Universidade de Coimbra
Faculdade de Ciências e Tecnologia

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

TRABALHO PRÁTICO

Implementação do Protocolo "PowerUDP"

Ano Letivo de 2024/2025

1 Objetivos

O objetivo deste trabalho é desenvolver um protocolo de comunicação baseado em UDP, mas com de garantias adicionais para o dotar de alguma confiabilidade. Devem implementar-se os mecanismos necessários ao funcionamento pretendido para o novo protocolo, bem como uma aplicação de testes que dele faz uso. No presente documento, o protocolo é referido como "PowerUDP".

2 Cenário de comunicações

A Fig. 1 ilustra a Rede de comunicação a configurar, para suportar a aplicação e testar o funcionamento do novo protocolo. De notar que se trata do mesmo cenário da Ficha 2, dotado de algumas funcionalidades adicionais, que se detalham a seguir.

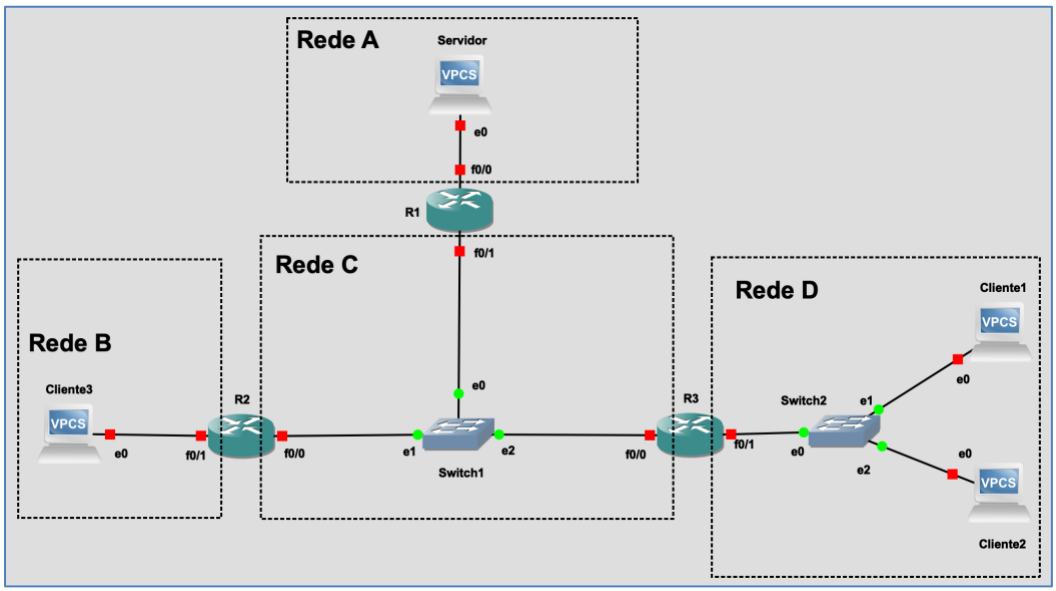

Fig. 1 - Rede de comunicação de suporte

Equipamentos

O cenário de rede faz uso de 3 *routers* e 2 *switches*. Os equipamentos devem ser configurados de forma a garantir que todos os *hosts* (clientes e servidor) conseguem comunicar entre si, bem como com o servidor, atendendo às regras de NAT referidas a seguir. Ao contrário da Ficha 2, todos os PCs (clientes e servidor) deverão utilizar Linux (em substituição dos VPCS), recorrendo a uma imagem criada em *docker*.

Endereçamento e NAT

As regras a ter em atenção no tocante ao endereçamento e NAT são as seguintes:

- A rede a que pertence o Cliente1 e o Cliente2 (Rede D) usa endereços privados, e o *router* R3 implementa SNAT, em que a rede do Cliente 1 e do Cliente 2 é a rede interna. Use a rede 10.5.2.0/26 para endereçar todos as interfaces na Rede D.

- O router R3 deve implementar igualmente DNAT, para permitir comunicações que tenham com destino o Cliente 1 e o Cliente 2.

- Utilize o espaço de endereçamento da rede 193.137.100.0/23 para obter 3 sub-redes, aproveitando todo o espaço endereçamento disponível na rede original. A primeira sub-rede (com endereços mais baixos na gama) deverá ser a que permite mais endereços, e utilizada na rede B. Das restantes 2 sub-redes, utilize a primeira (com endereços mais baixos na gama) na rede C.

- A rede a que pertence o Servidor também deve usar endereços privados, use a rede 10.20.0.128/25 para endereçar todas as interfaces na Rede A.

- Deverá atribuir a todos os equipamentos endereços IP apropriados, na gama da sub-rede convencionada.

- Adicione todas as rotas de encaminhamento que considere necessárias, sempre tendo em conta que não existe encaminhamento direto entre redes de domínio privado e redes de domínio público.

- Para além de DNAT, precisará igualmente de configurar SNAT no router R1.

3 Arquitetura

Visão geral

A Figura 2 apresenta uma visão geral da solução a implementar. Podemos ver que os vários clientes comunicam entre si com recurso ao PowerUDP, e com o servidor por TCP.

Por sua vez, o servidor é responsável por transmitir aos clientes uma nova configuração para o PowerUDP, e para isto faz uso de *multicast*.

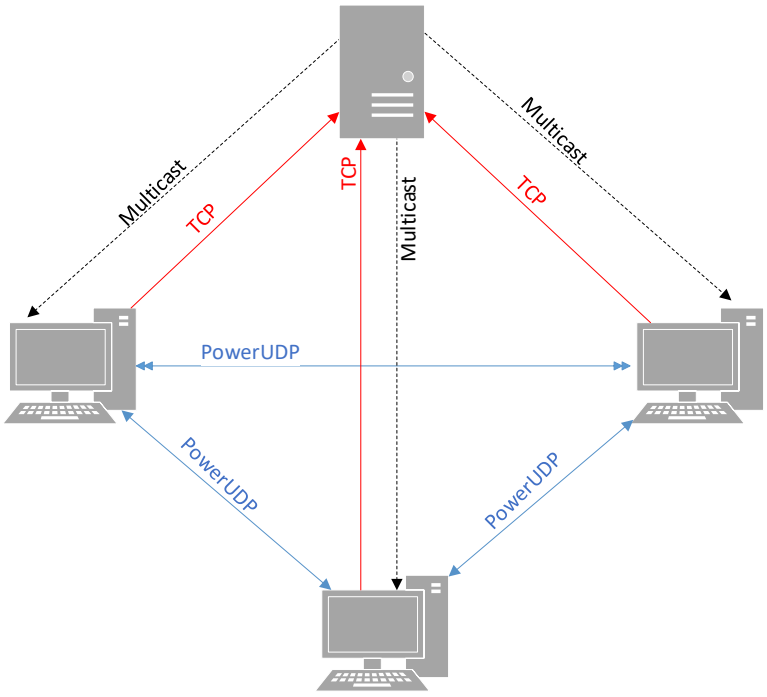

Figura 1 - Arquitetura

Comunicações

As comunicações ilustradas na arquitetura anterior permitem suportar as seguintes funcionalidades:

- Comunicações entre clientes usando o PowerUDP: comunicações *unicast* entre *hosts*, geridas pelo protocolo a implementar;

- Comunicações TCP entre clientes e o servidor: cada cliente regista-se no servidor, e envia pedidos para aplicação de uma nova configuração de funcionamento do protocolo;

- Comunicações *multicast*: são usadas pelo servidor para comunicar a todos os clientes a nova configuração de funcionamento do protocolo a aplicar;

Funcionalidades do cliente

Os clientes devem suportar o protocolo PowerUDP, cujas funcionalidades devem ser suportadas em linha com a API descrita mais à frente. O PowerUDP permitirá suportar as seguintes funcionalidades:

1. Registo da aplicação cliente no servidor, com recurso a chave pré-configurada;

2. Envio para o servidor de pedidos de alteração à configuração do protocolo ativa na rede;

3. Envio e receção de mensagens UDP para outros *hosts*, com garantias de confiabilidade de acordo com a configuração ativa;

Funcionalidades do servidor

O servidor é responsável por coordenar o registo de clientes que usam o PowerUDP, bem como disseminar pelos vários clientes a configuração ativa no momento. Em resumo, o servidor deve suportar as seguintes funcionalidades:

1. Registo de clientes: recebe de um cliente um pedido para registo, que deverá autenticar através de uma chave pré-definida (igual em todos os *hosts*)

2. Gestão da configuração ativa do PowerUDP na rede:
    a. O servidor pode receber de um cliente um pedido para alteração da configuração ativa no momento (assume-se que deverá ser sempre a mesma em todos os *hosts*);
    b. O servidor informa todos os clientes ativos (registados) na rede da nova configuração, recorrendo para o efeito a uma mensagem *multicast*;

4 O protocolo PowerUDP

Tal como referido anteriormente, pretende-se com o novo protocolo implementar alguns mecanismos que garantem confiabilidade na transmissão de mensagens UDP. Ou seja, o PowerUDP permite comunicar entre *hosts* utilizando UDP, mas ao mesmo tempo garante a retransmissão de mensagens (de acordo com os critérios detalhados a seguir) e também sua transmissão com ordenação. A figura seguinte ilustra a lógica considerada.

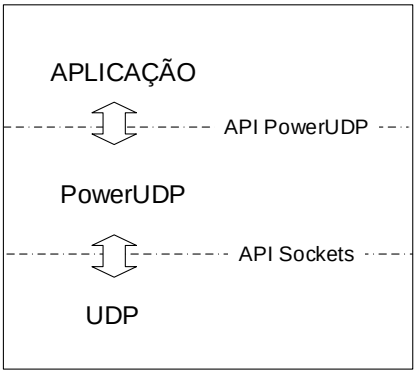

Figura 2 - Utilização da API do PowerUDP

A seguir descreve-se a estratégia considerada para cada um dos mecanismos a implementar.

Modelo de comunicação

Assume-se que o PowerUDP só envia uma mensagem quando a anterior tiver sido confirmada. Ou seja, a mensagem seguinte poderá ter de esperar pela retransmissão da mensagem anterior.

A receção de mensagens é confirmada pelo recetor (com uma mensagem ACK) ou negada a sua receção (com uma mensagem NAK), caso a mesma seja rejeitada.

Retransmissão de mensagens

O objetivo é permitir ao PowerUDP detetar que uma mensagem não foi recebida no destino, e retransmiti-la. Para isso, o protocolo deverá fazer uso dos seguintes mecanismos:

Temporizador: sempre que envia um pacote UDP, deverá ser ativado um temporizador. Caso não seja recebida uma confirmação (mensagem ACK), a mensagem deverá ser retransmitida, até um número máximo de retransmissões.

Exponential *backoff*: o tempo entre retransmissões (até um máximo de retransmissões pré-definido) deverá ser calculado de acordo com esta lógica.

A fórmula geral para calcular o tempo de espera (TTT) na tentativa *n* é:

$$T_n = T_{min} \times 2^n$$

Onde:

- T_{min} é o tempo de espera mínimo (base)
- n é o número de tentativas de transmissão falhadas

Ordenação de mensagens

Outra funcionalidade do PowerUDP é garantir a entrega de mensagens UDP no recetor pela ordem correta. Para tal, o emissor deverá adicionar num número de sequência a cada mensagem, e quando o recetor receber uma mensagem com o número de sequência repetido ou fora de ordem, deverá rejeitar a mesma, transmitindo uma mensagem NACK ao emissor.

Encapsulamento

Para implementação dos mecanismos no PowerUDP, deverá adicionar um cabeçalho com os campos que considerar necessários. Os dados da aplicação, conjuntamente com o cabeçalho, serão transmitidos no campo de dados da mensagem UDP enviada ao *host* de destino, tal como ilustra a figura seguinte.

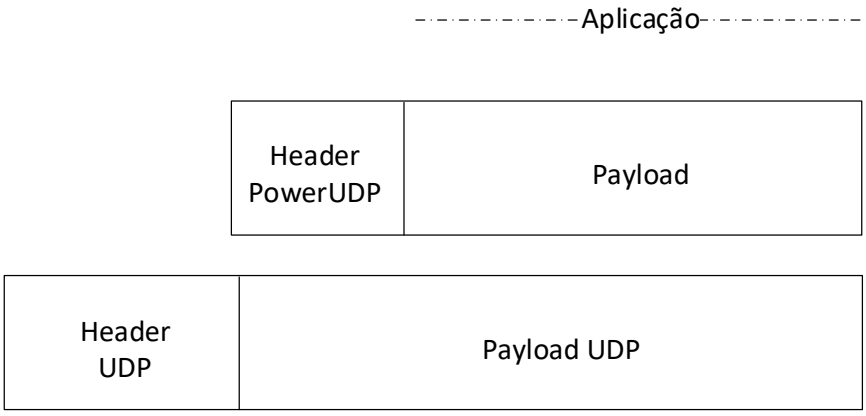

Figura 3 - Lógica de encapsulamento no PowerUDP

5 Estrutura de informação e API

Na sua implementação, deverá considerar a estrutura de informação e a especificação da API que a sua implementação do PowerUDP deverá implementar.

5.1 Estruturas de informação para mensagens

Mensagem de configuração do servidor enviada para os clientes (via *multicast*) ou de um cliente a solicitar uma alteração à configuração atual:

```
struct ConfigMessage {
   uint8_t enable_retransmission;     // 0 = Desativado, 1 = Ativado
   uint8_t enable_backoff;            // 0 = Desativado, 1 = Ativado
   uint8_t enable_sequence;           // 0 = Desativado, 1 = Ativado
   uint16_t base_timeout;             // Tempo base para timeouts (ms)
   uint8_t max_retries;               // Número máximo de retransmissões
};
```

Pedido de registo de um cliente, enviado para o servidor (via TCP)

```
struct RegisterMessage {
   char psk[64];  // Chave pré-definida para autenticação
};
```

5.2 API do Cliente

A implementação do protocolo PowerUDP deverá fornecer algumas funções que disponibilizam, à aplicação, um conjunto de funcionalidades, tal como a seguir se descreve:

Inicialização e configuração do protocolo no cliente

```
// Inicializa a stack de comunicação e regista-se no servidor
int init_protocol(const char *server_ip, int server_port, const char *psk);

// Termina a stack de comunicação e apaga o registo no servidor
void close_protocol();

// Solicita mudança na configuração do protocolo ao servidor
int request_protocol_config(int enable_retransmission, int enable_backoff, int enable_sequence, uint16_t base_timeout, uint8_t max_retries);
```

Envio e receção de mensagens

```
// Envia uma mensagem UDP
int send_message(const char *destination, const char *message, int len);

// Recebe uma mensagem UDP
int receive_message(char *buffer, int bufsize);
```

Estatísticas e injeção de erros

```
// Obtém estatísticas da última mensagem enviada
int get_last_message_stats(int *retransmissions, int *delivery_time);

// Simula a perda de pacotes para testar retransmissões
void inject_packet_loss(int probability);
```

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).

- Data limite de entrega por *upload* no Inforestudante: dia 16 de Maio de 2025.

- Defesas na semana de 19 de Maio 2025.

Notas importantes:

- Na realização do trabalho deverá recorrer à **linguagem de programação C**, utilizando sockets TCP/UDP.

- 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.

- 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 são obrigatórias para todos os elementos do grupo.

- Todos os trabalhos serão escrutinados para deteção de cópias de código.

Testes a considerar (relatório e defesa):

- Demonstrar a capacidade de mudar dinamicamente as configurações via *multicast*
- Apresentar testes práticos com diferentes configurações do protocolo
- Simular perda de pacotes e verificar o funcionamento das retransmissões
- Medir tempos de resposta com e sem lógica de retransmissão