

Trabalho Prático Nº2 – Transferência rápida e fiável de múltiplos servidores em simultâneo

Duração: 8 aulas (ver planeamento das aulas práticas)

Motivação

A partilha de ficheiros é um dos serviços essenciais de uma rede. Permite que um programa cliente identifique o ficheiro e o descarregue de um servidor que o disponibiliza. O serviço tem como requisito base que a transferência seja fiável. Os dados devem ser descarregados e guardados sem erros. A réplica descarregada pelo cliente é uma cópia integral da existente no servidor, cuja integridade pode ser verificada. Os protocolos desenhados para a transferência de ficheiros procuram garantir os requisitos base, mas também assegurar um bom desempenho na transferência. Neste trabalho pretende-se desenhar um serviço avançado de transferência de ficheiros, numa rede *peer-to-peer* (P2P) com múltiplos servidores, que são também clientes do mesmo serviço. São por isso pares que dialogam entre si. Num dado instante, um ficheiro pode estar disponível em mais que um *peer*, e pode ser transferido de qualquer um deles ou mesmo de vários deles ao mesmo tempo, por partes, com o intuito de melhorar a disponibilidade e o desempenho. Um *peer* que inicie a descarga de um dado ficheiro pode de imediato disponibilizá-lo a outros *peers*, mesmo que ainda esteja incompleto, em blocos devidamente identificados. O desempenho global melhora substancialmente com o número de réplicas disponíveis na rede. O serviço é levemente inspirado nas redes do tipo “*BitTorrent*”, mas espera-se que cada grupo proponha uma solução própria e inovadora.

Objetivos

O objetivo deste trabalho é desenhar, implementar e testar um serviço de partilha de ficheiros *peer-to-peer* de alto desempenho, tal como ilustrado na Figura 1. Para garantir um bom desempenho, a transferência deve ser feita com base num protocolo de transferência a correr sobre o protocolo de transporte UDP. Adicionalmente, o ficheiro deve ser previamente dividido em blocos devidamente identificados, que podem ser descarregados em paralelo de múltiplas localizações diferentes. No limite os N blocos que constituem um ficheiro podem ser descarregados ao mesmo tempo de N pares. Ao receber um dado bloco sem erros, um nó deve disponibilizá-lo imediatamente a outros possíveis pares da rede P2P.

Para ter acesso ao serviço, cada nó deve executar uma aplicação designada por **FS_Node**, que é simultaneamente cliente e servidor. Mal inicia a execução, o *FS_Node* conecta-se a um servidor de registo de conteúdos designado por **FS_Tracker**, e informa-o do seu endereço e os ficheiros ou blocos que possui na sua pasta partilhada. O *FS_Tracker* tem, pois, conhecimento atualizado de toda a rede, dos nós que a constituem, e dos ficheiros ou blocos disponíveis em cada um deles. De igual modo, quando pretende localizar e descarregar um ficheiro, um *FS_Node* deve em primeiro lugar interrogar o *FS_Tracker* para obter uma lista atualizada de localizações. De seguida corre um algoritmo de seleção de *FS_Node* e de escalonamento da transferências dos vários blocos que formam o ficheiro.

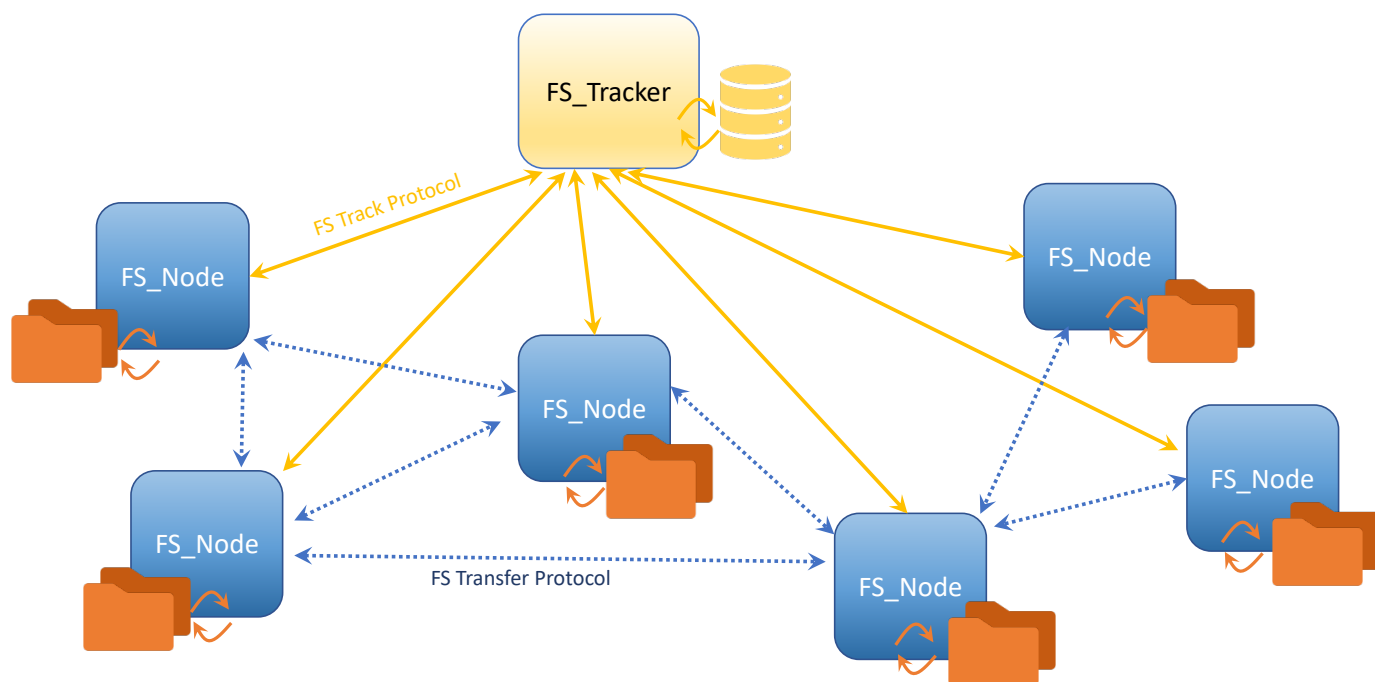


Figura 1: Esquema geral de funcionamento

O protocolo de localização de conteúdos que é usado na interação entre os *FS_Node* e o *FS_Tracker* é designado por **FS Track Protocol** e deve funcionar sobre uma conexão de transporte fiável TCP. Se essa conexão terminar, o *FS_Node* é declarado inativo e removido da base de informação do *FS_Tracker*. O protocolo de transferência, que regula as interações de troca de dados entre os *FS_Node*, designa-se por **FS**

Transfer Protocol e deve funcionar sobre UDP para maior eficiência. Como se trata da transferência de ficheiros, este deve suportar a recuperação de blocos perdidos, emulando a fiabilidade presente em TCP. Adicionalmente, por modo a garantir uma performance adequada, poderá ser implementado também um sistema de monitorização da rede que priorize os utilizadores que oferecem as melhores condições de transferência para o ficheiro desejado (uma implementação simples poderá ser seleccionar os *peers* pela ordem recebida pelo *FS_Tracker*).

Descrição

Para facilitar o desenho do sistema, detalha-se na Figura 2 um possível caso de uso. Neste caso de uso estão envolvidas 4 máquinas diferentes. Numa máquina coloca-se a funcionar a aplicação *FS_Tracker* que fica à escuta de conexões *FS Track Protocol* na porta **TCP 9090**. Este servidor manterá uma conexão ativa por cada *FS_Node* registado. Inicialmente não tem conhecimento de nenhum *FS_Node* e a sua base de informação interna está vazia. Em três outros sistemas colocam-se em execução instâncias da aplicação *FS_Node*. Quando inicia a execução, a aplicação começa por ler a pasta partilhada a seu cargo, que lhe é indicada como parâmetro ou via ficheiro de configuração, e conecta-se à porta TCP 9090 do *FS_Tracker*. Utiliza depois o protocolo *FS Track Protocol* para informar dos ficheiros ou blocos a seu cargo. Ao mesmo tempo, coloca-se à escuta, em permanência, na porta **UDP 9090** por pedidos de blocos segundo o protocolo *FS Transfer Protocol*. A aplicação possui um pequeno interface em modo texto, de linha de comando, para permitir a interação com o utilizador.

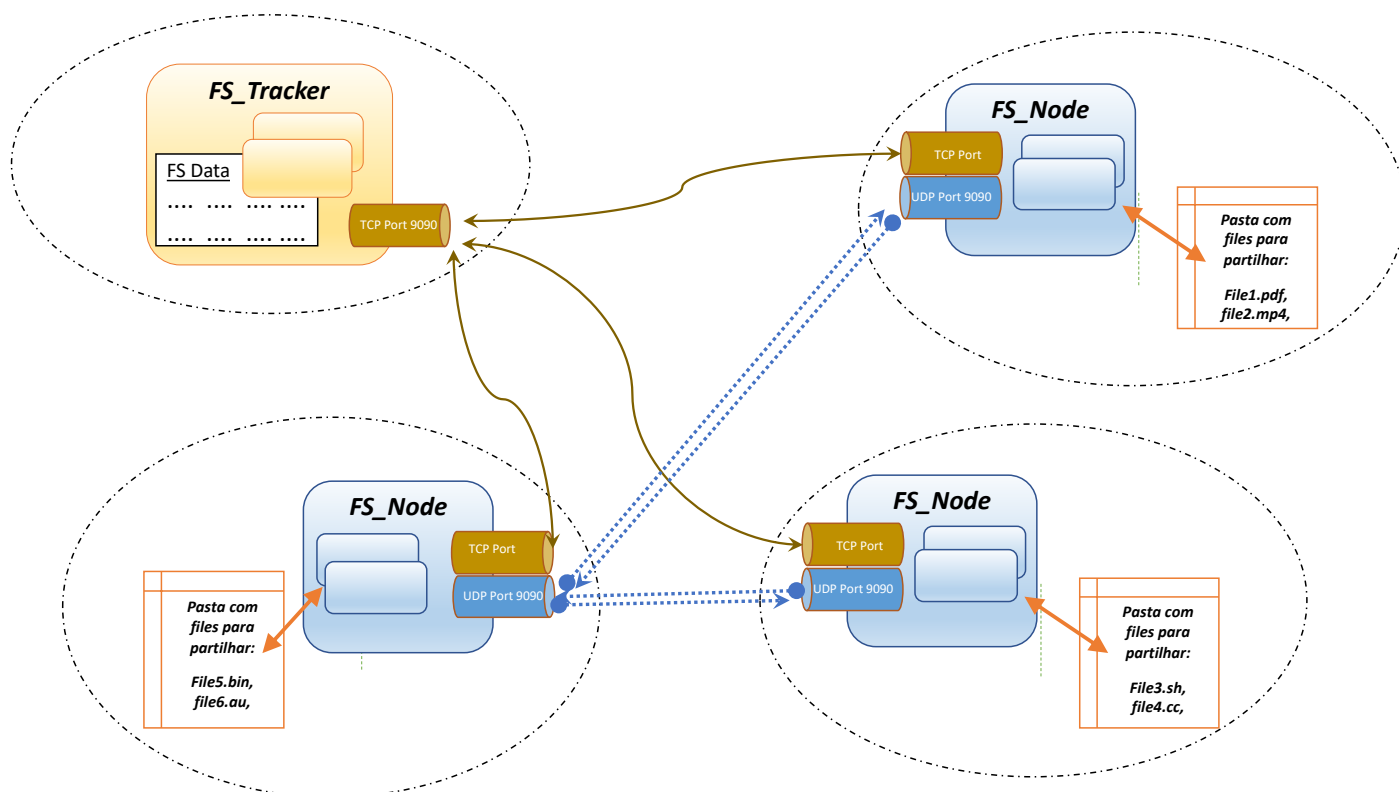


Figura 2: Visão detalhada

A descarga de ficheiros pode ser iniciada por qualquer *FS_Node* da rede, a pedido do utilizador, que pode simplesmente digitar o nome do ficheiro (ex: GET File1). Ao receber uma indicação de descarga, o *FS_Node* interroga o *FS_Tracker* que lhe devolve uma lista dos *FS_Node* que possuem blocos desse ficheiros. De seguida corre um algoritmo de seleção que os ordena com vista ao máximo desempenho e um algoritmo de escalonamento que decide em cada instante quantos e quais blocos deve pedir a cada *FS_Node*. O objetivo é sempre maximizar o desempenho. Os algoritmos e protocolos devem ser desenhados pelos grupos de trabalho.

Cenário de teste

Pode-se usar como plataforma de teste o emulador CORE e a topologia CC-Topo-2023.imn (embora tal não seja obrigatório). Para o cenário de teste mínimo é necessário arrancar o servidor **FS_Tracker** e três ou mais **FS_Node** em diferentes localizações. As localizações devem ser escolhidas de modo a incluir ligações com perdas. O primeiro passo é executar o *FS_Tracker*:

```
Servidor1# FS_Tracker
Servidor ativo em 10.4.4.1 porta 9090.
....
```

Esta aplicação poderá aceitar a porta como parâmetro adicional na linha de comando. Por omissão usa-se a porta TCP 9090. De seguida, iniciam-se várias instâncias do *FS_Node*, por exemplo no **Portatil1**, no sistema **Paris** e no **PC1**:

```
Portatil1# FS_Node folder_portatil1 10.4.4.1 9090
Conexão FS Track Protocol com servidor 10.4.4.1 porta 9090.
FS Transfer Protocol: à escuta na porta UDP 9090.
...
➤ GET file1
...
➤
```

Os parâmetros indicados e a interação são meros exemplos ilustrativos. Cabe a cada grupo decidir a melhor implementação. Neste caso assume-se que o *FS_Node* precisa de receber como parâmetros a pasta partilhada (*folder_portatil1*), o endereço (10.4.4.1) e porta (9090) do *FS_Tracker*.

Desenvolvimento

Requisitos obrigatórios:

- O protocolo *FS Track Protocol* deve funcionar sobre TCP e suportar:
 - Registo de um *FS_Node*
 - Atualização de lista de ficheiros e blocos disponíveis num *FS_Node*
 - Pedido de localização de um ficheiro devolvendo uma lista de *FS_Node* e blocos neles disponíveis
- O protocolo *FS Transfer Protocol* deve funcionar sobre UDP e suportar:
 - Aceitar pedidos de blocos, em paralelo, de múltiplos outros *FS_Node*
 - Pedir, em paralelo, blocos do mesmo ficheiro a múltiplos *FS_Node*
 - Suportar cenários de perda de blocos, garantindo uma entrega fiável
- O *FS_Node* deve registar-se e manter-se conectado a um *FS_Tracker*
- O *FS_Node* deve manter o *FS_Tracker* atualizado em relação à lista de blocos a seu cargo
- O sistema deve permitir a entrada e saída de novos servidores *FS_Node* a qualquer momento

Sugere-se que o desenvolvimento tenha três fases, uma primeira fase mais focada no desenho e implementação do protocolo ***FS Track Protocol*** que deve funcionar sobre TCP, uma segunda fase dedicada ao protocolo *FS Transfer Protocol* sobre UDP, e finalmente uma terceira fase onde os servidores devem ser identificados pelos nomes e não pelos seus endereços. Esta organização do trabalho é apenas uma sugestão. Caberá a cada grupo definir as tarefas, estudar as dependências entre elas e organizar o trabalho em equipa da forma mais adequada.

FASE 1: Desenho, implementação e teste do protocolo *FS Track Protocol*

(três semanas)

Etapas sugeridas para esta fase:

- Especificar o protocolo *FS Track Protocol* para funcionar sobre TCP
 - formato das mensagens protocolares (sintaxe)
 - função e significado dos campos (semântica)
 - diagrama temporal ilustrativo (comportamento)
- Implementação e teste do protocolo *FS Track Protocol*
 - exemplo de teste: registar um *FS_Node* (endereço) e ficheiros/blocos a seu cargo; pedir localização de um ficheiro;

FASE 2: Desenho, implementação e teste do protocolo *FS Transfer Protocol*

(três semanas)

Etapas sugeridas para esta fase:

- Especificar o protocolo *FS Transfer Protocol* para funcionar sobre UDP
 - Formato dos datagramas
 - Modo de funcionamento
- Implementação e teste no cenário proposto

FASE 3: Utilização de um serviço de resolução de nomes

(uma a duas semanas)

Nesta última fase pretende-se fazer uso de um serviço de resolução de nomes (como o DNS – *Domain Name System*), que permita que os *FS_Node* e o *FS_Tracker* se identifiquem com os seus nomes (*Servidor1*, *Portatil1*, *Paris*, *PC1*, ..) e não pelos endereços IPv4, como sugerido

no cenário de teste. O *FS_Tracker* deve manter a lista dos nomes do *FS_Node*. Ao iniciar uma transferência, os *FS_Node* devem resolver os nomes em endereços, antes de tentarem comunicar com os outros *FS_Node*. Esta alteração não implica alteração nas funcionalidades de base do serviço de partilha, nem alterações aos protocolos já desenhados. Apenas alterações na implementação e nos serviços de suporte.

Relatório

O relatório deve ser escrito em formato de artigo com um máximo de 10 páginas (recomenda-se o uso do formato LNCS - *Lecture Notes in Computer Science*, instruções para autores em <http://www.springer.com/computer/lncs?SGWID=0-164-6-793341-0>). Deve descrever o essencial do desenho e implementação com a seguinte estrutura recomendada:

- Introdução
- Arquitetura da solução
- Especificação do(s) protocolo(s) propostos
 - * formato das mensagens protocolares
 - * interações
- Implementação
 - * detalhes, parâmetros, bibliotecas de funções, etc.
- Testes e resultados
- Conclusões e trabalho futuro

Regras de submissão

Cada grupo deve fazer a submissão (*upload*) do trabalho, usando a opção de “troca de ficheiros” associada ao seu grupo nos “Grupos” do Blackboard (*elearning.uminho.pt*), da seguinte forma:

- **CC-TP2-PLx-Gy-Rel.pdf** para o relatório
- **CC-TP2-PLx-Gy-Codigo.zip** ou **CC-TP2-PLx-Gy-Codigo.rar** para a pasta com o código desenvolvido (devidamente documentado)

em que **x** é o identificador do turno PL e **y** o número do grupo (e.g. CC-TP2-PL2-G2-Rel.pdf para o relatório TP2 do grupo 2 do PL2)