



DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

TRABALHO PRÁTICO

Turmas online

Ano Letivo de 2023/2024

1 Objetivos

O objetivo deste trabalho prático é implementar um sistema de turmas online para difusão de conteúdos, 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 Fig. 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.

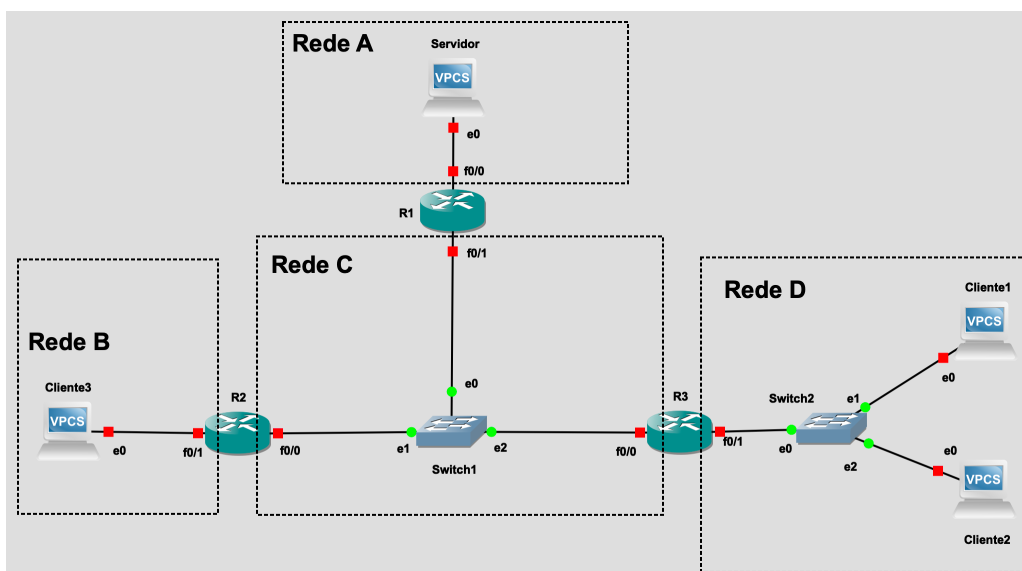


Fig. 1 - Rede de comunicação de suporte à aplicação.

O trabalho compreende duas fases de implementação e avaliação, estando os objetivos para cada fase descritos ponto 7. 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 na gama. Nesta divisão 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 gestão e difusão dos conteúdos, através do desenvolvimento das suas várias componentes (ver Fig. 2).

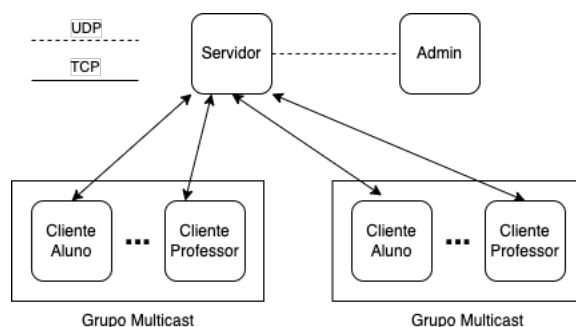


Fig. 2 – Arquitetura da aplicação (componentes)

Servidor:

- O programa servidor é responsável por autenticar e aceitar ligações de clientes, bem como servir os seus pedidos.
- O servidor informa os clientes das turmas existentes, e gere a participação dos clientes (professores ou alunos) nas várias turmas.
- As comunicações com os vários clientes são efetuadas por TCP. As comunicações entre clientes pertencentes à mesma turma são efetuadas via multicast. Para este efeito, o servidor envia aos clientes (professores ou alunos) os endereços *multicast* necessários à receção de conteúdos numa determinada turma (grupo).
- O servidor suporta igualmente comunicações UDP com a consola de administração.

¹ Consultar tutorial de apoio.

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.
- O funcionamento deste será diferente após autenticação, dependendo se esta foi realizada por um cliente/aluno ou um cliente/professor.
- Aos alunos será permitida a inscrição numa turma e a consequente receção de conteúdos, para cada uma das turmas em que esteja inscrito (através de comunicações *multicast*).
- Caso o utilizador seja um professor, este poderá criar turmas (indicando nome e número máximo de alunos) ou enviar conteúdos (frases de texto) para as turmas já criadas, sendo esses conteúdos recebidos só pelos alunos nelas inscritos.

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 é um **aluno**, **professor** ou **administrador**. De seguida descrevem-se com detalhe as funcionalidades de cada um dos componentes.

4.2 Descrição das funcionalidades a suportar

4.2.1 Funcionalidades suportadas pelo servidor

O servidor será responsável pelas seguintes funcionalidades:

- Gerir turmas:
 - Informar os clientes das várias turmas existentes, bem como gerir os clientes inscritos em cada turma.
 - Garantir que a capacidade das turmas não é excedida.
 - Autenticar os clientes, bem como gerir as suas permissões (que clientes podem criar turmas e publicar conteúdos, ou apenas receber os conteúdos nas turmas em que estão inscritos).
 - Informar os clientes que se inscrevem numa determinada turma qual o respetivo endereço *multicast*, a usar para receber conteúdos enviadas para essa turma.
- Gestão de utilizadores:
 - Gerir utilizadores identificados pelo seu login, password e permissões. Esta informação deverá ficar armazenada em ficheiro de texto.
 - 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 de utilizador.
 - 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/aluno/professor}`
 - Eliminar um utilizador
 - `DEL {username}`
 - Lista utilizadores
 - `LIST`

- Desligar servidor, que deverá fechar a aplicação “class_server”
 - `QUIT_SERVER`
 - Para aceder à CLI o administrador terá de se autenticar previamente, através de um utilizador com privilégios de administrador, através do seguinte comando
 - `LOGIN {username} {password}`
 - Nota: Para aceder à CLI de administração não necessita de criar um programa cliente, poderá usar o comando `netcat`.
 - Opção alternativa: O acesso à CLI administração poderá, em alternativa, ser realizada por uma aplicação cliente UDP a desenvolver. Esta situação será valorizada.
- O acesso de clientes previamente registados no serviço é feito através do `PORTO_TURMAS`, recorrendo a um conjunto de mensagens em formato texto:
 - Validação do acesso dos clientes através de *username* e *password*, sendo que um utilizador deve ter sido previamente registado no serviço:
 - `LOGIN {username} {password}`
 - Respondendo:
 - `OK` ou `REJECTED`
 - Após a autenticação, permitir ao cliente listar as turmas disponíveis:
 - `LIST_CLASSES`
 - Respondendo:
 - `CLASS {name1}, {name2}, ...`
 - Obter informação sobre as turmas nas quais ele já está inscrito:
 - `LIST_SUBSCRIBED`
 - Respondendo:
 - `CLASS {name1/multicast1}, {name2/multicast2}, ...`
 - Receber pedidos de inscrição em turmas por parte dos alunos, de notar que deverá ser verificada a capacidade da turma.
 - `SUBSCRIBE_CLASS <name>`
 - Resposta:
 - `ACCEPTED <multicast>`
 - ou
 - `REJECTED`
 - Aos professores será ainda permitido criar turmas, sendo internamente gerado e armazenado um endereço *multicast*, para o envio/receção posterior de conteúdos, este será enviado ao professor após a criação da turma:
 - `CREATE_CLASS {name} {size}`
 - Resposta:
 - `OK <multicast>`
 - Aos professores será ainda permitido o envio de conteúdos, recorrendo ao seguinte comando:
 - `SEND {name} {text that server will send to subscribers}`
- O servidor suporta acesso simultâneos de diversos clientes.
- A interface da aplicação cliente será livremente escolhida por cada grupo, sendo que as mensagens anteriormente descritas correspondem ao protocolo de comunicação a implementar entre os clientes e o servidor.

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

```
class_server {PORTO_TURMAS} {PORTO_CONFIG} {ficheiro configuração}
```

Ao sair, o servidor descarta toda a informação, não é necessário escrever o estado no ficheiro, com exceção do ficheiro de utilizadores.

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.
 - `LOGIN {username} {password}`
- O utilizador poderá ser de dois tipos, tal como se passa a descrever:

Cliente do tipo “aluno”:

- Com permissões para listar turmas, inscrever-se numa turma e receber conteúdos relativos às turmas em que esteja inscrito, com os seguintes comandos:

 Listar turmas

- `LIST_CLASSES`

 Listar turmas subscritas

- `LIST_SUBSCRIBED`

 Subscrever tópicos

- `SUBSCRIBE_CLASS <nome>`

- Ao subscrever uma turma, o cliente recebe do servidor a confirmação ou rejeição da inscrição (ver descrição servidor), em caso positivo receberá também o endereço *multicast* através do qual poderá passar a receber conteúdos, devendo subscrever esse endereço e imprimir no ecrã todos os textos recebidos.
- Ao sair, o cliente deve ser removido de todos os grupos *multicast* que tenha subscrito.

Cliente do tipo “professor”:

- Para além das funcionalidades anteriores, o professor poderá criar turmas, com recurso ao comando seguinte:

 Criar turma

- `CREATE_CLASS <nome> <size>`

- O cliente poderá também enviar para o grupo *multicast*, correspondente a uma turma, textos a difundir para todos os clientes inscritos nessa turma.

- `SEND {name} {text that server will send to subscribers}`

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

```
class_client {endereço do servidor} {PORTO_TURMAS}
```

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 texto 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;aluno
joana;azq1212;professor
```

De notar que no exemplo anterior o primeiro utilizador tem permissões de administrador, o segundo apenas pode inscrever nas turmas e receber conteúdos e o terceiro pode criar turmas e enviar conteúdos, 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 Configuração Multicast

A configuração dos routers CISCO para o reencaminhamento das mensagens *multicast* necessita da utilização dos seguintes comandos:

```
ip multicast-routing      # ativar multicast routing na configuração
                           global de cada router
ip pim sparse-dense-mode # em cada interface, ativar o PIM
```

7 Entrega do trabalho

- Realização do trabalho: Trabalho em grupos de dois alunos da mesma turma PL (entregas individuais só em casos excecionais e mediante aprovação prévia pelos docentes)
- Em casos excecionais (a confirmar com o respetivo docente da PL) poderão ser aceites trabalhos de grupos em que os dois alunos são de PL diferentes (mas sempre lecionadas pelo mesmo professor).
- 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.
 - Criação de uma primeira versão do servidor que permita gerir a consola de administração e receber os diversos comandos dos clientes. Para testes será usado como cliente o **netcat**. Nota: Nesta fase bastará apenas receber os comandos, não sendo necessário implementar o seu tratamento.

- Criação de uma primeira versão do cliente TCP que permita a realização e confirmação do login e enviar os comandos ao servidor para efeitos de teste.

Entrega de um arquivo contendo o código do servidor e do cliente e o relatório, por *upload* no Inforestudante. O relatório deverá descrever todos os comandos de configuração usados para cada dispositivo. As defesas serão efetuadas em *slots* a definir pelo docente:

- **Data-limite de entrega: 30 de Março de 2024.**
 - **Defesas entre 1 e 5 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, bem como o ficheiro do GNS3:
- **Data limite de entrega por *upload* no Inforestudante: dia 19 de Maio de 2024.**
 - **Defesas na semana de 20 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.