



DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

TRABALHO PRÁTICO

Serviço de notícias

Ano Letivo de 2022/2023

1 Objetivos

O objetivo deste trabalho prático é implementar um sistema de difusão de notícias, recorrendo a diversas técnicas de comunicação e com recurso aos protocolos da pilha TCP/IP. Em particular, iremos fazer uso dos protocolos UDP e TCP, bem como das comunicações IP *multicast*. O trabalho compreenderá duas fases, avaliadas separadamente.

2 Cenário de comunicações

A Figura 1 ilustra a Rede de comunicação a configurar, para suportar a aplicação a desenvolver no trabalho. Para suportar as comunicações UDP e TCP, a aplicação fará uso de uma rede com 3 *routers*, que deverão ser configurados para suportar as necessárias operações de encaminhamento e NAT (*Network Address Translation*), tal como descrito a seguir.

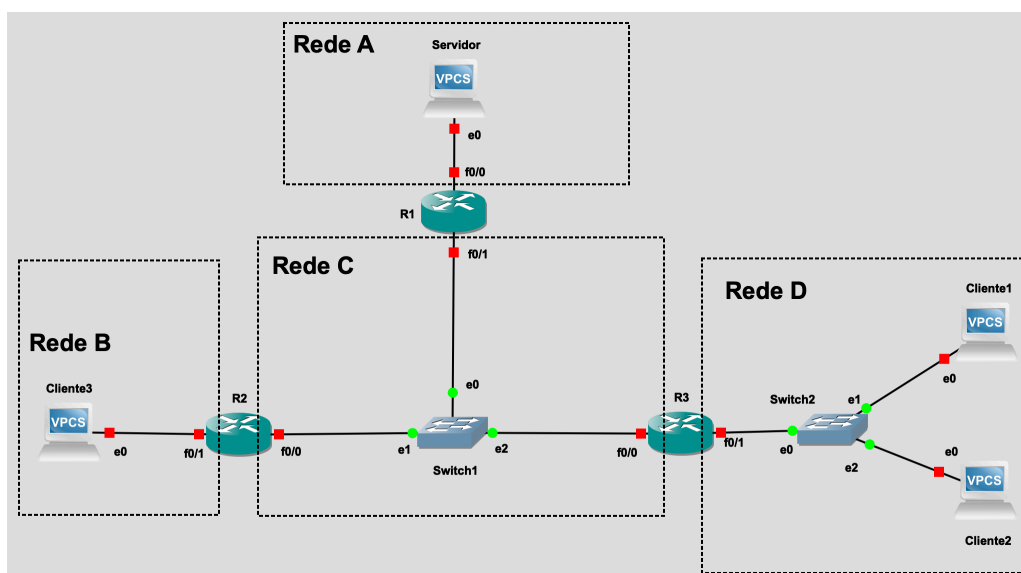


Fig. 1 - Rede de comunicação de suporte à aplicação.

O trabalho compreende duas fases de implementação e avaliação, estando os objetivos para cada fase descritos ponto 6. A seguir descrevem-se os detalhes a ter em conta na implementação do cenário da rede de comunicação de suporte à aplicação, com recurso ao GNS3.

3 Rede de comunicação de suporte à aplicação

Tal como a Figura 1 ilustra, o cenário de rede faz uso de 3 *routers* e 2 *switches*. Os equipamentos devem ser configurados de forma a garantir que os clientes (PCs) conseguem comunicar entre si, bem como com o servidor. **Todos os PCs (clientes e servidor) deverão utilizar Linux, recorrendo à imagem criada em *docker* e já disponível no GNS3 na VM fornecida aos alunos**¹.

Ao nível do endereçamento IPv4, deverá ter em conta os seguintes requisitos na configuração do cenário de comunicações:

- A rede a que pertence o Cliente1 e o Cliente2 (Rede D) usa endereços privados, e o router R3 implementa NAT. Use a rede **10.5.2.0/26** para endereçar todas as interfaces na Rede D.
- Use a rede **193.137.100.0/23** para endereçar as redes A, B e C. Na divisão que tiver de fazer, garanta que a Rede C fica com mais endereços disponíveis que as restantes e que a Rede A fica com os endereços mais baixos. Não deve desperdiçar endereços.
- Deverá atribuir a todos os equipamentos endereços IP apropriados, na gama da sub-rede convencionada.

Para além da comunicação, o router **R3** deverá suportar SNAT (*Source NAT*), em que a rede do Cliente1 e Cliente2 é a rede interna.

4 Funcionalidades da aplicação

4.1 Arquitetura da aplicação

Após a configuração da rede de comunicação de suporte à aplicação, será necessário construir a aplicação de notícias através do desenvolvimento das suas várias componentes (ver Fig. 2).

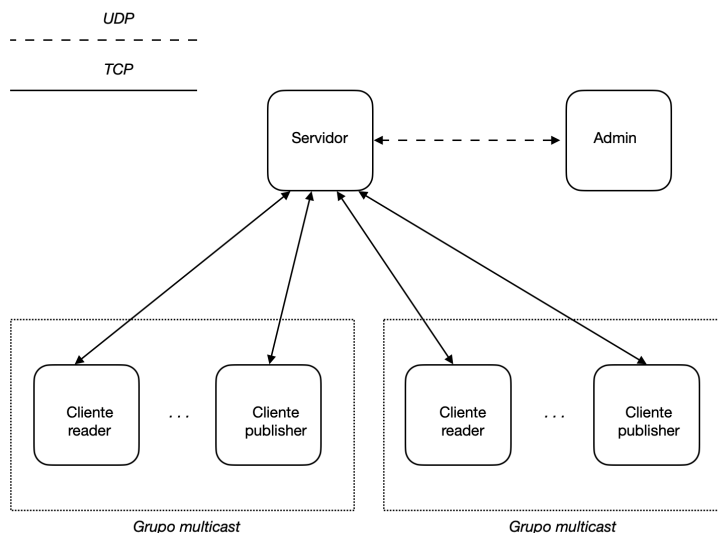


Fig. 2 – Arquitetura da aplicação (componentes)

Servidor:

- O programa servidor é responsável por autenticar e aceitar ligações de clientes, bem como servir os seus pedidos.

¹ Consultar tutorial de apoio.

- O servidor informa o cliente dos tópicos existentes, e gere a participação dos clientes nos vários tópicos.
- O servidor envia aos clientes os endereços *multicast* necessários à receção de notícias num determinado tópico (grupo).
- As comunicações com os vários clientes são efetuadas por TCP.
- O servidor suporta igualmente comunicações UDP com a consola de administração.

Consola de administrador:

- Comunica com o servidor por UDP.
- Permite gerir os utilizadores e configurar a aplicação.

Cliente:

- O programa cliente comunica com o servidor por TCP.
- Permite subscrever tópicos de notícias e receber notícias para cada um dos tópicos subscritos (através de comunicações *multicast*);
- Caso o utilizador tenha permissão para tal, permite enviar notícias dentro de um determinado tópico, aos vários clientes associados ao tópico.

Cada cliente deve inicialmente autenticar-se perante o servidor, recorrendo para tal ao seu *username* e *password*, sendo que estas credenciais devem ter sido previamente registadas no servidor pelo administrador do serviço. Juntamente com o *username* e *password*, o registo indica igualmente se o cliente tem permissão apenas para subscrever tópicos e receber notícias, ou se tem permissão para enviar (publicar) notícias. De seguida descrevem-se com detalhe as funcionalidades de cada um dos componentes em maior detalhe.

4.2 Descrição das funcionalidades a suportar

4.2.1 Funcionalidades suportadas pelo servidor

O servidor será responsável pelas seguintes funcionalidades:

- Gerir tópicos de notícias:
 - Informar os clientes dos vários tópicos existentes, bem como gerir os clientes associados a cada um dos tópicos.
 - Autenticar os clientes, bem como gerir as suas permissões (que clientes podem criar tópicos e publicar notícias, ou apenas receber notícias para os tópicos subscritos).
 - Informar os clientes que subscrevem um determinado tópico do endereço *multicast* a usar para receber notícias enviadas para esse tópico.
- Gestão de utilizadores:
 - Gerir utilizadores identificados pelo seu login, password e permissões. Esta informação deverá ficar armazenada em ficheiro.
 - Permitir o acesso do administrador através da consola de administração.
- Através de uma consola de administração disponível no PORTO_CONFIG do servidor, um utilizador com privilégios de administração na aplicação pode aceder ao servidor usando UDP e, através de uma *command line interface* (CLI), realizar as seguintes operações:
 - Adicionar um utilizador identificado pelo seu *username*, *password*, e tipo.
 - Armazenar informação sobre os utilizadores num ficheiro, para que fique sempre disponível.
 - A consola de administração deverá implementar os seguintes comandos, que o servidor deverá suportar:
 - Adicionar utilizador
 - `ADD_USER {username} {password}`
`{administrador/cliente/jornalista}`

Eliminar um utilizador

- `DEL {username}`

Lista utilizadores

- `LIST`

Sair da consola

- `QUIT`

Desligar servidor

- `QUIT_SERVER`

- Para aceder à CLI o administrador terá de se autenticar previamente, através de um utilizador com privilégios de administrador.
 - Nota: Para aceder à CLI não necessita de criar um programa cliente, poderá usar o comando `netcat`.
- Acesso de clientes previamente registados no serviço. Esse acesso é feito através do `PORTO_NOTICIAS`:
 - Validação do acesso dos clientes através de *username* e *password*, sendo que um utilizador deve ter sido previamente registado no serviço.
 - Após a autenticação, enviar ao cliente informação sobre os tópicos aos quais ele já está associado.
 - Receber pedidos de subscrição de tópicos por parte de clientes. Após receber um pedido de subscrição de um novo tópico, envia ao cliente que o pediu o endereço do grupo *multicast* a usar para receber as notícias desse tópico.
 - O servidor suporta acesso simultâneos de diversos clientes.

O programa servidor deverá ser executado na linha de comandos do Linux recorrendo à seguinte sintaxe:

```
news_server {PORTO_NOTICIAS} {PORTO_CONFIG} {ficheiro configuração}
```

Ao sair, o servidor descarta toda a informação, não é necessário escrever o estado no ficheiro.

4.2.2 Funcionalidades suportadas pelo Cliente

O cliente será responsável pelas seguintes funcionalidades:

- Deverá solicitar inicialmente ao utilizador o seu *username* e *password*, que deverá enviar ao servidor para se autenticar no serviço. O utilizador poderá ser de dois tipos, tal como se passa a descrever:

Cliente do tipo “leitor”:

- Com permissões para listar tópicos, subscrever um tópico e receber notícias nesse tópico, com os seguintes comandos:

Listar tópicos

- `LIST_TOPICS`

Subscrever tópicos

- `SUBSCRIBE_TOPIC <id do tópico>`

- Ao subscrever um tópico, o cliente recebe do servidor o endereço multicast através do qual poderá passar a receber notícias.
- Ao sair, o cliente deve ser removido de todos os grupos *multicast* que tenha subscrito.

Cliente do tipo “jornalista”:

- Para além das funcionalidades anteriores, o cliente permitirá ao utilizador criar um tópico, com recurso ao comando seguinte:

Criar tópico

- `CREATE_TOPIC <id do tópico> <título do tópico>`

- O cliente poderá também enviar para o grupo *multicast* correspondente ao tópico notícias para todos os clientes subscritos nesse tópico.

- `SEND_NEWS <id do tópico> <notícia>`

O programa cliente deverá ser executado na linha de comandos do Linux recorrendo à seguinte sintaxe:

```
news_client {endereço do servidor} {PORTO_NOTICIAS}
```

As comunicações entre o cliente e o servidor deverão utilizar TCP.

4.2.3 Ficheiro de configuração inicial

Ao arrancar o servidor lê os dados iniciais de um ficheiro de configuração. Na figura seguinte apresenta-se o formato e um exemplo desse ficheiro.

```
{username utilizador;password utilizador;tipo de utilizador}
{username utilizador;password utilizador;tipo de utilizador}
(..)
```

```
jorge;mtu67453;administrator
manuel;zpt567;leitor
joana;azq1212;2;jornalista
```

De notar que no exemplo anterior o primeiro utilizador tem permissões de administrador, o segundo apenas pode subscrever tópicos e receber notícias e o terceiro pode criar tópicos e publicar notícias, em linha com o descrito anteriormente.

5 Netcat

A aplicação **netcat** (**nc**) poderá ser bastante útil para efetuar testes ao cenário da rede de comunicação, bem como ao nível da própria aplicação. A seguir descreve-se de que forma esta aplicação pode ser usada para testar uma comunicação UDP ou TCP entre dois hosts. O **nc** pode ser usado diretamente na linha de comandos do Linux, portanto nos clientes e no servidor do cenário de comunicação. Para utilizar o **netcat** como **servidor** deverá utilizar a seguinte sintaxe:

```
nc -l {IP do servidor} {porto}      # Escuta num porto TCP
nc -u -l {IP do servidor} {porto}   # Escuta num porto UDP
```

Para utilizar o **netcat** como **cliente** deverá utilizar a seguinte sintaxe:

```
nc -v {IP do servidor} {porto}      # Ligação TCP ao servidor
nc -v -u {IP do servidor} {porto}    # Comunicação UDP com o servidor
```

Após o estabelecimento da comunicação entre o cliente e o servidor, para testar se a ligação está a funcionar corretamente basta escrever num dos terminais (cliente ou servidor), sendo que o texto introduzido deverá ser enviado e mostrado no outro terminal.

6 Entrega do trabalho

- Realização do trabalho: Trabalho em grupos de dois alunos (entregas individuais só em casos excecionais e mediante aprovação prévia pelos docentes)
- A entrega decorre em duas metas, com os seguintes objetivos e datas a ter em atenção:
 - **Meta 1:**
 - Implementação do cenário de rede com recurso ao GNS3, com todas as configurações necessárias ao suporte do encaminhamento entre as redes. Nesta meta não é necessário ter o NAT ou o *multicast* configurados (a rede D continuará privada).
 - Criação de uma primeira versão do servidor que permita gerir a consola de administração. Para testes será usado como cliente o `netcat`. Nota: Neste fase bastará apenas receber os comandos, não sendo necessário implementar o seu tratamento.

Entrega do relatório por *upload* no Infoestudante, com todos os comandos de configuração usados para cada dispositivo. As defesas serão efetuadas em *slots* a definir pelo docente:

- **Data limite de entrega: 9 de Abril de 2023.**
- **Defesas entre 10 e 14 de Abril 2023.**
- **Meta 2:** a entrega final consistirá do relatório final do trabalho, e dos ficheiros com o código fonte em C da aplicação desenvolvida:
 - **Data limite de entrega do relatório final por *upload* no Infoestudante: dia 14 de Maio de 2023.**
 - **Defesas na semana de 15 de Maio 2023.**

Notas importantes:

- Na realização do trabalho deverá recorrer à linguagem de programação C.
- O relatório final deve ser sucinto (no máximo 4 páginas A4), no formato PDF (não serão aceites outros formatos). No relatório deve explicar as opções tomadas na construção da solução e o modo de funcionamento.
- Crie um arquivo no formato ZIP (não serão aceites outros formatos) com todos os ficheiros do trabalho.
 - Inclua todos os ficheiros fonte e de configuração necessários.
 - Não inclua quaisquer ficheiros não necessários para a compilação ou execução do programa (ex. diretórios ou ficheiros de sistemas de controlo de versões)
 - Não serão admitidas entregas por e-mail.
- As defesas (meta1 e final) do trabalho são obrigatórias para todos os elementos do grupo.
- Todos os trabalhos serão escrutinados para deteção de cópias de código.