Licenciatura em Engenharia e Gestão de Sistemas de Informação

Universidade do Minho

Introdução às Redes de Computadores

2024/2025

Grupo 6 PL3

Afonso Leite

A108552

@uminho.pt

Rui Mendes

A108720

@uminho.pt

Zelong Ma

A108448

@uminho.pt

Índice

Índice de Ilustrações

Introdução

O presente trabalho prático tem como principal finalidade o desenvolvimento de competências na área das redes de computadores, através da conceção e implementação de um projeto baseado nas tecnologias Ethernet e TCP/IP.

Recorreu-se ao emulador de redes CORE (Common Open Research Emulator) para planear, configurar e testar uma rede de computadores funcional, aplicando os conhecimentos adquiridos nas aulas teóricas e teórico-práticas da unidade curricular.

Complementarmente, foi utilizada a ferramenta Wireshark para realizar a captura e análise de tráfego, permitindo observar o comportamento dos principais protocolos de comunicação nas diferentes camadas da pilha TCP/IP.

Este trabalho permitiu consolidar os conhecimentos técnicos adquiridos, promovendo também o desenvolvimento da capacidade de diagnóstico, resolução de problemas e trabalho colaborativo, num contexto de simulação próximo da realidade.

1-Emulação de LANs Ethernet

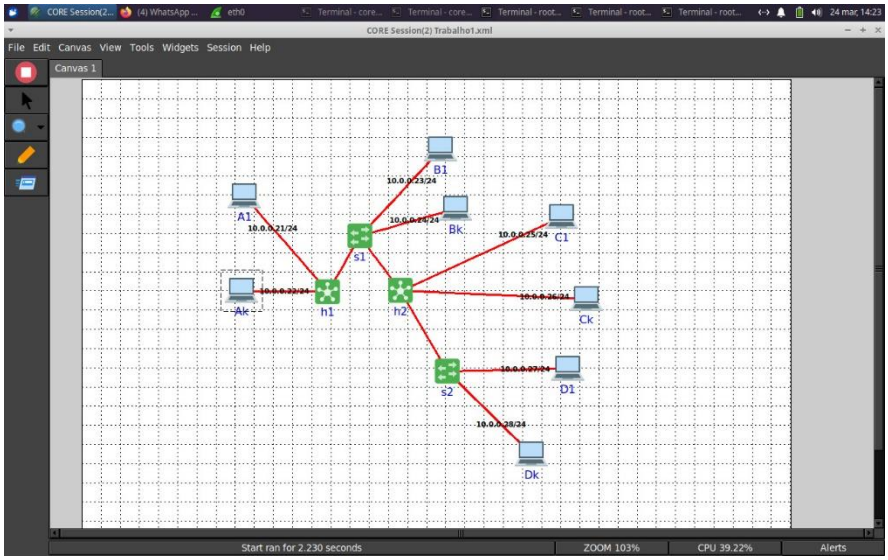


Figura 1-Topologia emulada no CORE

Neste exercício foi construída, no emulador CORE, uma rede local (LAN) com o objetivo de estudar o comportamento de dispositivos de camada 2 (Hubs e Switches), utilizando ferramentas de diagnóstico como o ping e o Wireshark para analisar os protocolos ARP e ICMP.

Construção da Topologia

A topologia emulada segue o modelo proposto no enunciado e inclui dois Hubs (H1 e H2) e dois Switches (S1 e S2), interligando um total de oito PCs, com pares de nós ligados a cada Hub ou Switch. Os PCs estão nomeados de forma a evidenciar a associação a cada segmento da rede:

Ao Hub H1 estão ligados os nós A1, Ak e B1, Bk, via Switch S1.

Ao Hub H2 estão ligados os nós C1, Ck e D1, Dk, via Switch S2.

Todos os nós têm endereços IP do mesmo espaço de endereçamento (ex.: 10.0.0.X/24), assegurando que pertencem à mesma rede e que existe conectividade IP entre eles.

Atribuição de Endereços IP e Configuração

Os endereços IP foram atribuídos automaticamente pelo CORE. Verificou-se que todos os terminais pertencem à mesma sub-rede, permitindo comunicação sem necessidade de gateways ou routers.

Comportamento dos Switches: Os Switches aprendem os endereços MAC com base no tráfego recebido e constroem uma tabela de encaminhamento. Quando recebem um ARP Request (que é um broadcast), este é transmitido a todas as portas. No entanto, respostas ARP ou pacotes ICMP unicast são enviados apenas à porta correta. Isso reduz significativamente o tráfego na rede e aumenta a eficiência.

Testes de Conectividade e Captura de Tráfego

Foram realizados testes de conectividade entre os vários nós da rede utilizando o comando ping. Verificou-se que os pacotes ICMP são corretamente enviados e recebidos entre todos os nós da rede, demonstrando a correta configuração IP e funcionamento da camada de ligação de dados (camada 2).

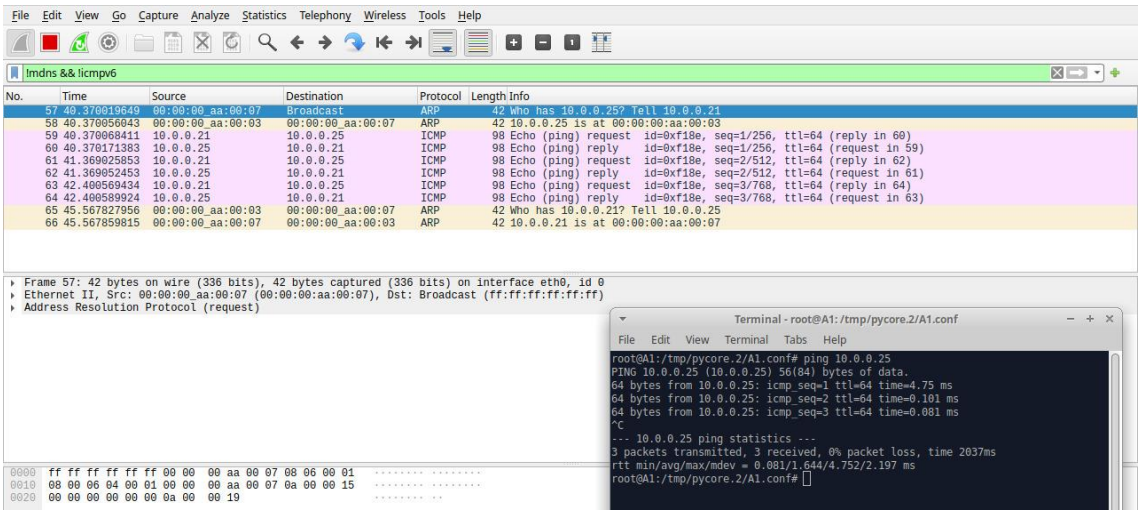


Figura 2-Resultado de ping no Wireshark

Durante a análise do tráfego de rede, foi possível observar uma troca de mensagens ARP entre dois dispositivos da rede local. Inicialmente, o host com endereço IP 10.0.0.21 envia uma solicitação ARP **(ARP Request**) a perguntar "Who has 10.0.0.25? Tell 10.0.0.21", ou seja, está a tentar descobrir o endereço MAC que corresponde ao IP 10.0.0.25. Em seguida, o host com IP 10.0.0.25 responde com uma mensagem ARP (**ARP Reply**) a informar que "10.0.0.25 is at 00:00:00:aa:00:03", revelando o seu endereço físico. Posteriormente, ocorreu uma nova solicitação ARP originada no host 10.0.0.25, agora a questionar "Who has 10.0.0.21? Tell 10.0.0.25", invertendo os papeís da consulta. O host 10.0.0.21 responde "10.0.0.21 is at 00:00:00:aa:00:07". Essa troca de ARP é fundamental para que os dispositivos possam associar os endereços IP aos endereços MAC correspondentes, permitindo a comunicação correta em uma rede local baseada em Ethernet.

Na imagem seguinte entramos no terminal do PC B1 e fizemos ping para o PC CK (ping 10.0.0.28). Em seguida, aplicamos o filtro !mdns && ! icmpv6 no Wireshark, para ignorarmos os pacotes que não nos interessam e ficamos à escuta a partir do PC D1. Este foi o nosso resultado:

Figura 3-Resultado de ping no Wireshark com filtro

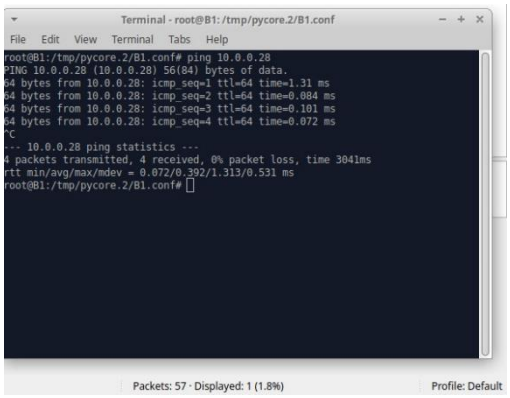


Figura 4-Ping para analise no Wireshark

Captura com Wireshark

As capturas de tráfego com o Wireshark permitiram analisar os seguintes comportamentos:

Os Hubs, por não terem qualquer tipo de inteligência ou tabela de endereços, replicam todos os pacotes recebidos para todas as suas portas. Assim, ao enviar um ping ou um ARP Request de um PC ligado a um Hub, todos os restantes dispositivos ligados a esse mesmo Hub recebem esses pacotes, mesmo que não sejam os destinatários.

Usando os Switches, a captura de tráfego mostrou que os pacotes foram encaminhados apenas para o destinatário correto, reduzindo o tráfego desnecessário. Apenas o primeiro pacote ARP foi transmitido para toda a rede, permitindo a resolução de endereços MAC/IP. Após isso, os Switches aprenderam os endereços e encaminharam os pacotes diretamente.

2- Interligação de LANs e redes IP

Rede	**Cálculo do número de endereços a atribuir a cada rede**	**M = 19 Resultado**
A	$2^{32-(19+1)}-3$	4093
B	$2^{32-(19+2)}+2^{32-(M+4)}-6$	2554
C	$2^{32-(19+3)})-3$	1021
D	$2^{32-(19+5)}-3$	253
E	$2^{32-(19+5)}-3$	253

Tabela de Endereçamento das Redes LAN (Gama 10.6.0.0/19)

	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}
	/19	/20	/21	22/	/23	/24	/25	/26	/27	/28	/29	/30	/31	/32
A	0	0	H	H	H	H	H	H	H	H	H	H	H	H
B1	0	1	0	H	H	H	H	H	H	H	H	H	H	H
C	0	1	1	0	H	H	H	H	H	H	H	H	H	H
B2	0	1	1	1	0	H	H	H	H	H	H	H	H	H
E	0	1	1	1	1	0	H	H	H	H	H	H	H	H
D	0	1	1	1	1	1	H	H	H	H	H	H	H	H

H - host

Rede	Número de Endereços	Endereço de Rede	Endereço de Difusão	Máscara	Gama de Endereços
A	4096	10.6.0.0	10.6.15.255	/20	10.6.0.1 - 10.6.15.254
B1	2048	10.6.16.0	10.6.23.255	/21	10.6.16.1 - 10.6.23.254
C	1024	10.6.24.0	10.6.27.255	/22	10.6.24.1 - 10.6.27.254
B2	512	10.6.28.0	10.6.29.255	/23	10.6.28.1 - 10.6.29.254
E	256	10.6.30.0	10.6.30.255	/24	10.6.30.1 - 10.6.30.254
D	256	10.6.31.0	10.6.31.255	/24	10.6.31.1 - 10.6.31.254

Ligações entre Routers (Gama 192.168.0.0/24)

Para o encaminhamento, optamos por seguir a métrica do menor número de saltos. Algumas rotas podem ter mais que um caminho com o mesmo número de saltos. Colocamos na tabela identificado com um * aqueles que são os caminhos alternativos. Esta redundância contribui para uma maior resiliência da rede, permitindo contornar falhas e manter a conectividade através de rotas alternativas.

	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1
	/25	/26	/27	/28	/29	/30	/31	/32
RA-R1	0	0	0	0	0	0	H	H
RB-R2	0	0	0	0	0	1	H	H
R1-R2	0	0	0	0	1	0	H	H
R2-R3	0	0	0	0	1	1	H	H
R2-R4	0	0	0	1	0	0	H	H
R4-R3	0	0	0	1	0	1	H	H
R4-RD	0	0	0	1	1	1	H	H
R1-R3	0	0	1	0	0	0	H	H
R1-RC	0	0	1	0	0	1	H	H
R3-RC	0	0	1	0	1	1	H	H
R3-RE	0	0	1	1	1	1	H	H
RC-RE	0	1	0	0	0	0	H	H

	Rede	Endereço de Rede	Endereço de Difusão	Máscara	Gama de Endereço
1	RA-R1	192.168.0.0	192.168.0.3	/30	192.168.0.1 - 192.168.0.2
2	RB-R2	192.168.0.4	192.168.0.7	/30	192.168.0.5 - 192.168.0.6
3	R1-R2	192.168.0.8	192.168.0.11	/30	192.168.0.9 - 192.168.0.10
4	R2-R3	192.168.0.12	192.168.0.15	/30	192.168.0.13 - 192.168.0.14
5	R2-R4	192.168.0.16	192.168.0.19	/30	192.168.0.17 - 192.168.0.18
6	R4-R3	192.168.0.20	192.168.0.23	/30	192.168.0.21 - 192.168.0.22
7	R4-RD	192.168.0.24	192.168.0.27	/30	192.168.0.25 - 192.168.0.26

8	R1-R3	192.168.0.28	192.168.0.31	/30	192.168.0.29 - 192.168.0.30
9	R1-RC	192.168.0.32	192.168.0.35	/30	192.168.0.33 - 192.168.0.34
10	R3-RC	192.168.0.36	192.168.0.39	/30	192.168.0.37 - 192.168.0.38
11	R3-RE	192.168.0.40	192.168.0.43	/30	192.168.0.41 - 192.168.0.42
12	RC-RE	192.168.0.44	192.168.0.47	/30	192.168.0.45 - 192.168.0.46
13	RD-RE	192.168.0.48	192.168.0.51	/30	192.168.0.49 - 192.168.0.50

Tabela de Encaminhamento das Redes LAN

LAN A	Rede Destino	Máscara	Interface de Saída	Próximo Nó
A	10.6.0.0	/20	10.6.0.1	----------
B1	10.6.16.0	/21	192.168.0.1	192.168.0.2
C	10.6.24.0	/22	192.168.0.1	192.168.0.2
B2	10.6.28.0	/23	192.168.0.1	192.168.0.2
D	10.6.31.0	/24	192.168.0.1	192.168.0.2
E	10.6.30.0	/24	192.168.0.1	192.168.0.2

LAN B	Rede Destino	Máscara	Interface de Saída	Próximo Nó
A	10.6.0.0	/20	192.168.0.5	192.168.0.6
B1	10.6.16.0	/21	10.6.16.1	----------
C	10.6.24.0	/22	192.168.0.5	192.168.0.6
B2	10.6.28.0	/23	10.6.28.1	----------
D	10.6.31.0	/24	192.168.0.5	192.168.0.6
E	10.6.30.0	/24	192.168.0.5	192.168.0.6

LAN C	Rede Destino	Máscara	Interface de Saída	Próximo Nó
A	10.6.0.0	/20	192.168.0.34	192.168.0.33
B1	10.6.16.0	/21	192.168.0.34	192.168.0.33
B1*	10.6.28.0	/23	192.168.0.34	192.168.0.33

C	10.6.24.0	/22	10.6.24.1	----------
B2	10.6.28.0	/23	192.168.0.38	192.168.0.37
B2*	10.6.28.0	/23	192.168.0.38	192.168.0.37
D	10.6.31.0	/24	192.168.0.45	192.168.0.46
E	10.6.30.0	/24	192.168.0.45	192.168.0.46

LAN D	Rede Destino	Máscara	Interface de Saída	Próximo Nó
A	10.6.0.0	/20	192.168.0.26	192.168.0.25
A*	10.6.0.0	/20	192.168.0.49	192.168.0.50
B1	10.6.16.0	/21	192.168.0.26	192.168.0.25
C	10.6.24.0	/22	192.168.0.49	192.168.0.50
B2	10.6.28.0	/23	192.168.0.26	192.168.0.25
D	10.6.31.0	/24	10.6.31.1	----------
E	10.6.30.0	/24	192.168.0.49	192.168.0.50

LAN E	Rede Destino	Máscara	Interface de Saída	Próximo Nó
A	10.6.0.0	/20	192.168.0.42	192.168.0.41
A*	10.6.0.0	/20	192.168.0.46	192.168.0.45
B1	10.6.16.0	/21	192.168.0.42	192.168.0.41
C	10.6.24.0	/22	192.168.0.46	192.168.0.45
B2	10.6.28.0	/23	192.168.0.42	192.168.0.41
D	10.6.31.0	/24	192.168.0.50	192.168.0.49
E	10.6.30.0	/24	10.6.30.1	----------

LAN R1	Rede Destino	Máscara	Interface de Saída	Próximo Nó
A	10.6.0.0	/20	192.168.0.2	192.168.0.1
B1	10.6.16.0	/21	192.168.0.9	192.168.0.10
C	10.6.24.0	/22	192.168.0.33	192.168.0.34
B2	10.6.28.0	/23	192.168.0.9	192.168.0.10

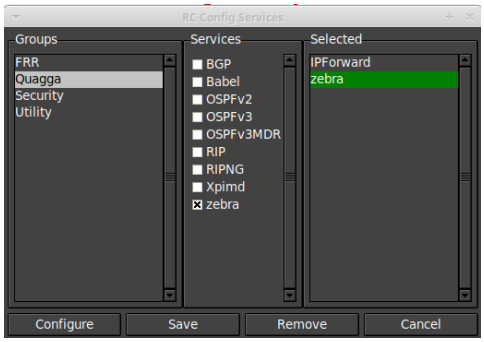
D	10.6.31.0	/24	192.168.0.29	192.168.0.30
E	10.6.30.0	/24	192.168.0.29	192.168.0.30

LAN R2	Rede Destino	Máscara	Interface de Saída	Próximo Nó
A	10.6.0.0	/20	192.168.0.10	192.168.0.9
B1	10.6.16.0	/21	192.168.0.6	192.168.0.5
C	10.6.24.0	/22	192.168.0.10	192.168.0.9
B2	10.6.28.0	/23	192.168.0.6	192.168.0.5
C*	10.6.24.0	/22	192.168.0.13	192.168.0.14
D	10.6.31.0	/24	192.168.0.17	192.168.0.18
E	10.6.30.0	/24	192.168.0.13	192.168.0.14

LAN R3	Rede Destino	Máscara	Interface de Saída	Próximo Nó
A	10.6.0.0	/20	192.168.0.30	192.168.0.29
B1	10.6.16.0	/21	192.168.0.14	192.168.0.13
C	10.6.24.0	/22	192.168.0.37	192.168.0.38
B2	10.6.28.0	/23	192.168.0.14	192.168.0.13
D	10.6.29.0	/24	192.168.0.22	192.168.0.21
D*	10.6.31.0	/24	192.168.0.41	192.168.0.42
E	10.6.30.0	/24	192.168.0.41	192.168.0.42

LAN R4	Rede Destino	Máscara	Interface de Saída	Próximo Nó
A	10.6.0.0	/20	192.168.0.18	192.168.0.17
A*	10.6.0.0	/20	192.168.0.21	192.168.0.22
B1	10.6.16.0	/21	192.168.0.18	192.168.0.17
C	10.6.24.0	/22	192.168.0.21	192.168.0.22
B2	10.6.28.0	/23	192.168.0.18	192.168.0.17
D	10.6.31.0	/24	192.168.0.25	192.168.0.26
E	10.6.30.0	/24	192.168.0.21	192.168.0.22

E*	10.6.30.0	/24	192.168.0.25	192.168.0.26



Desativado o encaminhamento dinâmico.

Nesta etapa, foi desativado o encaminhamento dinâmico por meio da desativação dos protocolos OSPFv2 (para IPv4) e OSPFv3 (para IPv6) em todos os routers da topologia. Essa medida teve como objetivo substituir o roteamento dinâmico por rotas estáticas, conforme o esquema de encaminhamento definido anteriormente.

Figura 5-Ativação zebra

Nas imagens abaixo encontram-se as configurações zebra de todos os routers para o encaminhamento estático.

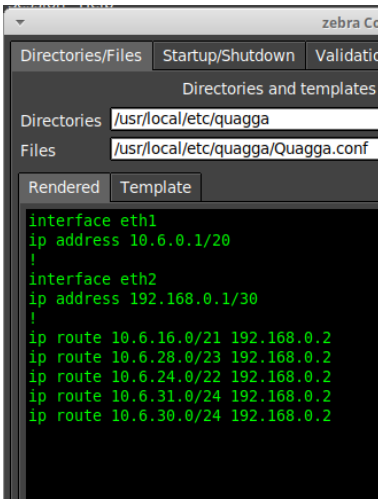


Figura 6- Router RA

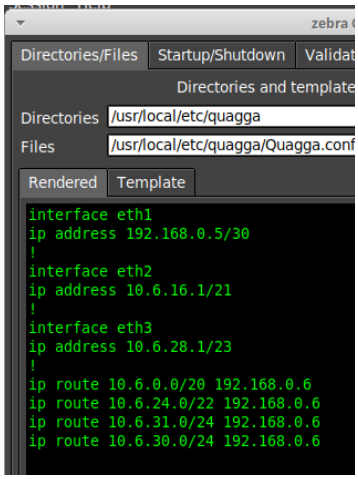


Figura 7- Router RB

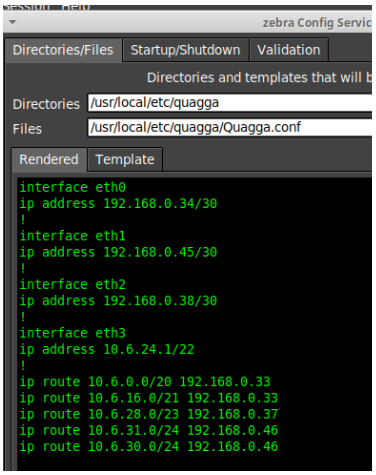


Figura 8- Router RC

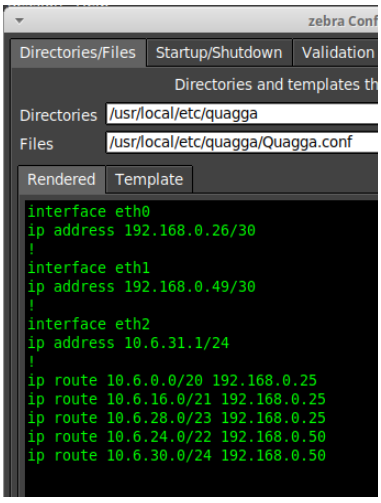


Figura 9- Router RD

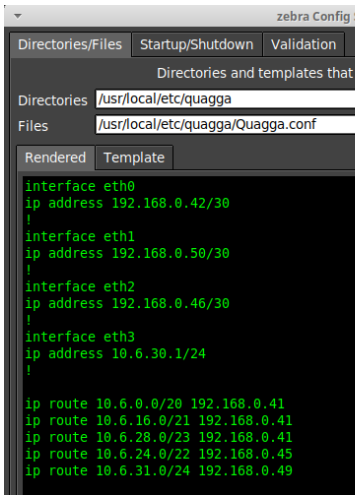


Figura 10- Router RE

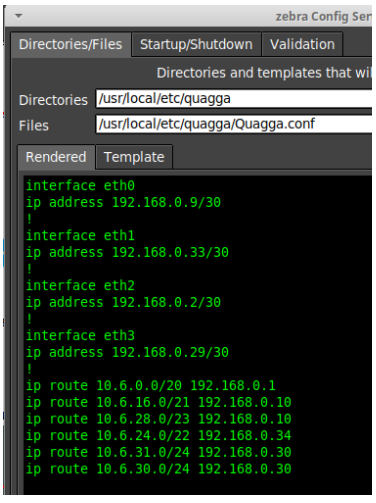


Figura 11- Router R1

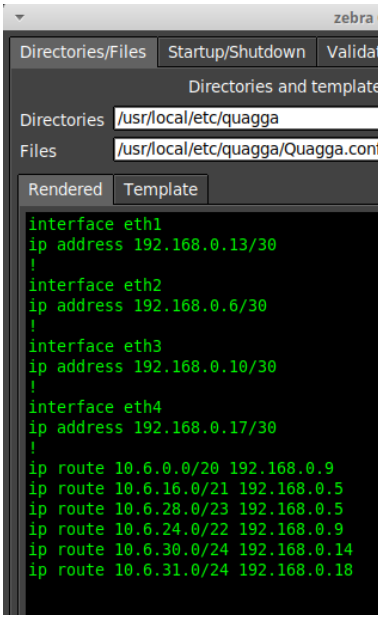


Figura 12 Router R2

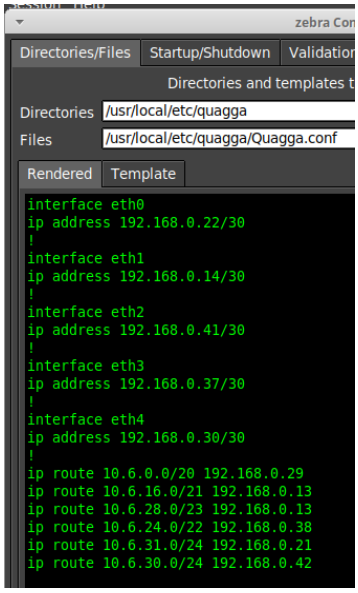


Figura 13- Router R3

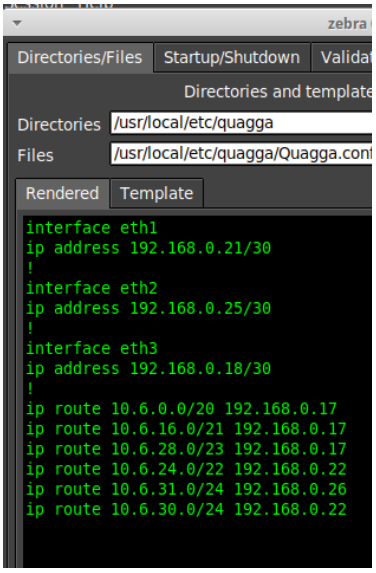


Figura 14- Router R4

Teste de Conectividade

Para testar a conectividade entre as diferentes redes, utilizamos os comandos ping e traceroute. No Wireshark para capturar e analisar o tráfego de rede no pc para o qual o ping é enviado.

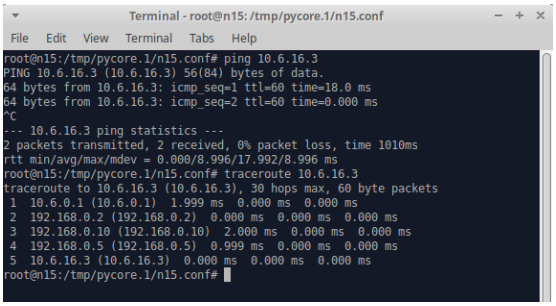


Figura 15-Conectividade entre RA-RB1

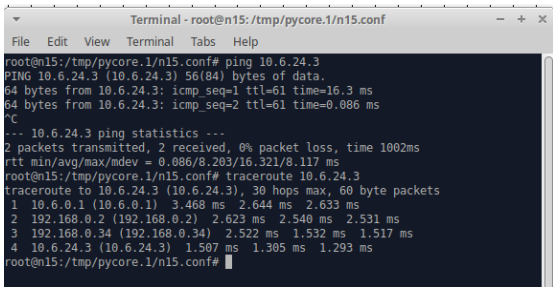


Figura 16-Conectividade entre RA-RC

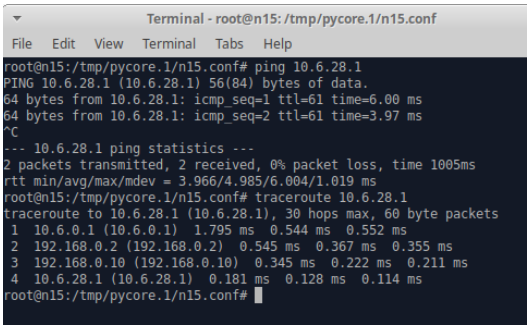


Figura 17--Conectividade entre RA-RB2

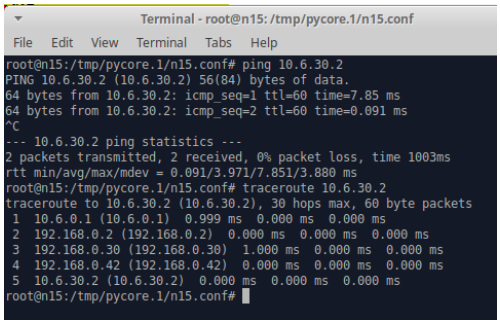


Figura 18--Conectividade entre RA-RE

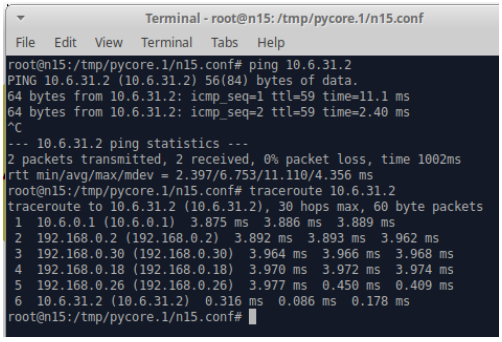


Figura 19-Conectividade entre RA-RD

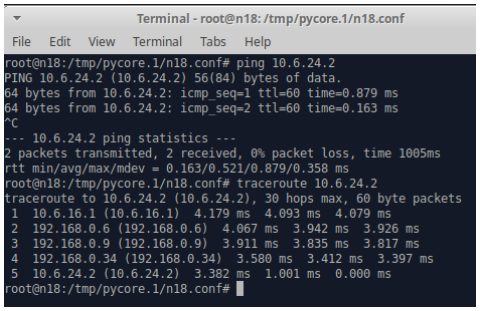


Figura 20-Conectividade entre RB1-RC

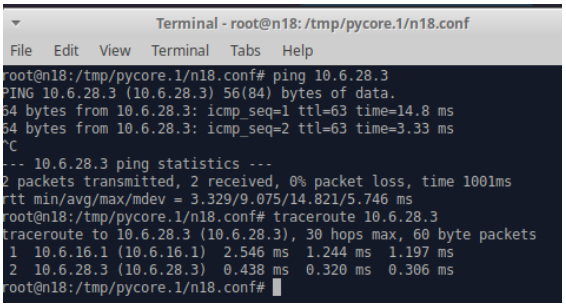


Figura 21-Conectividade entre RB1-RB2

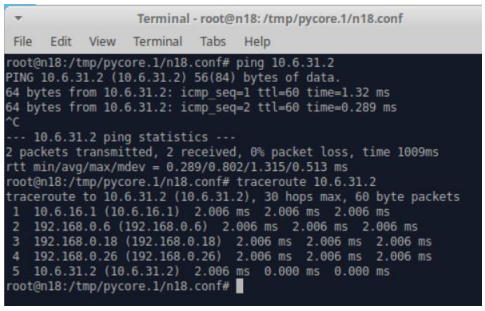


Figura 22-Conectividade entre RB1-RD

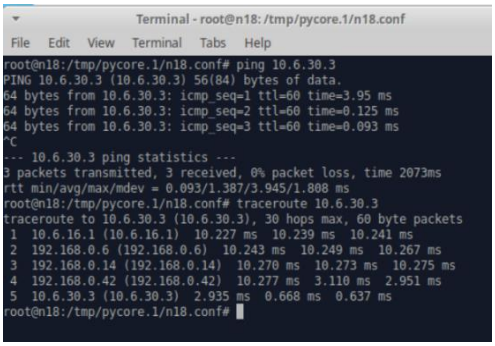


Figura 23-Conectividade entre RB1-RE

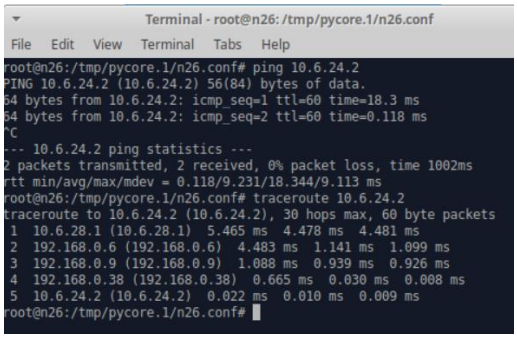


Figura 24-Conectividade entre RB2-RC

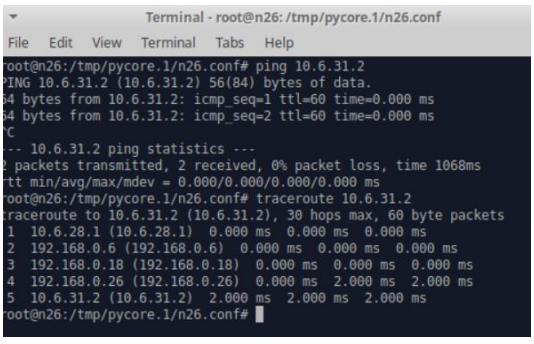


Figura 25-Conectividade entre RB2-RD

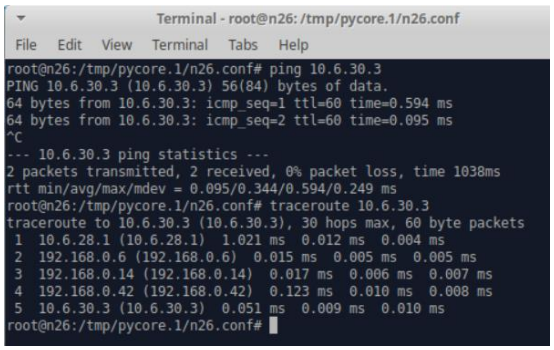


Figura 26-Conectividade entre RB2-RE

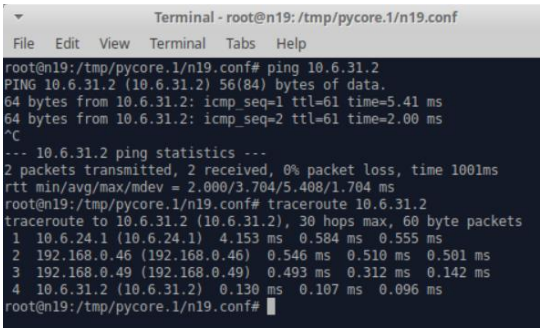


Figura 27-Conectividade entre RC-RD

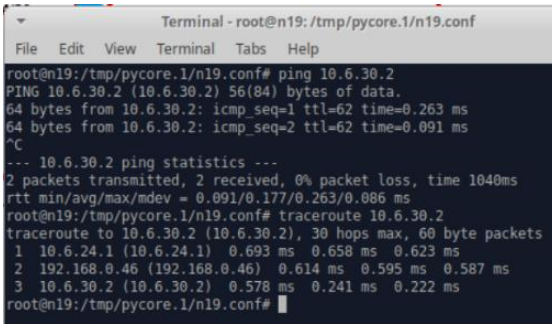


Figura 28-Conectividade entre RC-RE

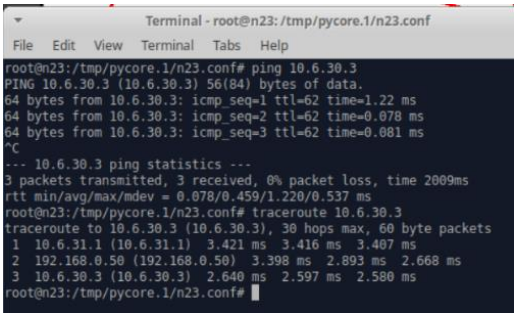


Figura 29-Conectividade entre RD-RE

Adição de mais dispositivos na LAN E utilizando a rede 192.168.100.0/26

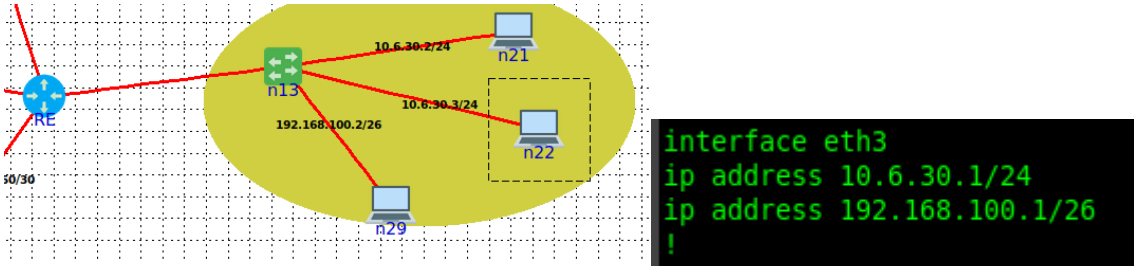


Figura 30-Nova rede na LanE

Para permitir que uma interface de rede pertença simultaneamente a duas redes IP distintas na mesma LAN física, utilizou-se a configuração do Zebra, que permite a atribuição de múltiplos endereços IP à mesma interface. Para que haja comunicação entre dispositivos de redes diferentes na mesma LAN, é necessário um router com os endereços configurados em ambas as redes. Antes de aplicar a configuração no Zebra, foi necessário remover o IP previamente atribuído pelo sistema, para evitar conflitos ou sobreposição, garantindo que a nova configuração seja corretamente aplicada.

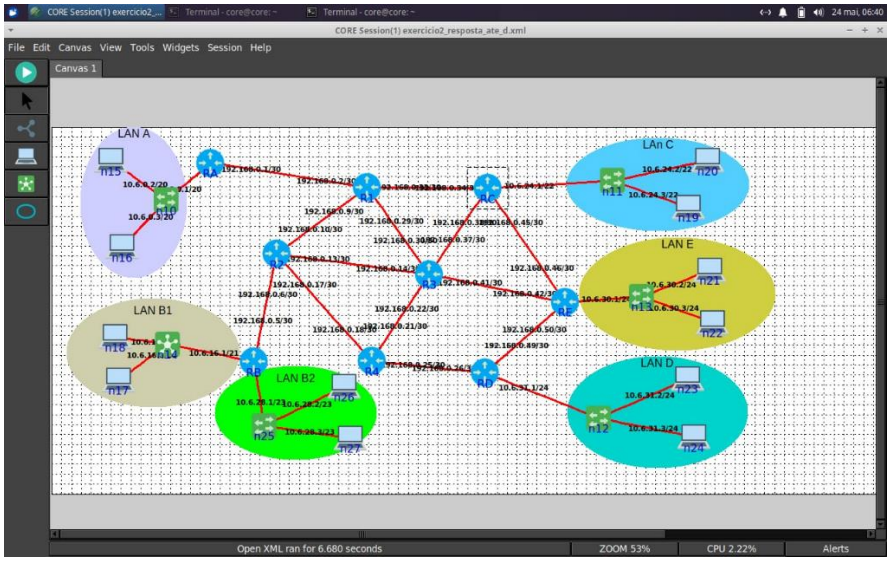


Figura 31-Topologia final 2

3-DHCP–Dynamic Host Configuration Protocol

O DHCP é um protocolo que atribui dinamicamente um IP a cada cliente da mesma rede. Este protocolo facilita a ligação de um dispositivo a uma nova rede, dado que, sendo um dispositivo novo, não é necessário configurar um IP estático, aumentando assim a simplicidade da utilização.

Configuração:

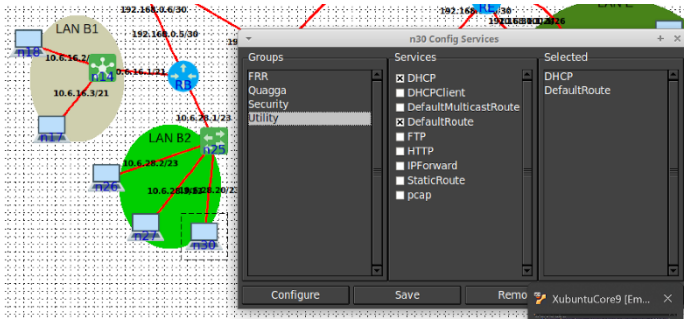


Figura 32-Ativação DHTP

Em primeiro lugar, foi feita a instalação a partir do comando “sudo apt-get install isc-dhcp-server” e de seguida a configuração dos sistemas nos terminais.

A máquina n30 foi configurada como servidor DHCP.

As restantes máquinas foram definidas como clientes DHCP, recebendo os IPs dinamicamente. Também foi feita a configuração de range para o endereço de gama do B1 definido no exercício anterior.

```
# auto-generated by DHCP service (utility.py)
# NOTE: move these option lines into the desired pool { } block(s) below
#option domain-name "test.com";
#option domain-name-servers 10.0.0.1;
#option routers 10.0.0.1;

log-facility local6;

default-lease-time 600;
max-lease-time 7200;

ddns-update-style none;

subnet 10.6.28.0 netmask 255.255.254.0 {
  pool {
    range 10.6.28.1 10.6.29.254;
    default-lease-time 600;
    option routers 10.6.28.20;
  }
}
```

Figura 33-Configuração DHCP

Processo de Atribuição de IP via DHCP

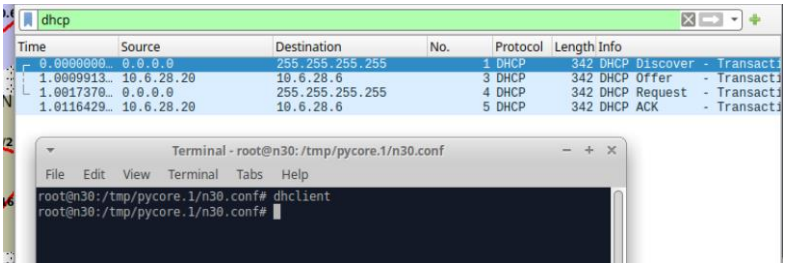


Figura 34-Leitura DHCP no Wireshark

Utilizando o filtro “dhcp” para uma melhor visualização, podemos observar as quatro fases de dhcp, Discover - Offer - request - Ack.

O processo de atribuição dinâmica de endereços IP através do protocolo DHCP inicia-se quando um cliente envia uma mensagem DHCP Discover em broadcast para a rede local. Nesta fase inicial, como o cliente ainda não possui um endereço IP configurado, o campo de endereço de origem aparece como 0.0.0.0. Esta mensagem é recebida pelo servidor DHCP configurado na rede.

Na fase seguinte, o servidor DHCP responde com uma mensagem DHCP Offer, também enviada em broadcast. Esta mensagem contém a proposta de configuração de rede para o cliente (10.6.28.6). Ao receber a proposta, o cliente DHCP envia então uma mensagem DHCP Request, também em broadcast, indicando ao servidor que aceita os parâmetros propostos e solicita formalmente a atribuição desse endereço IP. Esta etapa permite que outros servidores DHCP eventualmente presentes na rede tomem conhecimento da seleção feita pelo cliente.

Finalmente, o servidor DHCP responde com uma mensagem DHCP ACK (acknowledgement), confirmando a atribuição do endereço IP e dos demais parâmetros ao cliente. Só após a receção desta mensagem final é que o cliente passa a utilizar a configuração de rede atribuída, completando assim o processo de leasing DHCP. Todo este processo ocorre automaticamente sem necessidade de intervenção manual, demonstrando a eficiência do protocolo DHCP na gestão dinâmica de endereços numa rede.

4-Uso das camadas de rede e transporte por parte das aplicações

Um servidor HTTP (HyperText Transfer Protocol) é um serviço de rede que permite disponibilizar conteúdos web (como páginas HTML, imagens ou ficheiros) através do protocolo HTTP. Quando é instalado num computador (ou nó) dentro de uma rede simulada no CORE, esse servidor

permite que os restantes nós da mesma rede acedam a esses conteúdos utilizando um navegador web ou comandos como curl ou wget.

Ou seja, instalar um servidor HTTP consiste em configurar um dos PCs virtuais da rede (isto é, um nó) para funcionar como servidor web. Esse nó irá então responder aos pedidos enviados por outros nós que pretendam aceder ao conteúdo web que ele disponibiliza.

Configuração:

Em primeiro lugar, foi feita a instalação a partir do comando " sudo apt install apache2 " e o " sudo apt install links2"

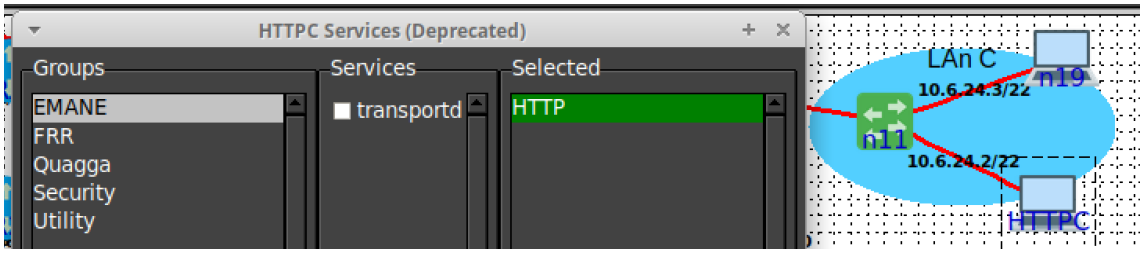


Figura 35-Ativação http

Adotamos um dos PCS como servidor HTTP, foi renomeado para HTTPC para uma melhor visualização. Para testar a conectividade usar um pc da rede D, usando comando wget 10.6.24.2.

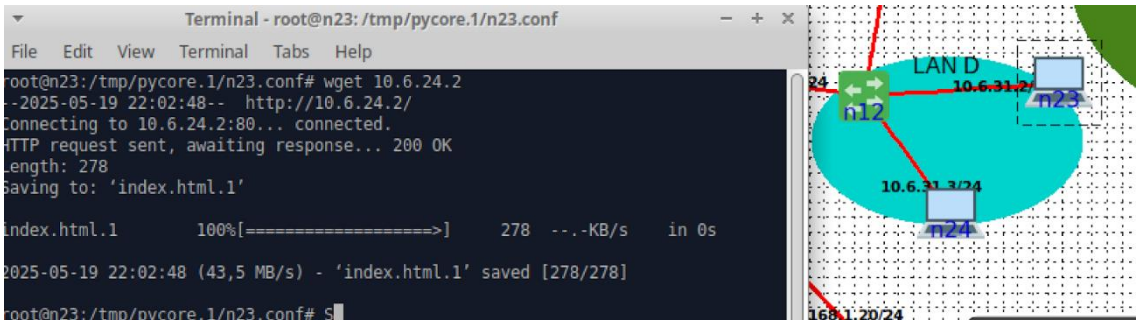


Figura 36-Testar conectividade http

10.6.31.2	10.6.24.2	13 TCP	74 49926 → 80 [SYN] Seq=0 Win=64240 Len=
10.6.24.2	10.6.31.2	14 TCP	74 80 → 49926 [SYN, ACK] Seq=0 Ack=1 Wi
10.6.31.2	10.6.24.2	15 TCP	66 49926 → 80 [ACK] Seq=1 Ack=1 Win=642
10.6.31.2	10.6.24.2	16 HTTP	190 GET / HTTP/1.1
10.6.24.2	10.6.31.2	17 TCP	66 80 → 49926 [ACK] Seq=1 Ack=125 Win=6
10.6.24.2	10.6.31.2	18 HTTP	604 HTTP/1.1 200 OK
10.6.31.2	10.6.24.2	19 TCP	66 49926 → 80 [ACK] Seq=125 Ack=539 Win=
10.6.31.2	10.6.24.2	20 TCP	66 49926 → 80 [FIN, ACK] Seq=125 Ack=53
10.6.24.2	10.6.31.2	21 TCP	66 80 → 49926 [FIN, ACK] Seq=539 Ack=12
10.6.31.2	10.6.24.2	22 TCP	66 49926 → 80 [ACK] Seq=126 Ack=540 Win=
00:00:00_aa:00:19	00:00:00_aa:00:16	23 ARP	42 Who has 10.6.24.1? Tell 10.6.24.2
00:00:00_aa:00:16	00:00:00_aa:00:19	24 ARP	42 Who has 10.6.24.2? Tell 10.6.24.1
00:00:00_aa:00:19	00:00:00_aa:00:16	25 ARP	42 10.6.24.2 is at 00:00:00:aa:00:19
00:00:00_aa:00:16	00:00:00_aa:00:19	26 ARP	42 10.6.24.1 is at 00:00:00:aa:00:16

Figura 37-Wireshark http

Na captura feita por wireshark, podemos verificar uma comunicação HTTP entre um cliente (10.6.31.2) e um servidor (10.6.24.2). Primeiro, ocorre o three-way handshake para estabelecer a conexão TCP: o cliente envia um SYN, o servidor responde com SYN-ACK e o cliente confirma com o ACK. Após isso, o cliente faz uma requisição HTTP GET(wget) no pacote 16. O servidor responde com HTTP/1.1 200 OK no pacote 18, indicando assim sucesso no carregamento da página. Finalmente, a conexão é encerrada com a troca de pacotes FIN e ACK entre o cliente e o servidor.

Foi criado um formulário simples em HTML contendo dois campos: username e password. E o conteúdo do mesmo é:

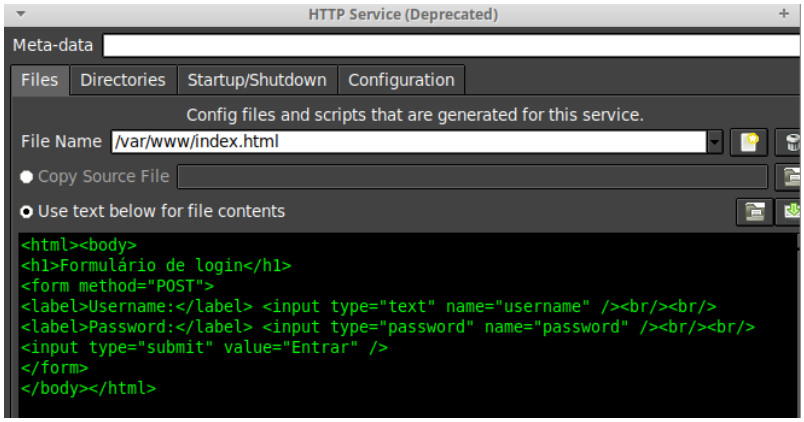


Figura 38-Formulário de login HTML

Usamos o comando "links2" e acedemos à página Web. Preenchemos o URL e seguidamente somos direcionados para o formulário de login, onde preenchemos os campos de username e da password e submetemos o formulário. Esta ação cria um pedido HTTP POST, visível na captura de tráfego realizada com o Wireshark.

Figura 39-Links2 execução de formulário

A imagem abaixo mostra os pacotes capturados durante o processo de envio do formulário. É possível identificar:

- Estabelecimento da ligação TCP através do handshake em três fases: SYN, SYN-ACK e ACK;
- Submissão do formulário através de um pedido HTTP POST com os dados username e password;
- Resposta do servidor com um HTTP 200 OK, indicando o sucesso na receção e processamento dos dados;
- Encerramento da ligação TCP com a troca de pacotes FIN e ACK entre o cliente e o servidor.

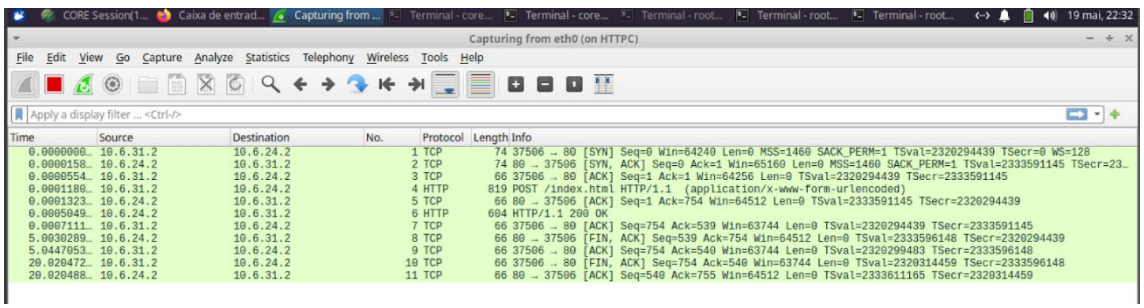


Figura 40-Captura HTTP POST do formulário de login

Para demonstrar o funcionamento de uma comunicação HTTP através do protocolo TCP, foi realizada uma captura de pacotes com o Wireshark. A análise mostra todas as fases essenciais da ligação entre cliente e servidor, desde o estabelecimento da sessão TCP até à troca de dados e ao encerramento da ligação.

O processo inicia-se quando o cliente, com o endereço IP 10.6.31.2, envia uma mensagem TCP com a flag SYN ativa para o servidor 10.6.24.2, utilizando a porta de origem 37566 e a porta de destino 80 (utilizada por defeito para HTTP). Esta mensagem indica a intenção de iniciar uma ligação. Em resposta, o servidor envia um segmento TCP com as flags SYN e ACK, confirmando a receção do pedido e aceitando estabelecer a ligação. Por fim, o cliente responde com um segmento ACK, completando assim o processo de three-way handshake que estabelece a sessão TCP entre as duas máquinas.

Imediatamente após a ligação estar estabelecida, o cliente envia um pedido HTTP do tipo GET, solicitando o ficheiro "/index.html". O servidor responde com uma mensagem HTTP contendo o código 200 OK, indicando que o pedido foi bem-sucedido, e envia o conteúdo da página solicitada. Durante esta fase, são trocadas várias mensagens de confirmação (ACK), assegurando que os dados são corretamente recebidos por ambas as partes.

Uma vez concluída a transferência dos dados, o cliente inicia o processo de encerramento da ligação TCP ao enviar uma mensagem com a flag FIN, sinalizando que não irá enviar mais dados. O servidor responde com uma mensagem FIN, ACK, aceitando terminar a comunicação. Por fim, o cliente envia um último ACK, concluindo assim a sessão.

Todo este processo decorre de forma automática, garantindo assim uma comunicação fiável e estruturada entre o cliente e o servidor. A análise desta captura evidencia claramente o funcionamento conjunto dos protocolos TCP e HTTP na troca de informação na Internet.

5-Interligação via Network Address Translation

O Network Address Translation (NAT) é uma técnica utilizada em redes de computadores que permite a tradução de endereços IP privados (internos) para um endereço IP público (externo).

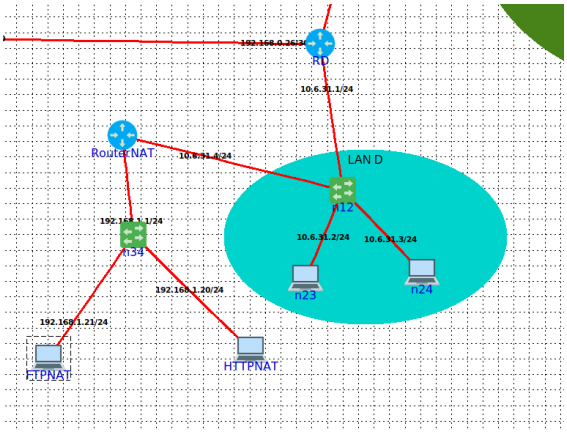


Figura 42-Topologia de NAT e FTP

Redireciona tráfego externo para o servidor HTTP interno
Redireciona tráfego externo para o servidor FTP interno (vsftpd)

```
#!/bin/bash

# Limpar todas as regras do iptables
iptables -F
iptables -t nat -F

# Habilitar o encaminhamento de IP
sysctl -w net.ipv4.ip_forward=1

# NAT - mascarar saída na interface externa eth0
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

# DNAT para HTTP no 192.168.1.20
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j DNAT --to-destination 192.168.1.20:80

# DNAT para FTP no 192.168.1.21
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 21 -j DNAT --to-destination 192.168.1.21:21

# Permitir encaminhamento para conexões estabelecidas/relacionadas
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT

# Permitir encaminhamento para os servidores HTTP e FTP
iptables -A FORWARD -p tcp -d 192.168.1.20 --dport 80 -j ACCEPT
iptables -A FORWARD -p tcp -d 192.168.1.21 --dport 21 -j ACCEPT

# Política padrão FORWARD (ajuste conforme sua necessidade)
iptables -P FORWARD ACCEPT

# Regras de roteamento via vtysh
vtysh -c 'configure terminal' \
  -c 'ip forwarding' \
  -c 'ip route 10.6.16.0/21 10.6.31.1' \
  -c 'ip route 10.6.28.0/23 10.6.31.1' \
  -c 'ip route 10.6.24.0/22 10.6.31.1' \
  -c 'ip route 10.6.31.0/24 10.6.31.1' \
  -c 'ip route 10.6.30.0/24 10.6.31.1' \
  -c 'ip route 192.168.100.0/26 10.6.31.1' \
  -c 'ip route 192.168.1.0/24 10.6.31.1' \
  -c 'exit'
```

Figura 41-Configuração Firewall da NAT

Como se pode observar na imagem incluída anteriormente, a topologia da rede foi expandida com a criação de uma rede local adicional usando a gama de endereços 192.168.1.0/24. Esta nova rede está ligada ao Router

A instalação do FTP foi feita com o comando “sudo apt install vsftpd” de seguida, a configuração do NAT (Network Address Translation) foi efetuada diretamente no router RouterNAT, utilizando iptables responsável pela criação de regras de firewall. O RouterNAT, seu ip público é 10.6.31.4, possui duas interfaces: uma ligada à rede privada com o endereço IP 192.168.1.1/24 e outra ligada às redes públicas com o endereço IP 10.6.31.4/24. Esta estrutura permite que o router funcione como intermediário entre as duas redes, aplicando as devidas regras de tradução de endereços e encaminhamento de tráfego.

Para corrigir o erro: "500 OOPS: vsftpd: refusing to run with writable root inside chroot()" usamos “chmod 755 var/run/vsftpd/empty” e “chmod 755 var/ftp” estes ,definem permissões de acesso que permitem ao servidor FTP funcionar corretamente, evitando erros de permissão durante conexões.

Relatório de Análise Técnica- Conexão FTP, HTTP e NAT

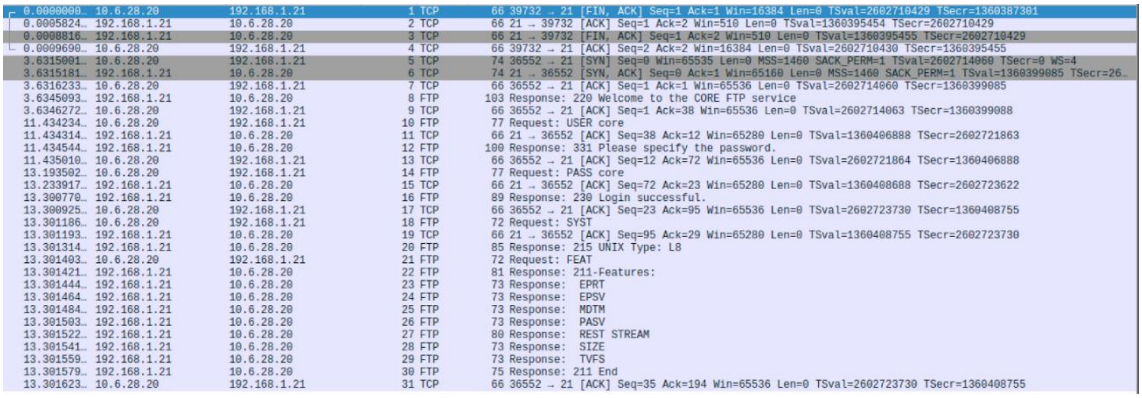


Figura 43-Conexão FTP,HTTP,NAT

Foi realizada com sucesso uma conexão FTP entre o cliente 10.0.26.20 (rede interna) e o servidor 10.6.31.4 (rede NAT). A análise da Wireshark revelou detalhes importantes sobre a comunicação. O cliente na rede interna (10.0.26.20) estabeleceu comunicação com o servidor FTP após NAT (192.168.1.21), demonstrando o correto funcionamento da tradução de endereços de rede. A captura mostrou claramente o handshake TCP inicial (SYN, SYN-ACK, ACK) na porta 21 (controlo FTP), seguido pela troca de mensagens de autenticação. O tráfego subsequente incluiu comandos adicionais como SYST (para identificação do sistema) e negociação de recursos (FEAT, EPSV), com respostas adequadas do servidor.

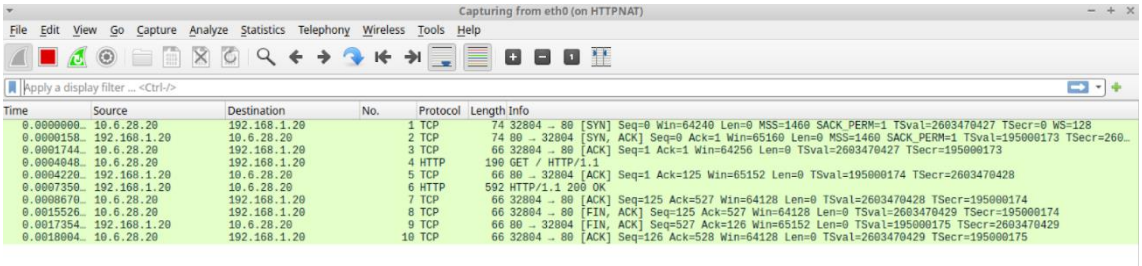


Figura 44-Fluxo de comunicação wireshark NAT

A análise simultânea no Wireshark capturou todo o fluxo de comunicação, A captura demonstra claramente o funcionamento adequado do NAT, com a tradução correta de endereços entre as redes interna (10.6.28.20) e externa (192.168.1.20), permitindo a comunicação bidirecional sem erros.