

Redes de Overlay para Aplicações sensíveis a atrasos

1. Introdução

Uma rede de overlay é uma rede lógica construída sobre uma rede física existente. Podem ser criadas por múltiplas razões. Por exemplo, as redes *Peer-to-Peer* (P2P) são redes de *overlay* aplicacional [3], criadas fundamentalmente para permitir a partilha eficiente de recursos, sejam eles conteúdos (ficheiros, *streams* multimédia, etc.) ou simplesmente CPU e memória (computação distribuída). Para ilustrar melhor este conceito apresenta-se a figura abaixo. Como se pode observar, uma ligação da rede de *overlay* não tem obrigatoriamente que corresponder a uma ligação da rede física. Se se tratar de uma rede de *overlay* aplicacional, os nós da rede são tipicamente sistemas terminais (*hosts*), que estão ligados entre si formando uma topologia lógica. Cada ligação da rede aplicacional, liga dois nós da rede de *overlay* recorrendo para isso a um conjunto de ligações da rede física, que não tem que corresponder à ligação que une esses dois nós na rede física, mesmo que essa ligação exista.

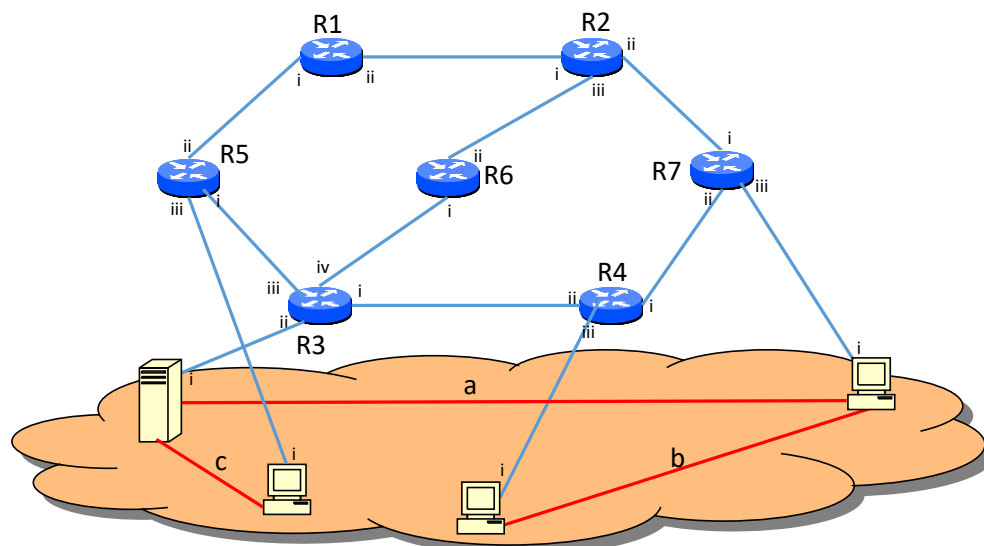


Figura 1: Exemplo de uma rede de overlay aplicacional

Como já foi referido em cima, os objetivos para a criação de uma rede de *overlay*, podem ser muito variados. Neste projeto vamos estudar a possibilidade de desenvolver uma rede de *overlay* aplicacional que tem como objetivo minimizar o atraso fim a fim, entre os nós que constituem a rede. Podemos supor que esta rede será usada por uma aplicação particularmente sensível a esta métrica, por exemplo um jogo interativo.

A simulação e emulação de redes permite-nos testar a validade de um protocolo de comunicações, antes de partirmos para o seu desenvolvimento num ambiente real. Normalmente as redes e os protocolos de comunicações são sistemas muito complexos, pelo que a utilização de ferramentas como emuladores e simuladores de redes é muito importante. Com estas ferramentas conseguimos perceber o comportamento do sistema e obter medidas de desempenho que nos ajudam a perceber a validade e a parametrizar corretamente as soluções que concebemos.

Neste projeto queremos perceber até que ponto é que a simulação e emulação de redes nos ajuda no desenvolvimento de soluções para as Redes de Telecomunicações. Assim, e perante um problema complexo (o desenvolvimento de uma rede de *overlay* aplicacional que nos permite minimizar o atraso fim-a-fim entre os nós que a compõem) vamos usar a emulação como ferramenta de auxílio à construção de uma solução, nas suas diferentes fases.

2. Descrição

A Figura 1 mostra um exemplo de uma rede de *overlay* aplicacional. Como já foi referido, nas redes de *overlay* aplicacional os nós da rede são sistemas terminais e funcionam também como encaminhadores na rede de *overlay*. Cada link que une dois nós na rede de *overlay*, é um link virtual e corresponde a um conjunto de links da rede física de suporte. Por exemplo, na Figura 1, o link **a**, pode corresponder ao caminho R3-R5-R1-R2-R7 ou ao caminho R3-R4-R7.

Por outro lado, como podemos observar também na Figura 1, nem todos os nós que integram a rede física fazem parte da rede de *overlay*. Uma rede de *overlay* aplicacional é muito dinâmica e tipicamente a sua constituição vai variando ao longo do tempo. Tipicamente um nó junta-se à rede de *overlay* quando o utilizador que controla esse nó ativa a aplicação que usa esta rede.

2.1 Rede de Overlay

A rede de *overlay* constrói-se e altera-se dinamicamente cada vez que um nó decide juntar-se ou abandonar a rede. O processo que permite a um nó juntar-se à rede designa-se por “*bootstrapping*” e implica sempre a capacidade de localizar algum outro nó que já faça parte da rede. Existem diferentes alternativas para o “*bootstrapping*” e é preciso averiguar qual é a melhor. O abandono da rede é por outro lado também uma operação comum que pode ser problemática já que obriga a rede a ser capaz de se reconfigurar. Dependendo da estratégia e do tipo de *overlay* que se pretende construir, cada nó da rede deve ter sempre conhecimento de N outros nós. Uma tarefa de manutenção comum consiste em manter esse N estável ao longo do tempo e à medida que vão ocorrendo eventos de entrada e saída de nós. Uma vez membro da rede, um nó pode começar a executar a aplicação para a qual a rede se destina.

2.2 Aplicação

As redes overlay podem ser construídas com múltiplos objetivos [4], sendo que um deles é a distribuição de conteúdos (partilha de ficheiros, *Content Delivery Networks*, etc.) com o objetivo de reduzir os tempos de descarga e os custos de tráfego no transporte pela rede. Outro objetivo importante é permitir desenhar novos protocolos e fazer evoluir a Internet, usando a própria Internet como rede de suporte, e construindo um novo overlay com novos protocolos por cima. Neste trabalho pretende-se, no entanto, usar a rede *overlay* para melhorar o suporte a aplicações sensíveis ao tempo de atraso fim a fim (*end-to-end delay*) ao mesmo tempo que se mantém a resiliência da rede. Há um conjunto de aplicações sensíveis ao tempo e ao atraso que poderiam ver o seu desempenho melhorado com o suporte desta rede *overlay*, nomeadamente as aplicações interativas de tempo real, como as chamadas de voz e de vídeo, ambientes virtuais, teleconferência ou jogos interativos com múltiplos jogadores. A aplicação fica em aberto, para definição posterior, mas sugere-se por exemplo um jogo simples interativo entre dois (ou mais) jogadores.

3. Planeamento e descrição das várias tarefas

O planeamento do trabalho é da responsabilidade de cada grupo. Não existe uma solução única esperada para todos os grupos. Cada grupo deve em primeiro lugar conceber, discutir e propor a sua própria solução. Há espaço para soluções novas, mas se elas não surgirem em tempo útil, o docente pode encaminhar o grupo para uma solução mais conduzida. Nesse sentido, as tarefas propostas são apenas exemplos que podem ser ajustados caso a caso.

Tarefa 0: Especificação do Projeto

Nesta tarefa inicial, de especificação do projeto, pretende-se

- Definição da arquitetura e funcionalidades a serem oferecidas pelo sistema.
- Identificação das tecnologias a serem utilizadas e fazer uma estimativa dos recursos que irão necessários (para o projeto inteiro).
- Discussão sobre as competências necessárias, indicando claramente o conjunto das que ainda não são dominadas na altura da escrita do documento.
- Planeamento temporal geral de todas as fases, tarefas e subtarefas através dum diagrama de Gantt.

O **relatório inicial de especificação do projeto (RI)** deverá abordar todos os tópicos referidos acima, em subsecções separadas.

Tarefa 1: Criação de uma topologia inicial

Criar uma topologia inicial, funcional, com routers ligados entre si e sistemas terminais ligados a redes locais que por sua vez têm conectividade através da topologia dos routers. Numa segunda versão devem ser introduzidos na topologia alguns constrangimentos de forma a criar situações em que o caminho escolhido pelo routing não seja o que conduz a um menor atraso fim a fim. Apesar disso, a topologia deve manter-se funcional, com conectividade, recorrendo aos protocolos já existentes e disponíveis na plataforma de teste;

Tarefa 2: Conceção e desenvolvimento do protocolo de construção da rede de overlay

Para construir a rede *overlay*, e mantê-la em funcionamento, os nós que a integram devem comunicar entre si usando um protocolo específico para o efeito.. Nesta tarefa, pretende-se em primeiro lugar definir os requisitos desse protocolo e, a partir deles, produzir uma especificação do mesmo em termos da semântica e sintaxe das mensagens a serem trocadas e diagramas temporais associados. O protocolo permitirá definir e concretizar uma estratégia distribuída de construção e manutenção conjunta do *overlay*. Enumeram-se, a título de exemplo, alguns requisitos gerais de uma rede *overlay* auto-organizada, que podem ou não ser considerados:

1. Um nó que pretenda fazer parte da rede, deve iniciar uma aplicação para o efeito, fácil de instalar e independente do sistema operativo de suporte;
2. Um nó deverá abandonar de forma adequada o *overlay* quando a aplicação terminar a sua execução;
3. Modularidade: é desejável que haja uma separação entre o protocolo de construção e manutenção do *overlay* e a aplicação que vai funcionar sobre o *overlay*, ainda que o *overlay* suporte uma única aplicação;
4. A aplicação não deve necessitar de configuração manual (idealmente, configuração zero); um nó deve conseguir ligar-se à rede *overlay* e manter-se bem conectado com um mínimo de configuração;
5. Cada nó deve manter de forma autónoma um número mínimo N de ligações a outros nós, sendo N o invariante da rede;
6. Cada nó deve ter capacidade de monitorização das condições mínimas de operacionalidade, especificadas ao nível do nó, do link, ou do overlay (a partir de um conjunto de métricas pré-definidas);
7. O objetivo do *overlay* é minimizar o atraso na comunicação entre os nós que o constituem, independentemente da rede física de suporte (podem ser consideradas outras métricas);
8. Idealmente, a rede deve operar no espaço do utilizador (*user level*) sem necessidade de privilégios especiais ao nível do kernel;
9. Para melhor independência da rede física, a rede deve poder funcionar sobre protocolos de transporte normalizados como TCP ou UDP;

De salientar que cada nó na rede *overlay* pode assumir múltiplas funções. Além de poder ser cliente aplicacional, e suportar o funcionamento da aplicação, tem também de conseguir encaminhar pacotes de dados, seus ou de outros nós para o destino certo, escolhendo próximo salto na rede *overlay*. Esta função de *routing*, não deve ser confundida com a função de *routing* da rede física, que correrá os seus próprios protocolos de forma independente e transparente. Significa que ao receber um pacote destinado a outro nó da rede *overlay*, tem de decidir o que fazer com ele. Esta funcionalidade é distinta da função de junção à rede, onde cada nó procura conhecer nós vizinhos e manter links ativos com eles, permitindo uma comunicação direta. A descoberta de caminhos pode ser feita a pedido, de forma reativa, iniciando a descoberta da rota quando necessário, ou por antecipação, de forma proactiva, mantendo informação em tabelas. Enumeram-se, a título de exemplo, algumas funcionalidades, que podem ser consideradas:

- i. Identificação dos nós
- ii. Localização inicial de um ou mais nós da rede (bootstrap)
- iii. Estabelecimento de um link virtual com nó vizinho (coligindo métricas de qualidade da ligação)
- iv. Lidar com perdas e recuperação de links com vizinhos
- v. Encaminhar dados no overlay
- vi. Segurança

No final desta tarefa deve ser possível enviar pacotes de teste na rede overlay.

Tarefa 3: Construção da aplicação que vai assentar sobre a rede de overlay

Pretende-se que a rede *overlay* seja otimizada para minimizar atrasos entre os nós do *overlay*, de modo a poder suportar de forma adequada aplicações sensíveis ao tempo de atraso. Um exemplo são os jogos interativos. Nesta tarefa pretende-se desenhar uma aplicação desse tipo que possa funcionar sobre a rede *overlay* desenhada na tarefa anterior. Como já foi dito, a *aplicação* poderia confundir-se com o próprio *overlay*, sendo implementada de forma conjunta. Mas o ideal será poder ter uma separação clara entre a camada aplicacional e a camada do *overlay*. No limite, poder-se-ia testar a mesma aplicação diretamente sobre a rede física e sobre a rede *overlay*, coligindo medidas e comparando o desempenho num e noutro cenário. A escolha da aplicação fica ao critério de cada grupo de trabalho. Uma proposta concreta, que pode ou não ser considerada para testes, é a implementação de um jogo interativo, onde o objetivo é ser mais rápido que o adversário e dirigir-lhe uma sequência de N notificações seguidas sem resposta (ex: N=4). Cada jogador identifica-se com uma letra atribuída pela ordem de entrada no jogo. Havendo mais que um jogador, cada um tem uma letra diferente. Para enviar notificações ao jogador A, um outro jogador qualquer B deve pressionar sucessivamente a tecla A. A cada pressionar da tecla é enviada uma notificação de B para A. Se A não responder a B antes de N notificações deste, perde um ponto. Ganha quem tiver menos pontos ao fim de X tempo ou quando alguém atingir Y pontos. Com mais que um jogador, o jogo complica-se, pois é preciso dirigir as notificações aos jogadores certos.

Tarefa 4: Estudo do desempenho da solução desenvolvida.

Nesta tarefa pretende-se fazer um estudo de desempenho da solução desenvolvida. Em primeiro lugar será necessário escolher as métricas que vão ser usadas. Para cada uma dessas métricas, importa definir a sua importância e a forma como pode ser coligida. Finalmente, e tirando partido do ambiente de desenvolvimento emulado, devem ser criados alguns cenários que permitam comparar o desempenho da aplicação desenvolvida com ou sem a rede de *overlay*.

4. Principais metas

Fase 1: Especificação inicial

Fase 2: Demonstração da rede de overlay

Fase 3: Demonstração com a aplicação a funcionar através da rede de overlay

Fase 4: Apresentação Final

5. Ambiente de Desenvolvimento e Teste

A implementação e teste da rede, bem como dos protocolos de encaminhamento e aplicação propostos, deverá ser efetuada na plataforma de emulação CORE (*Common Open Research Emulator*);

6. Entrega do trabalho

O trabalho deve ser realizado em grupo (de três elementos) sendo a constituição dos grupos da inteira responsabilidade dos alunos.

O trabalho deverá ser demonstrado em duas fases (*demonstração da rede de overlay*, *demonstração da aplicação*). Além disso os alunos deverão submeter dois relatórios: um relatório inicial de especificação e planeamento do projeto e um relatório final, que deve ser escrito em formato de artigo com um máximo de 8 páginas (recomenda-se o uso do formato LNCS - *Lecture Notes in Computer Science*). Este último deve descrever o essencial do desenho e implementação com a seguinte estrutura recomendada: Introdução; Especificação dos protocolos (primitivas de comunicação, formato das mensagens protocolares (PDU), interações); Implementação (detalhes, parâmetros, bibliotecas de funções, etc); Testes e resultados; Conclusões e trabalho futuro. Estes relatórios e o respetivo código deverão ser submetidos, na plataforma *elearning.uminho.pt*.

Na apresentação final, além de uma apresentação da solução, os alunos deverão ser capazes de discutir o seu desempenho, mostrando os pontos fortes e fracos da solução desenvolvida.

7. Avaliação

A avaliação da UC será individual e terá em conta a contribuição de cada aluno para o seu grupo de trabalho. Os elementos de avaliação incluem um **relatório inicial de especificação do projeto (RI)**, uma **demonstração intermédia do trabalho realizado até a Fase 2** (inclusive) **(DI)**, um **relatório final (RF)** e uma **demonstração final (DF)** ao término da Fase 4.

Junto com a demonstração final deverá ser feita uma apresentação oral (cerca de 15 minutos) baseada num ficheiro Powerpoint, que deverá fornecer uma visão global e objetiva do sistema desenvolvido, incluindo a descrição de todas as partes que compõem o sistema final. Neste sentido, a apresentação não deve incluir conteúdo sobre o planeamento do projeto, mas sim sobre a implementação e testes. Também não deve organizada de forma cronológica nem deve focar em implementações falhadas anteriores, mas sim na solução otimizada final, embora possam ser referidos brevemente os problemas encontrados e a sua resolução. A apresentação deverá incluir os principais resultados da análise de desempenho, bem como um slide final com as principais conclusões técnicas sobre o sistema implementado no projeto.

A UC inclui ainda um elemento de avaliação contínua, que terá em consideração parâmetros como a assiduidade (faltas), pontualidade (atrasos), participação, conhecimentos demonstrados, empenho e comportamento nas aulas. Os pesos dos elementos de avaliação são indicados abaixo:

- RI - Relatório inicial (10%)
- DI - Demo intermédia (20%)
- RF - Relatório final (20%)
- DF - Demo final (25%)
- AO - Apresentação oral (15%)
- AC - Avaliação contínua (10%)

O calendário com as datas das avaliações é disponibilizado num ficheiro à parte, na plataforma Blackboard.

Referências

- [1]. Ermakov and L. Suchkova, "Development of Data Exchange Technology for Autonomous Robots Using a Self-Organizing Overlay Network," *2019 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon)*, 2019, pp. 1-5, doi: 10.1109/FarEastCon.2019.8934727.
- [2]. G. Tato, M. Bertier and C. Tedeschi, "Designing Overlay Networks for Decentralized Clouds," *2017 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, 2017, pp. 391-396, doi: 10.1109/CloudCom.2017.64.
- [3]. Naik, A.R. and Keshavamurthy, B.N., 2020. Next level peer-to-peer overlay networks under high churns: a survey. *Peer-to-Peer Networking and Applications*, 13(3), pp.905-931.
- [4]. Clark, D., Lehr, B., Bauer, S., Faratin, P., Sami, R. and Wroclawski, J., 2006. Overlay Networks and the Future of the Internet. *Communications and Strategies*, 63, p.109.
- [5]. Kurose, J. (2016). *Computer Networking: A Top-Down Approach* (7th edition.). Pearson. Retrieved from <https://www.pearson.com/us/higher-education/program/Kurose-Computer-Networking-A-Top-Down-Approach-7th-Edition/PGM1101673.html?tab=resources>