

EMULAÇÃO E SIMULAÇÃO DE REDES DE TELECOMUNICAÇÕES

## **RELATÓRIO DE ESPECIFICAÇÃO**

JOÃO CARLOS TEIXEIRA CERCA DA CUNHA(PG47301)

JOÃO MIGUEL CERCA ALVES (PG47318)

JOSÉ PEDRO MARQUES MACEDO (PG47378)

Escola de Engenharia

Mestrado em Telecomunicações e Informática

# ÍNDICE

<b>1</b>	<b>Introdução</b>	<b>3</b>
<b>2</b>	<b>Planeamento</b>	<b>4</b>
<b>3</b>	<b>Arquitetura e Funcionalidades</b>	<b>5</b>
3.1	Arquitetura do Software . . . . .	5
3.2	Arquitetura de Rede . . . . .	7
<b>4</b>	<b>Tecnologias Utilizadas</b>	<b>9</b>
<b>5</b>	<b>Conclusão</b>	<b>10</b>

## 1 Introdução

Este projeto, inserido na unidade curricular de Emulação e Simulação de Redes de Telecomunicações, tem como objetivo o desenvolvimento de uma rede de overlay aplicacional com o intuito de minimizar o atraso fim a fim entre os nós que constituem a rede.

Outro dos objetivos do projeto é perceber a importância da emulação e a simulação de redes e como esta nos ajuda no desenvolvimento de soluções para as Redes de Telecomunicações, pelo que iremos explorar parte deste tópico, nomeadamente a emulação, como ferramenta de auxílio à construção de uma solução.

Neste documento iremos explorar a nossa abordagem inicial do problema, propor uma metodologia de trabalho que nos parece mais adequada tendo em conta os recursos e ferramentas necessários para tal e, por fim, apresentar uma arquitetura geral do sistema a ser concebido mas ainda sem detalhes técnicos.

## 2 Planeamento

O projeto divide-se naturalmente em 4 fases. Na primeira fase o grupo focou-se em analisar o problema, discutir e idealizar uma possível solução dividindo tarefas entre os membros. A divisão temporal de cada tarefa está estruturada de forma visual no diagrama de Gantt abaixo [Figura 1].

Na segunda fase foi onde decidimos atribuir mais tempo para o desenvolvimento sendo que se trata da principal parte do projeto, nesta iremos abordar a conceção da rede overlay, assim como a configuração dos nós e também a criação do protocolo de comunicação. No final desta fase temos ainda duas semanas reservadas para a otimização da rede.

Na fase seguinte é onde vamos criar a aplicação que irá assentar sobre a nossa rede overlay e achamos necessário três semanas para o estudo, conceção e desenvolvimento desta aplicação.

Por fim, temos uma fase focada especialmente na especificação e realização de testes, onde pretendemos obter resultados que nos permitam fazer uma análise qualitativa do desempenho da aplicação com a rede overlay em comparação com a rede estática.

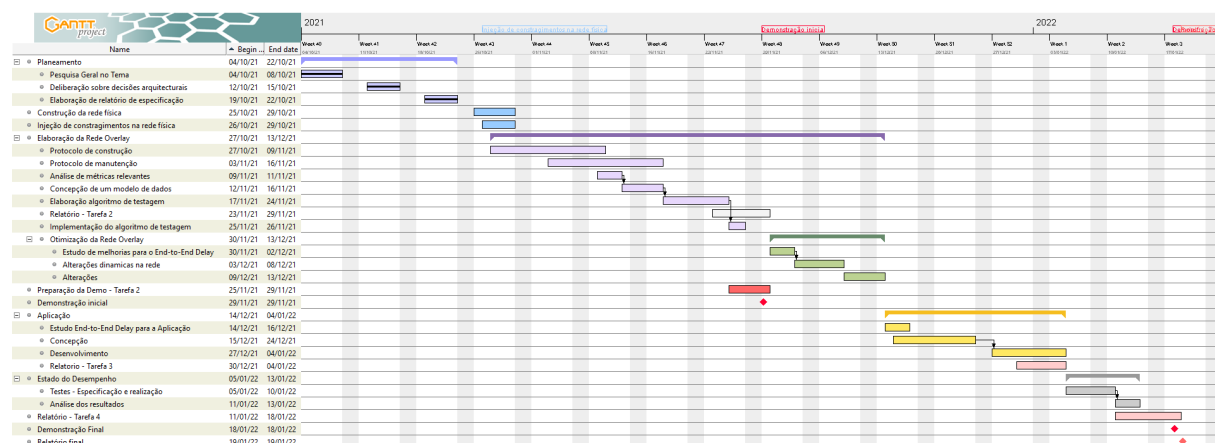
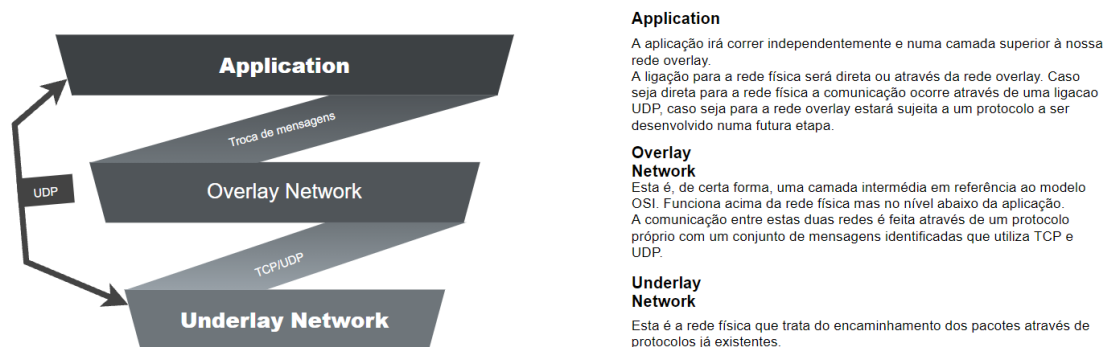


Figura 1: Diagrama de Gantt com as tarefas

## 3 Arquitetura e Funcionalidades

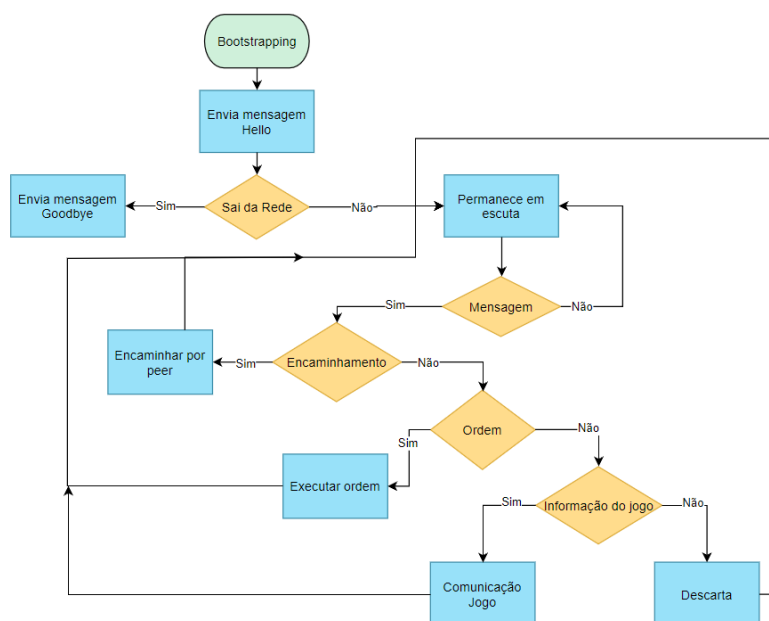
### 3.1 Arquitetura do Software

A premissa da nossa aplicação será sempre a abstração da camada implementada, ou seja, não deverá ser aparente ao utilizador que este software resolve problemas de atraso, ou que se encontra sequer em funcionamento. Deste modo tencionamos implementar uma aplicação que o que é normalmente denominado no campo informático de *Bootstrapping*, esta metodologia auto-sustentável de software permite que o utilizador tenha essa abstração da rede overlay mas a utilize de forma a tirar proveito das suas vantagens.



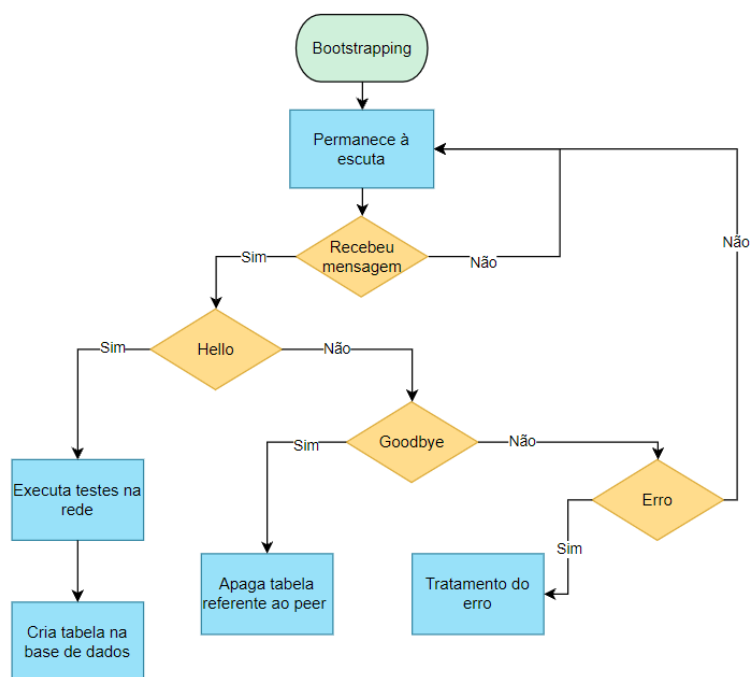
**Figura 2:** Arquitectura simplificada do projecto a ser desenvolvido

Teorizamos um sistema que se adapte automaticamente à entrada e saída de elementos, ainda que numa linguagem rudimentar o funcionamento base poderia ser descrito da seguinte maneira, quando um peer entra na rede, envia um pacote introdutório ao gestor, o qual, através de um modelo de dados, constrói uma tabela de encaminhamento referente a esse peer. De seguida, envia um ID que permita ao peer ser associado com a respectiva tabela de encaminhamento criada na base de dados, possivelmente também a servir de autenticação.



**Figura 3:** Protótipo de funcionamento da vertente comunicacional do peer.

Na entrada de um novo nó na rede é enviada uma mensagem introdutória em broadcast para a rede que anuncia ao servidor. A saída de um peer também deve ser efectuada de forma simples, então da mesma forma que se une, o peer que abandona a rede envia uma mensagem a informar a saída ao gestor e este procede à remoção da tabela desse peer na base de dados assim como qualquer outra entrada na mesma que tenha presente esse peer. O gestor tem ainda outra função que será a atualização de toda a base de dados consoante as informações recebidas sobre o estado da rede.

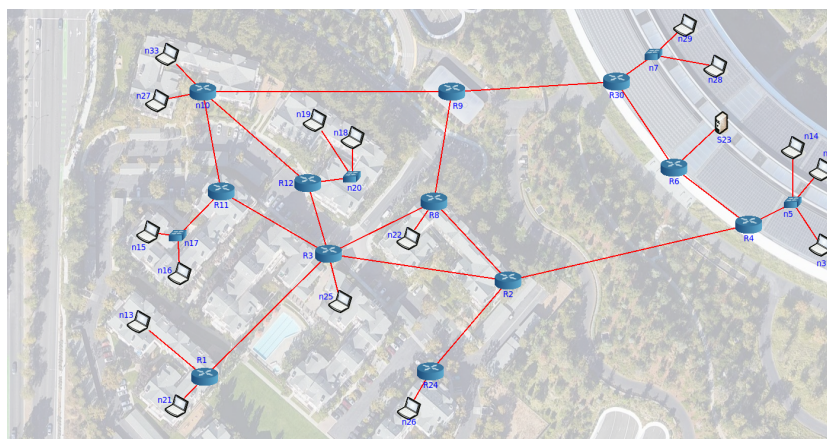


**Figura 4:** Protótipo de funcionamento do gestor

A fundamentação para a conceção da ideia acima tem de origem um documento apresentado na *International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon)* em 2019 que apresenta um estudo em redes de organização automática [2].

### 3.2 Arquitetura de Rede

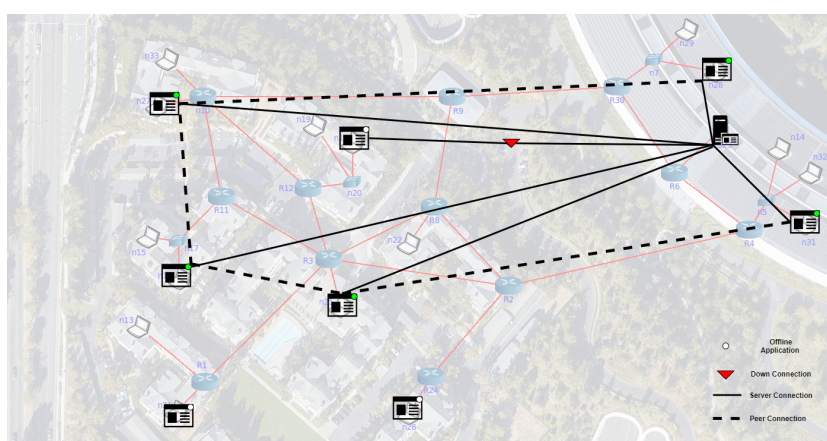
Para a elaboração da arquitetura de rede deste projeto começamos pela concepção de uma rede estática no ambiente de emulação do CORE [Figura 5]. Esta rede serve o propósito de representar uma rede física real, em que os nós se encontram fisicamente conectados e assim permanecerão de forma estática.



**Figura 5:** Mapeamento de uma área residencial com uma proposta de rede física.

Esta rede underlay [Figura 5] irá usar como protocolo de encaminhamento o OSPF (*Open Shortest Path First*) visto que este apresenta uma convergência rápida e funciona através do protocolo IP (*Internet Protocol*) o que se assemelha a grande parte das redes atualmente em funcionamento, ao oposto de, por exemplo, o protocolo RIP (*Routing Information Protocol*), que apesar da sua conotação histórica, apresenta bastantes problemas ao nível de escalonamento. Para além disso, o OSPF apresenta a vantagem de suportar várias métricas e de poder ter múltiplos caminhos para um determinado nó da rede.

Por cima da rede física temos a nossa rede de overlay [Figura 6]. Esta rede, ao contrário da que assenta por baixo, é uma rede dinâmica que se vai adaptando conforme entrem ou saiam nós da mesma e terá um protocolo de encaminhamento a ser futuramente desenvolvido pelo grupo. No entanto, também existirão nós fixos, sendo atribuído a eles o cargo de servidor/gestor que tomarão as funções de gestão da rede. Ambas estas funcionalidades, protocolo de encaminhamento e a gestão da rede overlay são temas em desenvolvimento e exploramos as presentes ideias abaixo.



**Figura 6:** Proposta de rede Overlay.

Ideologicamente, o servidor terá de ter alguma concorrência, de forma a conseguir obter uma medição da métrica desejada de cada uma das ligações existentes nesse momento na rede de overlay, e acompanhar

as alterações que acontecem a nível de membros da rede. Esta rede de overlay irá servir uma aplicação a ser desenvolvida numa fase posterior deste projeto.

A nível da rede de overlay os protocolos de comunicação a serem usados serão o TCP e o UDP, sendo que o UDP terá função de apenas enviar pacotes de forma periódica em que o objetivo é a rapidez e não a integridade da mensagem em si, ao contrário do TCP que será ser utilizado em trocas de informação com o servidor e porventura algumas comunicações entre os peers, toda a informação não descartável. Tudo isto dependerá do protocolo a ser desenvolvido por nós, o conteúdo e o uso das unidades de dados protocolares serão ditadas pelo objetivo do programa.



## 4 Tecnologias Utilizadas

A nível de software a ferramenta principal que iremos utilizar para emulação é o **CORE (Common Open Research Emulator)** [1], acompanhada pela ferramenta **Git** e a plataforma do **Github** de forma a que seja mantido um controlo durante toda a fase de desenvolvimento do software. Isto permite cada elemento do grupo trabalhar de forma independente em branches separados, reduzindo significativamente problemas de compatibilidade. Vamos usar também o **Discord** como plataforma de comunicação entre os elementos do grupo de modo a trocarmos de ideias mais facilmente. Como editor de texto/ planeamos usar o **Visual Studio Code** porque já nos encontramos bastantes familiarizados com este IDE e também temos a vantagem de ter um bom suporte da plataforma Git. A nível de linguagens de programação iremos usar o **Python** para configuração dos servidores e dos peers, porque é uma linguagem de alto nível e bastante fácil de utilizar. Relativamente à base de dados iremos usar o **MySQL** porque já temos alguma experiência com essa e assim reduz o nosso tempo de aprendizagem.

Relativamente ao conjunto de competências ainda não dominadas de momento estas são:

- Otimização de um protocolo de comunicação de rede de overlay orientado para diminuição do *End-to-End delay* [3];
- Conjunto de bibliotecas que iremos usar em python para configurar os peers e os servidores da rede;
- Estudo das métricas importantes para o caso e a transposição para o mundo real da implementação de uma solução destas;
- Desenvolvimento de uma arquitetura da base de dados que suporte toda a informação necessária mas que, simultaneamente, não necessite muito poder computacional;
- A aplicação devido à natureza aberta, ainda nos encontramos com algumas incertezas relativamente à direção que pretendemos seguir e a que será mais adequada para a demonstração do propósito da rede overlay, ou seja, evitar o delay *End-to-End*.

## **5 Conclusão**

Em suma este primeiro relatório serviu para fazermos um esboço inicial do que será o nosso projeto e nosso trabalho ao longo do semestre. Começamos por interpretar o trabalho proposto lendo a bibliografia disponibilizada, alguma pesquisa extra e tirando todas as dúvidas que surgiram após isso com os docentes. Tentamos também perceber os recursos e conhecimentos que iremos precisar para concluir o projeto com sucesso de forma a conseguirmos dividir bem as tarefas e perceber o tempo que será despendido por cada elemento, facilitando o trabalho do grupo.

## Referências

- [1] *Documentação da ferramenta CORE*. URL: <https://coreemu.github.io/core/>. (accessed: 12.10.2021).
- [2] A. Ermakov e L. Suchkova. «Development of Data Exchange Technology for Autonomous Robots Using a Self-Organizing Overlay Network». Em: *2019 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon)*. Out. de 2019, pp. 1–5. DOI: 10.1109/FarEastCon.2019.8934727.
- [3] Shu-NGai Yeung e J. Lehoczky. «End-to-end delay analysis for real-time networks». Em: *Proceedings 22nd IEEE Real-Time Systems Symposium (RTSS 2001) (Cat. No.01PR1420)*. Dez. de 2001, pp. 299–309. DOI: 10.1109/REAL.2001.990628.