

UNIVERSIDADE DE COIMBRA

ENGENHARIA INFORMÁTICA

REDES DE COMUNICAÇÃO

RELATÓRIO - Serviço de Notícias

Maio 23/24

Diogo Costa - 2022213506

Tomás Caçoete - 2022214188



UNIVERSIDADE D
COIMBRA

Simulação com um cenário no GNS3

No desenvolvimento deste projeto foi utilizado um cenário de GNS3 para simular a utilização do programa através de diferentes máquinas. Este sistema conta com três routers que fazem a ligação entre as quatro redes e utiliza ainda quatro containers de Docker para simular os diferentes tipos de utilizadores onde é possível executar o programa cujas configurações são as seguintes.

Servidor:

```
auto eth0
iface eth0 inet static
    address 193.137.100.1
    netmask 255.255.255.128
    gateway 193.137.100.126
```

Cliente Admin:

```
auto eth0
iface eth0 inet static
    address 193.137.100.129
    netmask 255.255.255.128
    gateway 193.137.100.254
```

Cliente 1:

```
auto eth0
iface eth0 inet static
    address 10.5.2.1
    netmask 255.255.255.192
    gateway 10.5.2.62
```

Cliente 2:

```
auto eth0
iface eth0 inet static
    address 10.5.2.2
    netmask 255.255.255.192
    gateway 10.5.2.62
```

Router 1:

```
config terminal
ip multicast-routing
interface f0/0
    ip pim spartse-dense-mode
    ip address 193.137.100.126 255.255.255.128
    no shutdown
exit
interface f0/1
    ip pim spartse-dense-mode
    ip address 193.137.101.254 255.255.255.0
    no shutdown
exit
ip route 10.5.2.0 255.255.255.192 193.137.101.252
ip route 193.137.100.128 255.255.255.128 193.137.101.253
exit
copy running-config startup-config
```

Router 2:

```
config terminal
ip multicast-routing
interface f0/0
    ip pim spartse-dense-mode
    ip address 193.137.101.253 255.255.255.0
    no shutdown
exit
interface f0/1
    ip pim spartse-dense-mode
    ip address 193.137.100.254 255.255.255.128
    no shutdown
exit
ip route 10.5.2.0 255.255.255.192 193.137.101.252
ip route 193.137.100.0 255.255.255.128 193.137.101.254
exit
copy running-config startup-config
```

Router 3:

```
config terminal
ip multicast-routing
interface f0/0
    ip pim spartse-dense-mode
    ip address 193.137.101.252 255.255.255.0
    no shutdown
exit
interface f0/1
    ip pim spartse-dense-mode
    ip address 10.5.2.62 255.255.255.192
    no shutdown
exit
ip route 193.137.100.0 255.255.255.128 193.137.101.254
ip route 193.137.100.128 255.255.255.128 193.137.101.253
exit
access-list 1 permit 10.5.2.0 0.0.0.63
ip nat inside source list 1 interface FastEthernet0/1 overload
copy running-config startup-config
```

Funcionamento do código

O sistema funciona através de três executáveis. Primeiramente temos o servidor que possui duas threads principais. Uma das threads lida com os clientes do protocolo TCP (alunos e professores) e a outra lida com os do protocolo UDP (administradores). Temos também o cliente admin que utiliza o protocolo UDP para comunicar com o servidor e o cliente da turma que é utilizado pelos alunos e professores para se ligarem ao servidor através do protocolo TCP.

Falando mais em detalhe da thread do servidor que lida com as comunicações TCP fizemos também com que este pudesse lidar com vários clientes TCP em simultâneo ao criar uma nova thread para lidar com cada conexão nova. Já a do UDP, através da própria natureza do protocolo consegue fazer isso por si só. As threads que lidam com as conexões possuem execuções bastante semelhantes após o sucesso de uma nova conexão. O servidor lê uma mensagem e faz o seu processamento para validar a sua sintaxe e permissões. Caso o comando não seja validado é enviada uma mensagem descritiva do erro ocorrido, caso contrário o comando é executado. É de frisar que numa fase inicial o servidor tentará validar o login do cliente não o deixando executar qualquer outra funcionalidade enquanto este não for sucedido.

O servidor possui também três listas ligadas para servir a sua funcionalidade. Uma delas inclui todos os utilizadores e é utilizada para adicionar novos, remover existentes e fazer a verificação do login. Outra serve para manter em conta todas as ligações atuais do servidor para lidar com um encerramento controlado do servidor onde todas as sockets são fechadas e todas as threads canceladas. A última é uma lista com todas as turmas onde temos as informações do multicast de cada turma bem como um registo dos alunos que nela estão inscritos e algumas informações extra. Esta auxilia-nos a gerir a quantidade de alunos nas turmas e a enviar as mensagens multicast.

O cliente TCP possui também a capacidade de receber mensagens multicast das turmas a que esteja subscrito. Esta funcionalidade está implementada através da criação de uma nova thread que fica à escuta das mensagens multicast vindas da turma que o aluno se inscreve.

Processo de Execução

Para executar o código é necessário executar o servidor que irá ficar à escuta de novas conexões de clientes. Quando o servidor estiver a correr então pode-se executar os clientes e fazer pedidos para o servidor.