

## **Trabalho Prático N°2 – Desenho e implementação de um sistema de partilha de conteúdos áudio**

Duração: 7 aulas (não consecutivas, ver calendário)

---

Neste trabalho pretende-se implementar um protótipo de um ambiente distribuído de troca de conteúdos áudio entre utilizadores registados num servidor existente em cada domínio de rede, com posterior interação direta entre os utilizadores que escolhem com quem irão trocar conteúdos com base em indicadores de desempenho da rede, nomeadamente a métrica One-way Delay (OWD). Sempre que um conteúdo não esteja disponível para partilha num domínio, a comunicação inter-domínio entre servidores e, posteriormente, entre os seus utilizadores num âmbito mais alargado, irá permitir o acesso a esse conteúdo existente noutra domínio.

O protocolo UDP, caracterizado por: (i) oferecer um serviço de transporte não fiável; (ii) não possuir mecanismos para controlo de fluxo e para controlo da congestão na rede; e (iii) disponibilizar um controlo de erros diminuto e opcional, é mais adequado para suportar tráfego aplicacional em que uma baixa latência é conveniente (e.g., aplicações com requisitos de tempo-real, como jogos interactivos, áudio e vídeo *streaming*). Contudo, no caso da transferência de um elevado volume de dados, é conveniente desenvolver uma camada protocolar sobre UDP que permita a ordenação e recuperação de dados perdidos.

Neste trabalho, pretende-se que: (i) a interação entre um servidor e seus utilizadores seja suportada por TCP; (ii) a interação direta entre utilizadores para troca de mensagens e conteúdos de áudio recorra ao protocolo UDP, com funcionalidades acrescidas; (iii) a interação entre servidores considerando o sistema distribuído inter-domínio recorra a comunicações TCP.

O enunciado abaixo apresenta as sugestões base para o desenvolvimento deste protótipo, podendo ser consideradas alterações, optimizações e novas funcionalidades desde que devidamente justificadas.

### **1. Objetivo Geral**

Planeamento e implementação de um serviço de distribuição de conteúdos áudio que otimize a transmissão considerando o desempenho da rede aquando da escolha do fornecedor do conteúdo e usando comunicações UDP com funcionalidades acrescidas (ordenação e retransmissão). As comunicações previstas no protótipo têm que ser implementadas recorrendo a programação em *sockets* TCP e UDP, não devendo ser utilizada nenhuma tecnologia baseada em *http* ou implementados interfaces baseados em serviços web.

### **2. Descrição**

Em cada rede local onde o serviço de partilha de conteúdos estará disponível, deverá ser instalado um servidor num endereço IP e porta fixos e pré-definidos, para que os clientes (programas que os utilizadores utilizam para partilhar ficheiros entre eles) possam registar-se no respetivo servidor. Toda a comunicação entre cliente e servidor e entre servidores é feita usando TCP (*unicast*). Já a comunicação entre clientes é feita usando uma versão otimizada do UDP (que forneça ordenação e retransmissão), também em *unicast*. Todas as comunicações aplicacionais do sistema devem utilizar o mesmo PDU base:

PDU base

Versão [1 byte] (por defeito, 1)
Segurança [1 byte] (por defeito, 0 - sem segurança)
Tipo [1 byte] (1 - REGISTER, 2 - CONSULT_REQUEST, ...)
Opções [4 bytes] (para otimização do UDP)
Dados

O funcionamento geral do serviço é a seguir descrito:

- \* Cada cliente, ao entrar no domínio onde o serviço é prestado regista-se no respetivo servidor através de um pedido **REGISTER** por unicast, informando o seu ID, endereço IP e porta TCP onde receberá consultas enviadas pelo servidor;
- \* O servidor da rede local mantém uma lista com todos os clientes ativos associados a ele e de outros servidores existentes em domínios diferentes;
- \* Um cliente interessado num conteúdo (ficheiro) específico, envia uma consulta ao seu servidor questionando quem o possui, através de uma mensagem do tipo **CONSULT\_REQUEST**;
- \* O servidor, por sua vez, envia um **CONSULT\_REQUEST** a todos os clientes atualmente ativos no seu domínio (exceto para o cliente que enviou a consulta) e aguarda por resposta durante um intervalo de tempo pré-definido. Caso nenhum dos clientes associados possua o respetivo ficheiro ou esgote o limite de tempo de resposta, o servidor enviará a mesma consulta para os outros servidores que possui na sua lista. Se ainda assim o ficheiro não for encontrado em nenhum cliente, o servidor responderá ao cliente com uma mensagem do tipo **CONSULT\_RESPONSE** informando a não existência do ficheiro (NOT\_FOUND);
- \* Cada cliente, ao receber a consulta do servidor responde com uma mensagem do tipo **CONSULT\_RESPONSE**, contendo o seu ID, endereço IP, e porta UDP pela qual aceitará comunicação com outros clientes, caso possua o respetivo ficheiro (FOUND), ou contendo a indicação de que não possui o ficheiro (NOT\_FOUND);
- \* O servidor então envia ao cliente que efetuou a consulta pelo ficheiro uma mensagem do tipo **CONSULT\_RESPONSE** (FOUND), contendo a lista de clientes (ID, IP e porta) que possuem o respetivo ficheiro;
- \* O cliente, por sua vez, envia uma mensagem do tipo **PROBE\_REQUEST** para todos os clientes que possuem o ficheiro de interesse, por forma a definir qual deles apresenta melhores condições para a transferência (ver detalhes do PROBE na Secção 3);

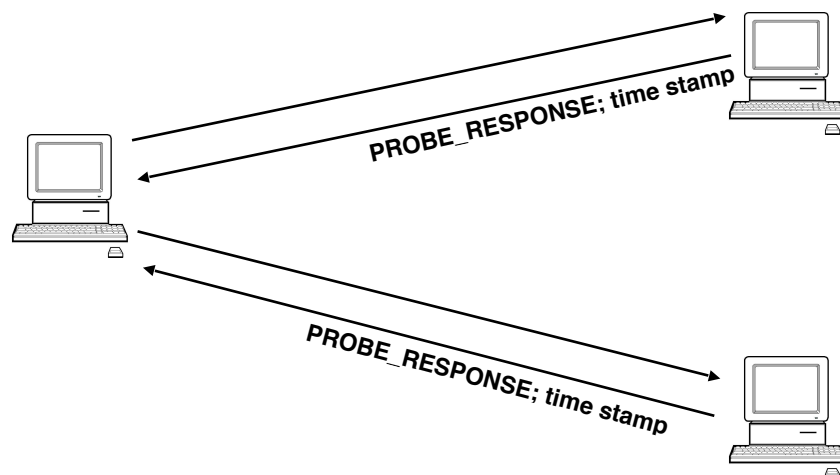
- \* Após definir qual cliente fornece as melhores condições, o cliente interessado no ficheiro envia uma mensagem do tipo **REQUEST** contendo o nome do ficheiro (ver semântica do pedido na Tabela 1);
- \* O cliente que recebe o pedido inicia o envio do ficheiro através de mensagens do tipo **DATA** seguindo as restrições e requisitos apresentados na Secção 4;
- \* A descoberta por novos servidores é feita através de um servidor central que mantém o registo de todos os servidores ativos. Tal registo é feito seguindo o mesmo processo para o registo de clientes (*i.e.*, através de um pedido **REGISTER**), o que implica usar um atributo que permita diferenciar clientes de servidores.

Tabela 1: Tipos e semântica de mensagens

Tipo de mensagem	Código	Descrição	Parâmetros - Dados
<b>REGISTER</b>	1	Registo de um cliente no servidor do domínio ou do servidor no serviço central	Tipo;ID;IP;Porta Tipo = in/out (para ingresso ou saída)
<b>CONSULT_REQUEST</b>	2	Consulta de um ficheiro de interesse	banda;musica.extensão
<b>CONSULT_RESPONSE</b>	3	Resposta sobre a existência ou não do ficheiro	Tipo;NumeroHosts;ID;IP;Porta Tipo = FOUND (1) / NOT_FOUND (0) NumeroHosts = número de clientes que possuem
<b>PROBE_REQUEST</b>	4	Mensagem de prova para medir as condições de transferência	
<b>PROBE_RESPONSE</b>	5	Mensagem de prova para medir as condições de transferência	Timestamp
<b>REQUEST</b>	6	Solicitação de transferência de ficheiro	banda;musica.extensão
<b>DATA</b>	7	Transmissão do ficheiro	*Ver requisitos e restrições na Secção 4

### 3. Mecanismo de PROBING

As mensagens do tipo **PROBE** serão usadas para decidir qual o cliente que apresenta a menor latência de transmissão (*One-way Delay* - OWD). Assim, como apresentado na Figura 1, o cliente envia uma mensagem do tipo **PROBE\_REQUEST** para todos os clientes que possuem o ficheiro. Cada um dos destinatários responderá com uma mensagem do tipo **PROBE\_RESPONSE** contendo o *timestamp* de envio da resposta. O cliente interessado no ficheiro usará este *timestamp* para calcular a latência aproximada para cada uma das possíveis fontes, decidindo para qual fará o **REQUEST** do ficheiro. A melhor condição de transferência é identificada para o cliente com o menor OWD, definido por:  $OWD = \text{timestamp (do PROBE\_RESPONSE)} - \text{timestamp (atual)}$ .



#### 4. Otimização do UDP

Uma vez que a transferência dos ficheiros áudio entre clientes será efetuada via UDP, a aplicação deverá fornecer mecanismos que incorporarem características relevantes a este tipo de serviço, nomeadamente tamanho máximo de pacote (48 kB), ordenação e retransmissão de pacotes perdidos. Para tal, deverá ser usado o campo “*Opções*” (descrito no PDU base), assim como implementados os mecanismos necessários de ordenação e recuperação na própria aplicação.

#### 5. Fases de Desenvolvimento

##### 1ª FASE

No arranque do trabalho será fornecido um conjunto de ficheiros áudio para testes do serviço a desenvolver.

Na primeira fase do trabalho deverão ser implementadas as versões base do servidor e cliente, envolvendo um ambiente para testes e demonstração com, pelo menos, um servidor e três clientes. Nesta fase deverão ser implementados os serviços de registo de clientes.

##### 2ª FASE

Nesta fase será implementado o sistema de descoberta de ficheiros áudio, incluindo:

- \* Consulta ao servidor local;
- \* Resposta sobre a existência ou não do ficheiro de interesse para o servidor e para o cliente interessado no ficheiro.

Esta fase será também dedicada à implementação do mecanismo de PROBING e da transmissão de dados entre os clientes, incluindo os mecanismos de otimização de transmissão.

##### 3ª FASE

Implementação do funcionamento inter-domínio, envolvendo o registo e interação entre servidores pertencentes a domínios distintos.

No final da 3ª Fase cada grupo fará uma demonstração do funcionamento global do serviço e de cada um dos seus componentes.

## 6. Relatório

O relatório deve ser escrito em formato de artigo com um máximo de dez páginas (não incluindo os anexos com exemplos da aplicação, listagens diversas e código fonte). Recomenda-se o uso do formato Lecture Notes in Computer Science (instruções para autores em <http://www.springer.com/computer/lncs?SGWID=0-164-6-793341-0>).

O relatório deve descrever o essencial do desenho e implementação, incluindo (mas não limitado):

- \* Introdução;
- \* Diferenças/adições na especificação do protocolo em relação às apresentadas no enunciado;
- \* Implementação (detalhes, parâmetros, bibliotecas de funções, etc);
- \* Testes e resultados;
- \* Conclusões e trabalho futuro.

Cada grupo deverá também fornecer o código fonte de toda a solução, devidamente comentado/documentado.

## 7. Calendarização

TP02 - Apresentação da ficha TP02 (semana 7.03.16)

1ª aula TP02 - Acompanhamento da implementação (semana 14.03.16)

2ª aula TP02 - Acompanhamento da implementação (semana 28.03.16)

3ª aula TP02 - Acompanhamento da implementação e Avaliação intermédia da 1ª Fase (semana 4.04.16)

4ª aula TP02 - Acompanhamento da implementação (semana 11.04.16)

5ª aula TP02 - Acompanhamento da implementação (semana 16.05.16)

6ª aula TP02 - Acompanhamento da implementação e Avaliação intermédia da 2ª Fase (semana 23.05.16)

7ª aula TP02 - Acompanhamento da implementação (semana 30.05.16)

Entrega – submissão electrónica do relatório (pdf) + projecto (código) na plataforma de *elearning* dia 5.06.16 até às 23:59

Avaliação final com demonstração, discussão e avaliação do trabalho (semana 6.06.16)