviços e comandos ativos no Router para fazer o OSPF

Serviços ativos	Comando (s) inserido (s)
Zebra	router ospf network 192.168.1.20/30 area 0.0.0.0 network 192.168.1.16/30 area 0.0.0.0 network 212.43.27.128/27 area 0.0.0.0
RIP	“ Por defeito “
Vtysh	“ Por defeito “
IPForward	“ Por defeito “

2.2.4. Interligação via NAT

Nesta parte, refizemos a topologia usada até aqui, removendo um dos sistemas terminais que estava ligado ao router G, e ligar em vez desse um router configurado em modo NAT, ligado a um *switch*, que por sua vez está ligado tanto a um servidor privado como a um sistema terminal. Essa nova sub-rede privada tem a gama de endereços privados 10.0.0.0/8. Essa nova topologia ficou como demonstrado na Figura 2.

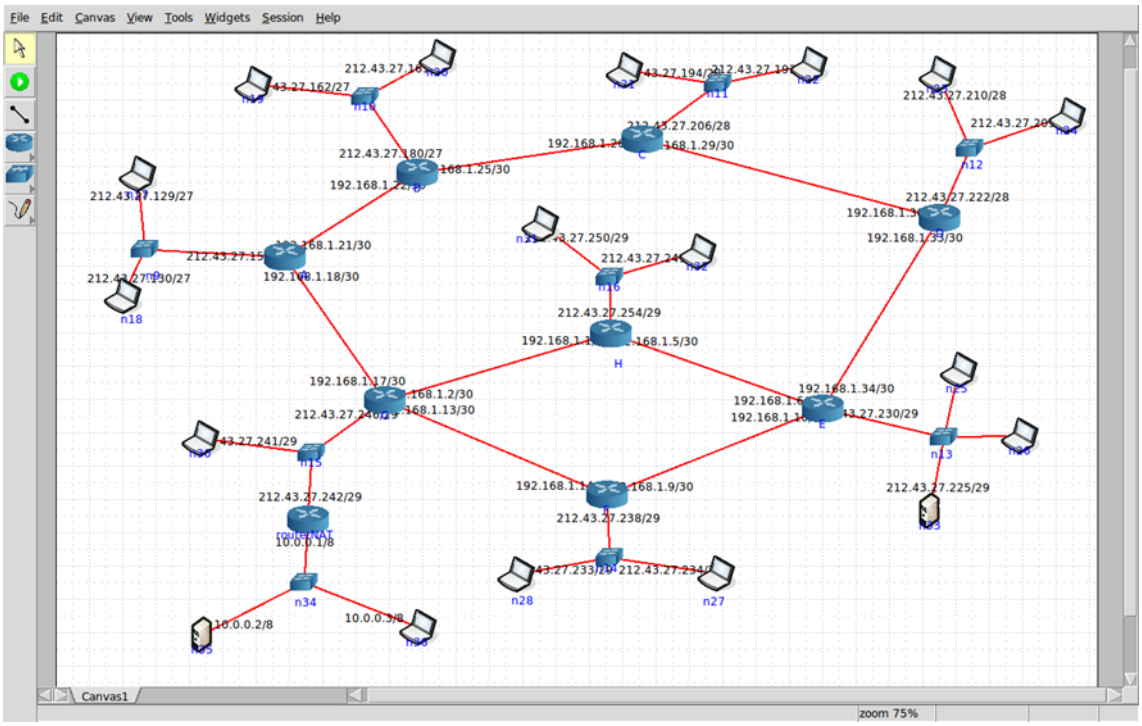

Figura 2 – Nova topologia após adição de uma sub-rede privada.

No novo *router* (aqui chamado de RouterNAT), tivemos de o configurar para funcionar em modo NAT. Tal pode ser visto na Tabela 4:

Tabela 4 - Serviços e comandos ativos no RouterNAT.

Serviços ativos	Comando (s) inserido (s)
Zebra	ip route 0.0.0.0/0 212.43.27.254
Vtysh	“ Por defeito “
IPForward	“ Por defeito “
DefaultRoute	ip route add default via 212.43.27.246 ip route add default via 10.0.0.1
Firewall	iptables -A FORWARD -i eth1 -j ACCEPT iptables -A FORWARD -o eth0 -j ACCEPT iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j DNAT --to 10.0.0.2:80

2.3. Testes

2.3.1. Desenho da topologia de REDE e endereçamento

Como o nosso servidor DHCP tem á sua disposição a gama de IPs 212.43.27.225 a 212.43.27.229, qualquer destes IPs pode ser atribuído a um sistema terminal pertencente a esta LAN. Como temos 2 sistemas terminais ligados ao servidor DHCP, eles terão um IP dentro dessa gama de IPs mas nunca iguais. Uma visualização deste fenómeno a acontecer pode ser vista na Figura 3. Com a ajuda do *Wireshark* podemos constatar que o servidor faz o processo de "offer" ao endereço 212.43.27.227, espera pelo "request" vindo do mesmo e envia um "ACK" assim que a conexão é estabelecida.

18	24.902416000	212.43.27.225	212.43.27.227	DHCP	342 DHCP Offer
19	24.902791000	0.0.0.0	255.255.255.255	DHCP	342 DHCP Request
20	24.947636000	212.43.27.225	212.43.27.227	DHCP	342 DHCP ACK

Figura 3 – Sequência de Eventos na atribuição de um IP pelo servidor DHCP

2.3.2. Encaminhamento Estático

Para testar a conectividade na rede, abrimos a *bash* num dos *hosts* dentro da rede ligada ao *router* B, de endereço 212.43.27.161. Em primeiro lugar tentamos "pingar" um *host* dentro da rede ligada ao *router* H, de endereço 212.43.27.250. Para tal executamos o comando ping seguido do endereço de destino, que é o endereço 212.43.27.250, tal como mostra a imagem seguinte:

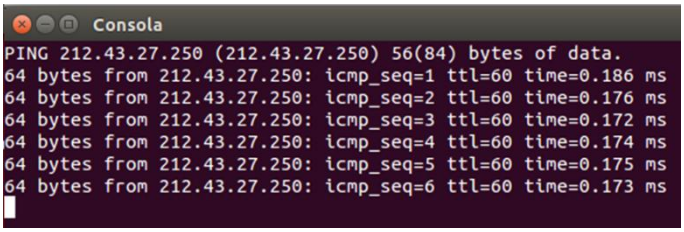

Figura 4 – A execução do comando *ping*.

Feito o ping com sucesso, como é observável na imagem acima, executamos o comando traceroute, seguido do mesmo endereço para verificar porque caminhos vão as mensagens trocadas entre ambos os *routers* B e H. Como podemos verificar pela figura a seguir, o host que se encontra no endereço 212.43.27.180 sai pelo router B, percorrendo o router A (endereço número 2 da imagem), o router G (endereço número 3) e o router H (endereço número 4) até chegar ao sistema terminal que se encontra na rede local do router H, de endereço 212.43.27.250, como desejado.

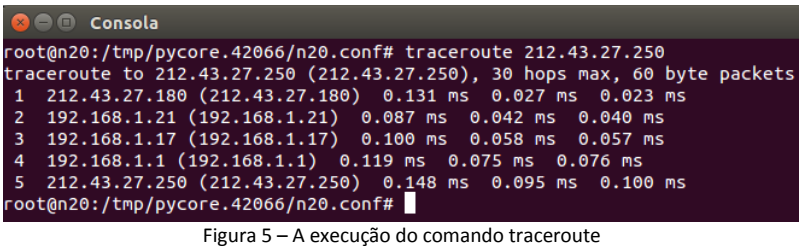

Figura 5 – A execução do comando traceroute

Através deste processo, foi-nos possível confirmar que a conectividade entre todos os nós da rede estava bem configurada.

2.3.3. Encaminhamento Dinâmico

Podemos ver o resultado, no router A, antes do corte (Figura 6, do lado esquerdo) e depois do corte (Figura 6, do lado direito), do protocolo RIP em ação entre os nós A e B. Podemos ver que depois do corte, como o caminho entre A e B desapareceu, aquando da construção da tabela de encaminhamento, o IP 192.168.1.20 já não aparece lá, devido a esse mesmo corte.

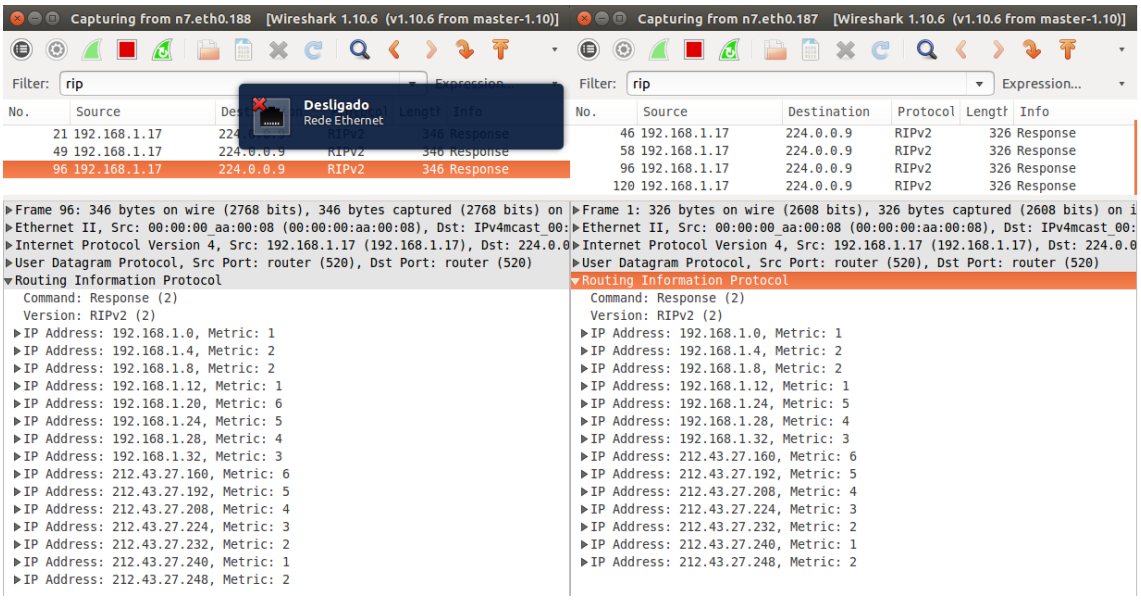

Figura 6 – O processo de RIP no nó A, antes e depois do corte

Finalmente, verificou-se pelo *Wireshark* que antes do corte entre o *router* A e o *router* B, há uma série de trocas de mensagens, como "Hello *Packet*", "*DB Description*", "*LS Request*" e "*LS Update*" em que vemos os protocolos a atualizar a tabela de encaminhamento (onde consegue encontrar todos os routers da topologia), iteração após iteração. Depois do corte, existe as mesmas trocas de mensagens, apenas fazendo mais iterações, para descobrir todos os *routers*.

2.3.4. Interligação via NAT

Para verificar a conetividade entre a rede privada e a rede externa, fizemos um ping entre o sistema terminal da rede privada e um sistema terminal fora dessa rede privada (um pertencente á LAN do router B por exemplo). Podemos ver isso a acontecer na Figura 7.

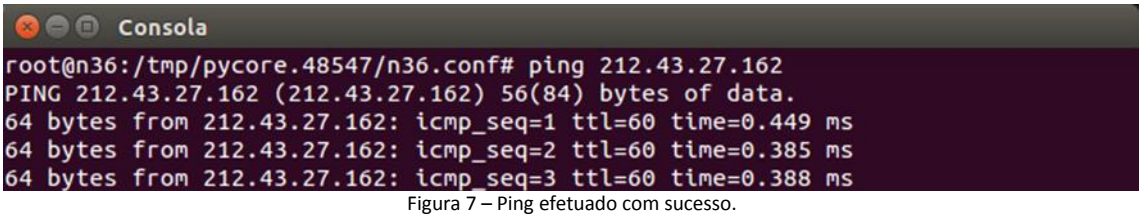

Figura 7 – Ping efetuado com sucesso.

Na figura seguinte podemos comprovar como a tradução é realizada com sucesso, dado que estamos a visualizar o mesmo evento a ocorrer (o processo de "request" e "repply" do protocolo ICMP) em ambas as janelas e o endereço "Source" muda, dado que na esquerda se encontra depois do router NAT e na direita se encontra dentro da nossa gama privada de endereços.

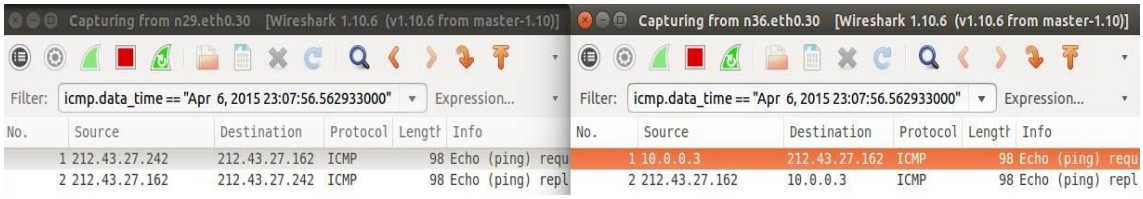

Figura 8 – Prova do funcionamento correto do NAT

Para criar um servidor HTTP, tivemos que instalar o mini-httpd na máquina nativa. Para isso fazemos, sudo apt-get install mini-httpd. Depois, vamos ao servidor com o endereço 10.0.0.2, abrimos a bash (o CORE tem de estar a correr a topologia) e digitamos, mini-httpd. Para testar a ligação ao serviço http, podemos abrir um sistema terminal e testamos a ligação através do seguinte comando, wget 10.0.0.2, se o resultado for 200 OK, quer dizer que temos uma ligação, tal como podemos observar pela Figura 9.

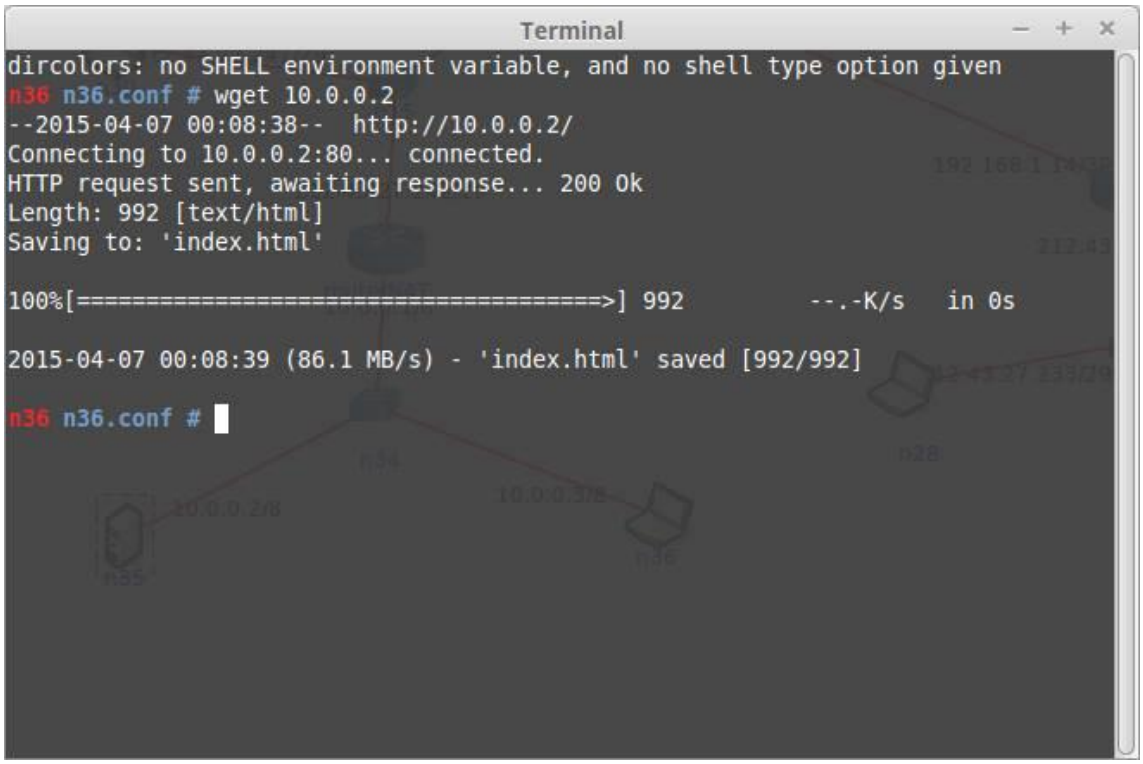

Figura 9 – Prova de que o servidor está funcional

3. Conclusão

Este primeiro trabalho prático de Redes de Computadores II, permitiu-nos consolidar vários conceitos fundamentais, tais como o Endereçamento, o Encaminhamento Estático e o Dinâmico. Também ajudou a melhorar a nossa habilidade e aptidão para com o emulador CORE e capturador de tráfego Wireshark.

Numa primeira parte, implementamos a topologia correspondente ao exercício teórico-prático 1 e elaboramos um protocolo de endereçamento dinâmico, utilizando o DHCP. Depois elaboramos um protocolo de encaminhamento estático, indo depois fazer uso de dois protocolos de encaminhamento dinâmico, o RIP e o OSPF. Na última parte deste trabalho, implementamos uma rede privada e configuramos o *router* com o NAT e criamos ainda um servidor HTTP.

Estamos convictos de que fizemos um trabalho bem-sucedido, cumprindo com a maioria dos requisitos pedidos. Saímos deste trabalho com a certeza de que alguns conceitos teóricos estão mais bem consolidados, pelo que o grau de conhecimento e à-vontade aumentou, o que indica que fizemos um trabalho competente, dentro das nossas possibilidades.

4. Referências

1 – “http://www.routeralley.com/guides/dhcp.pdf”, consultado em 01/04/2015

2 – “http://pplware.sapo.pt/tutoriais/networking/redes-encaminhamento-estatico-vs-encaminhamento-dinamico/”, consultado em 01/04/2015

3 – “http://www.routeralley.com/guides/rip.pdf”, consultado em 01/04/2015

4 – “http://www.routeralley.com/guides/ospf.pdf”, consultado em 01/04/2015

5 – “http://www.routeralley.com/guides/nat.pdf”, consultado em 01/04/2015