



Serviço de
Notícias



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE
COIMBRA

dei25

departamento
de engenharia informática
1998 - 2020

RELATÓRIO

Unidade Curricular: Redes de Comunicação

Data: 30/03/2024

Luana Carolina Reis, nº 2022220606

Gabriela Mendoza, nº 2022227025



Cenário de rede no GNS3

De modo a suportar as comunicações UDP e TCP, é necessário montar uma rede com 3 *routers*, configurados para suportar operações de encaminhamento e NAT (*Network Address Translation*).

As redes A, B e C são subdivisões da rede 193.137.100.0/23 (512 endereços totais), tendo em consideração que a rede C terá que ter mais endereços disponíveis que as restantes, que a rede A fica com os endereços mais baixos na gama e que não se podem desperdiçar endereços nesta divisão.

Já a rede D, utiliza a rede privada 10.5.2.0/26 (64 endereços totais) e o router R3 implementa SNAT (*Source NAT*) para modificar o endereço de origem dos pacotes de uma rede interna (*interface f0/1*) antes de serem enviados para uma rede externa (*interface f0/0*).

Para efeitos de teste das funcionalidades implementadas, consideram-se o Cliente1 e o Cliente2 como sendo dos tipos “Aluno/Professor” e o Cliente3 como sendo do tipo “Administrador”.

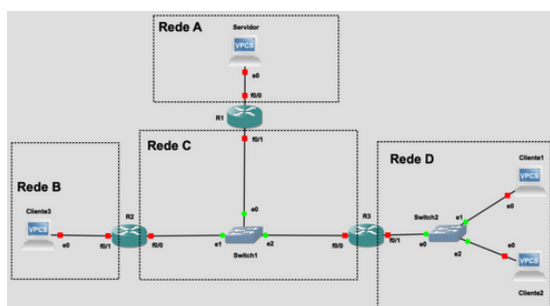


Fig. 1 - Rede de comunicação de suporte à aplicação

Para configurar os *containers*...

Por convenção, utilizamos os primeiros endereços disponíveis nas redes.

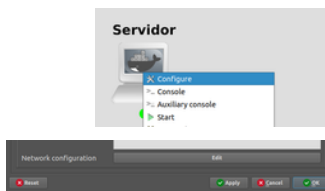


Fig. 2/3 - Network configuration nos containers

```
# Static config for eth0
auto eth0
iface eth0 inet static
    address 10.5.2.1
    netmask 255.255.255.192
    gateway 10.5.2.62
up echo nameserver 10.5.2.62 > /etc/resolv.conf
```

Fig. 4 - Alterações para o “Cliente1”

```
# Static config for eth0
auto eth0
iface eth0 inet static
    address 10.5.2.2
    netmask 255.255.255.192
    gateway 10.5.2.62
up echo nameserver 10.5.2.62 > /etc/resolv.conf
```

Fig. 5 - Alterações para o “Cliente2”

```
# Static config for eth0
auto eth0
iface eth0 inet static
    address 193.137.100.129
    netmask 255.255.255.128
    gateway 193.137.100.254
up echo nameserver 193.137.100.254 > /etc/resolv.conf
```

Fig. 6 - Alterações para o “Cliente3”

```
# Static config for eth0
auto eth0
iface eth0 inet static
    address 193.137.100.1
    netmask 255.255.255.128
    gateway 193.137.100.126
up echo nameserver 193.137.100.126 > /etc/resolv.conf
```

Fig. 7 - Alterações para o “Servidor”

Para configurar os *routers*...

Por convenção, utilizamos os últimos endereços disponíveis nas redes.

```
R1:
config terminal
ip multicast-routing
interface FastEthernet0/0
ip address 193.137.100.126 255.255.255.128
no shutdown
ip pim sparse-dense-mode
ip route 0.0.0.0 0.0.0.0 193.137.101.252

interface FastEthernet0/1
ip address 193.137.101.254 255.255.255.0
no shutdown
ip pim sparse-dense-mode
end

copy running-config startup-config
```

Fig. 8 - Comandos de configuração para “R1”

```
R2:
config terminal
ip multicast-routing
interface FastEthernet0/1
ip address 193.137.100.254 255.255.255.128
no shutdown
ip pim sparse-dense-mode
ip route 0.0.0.0 0.0.0.0 193.137.101.252

interface FastEthernet0/0
ip address 193.137.101.253 255.255.255.0
no shutdown
ip pim sparse-dense-mode
end

copy running-config startup-config
```

Fig. 9 - Comandos de configuração para “R2”

```

R3 (SNAT):
config terminal
ip multicast-routing
access-list 30 permit 10.5.2.0 0.0.0.63
ip nat inside source list 30 interface FastEthernet0/0 overload

interface FastEthernet0/0
ip address 193.137.101.252 255.255.255.0
no shutdown
ip pim sparse-dense-mode
ip nat outside
ip route 193.137.100.128 255.255.255.128 193.137.101.253
interface FastEthernet0/1
ip address 10.5.2.62 255.255.255.192
no shutdown
ip pim sparse-dense-mode
ip nat inside
end

copy running-config startup-config

```

Fig. 10 - Comandos de configuração para “R3”

Feitas todas as configurações necessárias, colocam-se os códigos desenvolvidos em linguagem C dentro da pasta “/home” de cada um dos *containers*, compilando e executando os mesmos. Assim sendo, dentro das pastas do “Cliente1” e “Cliente2” estarão os ficheiros “class_client.c” e “class_client”, na do “Cliente3” estarão “admin_client.c” e “admin_client” e na do “Servidor” estarão “class_server.c”, “config.txt” e “class_server.”

Servidor e clientes TCP/UDP

Servidor (“class_server”):

Comando >> ./class_server 6000 5000 config.txt

Cliente TCP (“class_client”):

Comando >> ./class_client 193.137.100.1 6000

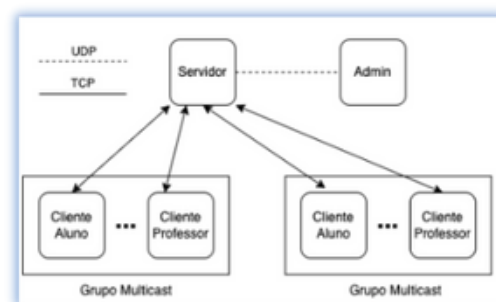


Fig. 11 - Arquitetura da aplicação (componentes)

Cliente UDP (“admin_client”):

Comando >> ./admin_client 193.137.100.1 5000

Para o desenvolvimento da aplicação, decidimos implementar também uma aplicação para o cliente UDP, utilizada apenas para efeitos de administração do sistema.

Como portos de escuta, definimos “PORTO_TURMAS” como 6000 e “PORTO_CONFIG” como 5000.

```

----- AUTENTICAÇÃO -----
Username: joao
Password: 123
Autenticação falhou. Impossível autenticar administradores por TCP!

----- AUTENTICAÇÃO -----
Username: ines
Password: ifar10
Autenticação bem sucedida!

----- OPÇÕES DISPONÍVEIS PARA PROFESSOR -----
[ 1 ] LIST_CLASSES
[ 2 ] CREATE_CLASS {name} {size}
[ 3 ] SEND {name} {text that server will send to subscribers}
[ 4 ] EXIT

```

Fig. 12 - Autenticação e comandos para um professor

```

----- AUTENTICAÇÃO -----
Username: henrique
Password: hcm04
Autenticação bem sucedida!

----- OPÇÕES DISPONÍVEIS PARA ALUNO -----
[ 1 ] LIST_CLASSES
[ 2 ] LIST_SUBSCRIBED
[ 3 ] SUBSCRIBE_CLASS {name}
[ 4 ] EXIT
Escolha uma opção:

```

Fig. 13 - Autenticação e comandos para um aluno

```

Username: lara
Password: 2203
Autenticação falhou. Impossível autenticar clientes aluno/professor por UDP!

Username: joao
Password: 123
Autenticação bem sucedida!

```

Fig. 14 - Autenticação e comandos para um administrador

Inicialmente, são pedidas as credenciais de *login* ao utilizador. Após a sua correta autenticação, são listadas as opções permitidas, consoante o tipo do mesmo.

Ao escolher um número como opção, serão pedidos os dados necessários ao comando, se for esse o caso. Após ter toda a informação necessária, o comando formatado é enviado ao servidor, que o irá tratar em conformidade.

Funcionalidades implementadas

Para o administrador (UDP)...

- (1) Adicionar um utilizador: envia um pedido com os parâmetros necessários para o servidor, que o processa e adiciona o novo utilizador ao ficheiro de configuração “config.txt” e à lista de utilizadores.
- (2) Remover um utilizador: envia um pedido com o nome de utilizador a remover e, caso este exista, é removido do ficheiro de configuração e da lista de utilizadores.
- (3) Listar utilizadores: pede ao servidor para listar todos os utilizadores registados no sistema.
- (4) Encerrar sessão: termina o cliente.
- (5) Encerrar servidor: termina o cliente e o servidor.

```
----- OPÇÕES DISPONÍVEIS PARA ADMINISTRADOR -----
[ 1 ] ADD_USER {username} {password} {administrador/aluno/professor}
[ 2 ] DEL {username}
[ 3 ] LIST
[ 4 ] EXIT
[ 5 ] QUIT_SERVER

Escolha uma opção: 1
Username a adicionar: antonio
Password a adicionar: pass
Tipo a adicionar: professor
ACCEPTED
```

```
----- OPÇÕES DISPONÍVEIS PARA ADMINISTRADOR -----
[ 1 ] ADD_USER {username} {password} {administrador/aluno/professor}
[ 2 ] DEL {username}
[ 3 ] LIST
[ 4 ] EXIT
[ 5 ] QUIT_SERVER

Escolha uma opção: 2
Username a remover: antonio
REMOVED
```

```
----- OPÇÕES DISPONÍVEIS PARA ADMINISTRADOR -----
[ 1 ] ADD_USER {username} {password} {administrador/aluno/professor}
[ 2 ] DEL {username}
[ 3 ] LIST
[ 4 ] EXIT
[ 5 ] QUIT_SERVER

Escolha uma opção: 3
----- REGISTERED USERS -----
USER jorge TYPE administrador
USER joana TYPE professor
USER henrique TYPE aluno
USER joao TYPE administrador
USER ines TYPE professor
USER lara TYPE aluno
USER luana TYPE aluno
```

Fig. 15 - Demonstração das funcionalidades essenciais de um administrador

Para o professor (TCP)...

- (1) Listar turmas: pede ao servidor para listar todas as turmas registadas no sistema.
- (2) Criar turma: caso a turma ainda não exista, vai ser criada uma nova e ser devolvido ao cliente o endereço multicast da mesma.
- (3) Enviar conteúdo: faz uma requisição ao servidor para enviar, através de grupos multicast, uma mensagem a todos os clientes (alunos) que estiverem inscritos na respetiva turma. Nota: a mensagem consegue ser recebida pelos alunos inscritos, através da utilização de uma thread que fica à escuta de mensagens. No entanto, não conseguimos fazer o print da mesma no terminal (por resolver).
- (4) Encerrar sessão: termina o cliente.

```
----- OPÇÕES DISPONÍVEIS PARA PROFESSOR -----
[ 1 ] LIST CLASSES
[ 2 ] CREATE_CLASS {name} {size}
[ 3 ] SEND {name} {text that server will send to subscribers}
[ 4 ] EXIT

Escolha uma opção: 1
----- CLASSES AVAILABLE -----
CLASS <rc>
CLASS <so>
```

```
----- OPÇÕES DISPONÍVEIS PARA PROFESSOR -----
[ 1 ] LIST CLASSES
[ 2 ] CREATE_CLASS {name} {size}
[ 3 ] SEND {name} {text that server will send to subscribers}
[ 4 ] EXIT

Escolha uma opção: 2
Nome da turma: so
Capacidade máxima (1-50): 10
OK <224.0.0.2>
```

```
----- OPÇÕES DISPONÍVEIS PARA PROFESSOR -----
[ 1 ] LIST CLASSES
[ 2 ] CREATE_CLASS {name} {size}
[ 3 ] SEND {name} {text that server will send to subscribers}
[ 4 ] EXIT

Escolha uma opção: 3
Nome da turma: rc
Texto a enviar: testing
SENT
```

Fig. 16 - Demonstração das funcionalidades essenciais de um professor

Para o aluno (TCP)...

- (1) Listar turmas: pede ao servidor para listar todas as turmas registadas no sistema.
- (2) Listar turmas subscritas: pede ao servidor para listar todas as turmas na qual o utilizador se encontra inscrito.
- (3) Subscrever turma: caso ainda não esteja subscrito e caso ainda hajam vagas disponíveis na turma, é feita a subscrição com o nome de utilizador atual e é devolvido o endereço multicast associado.
- (4) Encerrar sessão: termina o cliente.

```
----- OPÇÕES DISPONÍVEIS PARA ALUNO -----
[ 1 ] LIST CLASSES
[ 2 ] LIST_SUBSCRIBED
[ 3 ] SUBSCRIBE_CLASS {name}
[ 4 ] EXIT

Escolha uma opção: 2
----- CLASSES SUBSCRIBED BY henrique -----
CLASS <so/224.0.0.2>
```

```
----- OPÇÕES DISPONÍVEIS PARA ALUNO -----
[ 1 ] LIST CLASSES
[ 2 ] LIST_SUBSCRIBED
[ 3 ] SUBSCRIBE_CLASS {name}
[ 4 ] EXIT

Escolha uma opção: 3
Nome da turma: so
ACCEPTED <224.0.0.2>
```

Fig. 17 - Demonstração das funcionalidades essenciais de um alunor