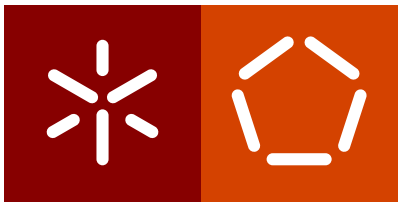
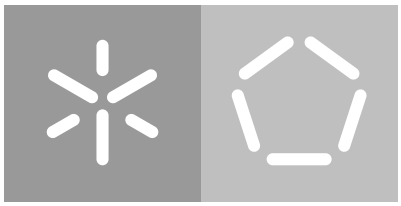


Universidade do Minho
Escola de Engenharia
Departamento de Informática

Miguel Gil Pires da Silva

Uma Rede Overlay Controlada pelo ISP para Partilha de Conteúdos

Outubro de 2019



Universidade do Minho
Escola de Engenharia
Departamento de Informática

Miguel Gil Pires da Silva

Uma Rede Overlay Controlada pelo ISP para Partilha de Conteúdos

Dissertação de Mestrado
Mestrado Integrado em Engenharia Informática

Dissertação efetuada sob a orientação do
Professor Doutor Pedro Nuno Sousa

Outubro de 2019

DIREITOS DE AUTOR E CONDIÇÕES DE UTILIZAÇÃO DO TRABALHO POR TERCEIROS

AGRADECIMENTOS

Gostaria de agradecer a ajuda de várias pessoas que me apoiaram na conclusão desta dissertação. Em primeiro, gostaria de agradecer ao meu orientador, Professor Doutor Pedro Nuno Sousa, por me ter guiado durante todo o percurso e pela motivação e suporte que me ofereceu ao longo deste projeto. Além disso, agradeço a disponibilidade apresentada e as inúmeras discussões que permitiram melhorar e finalizar esta dissertação. Por último, gostaria de agradecer também à minha família, à minha namorada e aos meus amigos, pelo apoio que prestaram ao longo do meu percurso académico, influenciando também a conclusão desta dissertação.

DECLARAÇÃO DE INTEGRIDADE

Declaro ter atuado com integridade na elaboração do presente trabalho académico e confirmo que não recorri à prática de plágio nem a qualquer forma de utilização indevida ou falsificação de informações ou resultados em nenhuma das etapas conducente à sua elaboração.

Mais declaro que conheço e que respeitei o Código de Conduta Ética da Universidade do Minho.

RESUMO

Atualmente, vivemos numa era extremamente tecnológica e exigente, traduzindo-se no incremento das necessidades dos utilizadores da *Internet*, face aos serviços que esta lhes disponibiliza. Neste sentido, existe uma crescente preocupação em desenvolver novos sistemas de rede virtuais, como é o caso das redes *overlay*, muitas delas ligadas à distribuição de conteúdos. Porém, como consequência deste crescimento adoptivo das redes *overlay*, as infraestruturas dos fornecedores de rede (Internet Service Provider (ISP)) são sobrecarregadas com tráfego Peer-to-Peer (P2P), produzindo maiores dificuldades em relação à sua administração. Assim, esta dissertação apresenta como objetivo primário conceber e promover o uso de uma rede *overlay* controlada pelo *ISP*, que disponha de mecanismos de controlo de tráfego *P2P*, e que, em simultâneo, possibilite a oferta de um serviço de partilha de ficheiros aos seus clientes. Como tal, foi especificada uma arquitetura de rede baseada no paradigma *P2P*, para suportar funcionalidades colaborativas com o *ISP*, sendo composta por quatro entidades principais: *peer* de acesso, *peer* de encaminhamento, coordenador e *ISP*. A partir desta, implementaram-se quatro aplicações correspondentes a cada uma das entidades, tendo sido criadas *interfaces* gráficas direcionadas para o *ISP* e para os clientes (*peers* de acesso). Foram implementados mecanismos básicos de partilha de conteúdos com suporte de dois modos de transferência (cliente-servidor e *P2P*), além de mecanismos de proteção e priorização de *links/routers* críticos (com a possibilidade de planeamento por datas), limitação de circulação de conteúdos na rede *overlay* e suporte a duas estratégias distintas de encaminhamento aplicacional, como principais medidas de apoio ao *ISP*. Posteriormente, a partir do emulador de rede Common Open Research Emulator (CORE), foram criados cenários distintos de teste, relativos aos mecanismos de colaboração com o *ISP* implementados. Nestes, além de ser analisado o tráfego transmitido por *router*, de modo a verificar os impactos dos mecanismos colaborativos com o *ISP*, foi efetuada a análise das rotas percorridas na rede física/aplicacional. Por fim, a partir dos cenários de teste efetuados, assume-se que os mecanismos de controlo de tráfego *P2P*, não afetam significativamente a qualidade do sistema da rede *overlay*, facilitando ao mesmo tempo as tarefas de administração do *ISP*.

Palavras-Chave: *Internet*, *ISP*, Partilha de Conteúdos, *P2P*, *Rede Overlay*

ABSTRACT

We are currently living in an extremely technological and demanding era, resulting in an increase in needs of Internet users about the services it offers them. In this sense, there is a growing concern to develop new virtual network systems, such as overlay networks, many of them linked to content distribution. However, as a consequence of this adoptive growth of overlay networks, network provider (ISP) infrastructures are overloaded with P2P traffic, producing more considerable difficulties concerning its administration. Thus, this dissertation presents as the primary objective to conceive and promote the use of an overlay network controlled by ISP, which has P2P traffic control mechanisms and enabling, at the same time, a file sharing service to customers. As such, a network architecture based on the P2P paradigm was specified to support collaborative functionality with ISP, consisting of four main entities: peer access, forwarding peer, coordinator and ISP. From this, four applications corresponding to each of the entities were implemented, having been created graphical interfaces directed for the ISP and the clients (access peers). Underlying content sharing mechanisms with support for two transfer modes (client-server and P2P) were implemented, as well as protection and prioritization mechanisms for critical links/routers with the possibility of planning by dates, limitation of content circulation on the overlay network and support for two distinct application routing strategies as main support measures for ISP. Subsequently, from the Common Open Research Emulator (CORE) network emulator, distinct test scenarios were created concerning the collaboration mechanisms with the ISP implemented. In these, besides analyzing the traffic transmitted by the router, to verify the impacts of the collaborative mechanisms with ISP, the analysis of the routes traveled in the physical/applicational network was performed. Finally, from the test scenarios performed, it is assumed that the P2P traffic control mechanisms do not significantly affect overlay network quality while facilitating ISP administration tasks.

Keywords: *Content Sharing, Internet, ISP, Overlay Network, P2P*

ÍNDICE

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ACRÓNIMOS

A

API Application Program Interface.

AS Autonomous System.

B

BGP Border Gateway Protocol.

C

CAN Content Addressable Network.

CDN Content Delivery Network.

CORE Common Open Research Emulator.

CSS Cascading Style Sheets.

D

DHT Distributed Hash Table.

DIFFSERV Differentiated Services.

DNS Domain Name System.

H

HTML HyperText Markup Language.

HTTP Hypertext Transfer Protocol.

I

IGP Interior Gateway Protocol.

IOT Internet of Things.

IP Internet Protocol.

ISP Internet Service Provider.

J

JSON JavaScript Object Notation.

M

MPLS Multi Protocol Label Switching.

O

OAM Operations, Administration and Maintenance.

OSI Open System Interconnection.

OSPF Open Shortest Path First.

P

P2P Peer-to-Peer.

P4P Provider Portal for Applications.

PA-CDN Peer-Assisted Content Delivery Networks.

PDU Protocol Data Unit.

Q

QOS Quality of Service.

R

REST Representational State Transfer.

S

SIS SmoothIT Information Service.

SLA Service Level Agreement.

T

TCP Transmission Control Protocol.

U

UDP User Datagram Protocol.

1

INTRODUÇÃO

1.1 ENQUADRAMENTO E MOTIVAÇÃO

Atualmente, com o crescimento exponencial da Internet, torna-se complexa a criação de serviços que consigam satisfazer as necessidades dos clientes, de forma rápida barata e eficaz. Cada vez mais, existe uma procura de serviços capazes de traduzir estas necessidades e para isso têm sido criadas novas soluções, como as redes *overlay*. O aparecimento das redes *overlay*, baseadas em abordagens *P2P*, trouxe inúmeras vantagens aos utilizadores, visto serem uma solução barata e escalável na execução de serviços. Estas, normalmente são implementadas diretamente na camada aplicacional de uma rede física já existente, daí se designarem redes de sobreposição. Porém, embora em termos de execução de serviços sejam escaláveis e eficientes, geram elevados volumes de tráfego, causando dificuldades para o *ISP* na gestão da infraestrutura.

No contexto deste trabalho, uma solução para a partilha de conteúdos, será a utilização de uma rede *overlay*/*P2P*, devidamente controlada pelo *ISP*, para providenciar este serviço a todos os utilizadores com acesso à rede. Com este sistema de rede, pretende-se utilizar as capacidades de processamento dos utilizadores (*peers*) como suporte de computação e comunicação, tornando assim este serviço escalável e eficiente. Deste modo, a partilha de conteúdos vai ser efetuada através da distribuição de informação por múltiplos *peers* que assumem o papel de *routers* virtuais, responsáveis pelas regras de encaminhamento de dados e mecanismos de partilha. Além disso, com a colaboração do *ISP* e consciencialização da rede física, pretende-se possibilitar ao próprio, mecanismos de controlo eficazes sobre como o tráfego da rede *overlay* atravessa a rede física do administrador. Estes, podem ou não, melhorar a qualidade de serviço do sistema de partilha de rede, dependendo do tipo de decisões de tráfego aplicacional efetuadas.

Na atualidade, já existem redes *overlay* destinadas a *streaming* [7], partilha de ficheiros [8], áudio [9], entre outros. Existem inúmeros modelos de arquiteturas implementadas para esta finalidade, assim como distintos tipos de sistemas de rede [10], que serão uma grande ajuda para promover este projeto, de forma escalável e organizada. No entanto, e contrariamente a outras soluções existentes, o foco principal deste projeto é o desenvolvimento

de mecanismos de controlo de tráfego decididos pelo *ISP*. Deste modo, os utilizadores têm acesso a uma plataforma básica de partilha de conteúdos, onde podem efetuar *downloads*, fazer *uploads* de conteúdos, visualizar os ficheiros na rede *overlay*, entre outras funcionalidades, enquanto que para o *ISP* será disponibilizada uma plataforma de administração da rede *overlay* com vários mecanismos de controlo, com vista a controlar o tráfego *P2P* do sistema. O foco principal deste trabalho baseia-se no estudo de mecanismos de encaminhamento aplicacional a utilizar, no planeamento de decisões de controlo de tráfego pelo *ISP*, na possibilidade de limitar a circulação de ficheiros no sistema de rede e na versatilidade da rede utilizar um modo de transferência (cliente-servidor e/ou *P2P*) que se adeque aos propósitos dos pedidos de *download* de cada utilizador.

1.2 OBJETIVOS

O objetivo desta dissertação é a implementação de uma rede *overlay* de partilha de conteúdos, disponibilizada pelo *ISP* aos utilizadores, de modo a evitar que estes usem aplicações *P2P* proprietárias que fujam ao controlo do administrador de rede. Este sistema de rede, visa facilitar a experiência de partilha de conteúdos pelos utilizadores que usem a rede, de forma simples e escalável. Além disso, visa disponibilizar ao *ISP* uma *interface* de administração da rede *overlay*, de modo a que o mesmo consiga beneficiar de mecanismos de controlo sobre como o tráfego *P2P* atravessa a rede física. Para atingir este objetivo, foram definidos alguns pontos essenciais para o correto desenvolvimento deste projeto:

- Investigação preliminar à área das redes *overlay* e fundamentalmente nas áreas relacionadas com a distribuição de conteúdos;
- Definição da arquitetura do sistema de rede, assim como as entidades envolvidas e as regras/mecanismos da rede *overlay* a serem desenvolvidos;
- Implementação de um protótipo da rede *overlay* de partilha de conteúdos e implementação de duas *interfaces* gráficas, uma dirigida para os clientes e outra para o administrador;
- Demonstração das funcionalidades do protótipo definidas em ambiente de testes, com a utilização de um emulador de rede.

1.3 ORGANIZAÇÃO DA DISSERTAÇÃO

A presente dissertação está dividida em 6 capítulos:

- **Introdução:** enquadramento e contextualização do trabalho, assim como os seus principais objetivos e organização;

- **Estado da Arte:** compreende a revisão bibliográfica das teorias relativas a esta dissertação, onde serão explicitados os conceitos e funções das redes *overlay* num caso genérico e posteriormente para o caso particular da aplicação das mesmas sob a prática de partilha de conteúdos, essenciais para o desenvolvimento da dissertação;

- **Arquitetura e Mecanismos do Sistema Proposto:** especificação geral da arquitetura do sistema de rede, onde são definidos teoricamente as suas entidades e funcionalidades a serem desenvolvidas;

- **Implementação:** implementação do protótipo, onde são apresentados as entidades realmente implementadas, assim como tecnologias utilizadas e *interfaces* gráficas para o utilizador/*ISP*;

- **Testes e Análise de Resultados:** realização de testes com a utilização de uma ferramenta de emulação de rede, com o objetivo de validar a rede *overlay* implementada, assim como as suas funcionalidades e arquitetura;

- **Conclusões:** apresentação das principais conclusões do trabalho desenvolvido, assim como futuras oportunidades de melhorias que poderão, eventualmente, vir a ser implementadas.

2

ESTADO DA ARTE

Este capítulo é estruturado em quatro secções que visam apresentar detalhadamente as redes *overlay*, com o fim de retirar informação útil para a produção da dissertação. A primeira, secção 2.1, apresenta o conceito de rede *overlay*, assim como detalhes sobre as suas funcionalidades, problemas de segurança e consciencialização da rede *underlay* de um sistema *overlay*. Na secção 2.2, é efetuado um estudo sobre a interação entre sistemas *overlay*/*P2P* e o *ISP*, sendo também analisados mecanismos de colaboração mútua existentes. Por fim, na secção 2.3 são estudados dois tipos de sistemas de entrega de conteúdos, sendo posteriormente efetuada a comparação entre estes, e os sistemas *P2P* tradicionais. Finalmente, na secção 2.4 é apresentado um sumário sobre o capítulo.

2.1 REDES OVERLAY/P2P

Esta secção aborda, de modo geral, as redes *overlay*/*P2P* de forma a conhecer a estrutura das mesmas, assim como as suas funcionalidades, problemas e desafios inerentes. Na subsecção 2.1.1, é efetuada uma introdução às redes *overlay* e, posteriormente, na subsecção 2.1.2 é explicitada a sua estruturação. Por fim, na subsecção 2.1.3, é efetuada a análise dos problemas e desafios existentes na implementação deste tipo de redes.

2.1.1 *Contextualização*

O conceito de redes *overlay*, designa a criação de uma rede virtual inserida sob uma rede física já existente [11]. Neste tipo de rede, os nós são conectados por ligações virtuais, que correspondem a caminhos na rede subjacente, Figura 2.1. Além disso, a rede *overlay* pode ser implementada em qualquer um dos níveis superiores à camada física, correspondente ao modelo Open System Interconnection (OSI). Porém, a rede de sobreposição pertinente para o estudo desta dissertação, situa-se na camada aplicacional, onde normalmente toma o nome de rede *overlay P2P*.

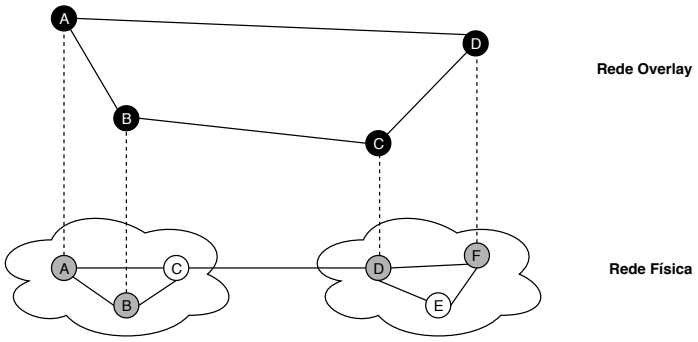


Figura 2.1: Topologia de uma rede física em relação à topologia de uma rede overlay

Este conceito traz benefícios vantajosos, relativamente a outro tipo de redes, como o baixo custo de implementação e escalabilidade do sistema. Estas propriedades fazem com que este conceito seja ideal para a construção de sistemas de redes de entrega de conteúdos, visto que as suas propriedades permitem a consistência e fiabilidade de um sistema de entrega de dados [12, 13].

As redes *overlay* podem ser classificadas, sendo divididas em categorias com diferentes estruturas, como será explicitado posteriormente. São bastante úteis para criar meios de organização automática e balanceamento de carga para determinadas redes físicas, como por exemplo os sistemas *Tapestry*, *Chord*, *Pastry*, etc. Além disso, as redes *overlay* também podem ser utilizadas em outras vertentes, como por exemplo, em sistemas de entrega de conteúdos (*BitTorrent*) sistemas de pesquisa descentralizados, (*Gnutella*), sistemas de armazenamento de conteúdos, *Freenet*, entre outros [14].

2.1.2 *Estruturação das Redes Overlay*

As redes *overlay* são divididas em redes estruturadas e não-estruturadas, dependendo das propriedades atribuídas à sua topologia. Esta classificação, foi elaborada para distinguir inúmeros sistemas deste tipo, dependendo da existência de entidades administradoras, e de fatores de armazenamento de estrutura do sistema em si. Além disso, também permite a criação de uma base de comparação entre estes sistemas e obtenção de uma informação da topologia mais detalhada, assim como as suas possíveis funcionalidades.

2.1.2.1 *Redes Overlay Estruturadas*

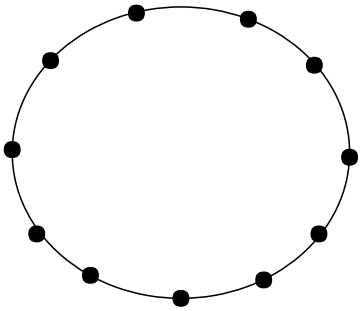


Figura 2.2: Topologia de uma rede overlay estruturada

As redes *overlay* estruturadas, tal como o nome indica, são sistemas capazes de identificar a sua topologia, na qual, o fornecedor de rede consegue conjugar os seus nós de modo organizado, Figura 2.2. O conceito mais importante nas redes *overlay* estruturadas, é a utilização de uma *Distributed Hash Table* (Distributed Hash Table (DHT)). A *DHT* retrata um sistema distribuído, que tem como função providenciar um serviço de pesquisa baseado numa tabela de *hash*, com atributos *"chave, valor"*. Este conceito, utiliza como estrutura auxiliar um grafo composto pelos nós da topologia, permitindo assim a descoberta rápida de qualquer nó da rede virtual. Para este tipo de redes é importante a definição de quatro conceitos:

- **Chave:** conjunto dos nós/*peers* da topologia, normalmente definidos pelos seus endereços de *IP* ou chaves públicas.
- **Valor:** conteúdo de cada nó/*peer*. Normalmente o seu conteúdo são ficheiros, pastas, *raw data*, etc.
- ***Item:*** par chave-valor armazenado da tabela de *hash*.
- **Espaço de Identificação:** é um conceito presente em todas as *DHTs* e refere-se à gama de valores que uma chave pode ter. É aplicado aos endereços *IP* relativos aos nós da topologia de uma função de *hash*, de modo a serem identificados através de um conjunto de valores dependentes do tipo da função utilizada.

O termo de redes *overlay* estruturadas, surgiu com o mapeamento da topologia de rede numa *DHT*, que associa a identificação dos nós na tabela de *hash*, com o seu conteúdo, sendo esta informação armazenado em um grafo. Este método garante pesquisas eficientes, que tornam a procura de um nó mais rápida, comparativamente às redes *overlay* não-estruturadas. Além disso, esta é bastante útil, pois além de facilitar a pesquisa de nós, permite tolerância a faltas e balanço de carga [15]. Um exemplo de uma aplicação que utiliza este conceito pode ser observado na Figura 2.3.

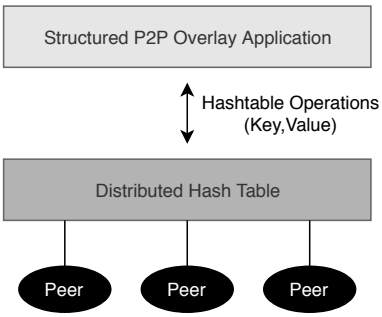


Figura 2.3: Interface de uma aplicação para redes overlay P2P estruturadas

A principal desvantagem das redes *overlay* estruturadas são os custos de manutenção sob a própria topologia, visto que ao utilizar uma *DHT*, existe um *trade-off* entre o estado da topologia e o seu desempenho. Além disso, existe sempre o problema de segurança da topologia, fiabilidade, anonimidade, comportamento malicioso por parte dos nós, etc.

Como exemplos de abordagens de redes *overlay* estruturadas, podem ser destacadas:

- **Content Addressable Network (CAN):** representa uma infraestrutura de indexação distribuída dedicada a sistemas *P2P* [16], sendo que providencia funcionalidades semelhantes a tabelas de *hash* (mapeamento chaves-valores) em sistemas *Internet* de larga escala.
- **Chord:** é uma infraestrutura desenhada para disponibilizar um sistema de pesquisa de nós eficiente, a partir da técnica de *consistent hashing* para atribuir chaves aos *peers* da infraestrutura [17].
- **Pastry:** designa uma infraestrutura escalável, com técnicas de *distributed object location* para aplicações *P2P*. Esta, realiza encaminhamento ao nível da aplicação e efetua a localização de objetos em redes *overlay* de larga escala [18].

2.1.2.2 *Redes Overlay Não-Estruturadas*

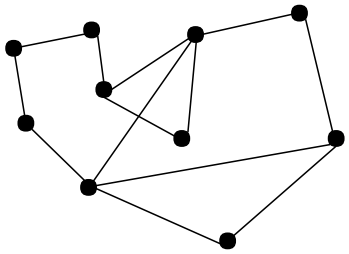


Figura 2.4: Exemplo de uma topologia de rede overlay não-estruturada

As redes *overlay* não-estruturadas, ao contrário das redes abordadas anteriormente, não dependem de qualquer invariante, sendo que a organização da topologia de rede é completamente aleatória, Figura 2.4. Embora este conceito pareça contraditório relativamente

às funções de organização das redes *overlay*, na verdade é bastante escalável e produz resultados eficientes no que diz respeito ao desempenho de serviços e aplicações executados num nível superior. Este conceito é muito utilizado em sistemas *P2P* modernos, visto que conseguem aproveitar a aleatoriedade na construção da topologia da rede para reduzir os custos de produção e manutenção da própria. Além disso, geralmente, são mais tolerantes e resilientes em termos de falha de nós e perda de mensagens, contrariamente às redes *overlay* estruturadas [19].

A maior desvantagem destas redes é a pesquisa pouco eficiente de nós, visto que não existe uma topologia organizada e armazenada em uma *DHT*, embora possam tentar conseguilo através de protocolos *gossip* [20]. Além desta grande desvantagem, não existe uma resposta a comportamentos maliciosos por parte dos nós, confiança no sistema e consequentemente a segurança do mesmo.

Como exemplos de abordagens de redes *overlay* não-estruturadas, podem ser destacadas:

- **Freenet:** designa uma aplicação de rede *P2P* que permite a publicação, replicação e recuperação de dados, de modo anónimo para os autores e utilizadores [21].
- **FastTrack:** trata-se de um sistema *P2P* descentralizado de partilha de ficheiros que suporta pesquisas de metadados [22]
- **BitTorrent:** é um sistema *P2P* centralizado de distribuição de ficheiros, que utiliza uma entidade central para gerir as transferências dos utilizadores [8].

2.1.3 *Problemas e Desafios*

Com o crescimento da *Internet* e com a criação de novos serviços, existem algumas preocupações nos sistemas de redes *overlay P2P*. Assim, existe uma grande preocupação com a manutenção do sistema e proteção contra ataques maliciosos, além da tentativa de colaboração com o *ISP* para se obter uma maior eficiência do serviço. Esta eficiência obtém-se através da aprendizagem do sistema físico adjacente da rede virtual. Como objetivo desta subsecção, pretende-se evidenciar vários aspetos considerados críticos para qualquer fornecedor de rede. Vão ser expostos e demonstrados posteriormente, possíveis vulnerabilidades apresentadas por sistemas *overlay* e os impactos obtidos a partir da consciencialização deste tipo de sistemas com a sua rede *underlay*.

2.1.3.1 *Vulnerabilidades do Sistema*

Os sistemas *overlay*/*P2P* foram desenhados de modo a evitar que todo o sistema desligue, quando vítimas de ataques efetuados em pontos singulares, que podem ser bastante vulneráveis em arquiteturas cliente-servidor [23, 24]. As principais razões para a criação

destes sistemas são o seu poder de distribuição e a sua escalabilidade. No entanto, essas duas características introduziram também um novo conjunto de vulnerabilidades que visam ameaçar o comportamento do sistema. A proteção destas estruturas torna-se complexa, visto que um ataque a um dos nós pertencentes, pode ser crucial e afetar o resto da rede. Neste sentido, com a expansão deste tipo de sistemas para a *Internet*, podem vir a existir nós não confiáveis, que podem começar uma ameaça em larga escala.

A Figura 2.5 representa uma noção do conjunto de possíveis ameaças em torno das diferentes camadas de rede, segundo o modelo *OSI*, sendo que podem verificar-se distintos tipos de ataque, desde as camadas inferiores até às camadas superiores.

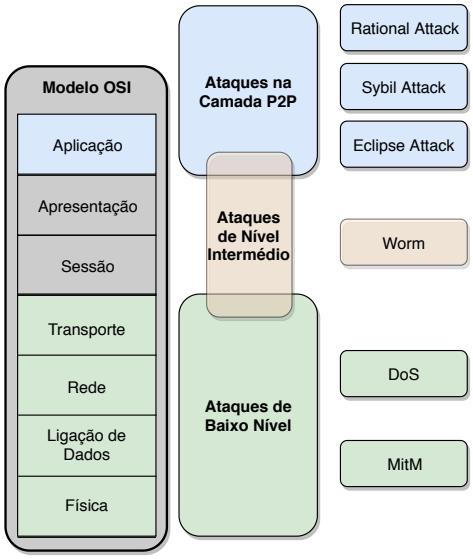


Figura 2.5: Representação do tipo de vulnerabilidades em sistemas de redes com base no modelo OSI

2.1.3.2 *Consciencialização da Rede Underlay de um Sistema P2P*

Com o desenvolvimento da tecnologia e da necessidade humana, cada vez mais surgem novos tipos de aplicações *P2P*, que atraem inúmeros utilizadores. Consequentemente, os sistemas *P2P* tradicionais tornam-se ineficientes, visto que estão implementados de forma a garantir o bom funcionamento do serviço e satisfazer os clientes, não conseguindo tirar proveito da informação provida pela rede *underlay* (física). Neste segmento vai-se discutir a importância da colaboração do sistema *P2P* com o *ISP* [6, 25, 26].

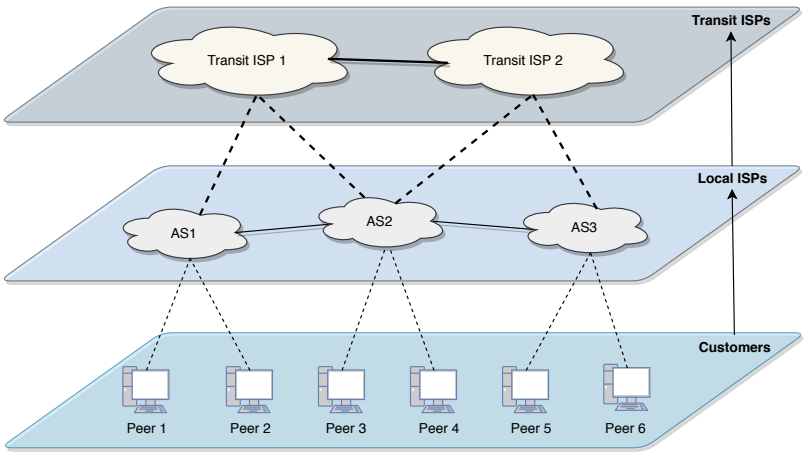


Figura 2.6: Exemplo de hierarquia de comunicação em sistemas P2P. As setas sólidas indicam fluxo monetário, e as linhas tracejadas e sólidas entre ISPs indicam conexões de trânsito e P2P, respectivamente. (adaptação de [1]

Tal como foi discutido anteriormente, os sistemas *P2P* ficaram famosos pela sua eficiência e baixo custo, sendo utilizados em diversos tipos de serviços *online* e, consequentemente produzindo uma elevada circulação de tráfego nas redes administradas pelos *ISPs*, provocando-lhes grandes problemas administrativos, dada também a falta de soluções referentes a este desafio. Grande parte destes serviços transferem quantidades massivas de conteúdos globalmente, o que provoca uma sobrecarga de tráfego *inter*-Autonomous System (AS), criando grandes custos para os *ISPs*. A Figura 2.6, define a hierarquia de comunicação existente em sistemas *P2P*. Nesta, podemos verificar o percurso da comunicação desde um consumidor até ao *transit ISP*, distinguindo as diferentes entidades que existem neste sistema. Os custos monetários provenientes existem entre a camada do consumidor e a camada do *AS*, e entre a camada do *AS* e a do *Transit ISP*, sendo que nesta última os custos são mais elevados. Neste âmbito, um importante conceito na colaboração dos sistemas *P2P* e o *ISP* é a consciêncialização da rede física (underlay). Este conceito é definido como a base física onde a rede *overlay* reside, abstraindo as quatro primeiras camadas do modelo *OSI*, ou seja, a camada física, ligação de dados, rede e transporte. Porém, quando se fala em consciencialização da rede física, os parâmetros importantes que influenciam o desempenho em sistemas *P2P* são:

- **ISP-Location awareness:** identificação do *ISP* através de uma conexão de um *peer* à *Internet*. Esta consciencialização para encontrar o *ISP* local é bastante importante, devido à enorme quantidade de tráfego *P2P* que atravessa os *ISPs* para a *Internet* em geral. Este parâmetro é bastante relevante, quer para o *ISP* manter o tráfego localmente e reduzir os custos *inter-AS*, quer para o consumidor ter uma melhor experiência na utilização de serviços.

- **Latency awareness:** A importância deste parâmetro reside no facto de que o atraso na transferência de informações pode prejudicar a experiência do utilizador e degradar a Quality of Service (QoS), especialmente para aplicações interativas. A relação entre a latência da rede e o seu *QoS* é inversamente proporcional. Em função do âmbito do serviço de um sistema *P2P*, existem certos intervalos aceitáveis de latência distintos, dependendo da importância do atraso dos pacotes que chegam ao destino. Na geração atual, o controlo da latência é muito importante, especialmente para as redes de partilha de conteúdos multimédia e ainda mais em serviços de comunicação em tempo real.

- **Peer resources awareness:** A consciencialização dos recursos dos *peers*, visa identificar as capacidades de cada *peer* (poder de processamento, largura de banda, espaço em disco), de modo a melhorar o desempenho geral do sistema. A partir desta consciencialização, a rede *overlay* pode ser distribuída, de forma a atribuir as tarefas apropriadas de acordo com a capacidade de cada *peer*, de modo a tornar o sistema mais estável.

- **Geolocation awareness:** Identificação da localização geográfica de um determinado nó da rede. O conceito de geolocalização também pode ser usado para calcular as distâncias geográficas entre dois nós da rede. A distância geográfica pode ser relacionada com o atraso de transmissão entre nós, porém, apenas no caso dos nós se situarem no mesmo *AS*, partilhando o mesmo *ISP*. Este parâmetro em conjunto com a consciencialização do *ISP*, permite ao sistema uma maior noção geográfica, permitindo oferecer às aplicações maior escalabilidade a partir do controlo de congestionamento e localização do tráfego.

A partir destes parâmetros consegue-se estabelecer uma percepção dos conhecimentos ideais a reter, para se implementar um sistema *P2P* estável. Na Tabela 2.1, são representados os impactos criados num sistema deste género, e a grande influência que a consciencialização abordada provoca. Além disso, foram definidos alguns parâmetros para comparar as duas entidades separadamente (utilizadores e *ISPs*) e posteriormente em conjunto, com os respetivos parâmetros de maior influência na utilização do sistema para cada entidade (tempo de download, atraso, *ISP* Operations, Administration and Maintenance (OAM), custos *ISP*, novas áreas de aplicação e resiliência).

Em termos dos benefícios que a consciencialização da rede *underlay* fornece aos sistemas *P2P*, e à Internet em si, podemos dividi-los em quatro categorias, em conjunto com o que foi referido na Tabela 2.1.

Em primeiro lugar, quanto ao efeito desta consciencialização aplicado aos utilizadores do sistema, pode-se prever um melhor *QoS*, ou seja, menores atrasos, sucesso das funcionalidades do sistema, melhor fluidez da aplicação, etc. Porém, para isto acontecer, os interesses do sistema *P2P* têm que ir de encontro com os interesses dos *ISPs*. Desta forma, os utiliza-

		Parâmetros de Consciencialização			
Impactos em		ISP-Location	Latency	Peer Resources	Geolocation
Utilizadores	Tempo de Download	Grande Efeito	Neutro	Grande Efeito	Neutro
	Delay	Neutro	Grande Efeito	Neutro	Pequeno Efeito
ISPs	ISP OAM	Grande Efeito	Neutro	Neutro	Neutro
	Custos ISP	Grande Efeito	Neutro	Pequeno Efeito	Neutro
Ambos	Novas Áreas de Aplicação	Neutro	Pequeno Efeito	Grande Efeito	Neutro
	Resiliência	Grande Efeito	Grande Efeito	Pequeno Efeito	Neutro

Tabela 2.1: Tabela de impactos num sistema P2P com consciencialização da rede underlay (adaptação de [6])

dores têm que confiar nos seus *ISPs* para lhes fornecerem informações privadas ao aceder ao serviço e o sistema tem que garantir a fiabilidade e legalidade do seu conteúdo, principalmente no caso de sistemas P2P para partilha de conteúdos. Em segundo lugar, focando agora o efeito da consciencialização da *underlay* criado nos *ISPs*, é de se compreender que estes estão mais interessados no facto de reduzir os custos *Inter-ISP*. Se os fornecedores de *Internet* tiverem consciência da atividade e do fluxo de dados *P2P*, assim como a sua localização geográfica, conseguem economizar bastantes recursos e melhorar a gestão interna da rede. Com base no que foi referido anteriormente, visto que os *ISPs* ambicionam uma condição vantajosa, podem ajustar-se a determinados sistemas *P2P* e melhorar o *QoS* das suas aplicações a partir de incentivos monetários por parte dos fornecedores da rede *overlay*.

Posteriormente, também é possível analisar um efeito considerável nos parâmetros de consciencialização abordados em relação aos fornecedores do sistema *P2P*. Com a consciencialização da rede *underlay* e cooperação com o *ISP* e os fornecedores do sistema, é possível a criação de uma condição vantajosa para ambas as entidades. Porém, devido à dificuldade dos sistemas *P2P* monitorizarem a legalidade dos conteúdos dentro da sua rede, esta ainda é uma tarefa bastante desafiadora para ambos.

Por fim, é possível apreender que a consciencialização da rede *underlay* de sistemas *overlay/P2P*, pode beneficiar mutuamente, os próprios sistemas *overlay/P2P* e os *ISPs*. Em adição, se os desafios e dificuldades de colaboração entre os fornecedores de rede e os fornecedores dos sistemas *overlay/P2P* forem superados, seria possível um melhor balanceamento de tráfego nas redes físicas, assim como possivelmente uma melhor qualidade de serviço, dependendo das prioridades anexadas aos distintos tipos de tráfego em circulação.

2.2 Estudo da Interação entre Redes Overlay/P2P e o ISP

Após ter sido explicitado o conceito de redes *overlay/P2P*, assim como as suas funcionalidades e problemas, esta secção visa explicar as dificuldades que os sistemas *P2P* causam às redes dos *ISPs*, assim como referir alguns desafios e respetivas soluções que permitam a coexistência de ambos. Inicialmente, na subsecção 2.2.1, é efetuada uma introdução geral

que visa detalhar o estudo da interação entre as entidades anteriormente abordadas. Posteriormente, são analisados os principais desafios na implementação de métodos de interação com o *ISP* (subsecção 2.2.2). De seguida, na subsecção 2.2.3, é efetuada a classificação da interação entre os sistemas *overlay/P2P* e o *ISP*, assim como a enumeração de abordagens que englobal cada uma delas. Por fim, na subsecção 2.2.4, irão ser explorados mecanismos de colaboração mútua entre sistemas *overlay/P2P* e o *ISP*, existentes na atualidade.

2.2.1 *Contextualização*

Como já foi referido anteriormente, os sistemas *P2P* ganharam muita atenção dos utilizadores da *Internet* e, consequentemente dos motores de pesquisa. O recurso a estes sistemas por parte dos utilizadores, inclui métodos de partilha de conteúdos, no entanto, é de ressalvar, que apenas lhes é concebida a permissão de efetuarem cópias e/ou impressão dos recursos, gratuitamente, se e só se a sua utilização não estiver associada a fins lucrativos. Posto isto, devido ao seu amplo uso, impõe-se um dilema aos *ISPs*. Se por um lado, as aplicações do sistema *P2P* resultam num aumento da receita dos *ISPs*, por serem uma das principais razões apontadas pelos usuários da *Internet* para atualizar o seu acesso à banda larga, por outro, os *ISPs* vêem o tráfego *P2P* como um desafio que afeta a qualidade das suas redes. Uma explicação para este facto, deve-se á invasão de tráfego *P2P* que se realiza em inúmeros pontos da rede física dos *ISPs*, utilizando uma quantidade massiva de largura de banda, o que torna a administração da rede física inviável. Este fenómeno ocorre, porque a maioria dos sistemas *P2P* depende do encaminhamento da camada aplicacional com base numa topologia de *overlay* na *Internet*, o que é independente do encaminhamento e da topologia da *Internet*. Em termos práticos, para criar uma topologia de rede *overlay*, as redes *P2P*, usam um procedimento arbitrário de seleção de vizinhos. No entanto, este procedimento, demonstra que o tráfego *P2P* atravessa os limites da rede do *ISP* inúmeras vezes, o que não é necessariamente o ideal, pois, não existe um aproveitamento geográfico na seleção dos vizinhos. Além disso, vários estudos têm vindo a demonstrar que o conteúdo desejado se encontra na proximidade dos utilizadores interessados [27, 28]. Facto, que ocorre, devido ao idioma do conteúdo e às regiões geográficas de interesse, ou seja, como um utilizador do serviço *P2P* está essencialmente interessado em encontrar de forma rápida o conteúdo desejado e, com bom desempenho, acredita-se que o aumento da localidade do tráfego *P2P* beneficiará os *ISPs* e os utilizadores *P2P* [29].

2.2.2 *Desafios*

A disputa entre sistemas *overlay/P2P* e *ISPs* sobre o crescimento explosivo do tráfego *P2P* na *Internet*, é originada pela falta de consciencialização existente entre a rede *overlay* e *un-*

derlay, na qual os sistemas *P2P* formam implicitamente/explicitamente topologias de rede *overlay* da camada aplicacional a partir dos *peers* participantes [30]. Isto, pois o encaminhamento aplicacional e o processo de seleção de vizinhos efetuado por este tipo de sistemas, é independente da topologia física do *ISP* e, por conseguinte, dos custos dos *links* da sua rede. Embora esta flexibilidade abordada seja um dos principais fatores que garantem a robustez e escalabilidade do paradigma *P2P*, também traz uma série de desafios e questões, relativamente ao tráfego *P2P* que circula nas redes dos *ISPs* e ao desempenho da rede *overlay*. Posto isto, a partir do que se referiu, e de modo a complementar o que foi analisado na secção anterior, foram retirados alguns dos principais desafios colocados entre sistemas *overlay/P2P* e *ISPs* [30, 2]:

- **Poupança de custos:** muitas das aplicações *P2P* geram elevados volumes de tráfego entre domínios de rede, criando custos adicionais ao *ISP*.
- **Dependência funcional:** mesmo que exista uma colaboração entre sistemas *overlay/P2P* e o *ISP*, a *QoS* dos serviços oferecidos por este tipo de sistemas, está dependente da condição dos *links* da rede física dos *ISPs*.
- **Utilização ineficiente dos recursos de rede:** normalmente, neste tipo de sistemas existe uma utilização ineficiente dos recursos de rede, o que provoca consequentemente, uma degradação de desempenho e qualidade de serviço.
- **Padrões dinâmicos de distribuição de tráfego:** os sistemas *overlay/P2P* têm padrões dinâmicos de distribuição de tráfego, complicando os processos de engenharia de tráfego na rede física.
- **Aplicações não regulamentadas:** sistemas *overlay/P2P* não regulamentados, geralmente, provocam um aumento significante de largura de banda da rede e por conseguinte um congestionamento na rede *backbone*. Como consequência do consumo de largura de banda *P2P*, outras aplicações da *Internet* podem ser bastante afetadas, chegando a ser interrompidas [31].
- **Conteúdo ilegal:** Podem surgir questões jurídicas quando um *ISP* ajuda na distribuição ilegal de conteúdos protegidos por direitos de autor.
- **Intercâmbio de dados:** os dados transmitidos por serviços de sistemas *overlay/P2P*, devem seguir uma semântica acordada com o *ISP*, de modo a que a troca de informação siga protocolos escaláveis. Estes, devem garantir métodos de autenticação, integridade e devem ser seguros.
- **Comportamentos não cooperativos:** o *ISP* ou o sistema *overlay/P2P*, pode tentar explorar a informação fornecida pela outra entidade, sem fornecer qualquer informação em troca.

2.2.3 *Classificação da Interação de Sistemas Overlay/P2P com o ISP*

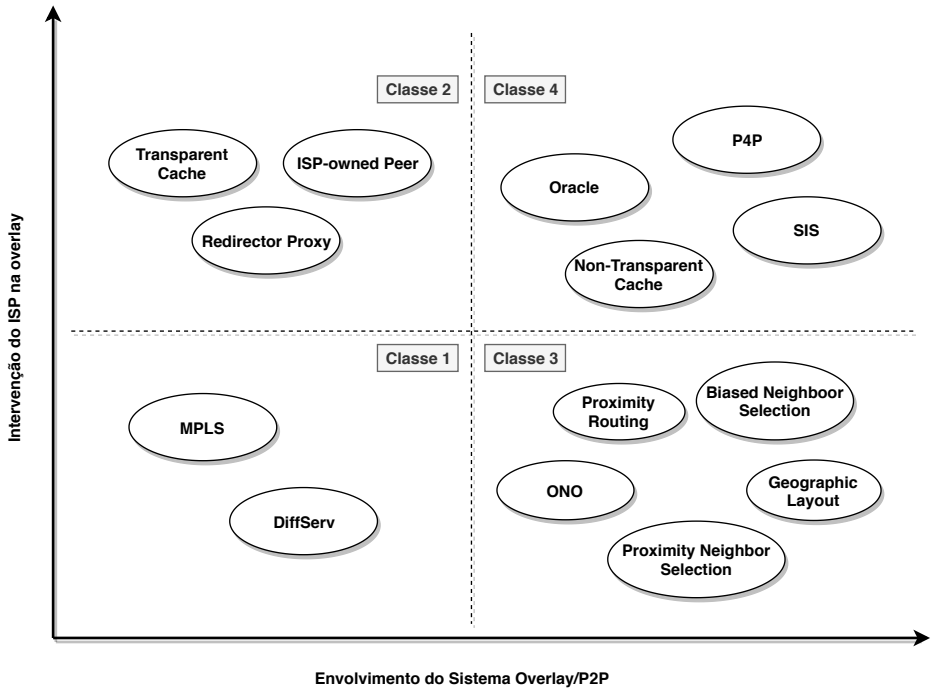


Figura 2.7: Exemplos de abordagens relacionadas com a classificação da interação entre sistemas overlay/P2P e o ISP (adaptação de [2])

Com base no artigo [2], efetuou-se uma classificação da interação entre sistemas *overlay/P2P* e o *ISP*, dependendo de duas dimensões: o envolvimento do sistema de rede *overlay* e do *ISP*. Tendo como base as dimensões abordadas, a classificação pode ser dividida em quatro classes, como demonstra a Figura 2.7.

A classe 1, diz respeito à influência indireta do *ISP* na rede *overlay*. Assim, o sistema de rede *overlay* pode ser influenciado por decisões de engenharia de tráfego que o *ISP* aplique na rede física. Neste caso, o fornecedor de rede, pode optar por utilizar mecanismos de tratamento e priorização de tráfego, de modo a optimizar o *QoS* da rede física e a utilização de recursos na sua rede. Deste modo, o fornecedor de rede pode utilizar métodos como o Differentiated Services (DiffServ) [32], que lhe permite efetuar a negociação dos distintos tipos de tráfego que percorrem a rede física, além de facilitar a classificação e administração do tráfego que percorre a sua rede. Por outro lado, o *ISP* pode também recorrer a mecanismos como o Multi Protocol Label Switching (MPLS) [33], de modo a etiquetar o tráfego que percorre a rede física e/ou associar-lhe requisitos de *QoS*, além de melhorar o desempenho do encaminhamento de pacotes.

As interações pertencentes à classe 2, relacionam-se com a influência direta do *ISP* e da rede *overlay*. Deste modo, o *ISP* executa operações que influenciam diretamente a rede *overlay*, de maneira a que os *peers* da mesma, não estejam cientes do seu envolvimento. Neste

cenário, o *ISP* pretende melhorar a eficiência do encaminhamento *P2P*, em termos de utilização de recursos da rede, com o propósito de diminuir os próprios custos existentes, sem tentar influenciar o *QoS* do sistema *overlay/P2P*. Posto isto, o *ISP* pode efetuar abordagens, tais como: utilização de um servidor *proxy* com o objetivo de redirecionar tráfego *P2P*, utilizar métodos de *transparent caching*, para fins de optimização do uso de recursos na rede do fornecedor de rede e do sistema de rede *overlay*, e por último, utilizar um *peer* proprietário do *ISP*, encarregue de administrar diretamente a rede *overlay*.

Contrariamente às classes anteriormente referidas, a classe 3, engloba o envolvimento unilateral do sistema *overlay/P2P*. Nesta, são efetuadas alterações ao protocolo de encaminhamento *P2P* do sistema, de forma a optimizar a *rede overlay*, tendo em consideração a rede *underlay*, sem ser necessário o envolvimento do *ISP*. Porém, neste cenário é difícil prever os ganhos que o *ISP* pode reter. Como exemplos de abordagens pertencentes a esta classe, existem algoritmos baseados na proximidade dos *peers* da rede *overlay*, que são calculados sem a ajuda do *ISP*, a partir de medições efetuadas pelos próprios *peers*, tais como: *proximity routing*, *biased neighboor selection*, *proximity neighboor selection*. Além destes, também existe um serviço de *software*, que oferece métodos eficientes de identificação de *peers* vizinhos, denominado *ONO* [34].

Por fim, interações da classe 4, englobam influência direta mútua, entre as duas entidades anteriormente referidas. Assim, estabelece-se uma colaboração mútua, entre a rede física do *ISP* e o sistema *overlay/P2P*, para que esta interação se reflita num aumento do desempenho de ambos. Neste cenário, o *ISP* opera uma infraestrutura que disponibiliza informação ao sistema *overlay/P2P*, resultando numa condição onde ambas as entidades ganham. Neste caso, o sistema *overlay* tem que ser modificado, de forma a conseguir colaborar com a infraestrutura do fornecedor de rede. Esta classe é a que mais se aproxima com o objetivo da dissertação, sendo que vão ser descritas seguidamente algumas abordagens com maior detalhe, como o *Oracle*, o SmoothIT Information Service (SIS) e o Provider Portal for Applications (P4P).

2.2.4 *Abordagens de Colaboração Mútua Existentes*

Neste tópico vão ser apresentadas soluções de colaboração mútua entre sistemas *overlay/P2P* e o *ISP*. Especificamente, como referido na subsecção anterior, vão ser detalhadas as abordagens pertencentes à classe 4, nomeadamente, o *SIS*, o *Oracle* e o *P4P*.

2.2.4.1 *SIS*

O *SIS* [3] representa um serviço de comunicação entre o *ISP* e aplicações *overlay/P2P*, disponibilizado a partir de uma arquitetura denominada *SmoothIT*. Este, inclui *interfaces* de comunicação bidirecional entre *ISPs* e aplicações deste género, além de possuir meios de

comunicação dedicados a distintos *ISPs*. O *SIS* pode fornecer informações sobre políticas de tráfego, localização, congestionamento, tarifação e *QoS*, a fim de ajudar as aplicações *overlay/P2P* a decidir como construir e manter a rede *overlay*.

A Figura 2.8 ilustra os componentes funcionais da arquitetura *SmoothIT*. O servidor *SIS* fornece o serviço *SIS* ao cliente, sendo o componente central da arquitetura. Este, tem de ser implantado em cada *ISP*, de modo a fornecer o serviço dentro da rede de cada um. As funções primárias de cada componente, são:

- **Componente administrativo:** permite ao *ISP* configurar parâmetros internos do *SIS* através de uma interface administrativa. Podem ser definidos parâmetros de classes de serviços de *QoS*, número de fluxos máximos que podem ser priorizados, entre outros.

- **Componente de medição:** responsável pela realização de medições na rede. Este, pode ser composto por um ou mais módulos que especificam diferentes medições a realizar.

- **Componente de segurança:** responsável por fornecer métodos de autenticação e autorização para o cliente *SIS*, o administrador e outros servidores *SIS*. Adicionalmente, a confidencialidade e integridade dos dados são asseguradas por canais de comunicação seguros entre os componentes.

- **Componente de configuração de bases de dados:** armazena as políticas dos fornecedores de rede e é responsável por qualquer informação que um *ISP* possa configurar para a arquitetura *SIS*. Além disso, dispõe de informação sobre diferentes tipos de relações comerciais entre *ISPs* e métricas relacionadas com o desempenho correspondente de cada rede.

- **Componente de QoS:** verifica a disponibilidade dos recursos da rede e garante que estes sejam solicitados pelas aplicações *overlay*, bem como a aplicação de políticas de *QoS* na rede.

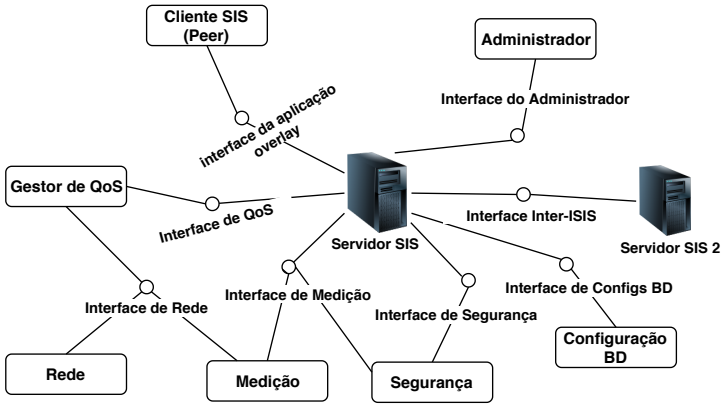


Figura 2.8: Arquitetura SmoothIT (adaptação de [3])

2.2.4.2 *Oracle*

O *Oracle* [35] é uma solução de colaboração entre sistemas *overlay/P2P* e o *ISP* que consiste na oferta de uma entidade colaborativa pertencente ao *ISP*, destinada para os utilizadores do sistema (*Oracle*). Assim, quando um utilizador do sistema fornece a esta entidade uma lista de possíveis vizinhos, esta classifica-os de acordo com certos critérios, tais como, fatores de proximidades ou *links* de maior largura de banda. Deste modo, esta solução tem como finalidade melhorar o desempenho do serviço dos sistemas *overlay*, em conjunto com, a administração de tráfego *P2P* efetuada pelo *ISP*. O objetivo desta abordagem consiste numa condição de ganho mútuo, de forma a que os recursos da rede sejam aproveitados. Uma melhor utilização da rede, irá por conseguinte, garantir ao sistema *overlay/P2P* e ao *ISP*, a oferta de um serviço com maior nível de *QoS* para cada um dos seus clientes.

Com base no descrito, em vez de um sistema *overlay/P2P* efetuar a seleção de vizinhos independentemente do *ISP*, este serviço denominado *Oracle*, oferece-lhe um serviço de *ranking* de vizinhos, de modo a melhorar o desempenho dos serviços disponibilizados por ambas as entidades. As métricas utilizadas no *ranking*, podem ser vistas como as preferências do *ISP* por determinados vizinhos *P2P*. As métricas de distância desta solução, enquadradas com base numa granularidade grossa, são: dentro/fora do *AS*, número de saltos entre *AS* (de acordo com o protocolo Border Gateway Protocol (BGP)) e distância à extremidade de *AS* (de acordo com o protocolo Interior Gateway Protocol (IGP)). Para os nós pertencentes a um determinado *AS*, o *Oracle* pode ainda definir a seguinte classificação: informação geográfica, (cidade ou outro ponto de referência), informações sobre o desempenho (como o atraso esperado, largura de banda) e congestão de *links* (engenharia de tráfego). Esta classificação pode ser utilizada pelos *peers* do sistema para selecionar um vizinho próximo, embora seja opcional.

Deste modo, existem múltiplos benefícios para os *peers* dos sistemas de rede *overlay*: não têm que medir o desempenho do caminho eles mesmos, podem tirar proveito do conheci-

mento do *ISP* e podem esperar melhor desempenho, pois existe uma diminuição de latência, assim como um maior aproveitamente dos recursos de rede.

Adicionalmente, o benefício para os *ISPs* é a capacidade de influenciar o processo de seleção da rede *P2P*, de modo vantajoso para os próprios. Desta forma, o *ISP*, pode oferecer um serviço com maior qualidade aos seus clientes e garantir a prioridade de outro tipo de tráfegos que circulem na sua rede, o que também lhes permite reduzir os custos do tráfego que sai da sua rede interna.

2.2.4.3 *P4P*

O *P4P* [4] é uma arquitetura flexível que permite aos *ISPs*, o fornecimento de orientações e recursos para aplicativos emergentes, como a distribuição de conteúdos *P2P*. Estas *interfaces* preservam a privacidade do fornecedor de rede, permitindo aos próprios e aos sistemas *overlay/P2P*, medidas de optimização conjunta.

Para que o mecanismo de colaboração funcione, o *P4P* introduz entidades, denominadas *iTrackers*, que servem como portais utilizados pelos fornecedores de rede. Esta entidade permite que o *P4P* divida as responsabilidades de controlo de tráfego entre os sistemas *overlay* e os *ISPs*. De modo específico, cada *ISP*, é tratado como um fornecedor de rede comercial convencional ou um fornecedor de serviços virtual que dispõe de um *iTracker* para a sua rede.

A Figura 2.9 mostra as principais entidades da arquitetura do *P4P*, assim como as *interfaces* pertencentes aos *iTrackers*, além de mostrar também o fluxo de informações destinado ao contexto de aplicações *P2P*. As entidades presentes na figura, são: os *iTrackers*, *appTrackers* de clientes *P2P* e clientes *P2P* (*peers*). Esta solução, suporta dois modos de funcionamento, respetivamente dedicados para redes *overlay* estruturadas e para redes *overlay* não-estruturadas. Para as redes *overlay* não-estruturadas, os *appTrackers* interagem com *iTrackers* e distribuem a informação *P4P* aos seus *peers*, enquanto que para as redes *overlay* estruturadas, nas quais não existem *appTrackers* centrais, mas sim métodos *DHT*, os *peers* obtêm a informação necessária diretamente dos *iTrackers*. Em ambos os casos, os *peers* podem auxiliar no processo de distribuição da informação.

Adicionalmente, também podem ser observadas na Figura 2.9 as *interfaces* fornecidas pelo *iTracker*, que são:

- **Interface de políticas de tráfego:** permite que as aplicações obtenham as políticas de utilização de uma rede. Podem ser definidas políticas de utilização, tais como: política de utilização de *links*, na qual é definido o padrão de utilização desejado em ligações específicas (por exemplo, evitar a utilização de *links* que estejam congestionados em certas horas) e definição de limites de congestionamento a partir de determinados *thresholds*.

- **Interface de distância P4P:** permite que sejam consultados os custos e a distância entre *peers*, de acordo com as redes físicas em que se encontram.
- **Interface de capacidade:** permite aos *peers* ou fornecedores de conteúdos, solicitar os recursos dos *ISPs*. Um *appTracker* pode consultar os *iTrackers* em domínios populares que possam disponibilizar servidores ou *caches* que podem ajudar a acelerar a distribuição de conteúdo *P2P*.

Deste modo, o *P4P* apresenta benefícios de ganho mútuos, quer para os *ISPs*, quer para os sistemas *overlay/P2P*, com a introdução de mecanismos de colaboração, que visam melhorar os aspetos de aproveitamento de recursos da rede e métodos de procura de vizinhos mais eficientes, nos sistemas *overlay*, além da possibilidade de aplicação de políticas de utilização de *links* pelo *ISP*, garantido um aumento do *QoS* de ambas as entidades abordadas.

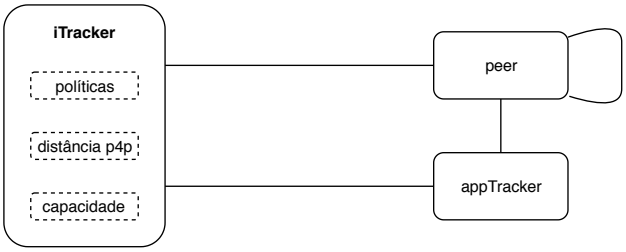


Figura 2.9: Interfaces iTracker e fluxo de dados (adaptação de [4])

2.3 Sistemas de Rede Específicos para Distribuição de Conteúdos

Neste segmento, inicialmente, vai ser efetuada uma análise da arquitetura de uma Content Delivery Network (CDN) [36]. De seguida, vai ser abordado outro sistema de rede, cujo estudo é apropriado para o desenvolvimento da aplicação que irá ser implementada nesta dissertação, denominado Peer-Assisted Content Delivery Networks (PA-CDN), em conjunto com os seus problemas de implementação [37]. Finalmente, será efetuada a comparação entre *CDN* e sistemas *P2P* tradicionais.

2.3.1 *Content Delivery Network*

Uma *CDN* é uma rede distribuída geograficamente, composta por servidores *proxy* e centrais de dados [36]. Este sistema de rede, efetua a replicação de conteúdo através da colocação estratégica dos servidores descritos anteriormente, de modo a melhorar o desempenho e escalabilidade do sistema. Esta arquitetura utiliza técnicas de maximização de largura de banda, de modo a melhorar a acessibilidade e manter a replicação correta de conteúdo, aumentando assim o seu desempenho geral. Além disso, este tipo de infraestruturas oferecem aplicações com alto desempenho, confiáveis e com serviços de distribuição de conteúdo

destinados a servidores de *cache* localizados perto dos utilizadores finais. Um exemplo da arquitetura deste tipo de rede, pode ser ilustrado pela Figura 2.10.

Os fornecedores das *CDNs* asseguram a entrega rápida de qualquer tipo de conteúdo. Estes, hospedam conteúdos *third-party* incluindo conteúdos estáticos (imagens, documentos, páginas HyperText Markup Language (HTML) estáticas, etc), conteúdos de *streaming* (áudio, *real time video*, etc) e outros serviços de conteúdo (serviço de diretórios, *e-commerce*, serviço de transferência de ficheiros, etc). Existem vários fornecedores de conteúdo, como por exemplo, grandes empresas, fornecedores de serviços de rede, emissoras de notícias, etc. Os utilizadores finais interagem com a *CDN* de modo a especificar o pedido de conteúdo/serviço via telefone, *smartphone*, *laptop*, *desktop*, etc.

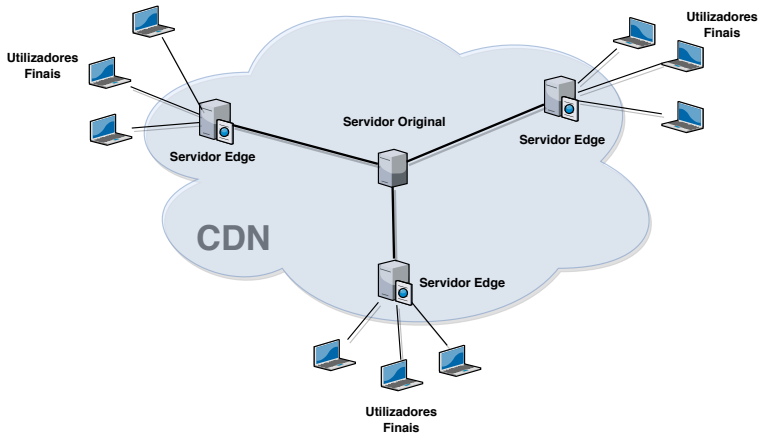


Figura 2.10: Arquitetura genérica de uma CDN

A funcionalidade das *CDNs* é dependente do sistema de *request-routing* que gere os pedidos dos clientes. Dois tipos de módulos foram incluídos para efetuar o encaminhamento dos pedidos dos utilizadores. Um deles, é responsável por selecionar o servidor *edge* mais apropriado para os pedidos dos utilizadores, enquanto que o outro, é responsável em direcionar os pedidos do utilizador para os servidores *edge* que forem selecionados. Todo este processo, foi desenhado para ser transparente aos utilizadores finais, sendo maioritariamente implementado ao nível do Domain Name System (DNS). Porém, as *CDNs* são sistemas complexos de distribuição de conteúdo que têm bastantes problemas técnicos e decisões envolvidas na gestão e administração de toda a rede. Algumas questões são colocadas quanto ao posicionamento dos servidores *edge*, o conteúdo certo para replicação e em que *cluster* de servidores copiar cada fragmento de conteúdo. Para a eventual resolução destes problemas, deve ser realizado um estudo aprofundado, visto que normalmente é bastante complexo arranjar uma solução estável.

Atualmente, existem diferentes tipos de *CDNs*, tais como: *Akamai* [38], *Limelight Networks* [39], *Netflix Open Connect* [40]. A empresa *Akamai Technologies* é uma das maiores empresas fornecedoras de serviços de distribuição de conteúdo. A abordagem da empresa, baseia-se

em combater os problemas de escalabilidade, desempenho e fiabilidade de fornecimento de conteúdo na *web* a partir de um único local e, através de um sistema criado para atender pedidos de um número variável de servidores nas extremidades da rede. Milhares de empresas criaram parcerias de confiança com a *Akamai*, aumentando a receita e reduzindo os custos dos seus serviços ao melhorar o desempenho dos seus negócios *online*. A *Limelight Networks* é uma rede de entrega de conteúdo desenhada para distribuir todo o tipo de conteúdo *web*, vídeo e aplicações de dados sobre Internet of Things (IoT). Garante uma experiência de serviço rápida, fiável e segura. Esta *CDN* é conhecida pela sua rede avançada de distribuição de conteúdo que fornece entregas com alto desempenho via *Internet*. A *Netflix Open Connect* retrata a rede de distribuição de conteúdo exclusiva de uma das empresas mais famosas em todo o mundo, a *Netflix*. Neste caso, o objetivo desta rede é fornecer especificamente aos assinantes da *Netflix*, a melhor experiência de visualização de vídeo possível. Sendo que a empresa conseguiu alcançar este objetivo a partir do estabelecimento de inúmeras parcerias com *ISPs* por todo o mundo, para localizar as dimensões de tráfego gerado com maior precisão possível e conseguir fornecer um nível ótimo de *QoS* a todos os seus subscritores.

Visto que a implementação de *CDNs* tem custos bastante elevados, e normalmente é operada por grandes empresas, surge então uma grande motivação para a criação de sistemas *CDN* assistidos por *peers*. Embora existam bastantes fornecedores comerciais de *CDNs*, alguns não cooperam na entrega dos conteúdos para os utilizadores finais de forma escalável. Assim sendo, este tipo de modelo fechado e não cooperativo, não é o mais apropriado para efetuar entregas de conteúdo a nível mundial. Visto que é bastante dispendioso para as empresas fornecer estas redes enormes de entrega de conteúdo, são criadas várias parcerias para conseguirem fornecer serviços que apenas um *CDN* sozinho não conseguiria. Além disso, as *CDNs* comerciais cobram aos consumidores pela utilização dos seus serviços, garantindo no entanto, um forte compromisso com os utilizadores finais de forma a cumprir o Service Level Agreement (SLA). Um *SLA* é um contrato entre os fornecedores do serviço e os utilizadores, descrevendo os compromissos dos fornecedores e especificando as penalidades a aplicar, caso estes não sejam cumpridos. O objetivo destas redes é satisfazer os seus clientes com serviços competitivos, porém, muitas empresas criam inconscientemente impactos negativos, quando existem violações no *SLA*, tais como a degradação do *QoS*. Existe também a preocupação e necessidade de um grande nível de conhecimento, para se conseguir distribuir os servidores *edge* geograficamente, de modo a ser obtido o melhor desempenho e garantias da entrega do conteúdo de maneira correta e estável. Toda esta informação revela a forte motivação que é estabelecida para a construção de um sistema descentralizado com maior escalabilidade e garantias de entrega de conteúdo, de modo mais barato e balanceado para os utilizadores conseguirem obter um bom nível de *QoS*.

Para isso foram implementadas estruturas híbridas que combinam as vantagens das *CDNs* e dos sistemas *P2P*, como irá ser abordado de seguida.

2.3.2 *Peer Assisted Content Delivery Networks*

A *PA-CDN* [37] é uma arquitetura híbrida que combina as vantagens das *CDNs* tradicionais com os sistemas *P2P*. Por um lado, o sistema híbrido necessita das contribuições de entrega de conteúdo pelos utilizadores finais, o que permite minimizar os custos da infraestrutura em relação a uma *CDN* tradicional. Por outro lado, as *PA-CDNs* aumentam o nível de *QoS* e disponibilidade na entrega de conteúdo dos sistemas *P2P* tradicionais, a partir da replicação de conteúdo em servidores *edge* espalhados pela rede. Deste modo, existe uma minimização de custos, visto que a contribuição dos *peers* é suficiente para diminuir o número de servidores *edge* da rede, tornando o serviço mais escalável e com um grau de *QoS* semelhante ao das *CDNs* tradicionais. Desta forma, o maior desafio é integrar os componentes deste sistema híbrido, de modo a serem garantidos os benefícios das *CDNs* e sistemas *P2P* tradicionais. A Figura 2.11 ilustra a arquitetura genérica de uma *PA-CDN*.

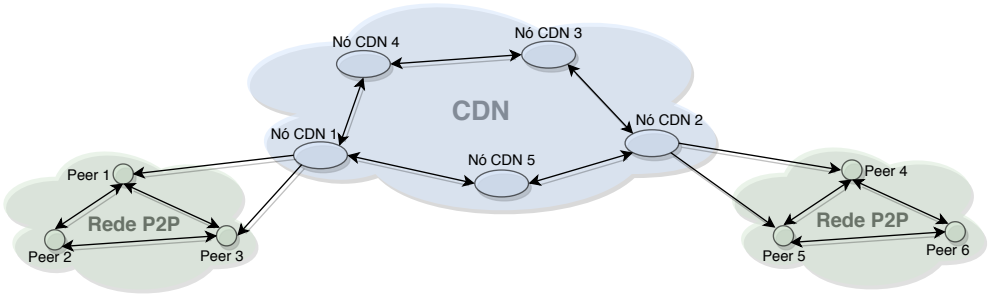


Figura 2.11: Arquitetura genérica de uma PA-CDN (adaptação de [5])

Estas arquiteturas híbridas, podem ser divididas em dois grupos: arquiteturas centralizadas e arquiteturas descentralizadas, tendo como base o grau de centralidade na gestão de sistemas *overlay P2P*. No grupo das arquiteturas centralizadas, os novos utilizadores contactam os nós *edge* mais próximos a partir de um sistema *request-routing* da *CDN*. Os servidores *edge* registam os *peers* e guardam os seus metadados (endereço Internet Protocol (IP), porta relativa ao conteúdo pedido, etc) numa base de dados centralizada. Além disso, os servidores *edge* encaminham vários segmentos de dados para os *peers* que efetuaram o pedido, seguidos por uma lista aleatória de *peers* selecionados que contêm esse conteúdo. Se um *peer* for incapaz de estabelecer conexão com o número de *peers* necessário, a conexão colapsa e o servidor *edge* volta a enviar uma lista atualizada de *peers*. Se um segmento de dados não for recebido no prazo certo, o servidor *edge* atende diretamente o pedido. Este tipo de arquiteturas *PA-CDN* puramente centralizadas têm muitas vantagens sobre sistemas *P2P* tradicionais. O nível de *QoS* é garantido a partir da replicação/recuperação dos

segmentos de dados armazenados em nós *CDN*. Adicionalmente, neste tipo de arquiteturas centralizadas a *CDN* tem controlo sobre a *P2P overlay*, visto que consegue simplificar significativamente as suas rotinas e gestão de conteúdo, aumentando o desempenho e estabilidade do sistema em si.

No caso das arquiteturas descentralizadas, a gestão da *overlay P2P* é efetuada por *Tracker peers*, acompanhando o estado dos *peers* na sua vizinhança. Quando um *peer* entra na rede, contacta de imediato o *Tracker* peer mais próximo, fornecendo-lhe uma lista de *peers* ativos. Se um *Tracker* não consegue fornecer a lista de *peers* ativos, o pedido é redirecionado para outro *Tracker* na sua vizinhança, até que o segmento de dados desejado seja encontrado. Caso não existam *peers* suficientes para assistir a transferência do conteúdo, o *Tracker* fica encarregue de fazer o pedido dos segmentos de vídeo em falta para um nó *edge*. Neste tipo de arquitetura descentralizada existe uma maior escalabilidade a um custo bastante menor em relação a *CDNs* tradicionais, sendo que os *peers* efetuam grande parte do trabalho, uma vez que só são utilizados nós *edge* em casos excecionais. Porém, *PA-CDNs* descentralizadas são mais vulneráveis a ataques maliciosos e a gestão da rede é mais difícil comparativamente com a alternativa centralizada.

Embora muitos ambicionem este tipo de arquiteturas híbridas, as *PA-CDNs* apresentam vários desafios de implementação, nomeadamente para sustentar um nível de *QoS* que consiga satisfazer os utilizadores finais. De modo a que esta arquitetura híbrida de distribuição de conteúdo funcione para um grande número de utilizadores, é necessária a existência de um grande número de *peers* em funcionamento, para que a carga do sistema seja distribuída de maneira uniforme. Porém, o tamanho de um sistema *P2P* pode ser afetado por diversos fatores, que podem ser classificados em três grupos de desafios:

- **Desafios de heterogeneidade**
  - **Heterogeneidade em padrões de acesso dos utilizadores:** a entrada/saída assíncrona de utilizadores no sistema e a instabilidade do tempo de duração no sistema, provoca uma grande dificuldade na gestão dos diversos grupos de *peers* encarregues pela entrega do conteúdo;
  - **Heterogeneidade em recursos:** os recursos de cada utilizador, como a ligação à *Internet* e o desempenho de diferentes dispositivos, influenciam a quantidade e eficiência da partilha de conteúdo com outros. Adicionalmente, é importante a consciencialização de como o tamanho e a sustentabilidade dos grupos de *peers* com o conteúdo é afetada pelos *peers* com poucos recursos.
- **Desafios técnicos**
  - **Qualidade de serviço:** ao contrário da garantia do nível de *QoS* inerente às *CDNs*, nos sistemas *PA-CDN*, dado que a maior parte do trabalho de partilha de conteúdo é efetuado pelos *peers*, torna-se bastante complicado adquirir o mesmo

nível de *QoS*. Para implementar este tipo de infraestruturas híbridas, têm que ser analisados métodos de entrega de conteúdo de alta qualidade, uma vez que devido à heterogeneidade de partilha de recursos por parte dos *peers*, por vezes, torna se num dos maiores desafios técnicos;

- **Inacessibilidade:** os problemas de inacessibilidade por parte dos *peers* são um desafio técnico bastante complexo de resolver. Quando um *peer* está dentro de uma rede privada, consegue iniciar a conexão com *peers* de redes públicas, porém o inverso pode ser impossível devido a certas políticas administrativas da rede privada. Este problema provoca a diminuição da taxa de *download* aos utilizadores que não conseguem estabelecer conexão com nós dentro de redes privadas.
- **Direitos de autor:** a preservação dos direitos de autor, juntamente com a permissão dada aos utilizadores para armazenar e partilhar conteúdo é um dos maiores desafios dos fornecedores das *PA-CDNs*. Para todos os utilizadores poderem armazenar e partilhar conteúdos nestes sistemas, muitas vezes são efetuadas técnicas de codificação para proteger os direitos de autor dos conteúdos em circulação no sistema. Outra solução visa armazenar esquemas de controlo em servidores *Tracker* para que apenas autorizem a distribuição de conteúdos legais no sistema.

- **Desafios de viabilidade comercial**
  - **Participação parcial e incentivos**: dado que o funcionamento das *PA-CDNs* depende maioritariamente dos recursos dos *peers*, estes têm que estar dispostos a participar e contribuir com os seus recursos para o bom funcionamento destes sistemas híbridos. Isto significa que, a escalabilidade do sistema é diretamente proporcional às contribuições dos utilizadores, pelo que se não houver participação por parte dos utilizadores, o funcionamento do sistema pode ficar em risco. Para tentar impedir este risco, os fornecedores das *PA-CDNs* oferecem aos utilizadores planos de incentivos, persuadindo cada utilizador a fazer uma certa taxa de *upload* em troca de ganhos em relação ao serviço em questão, como a oferta de vídeos de alta qualidade gratuitos, etc.
  - **Tráfico *inter-ISP*:** neste tipo de sistemas híbridos é importante existir uma consciencialização do tráfego que circula na rede *overlay*. Caso não exista tal consciencialização, pode surgir um problema quando os utilizadores partilham os seus recursos de largura de banda, e ultrapassam os limites da área do *ISP* local, trazendo-lhe prejuízo. Para resolver este problema são efetuados contratos com *ISPs* e são implementados protocolos de optimização para diminuir esta carga de tráfego entre *ISPs*.

Comparativamente com as *CDNs*, a implementação de *PA-CDNs* ainda é recente, existindo poucas destas arquiteturas híbridas. Como exemplo, vão ser analisados os serviços:

Spotify [41], *P2PTube* [42, 43] e *PeerCDN* [44]. O *Spotify* é um dos serviços de *streaming* de música, *podcast* e vídeo mais conhecido em todo mundo. Este, permite o acesso de baixa latência a uma biblioteca com mais de oito milhões de faixas de música, permitindo aos utilizadores a escolha livre de faixas que desejarem ouvir e um sistema de pesquisa bastante otimizado para procurar todo o tipo de músicas, álbuns, artistas, etc. Os dados são transmitidas pelos servidores e por redes *P2P*. O *P2PTube* é um sistema de distribuição de conteúdo, que aproveita o potencial técnico e económico do paradigma emergente na distribuição de media. É conhecido como um sistema *P2P* híbrido para difusão de conteúdo media e interação utilizador-artista suportado por doação e propaganda. Este sistema, como o *Youtube*, mantém um meio de interação social a partir do consumo de vídeo e da submissão de conteúdo por parte de cada utilizador. O *PeerCDN* é um serviço eficiente de *streaming*, cuja arquitetura é composta por duas camadas. A primeira camada é constituída por servidores *CDN* originais e de réplica. A segunda camada consiste no grupo de clientes que efetuam serviços de *streaming*, no qual, cada um deles é considerado um *peer* do sistema. A sua arquitetura tornou-se conhecida por conseguir reduzir drasticamente o custo de largura de banda na entrega de conteúdo.

2.3.3 *Comparação entre Arquiteturas CDN e Sistemas P2P*

Métrica Comparativa	**Comparação entre Arquiteturas CDN e P2P**	
	CDN	**P2P**
Capacidade de Serviço e Escalabilidade	Capacidade de serviço limitada e custo de expansão elevado	Capacidade de serviço aumenta consoante o aumento do número de peers e custo de expansão baixo
Confiabilidade e Estabilidade	Alta confiabilidade e boa estabilidade	Baixa confiabilidade e pouca estabilidade
Fluxo de Dados	Controlado em diferentes zonas. ISP-friendly	Desordem/expansão de tráfego entre ISPs. ISP-unfriendly
Monitorização de Conteúdo	Pode ser monitorado ao nível da fonte	É difícil de ser monitorado ao nível da fonte
Administração de Utilizadores	Centralizada	Menor capacidade de administração
Garantias de QoS	Pode ser garantido dentro da capacidade máxima de serviço	Não pode ser controlado (Best-Effort)
Direitos de Autor	Controlável	Incontrolável
Autenticação	Certificação central	Certificação distribuída ou inexistente
Nó de serviço	Homogéneo. CDN nodes efetuam o serviço e client nodes acedem aos serviços da CDN	Heterogéneo. Um client node pode efetuar/aceder um serviço

Tabela 2.2: Tabela de comparação entre arquiteturas CDN e P2P (adaptação de [5])

Com base nas arquiteturas *CDN* e *P2P* estudadas, este tópico visa compreender os aspetos que levaram à necessidade de implementação de arquiteturas *PA-CDN*. Para isso, vão ser explicitadas todas as informações técnicas a nível de capacidade de serviço, escalabilidade, fiabilidade, estabilidade, monitorização, garantias de *QoS*, administração, autenticação, proteção dos direitos de autor, entre outras. Na Tabela 2.2 estão representadas sucintamente as características dos sistemas abordados de modo comparativo, de forma a serem compreendidos os problemas e desafios da implementação das *PA-CDNs*, e das vantagens que estas

poderiam trazer relativamente às *CDNs* tradicionais, nomeadamente em termos monetários, arquiteturais e funcionais.

2.4 SUMÁRIO

Este capítulo abordou os conceitos essenciais necessários para o desenvolvimento do sistema de partilha de conteúdo proposto nesta dissertação. Inicialmente, foi efetuado um estudo geral do conceito de redes *overlay*, sendo especificada a sua arquitetura, estruturação da rede, funcionamento, consciencialização da rede *underlay*, tendo sido abordados problemas da rede com os *ISPs* e possíveis ameaças que estas redes possam sofrer. Seguidamente, em adição ao que foi abordado na secção anterior, sobre consciencialização da rede *underlay* em sistemas *overlay/P2P*, foi efetuado um estudo sobre a interação entre este tipo de sistemas e o fornecedor de rede, no qual também foram detalhadas soluções de colaboração atuais. Por fim, foram abordados sistemas de rede específicos para distribuição de conteúdo, onde foi efetuado o estudo de redes de entrega de conteúdo (*CDNs*) e redes de entrega de conteúdo assistidas por *peers* (*PA-CDNs*), assim como a comparação destes dois sistemas, ilustrada em uma tabela. Este capítulo ofereceu as bases necessárias para a implementação da rede *overlay* de partilha de conteúdos, controlada pelo *ISP*, proposta nesta dissertação, assim como tecnologias atuais que possam servir de meio de comparação com o protótipo a ser implementado.

3

ARQUITETURA E MECANISMOS DO SISTEMA DA REDE OVERLAY

Assim como referido anteriomente, este capítulo apresenta a arquitetura da rede *overlay* para partilha de conteúdos, assim como as suas funcionalidades básicas e de colaboração com o *ISP*. Na secção 3.1 é efetuada a apresentação da arquitetura geral do sistema proposto, na qual são representadas as entidades integrantes, em conjunto com as suas funções, assim como os protocolos utilizados. Posteriormente, na secção 3.2, são abordadas as funcionalidades da rede *overlay*, assim como os mecanismos de colaboração com o *ISP* de modo a possibilitar o controlo de tráfego *P2P* excessivo que percorre a rede física do fornecedor de rede. Por fim, é efetuado o sumário do capítulo na secção 3.3.

3.1 ARQUITETURA GERAL DO SISTEMA PROPOSTO

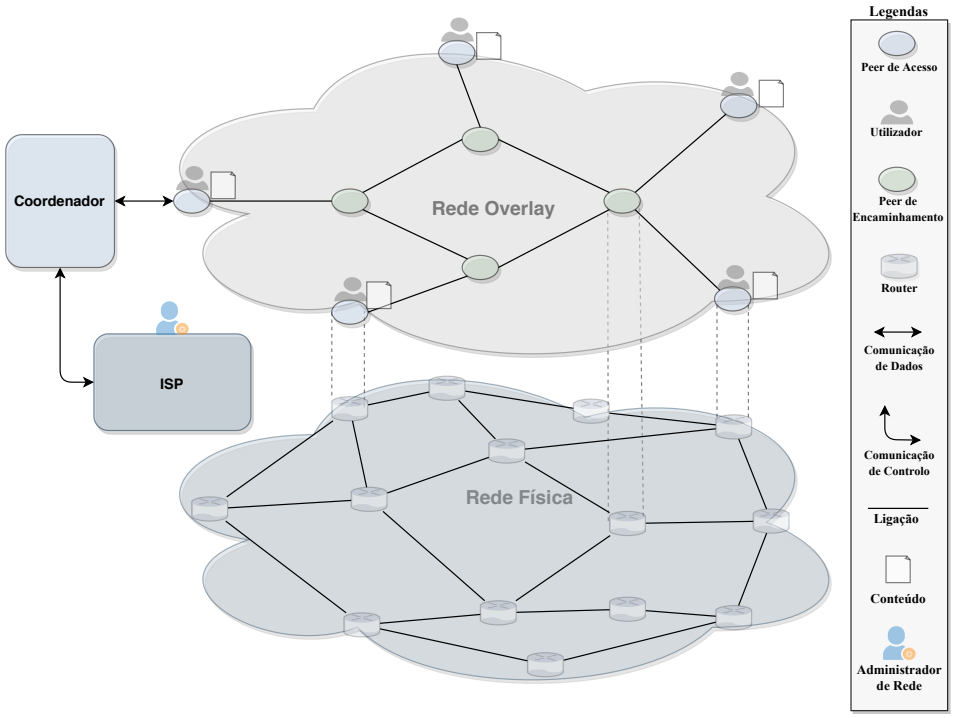


Figura 3.1: Arquitetura geral da rede overlay em conjunto com a rede física

Considerando os conceitos pesquisados e expostos no capítulo anterior, subentende-se que para elaborar uma rede *overlay* de modo funcional e responsivo, é essencial a existência de uma arquitetura concisa, organizada e confiável. Nesse sentido, e dado que o objetivo desta dissertação é a criação de uma rede *overlay* para partilha de conteúdos, controlada pelo *ISP*, é necessária a conciliação de dois ambientes de rede distintos: rede *overlay* e rede física, assim como demonstra a Figura 3.1. Para que esta conciliação seja produzida, a rede *overlay* tem que ter consciência da rede física. Desta forma, foi armazenada a topologia física de rede e o estado da rede *overlay* em dois grafos separados.

A rede *overlay* é caracterizada por três entidades essenciais para o funcionamento da partilha de conteúdos: os *peers*, o coordenador e o *ISP*. Neste sistema, existem dois tipos de *peers*, que foram classificados por: *peers* de encaminhamento e *peers* de acesso, como irá ser posteriormente explicado com maior detalhe. Estes, têm como função principal, efetuar o processo de encaminhamento a nível aplicacional. O coordenador é o responsável pela consciencialização de toda a estrutura da rede *overlay*. Posto isto, tem como principais funções: armazenar a topologia de rede física do *ISP*, armazenar e atualizar o estado da rede *overlay* à medida que entram novos *peers*, armazenar os conteúdos que percorrem a rede *overlay*, efetuar a recolha de pedidos de utilizadores. Além disso, também tem a função de administrar os *peers* de modo a que a partilha de ficheiros inclua as definições estipuladas pelo *ISP*. Além das duas entidades abordadas, a administração de rede não seria funcional, sem a terceira e última entidade, o próprio *ISP* da infraestrutura de rede. Esta entidade, tem a função de comunicar ao coordenador da rede *overlay*, as regras e políticas que irão ser utilizadas ao longo do funcionamento da rede *overlay*. Com a arquitetura proposta, vão ser definidas como funcionalidades base do sistema de partilha de conteúdos, dois modos de operação de transferência: o cliente-servidor, e o *P2P*. Este processo irá ser administrado pelo coordenador em conjunto com o *ISP*. Posto isto, as principais funcionalidades/objetivos a desenvolver serão: definição de regras de encaminhamento, políticas/mecanismos de engenharia de tráfego e planeamento de decisões de tráfego, definidas pelo *ISP* e aplicadas pelo coordenador da rede *overlay*.

3.1.1 *Rede Overlay*

Inicialmente, na especificação da arquitetura geral, é exposta a visão geral da rede *overlay*, indicada na parte superior da Figura 3.1. Esta, é o elo de ligação entre os *peers* e o coordenador, representando o local onde são efetuados os pedidos dos clientes ao coordenador e ocorre o encaminhamento de pacotes a nível aplicacional. Para garantir o bom funcionamento do sistema de partilha são fundamentais três entidades principais: *peer*, coordenador e *ISP*, que serão abaixo descritas. Nesta rede, existem dois modos de transferência de conteúdos: *P2P* e cliente-servidor. O modo de transferência vai ser automaticamente escolhido

pelo coordenador da rede, tendo em conta questões relacionadas com o número de *peers* que possuem o conteúdo a ser transferido, assim como o tamanho do mesmo. Para suportar as funcionalidades de partilha de conteúdos e controlo de tráfego (com a intervenção do *ISP*) são necessárias duas bases de dados distintas. Uma, para armazenar a topologia física e a organização da rede *overlay*, e outra para armazenar os *metadados* essenciais para a partilha de conteúdos entre utilizadores. Além disso, foram implementados quatro Protocol Data Unit (PDU)*s*, para organizar e controlar os pacotes aplicacionais encaminhados na rede, que serão descritos posteriormente.

3.1.2 *Peer*

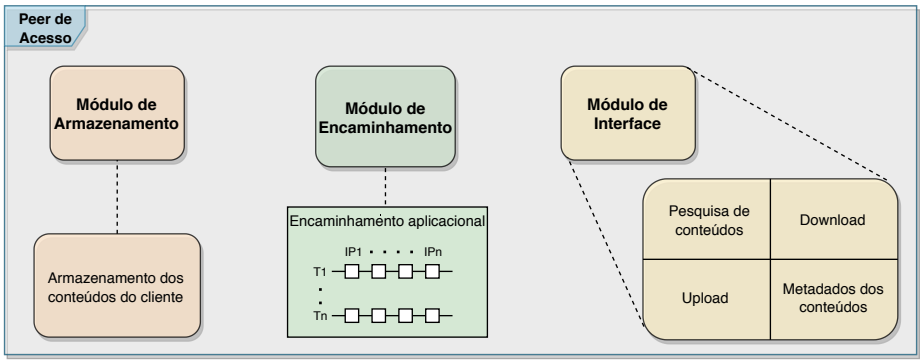


Figura 3.2: Visão modular do peer de acesso

O *peer* é a entidade evidenciada pela sua maior representatividade na rede *overlay* (Figura 3.1). Intuitivamente, percebe-se que a sua existência é essencial, quer para o funcionamento da rede *overlay*, quer para a execução do encaminhamento a nível aplicacional e para que o utilizador consiga partilhar todo o tipo de conteúdos. Além dissso, com o benefício de disponibilizar recursos computacionais para o sistema de partilha de conteúdos. Existem dois tipos de *peers*: os de acesso e os de encaminhamento. Os *peers* de acesso (utilizadores finais) são responsáveis por apresentar uma *interface* amigável para o cliente do sistema, efetuar o encaminhamento aplicacional e receber/efetuar ordens de partilha de conteúdo ao coordenador. Os *peers* de encaminhamento, têm apenas a função de distribuir os pacotes de conteúdo para o seu destino. Deste modo, com a ajuda do coordenador, os *peers* têm a função de conhecer a topologia de rede e efetuar o encaminhamento de nível aplicacional, de modo a partilhar o conteúdo entre os utilizadores da rede. As funcionalidades do *peer* de acesso, podem ser agregadas em três módulos principais, Figura 3.2:

- **Módulo de armazenamento:** armazenamento dos conteúdos que o cliente partilha na rede. O armazenamento é efetuado no sistema de ficheiros de cada *peer* de acesso (utilizador final) pertencente à rede *overlay*;

- **Módulo de encaminhamento:** após o coordenador receber um pedido de transferência por um cliente da rede, a rota aplicacional é transmitida aos *peers* de modo a que estes conheçam o seu destino. A comunicação é efetuada com a utilização de *PDUs* que visam organizar a estrutura dos pacotes de dados.

- **Módulo de visualização:** a *interface desktop* está presente nos *peers* de acesso. Nestes, irá ser apresentado um ambiente gráfico fácil e acessível, destinado ao utilizador, para que o mesmo possa usufruir dos serviços de partilha de conteúdos da rede *overlay*.

3.1.3 *Coordenador*

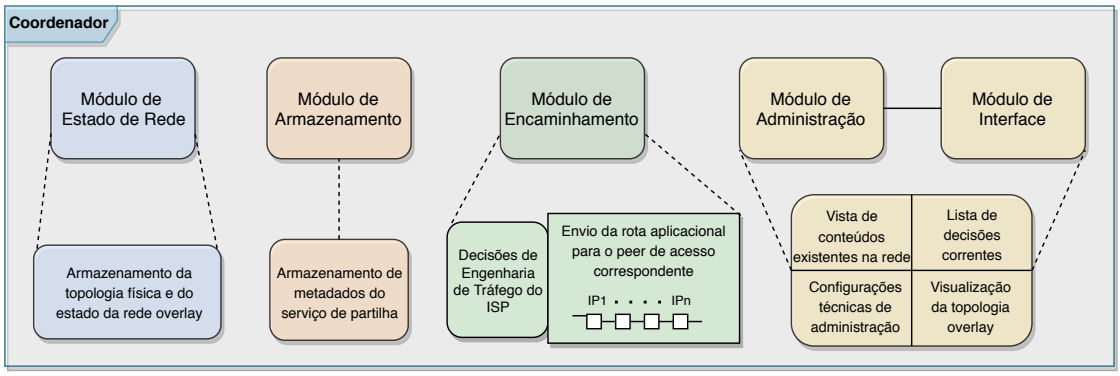


Figura 3.3: Visão modular do coordenador

O coordenador é a entidade mais importante e complexa da arquitetura do sistema de partilha de conteúdos. Tem várias funções, como: armazenar a topologia de rede física, o estado da rede *overlay* e informação acerca dos ficheiros incluídos na rede, além de processar os pedidos dos clientes, associando aos *peers* a respetiva rota de encaminhamento para o utilizador final e administrar a rede *overlay* com ajuda do *ISP*. A visão modular do coordenador pode ser ilustrada na Figura 3.3, sendo composta por:

- **Módulo de Estado de Rede:** de modo a que o coordenador tenha consciência sobre a rede física e *overlay*, este módulo efetua o armazenamento de todas as informações relativas à topologia física e à organização da rede *overlay* (consoante a entrada/saída de *peers*).

- **Módulo de Armazenamento:** responsável pelo armazenamento de metadados acerca dos conteúdos fornecidos pelos utilizadores. Estes, são depois associados a cada *peer* de acesso.

- **Módulo de Encaminhamento:** responsável pela gestão do encaminhamento aplicacional efetuado pelos *peers* do sistema. Após um pedido de transferência de um

utilizador final ser processado, é calculada a rota aplicacional, segundo os parâmetros indicados pelo *ISP* (com a ajuda do módulo de estado de rede, armazenamento e administração), sendo posteriormente encaminhada para os *peers* de acesso que irão começar o processo de partilha de conteúdo.

- **Módulo de Visualização:** oferece uma *interface web* visual para o *ISP* efetuar/alterar configurações. Esta, estabelece um canal de comunicação com o coordenador da rede *overlay*, de modo a que as configurações sejam efetuadas em tempo real.
- **Módulo de Administração:** processa e efetua as decisões do *ISP* após serem decididas no módulo de visualização.

3.1.4 *ISP*

O *ISP* é a entidade que estabelece a consciencialização entre a topologia de rede física e a *overlay*. Este, estabelece as configurações a aplicar na rede *overlay*, com base em determinados fatores analisados na rede física, enviando-as ao coordenador para serem aplicadas no sistema. Como foi evidenciado na Figura 3.3, a *interface* do administrador do *ISP* está conectada ao coordenador, onde são efetuadas em tempo real as devidas alterações. Em termos gerais, o *ISP* tem a possibilidade de:

- **Escolher a estratégia de encaminhamento da rede overlay:** a rede *overlay* poderá funcionar segundo duas estratégias de encaminhamento base: minimização de custos de encaminhamento e minimização do número de saltos. Na primeira, assume-se que o custo associado à comunicação entre um determinado par de *peers* está de acordo com o custo de encaminhamento associado ao próprio par, sendo assumido pelo nível da rede do *ISP*. No segundo caso, assume-se que o custo associado à comunicação entre um determinado par de *peers* representa o número de *links* (saltos) atravessados pelo caminho do nível de rede que interliga o próprio par. Tendo em consideração que o *ISP* poderá pretender minimizar o impacto que o tráfego da rede *overlay* tem na sua infraestrutura, a segunda estratégia de encaminhamento é aquela que é usada por defeito.
- **Adicionar/Remover proteção de links/routers:** ao adicionar um *link/router* à lista de alvos protegidos, esta opção permite ao *ISP* fazer com que os pacotes aplicacionais da rede *overlay* tentem evitar os percursos que contenham o link/router em questão.
- **Adionar/Remover priorização de links/routers:** ao adicionar um *link/router* à lista de alvos com prioridade, esta opção permite ao *ISP* fazer com que os pacotes aplicacionais da rede *overlay* tentem priorizar os percursos que contenham o *link/router* em questão.

- **Adicionar/Remover limite de circulação de conteúdos:** esta opção permite ao *ISP* adicionar um limite no coordenador da rede *overlay*, que é ativado quando a soma dos tamanhos dos ficheiros em circulação na rede atingir o valor especificado. Quando esse valor for atingido, todos os clientes que desejarem efetuar *downloads* serão colocados numa fila de espera.

3.1.5 *Especificação Protocolar do Sistema*

Os protocolos que foram utilizados no sistema de partilha de conteúdos, foram escolhidos tendo em consideração: o aproveitamento de recursos de rede e eficiência na entrega de conteúdo. No sistema de rede *overlay* foi definida a utilização do protocolo de transporte User Datagram Protocol (UDP), na comunicação entre o coordenador da rede *overlay* e os *peers*, exceto quando o *peer de acesso* (utilizador final) efetua pedidos ao coordenador, aí é utilizado o Transmission Control Protocol (TCP). A comunicação *ISP*-coordenador é inteiramente efetuada por *TCP*. Por fim, a comunicação entre *peers* utiliza o protocolo *UDP*. Deste modo, o *TCP* foi utilizado para os processos de controlo e gestão, de forma a garantir a entrega dos pacotes e a ser verificado o estado da conexão entre duas entidades. Por conseguinte, o *UDP* foi utilizado para distribuir grandes quantias de pacotes de um ponto para outro, devido à sua flexibilidade, simplicidade e menor latência comparativamente ao *TCP*, além de possibilitar a implementação de métodos de entrega de pacotes e outros mecanismos na camada aplicacional do sistema de rede *overlay* de partilha. Adicionalmente, foram definidos *PDUs*, responsáveis por organizar e armazenar a informação dos pacotes de nível aplicacional, de maneira estruturada e uniforme. Como tal, foram selecionados quatro *PDUs* utilizados para o sistema de rede *overlay* de partilha, nomeadamente: administração, informação, notificação e confirmação (*ACK*).

Figura 3.4: Ilustração do PDU administrativo

Inicialmente, foi desenhado o *PDU* de administração relativo às comunicações entre o coordenador da rede *overlay* e o *ISP*. A Figura 3.4 ilustra as características do *PDU* administrativo:

- ***ID:*** identificador único da tarefa enviada ao coordenador pelo *ISP*. O valor inicialmente estipulado para o campo foi de 4 bytes.

- ***Opção:*** tipo de decisão/configuração escolhida pelo *ISP* (proteção de *link/router*, priorização de *link/router*, definir limite,etc) cujo valor do campo é representado por 32 bytes.

- ***Valor da Opção*:** valor a alterar na configuração escolhida pelo *ISP* (nome do link/-router, valor do limite, etc) cujo valor do campo é representado por 64 bytes.

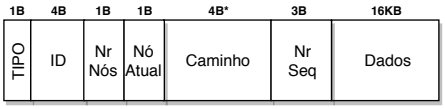


Figura 3.5: Ilustração do PDU de dados

Em segundo lugar, o *PDU* de dados foi desenhado para suportar os pacotes de nível aplicacional que transportam os conteúdos, e que serão encaminhados ao longo da rede *overlay*.

A Figura 3.5 ilustra as características do *PDU* de dados:

- ***Tipo:*** campo que específica o tipo do pacote, com tamanho fixo de 1 byte. O valor fixo do campo tipo para os pacotes de dados, foi definido com o valor 1.
- ***ID:*** identificador único de um pedido do utilizador. O valor inicialmente estipulado para o campo foi de 4 bytes.
- **Número de Nós:** representa o número de *peers* presentes no caminho calculado, pelo coordenador, até ao destino, com tamanho fixo de 1 byte.
- **Nó Atual:** representa o nó atual, onde o pacote que está a ser encaminhado na rede *overlay* se encontra. Este campo, em conjunto com o anterior, foi adicionado para aumentar a *performance* do algoritmo de encaminhamento, sendo que os *peers* conseguem perceber qual o destino de imediato, e/ou se é o último destino do pacote. O tamanho máximo também é de 1 byte, sendo que o seu conteúdo que representa o nó atual vai incrementando sempre que avança no encaminhamento para o próximo *peer* da rede *overlay*.
- **Caminho:** lista de *IPs* dos *peers* da rede *overlay*, previamente calculados pelo coordenador, após a recepção de um pedido de transferência por um utilizador do serviço de partilha. O tamanho deste campo é dinâmico e múltiplo de 4 bytes, visto armazenar múltiplos endereços *IP*, que variam dependendo do estado da rede e das decisões do *ISP*.
- **Número de Sequência:** este campo representa o número de sequência do pacote, para ser efetuada no fim a organização do conteúdo. O tamanho fixo máximo idealizado para este campo foi de 3 bytes.
- **Dados:** respetivos dados que circulam no pacote. Inicialmente foi definido o tamanho máximo de 16 *KB* para este campo do pacote.

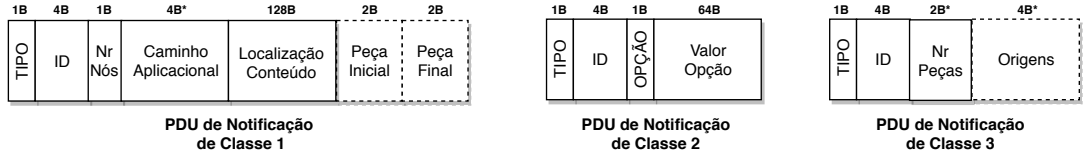


Figura 3.6: Ilustração das diferentes classes do PDU de notificação

Posteriormente, foi criado um tipo de *PDU*, que foi desenhado para efetuar as notificações do sistema, ou seja, para suportar três diferentes classes de notificações, essenciais para o bom funcionamento do serviço de entrega de conteúdo pelos *peers* da rede *overlay*. A Figura 3.6 ilustra as características dos diferentes *PDUs* de notificação, diferenciáveis por classes:

- **Classe 1:** Notificação entre o coordenador e o *peer* de acesso com um fragmento do conteúdo pedido pelo utilizador (caso o modo seja *P2P*) ou com a totalidade do conteúdo (caso o modo seja cliente-servidor). Esta notificação informa o *peer* com o conteúdo pedido pelo utilizador, do caminho que vai ser efetuado no encaminhamento, e caso esteja no modo *P2P*, também vai informar o início e fim da peça na totalidade do conteúdo, que o mesmo irá enviar para o destino:
  - ***Tipo:*** campo que específica o tipo do pacote, com tamanho fixo de 1 byte. O valor fixo do campo tipo para os pacotes de notificação de classe 1, foi definido com o valor 2.
  - ***ID:*** identificador único de um pedido do utilizador. Inicialmente foi definido o tamanho máximo de 4 bytes.
  - **Número de Nós:** representa o número de *peers* presentes no caminho até ao destino, com tamanho fixo de 1 byte.
  - **Nó Atual:** Representa o nó atual, onde o pacote se encontra. O nó atual é o conteúdo correspondente a uma posição no caminho aplicacional, sendo incrementada à medida que o pacote é encaminhado de um ponto para o outro. Este campo foi adicionado para que os *peers* conhecem sempre a sua posição atual e o seu destino (conseguindo distinguir se é o último ou não com a ajuda do campo, número de nós). O tamanho máximo também é de 1 byte.
  - **Caminho Aplicacional:** lista de (*IP*s) dos *peers* da rede *overlay*, previamente calculados pelo coordenador, após a recepção de um pedido de transferência por um utilizador do serviço de partilha. O tamanho deste campo é dinâmico e múltiplo de 4 bytes, visto armazenar múltiplos endereços *IP*, que variam dependendo do estado da rede e das decisões do *ISP*.
  - **Localização do Conteúdo:** Localização do conteúdo em determinado *peer* de acesso que começará a efetuar o envio aplicacional. O tamanho máximo foi fixado em 128 bytes.

  - **Peça Inicial:** campo opcional com tamanho fixo de 2 bytes, que indica o início da peça do conteúdo a ser enviada para o utilizador de um *peer* de acesso com um fragmento da sua totalidade.
  - **Peça Final:** campo opcional com tamanho fixo de 2 bytes, que indica o final da peça do conteúdo a ser enviada para o utilizador de um *peer* de acesso com o fragmento da sua totalidade.

- **Classe 2:** Notificação entre o *peer* de acesso (utilizador) e o coordenador. Esta, tem a função de informar o coordenador, sobre o tipo de pedido que o utilizador efetuou (pesquisa de ficheiro, *upload* ou *download*):
  - ***Tipo:*** tipo do pacote de dados, com tamanho fixo de 1 byte. O valor fixo do campo tipo para os pacotes de notificação de classe 2, foi definido com o valor 3.
  - ***ID:*** identificador único de um pedido do utilizador. Inicialmente foi definido o tamanho máximo de 4 bytes.
  - ***Opção:*** tipo de pedido que o utilizador efetuou à rede. Foi definido o tamanho fixo de 1 byte, caso o utilizador possa vir a ter no futuro mais opções, além de pesquisar um ficheiro, efetuar um *download* ou partilhar um conteúdo para a rede *overlay*.
  - ***Valor da Opção:*** valor do pedido que o utilizador efetuou, no qual, normalmente está estipulado o nome do ficheiro, ou a localização do mesmo. Foi definido o tamanho máximo de 64 bytes.

- **Classe 3:** Notificação entre o coordenador e o *peer* conectado com o utilizador. Esta, tem como função informar o utilizador do estado do seu pedido, a partir do número de peças, caso o modo seja cliente-servidor, ou pelo número de peças por cada origem (*peer* com conteúdo), caso o modo seja *P2P*:
  - ***Tipo:*** tipo do pacote de dados, com tamanho fixo de 1 byte. O valor fixo do campo tipo para os pacotes de notificação de classe 3, foi definido com o valor 4;
  - ***ID:*** Identificador único de um pedido do utilizador. Inicialmente foi definido o tamanho máximo de 4 bytes, capaz de representar mais de 4 biliões de *IDs*;
  - ***Número de Peças:*** número de peças de um dado conteúdo a enviar para o utilizador. Foi definido um tamanho dinâmico múltiplo de 2 bytes, dependendo do modo de transferência do sistema (*P2P* ou cliente-servidor);
  - ***Origens:*** este campo representa os *peers* que vão enviar fragmentos/totalidade do respetivo conteúdo pedido pelo utilizador. Foi definido um tamanho dinâmico múltiplo de 4 bytes, visto que cada endereço *IP* terá 4 bytes, dependendo do modo de transferência do sistema (*P2P* ou cliente-servidor).

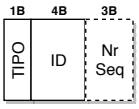


Figura 3.7: Ilustração do PDU de confirmação

Por fim, foi utilizado um *PDU* de confirmação, semelhante ao típico *ACK* utilizado em protocolos de comunicação, para efetuar a confirmação de pacotes, quer no âmbito *P2P*, quer entre o coordenador e os *peers*. A Figura 3.7 ilustra as características do *PDU* de confirmação:

- ***Tipo:*** Campo que especifica o tipo do pacote, com tamanho fixo de 1 byte. Este pacote específico foi declarado como sendo do tipo 5.
- ***ID:*** Identificador único de um pedido do utilizador. Inicialmente foi definido o tamanho máximo de 4 bytes.
- ***Número de Sequência:*** Campo opcional, visto que só é necessário nos casos do encaminhamento *P2P*. O tamanho fixo máximo idealizado para este campo foi de 3 bytes.

3.1.6 *Ilustração do Funcionamento Básico do Sistema de Rede*

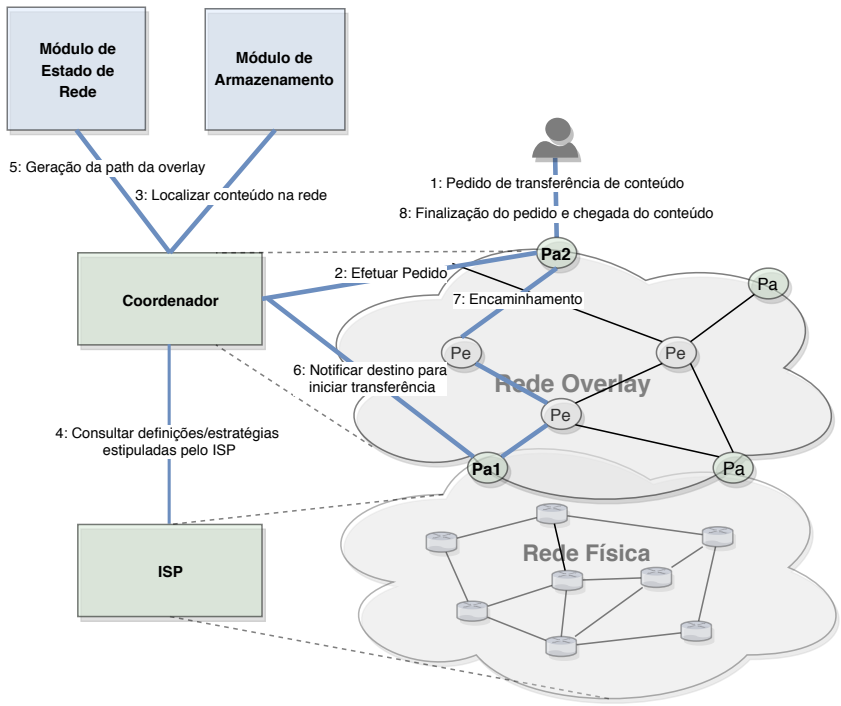


Figura 3.8: Ilustração do funcionamento básico do sistema que percorre a rota aplicacional desde o peer de acesso $Pa1$ até ao peer de acesso $Pa2$

Para compreender o funcionamento do sistema de partilha de conteúdos proposto nesta dissertação, foi criada a presente subsecção, de modo a ser possível demonstrar de forma ilustrativa e detalhada o funcionamento básico do sistema de rede *overlay*. Na Figura 3.8 é apresentado um exemplo de um pedido de transferência de conteúdo no modo de cliente-servidor, estando o percurso assinalado numa tonalidade azul. Inicialmente, o utilizador efetua a comunicação com o *peer* de acesso mais próximo, recorrendo aos recursos descritos no módulo de *interface* do *peer* de acesso, previamente referido na Figura 3.2. Na *interface*, o utilizador seleciona a opção de pesquisa de conteúdo (passo 1 da Figura 3.8). De seguida, no passo 2, este pedido é encaminhado para o coordenador da rede *overlay*, cuja caracterização modular pode ser visualizada na Figura 3.3. Após receber o pedido de transferência do cliente, o coordenador efetua uma *query* à base de dados incluída no módulo de armazenamento (passo 3), para determinar os *peers* que possuem o conteúdo em questão. No passo 5, o coordenador efetua uma *query* à base de dados que contém o estado da rede *overlay*, de modo a calcular o caminho mais curto até ao *peer* de acesso que efetuou o pedido, de acordo com as estratégias estipuladas pelo *ISP* (passo 4). Após a rota estar calculada, o coordenador notifica o *peer* de acesso com o conteúdo em questão e envia-lhe a rota que deverá ser seguida pelos pacotes na rede *overlay* com informações sobre o conteúdo que foi escolhido pelo utilizador (passo 6). Após esse processo, no passo 7, é efetuado o encaminhamento de nível aplicacional, compreendendo a rota enviada pelo coordenador, até chegar ao *peer* de acesso destino (utilizador que efetuou o pedido). Finalmente, quando os pacotes chegam ao *peer* de acesso mais próximo do utilizador, o pedido é finalizado e o conteúdo é recebido pelo utilizador (passo 8). É de notar, que este exemplo ilustra o modo de transferência cliente-servidor e que o coordenador apenas notifica um *peer* com o respetivo conteúdo pedido pelo utilizador. Porém, poderia notificar múltiplos *peers*, caso o *ISP* em conjunto com o coordenador, estimasse que seria mais vantajoso usar o modo de transferência *P2P*.

3.2 Funcionalidades da Rede Overlay

Nesta secção são abordadas as funcionalidades gerais do sistema, porém, vai ser atribuída uma maior importância aos mecanismos de colaboração com o *ISP*, uma vez que estes representam o foco principal do projeto. Inicialmente, é efetuada uma análise das funcionalidades dos *peers* de acesso (utilizadores finais) e do coordenador do sistema, com a utilização de diagramas de casos de uso. Posteriormente, vão ser abordados os modo de transferência suportados pelo sistema. Por fim, são detalhados os mecanismos de colaboração com o *ISP*.

3.2.1 *Funcionalidades Gerais do Sistema*

Uma vez que esta dissertação retrata a implementação de uma rede *overlay* de partilha de conteúdos, vão ser disponibilizadas funcionalidades básicas e típicas deste serviço. Seguidamente, serão disponibilizadas as funcionalidades do coordenador, em conjunto com os mecanismos de colaboração entre o sistema de rede *overlay* e o *ISP*, sendo-lhes atribuída uma maior relevância e prioridade de desenvolvimento.

As funcionalidades básicas que o utilizador tem disponíveis no seu campo visual, após conexão com o *peer* de acesso mais próximo, são:

- **Pesquisa de conteúdo:** permite ao utilizador pesquisar qualquer conteúdo existente na rede;
- ***Upload* de conteúdo:** permite ao utilizador permitir à rede *overlay* utilizar um conteúdo presente no seu dispositivo, a partir do *upload* de metadados do conteúdo em questão;
- ***Download* de conteúdo:** permite ao utilizador fazer *download* de um conteúdo presente na rede.

As funcionalidades que o coordenador exerce, são:

- **Administrar a base de dados de conteúdos:** efetuar alterações na base de dados com informações relativas aos conteúdos da rede *overlay* associados a cada *peer*;
- **Administrar a base de dados da topologia física/overlay:** efetuar alterações na base de dados relativa ao estado da rede física e *overlay*;
- **Gestão de *peers* e mecanismos de funcionamento da rede overlay:** permitir ao coordenador gerir o sistema da rede *overlay*, administrar pedidos de utilizador, calcular rotas *inter-peers*, assim como, modificar métodos de engenharia de tráfego da rede *overlay*.

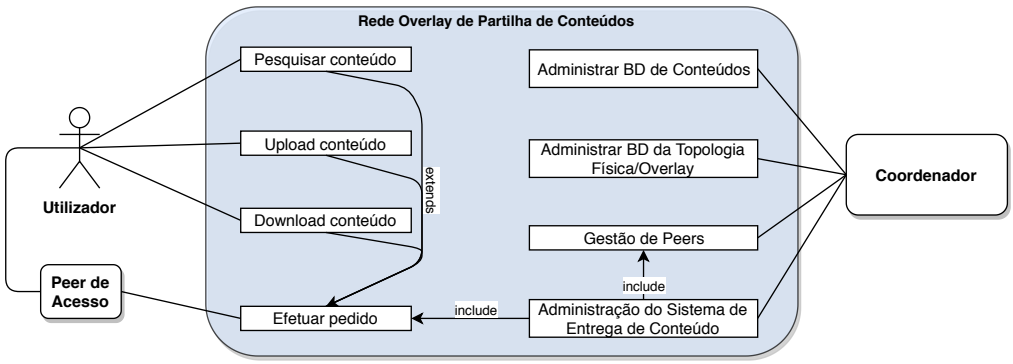


Figura 3.9: Diagrama de casos de uso do sistema de partilha de conteúdos

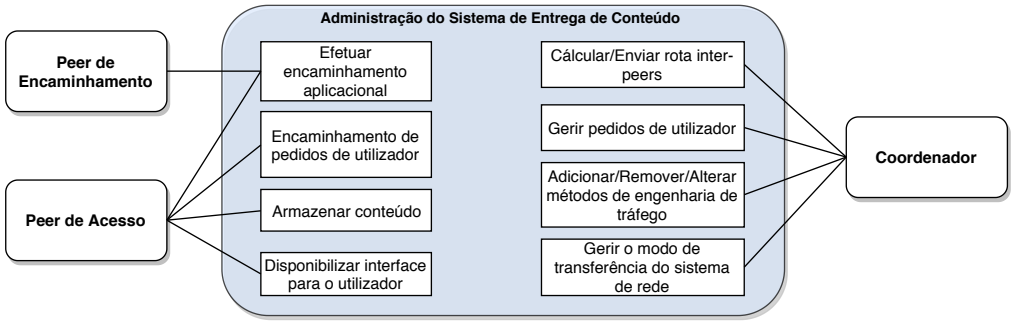


Figura 3.10: Diagrama de casos de uso sobre a administração do sistema de partilha de conteúdos

Na Figura 3.9 está ilustrado o diagrama geral de casos de uso do sistema, onde se pode verificar a vista do utilizador e do coordenador. Além disso, na mesma figura, são abordadas também as funcionalidades do coordenador da rede *overlay*, expressas com maior detalhe no diagrama de casos de uso sobre a administração do sistema, incluído na gestão de *peers*, Figura 3.10. Nesta figura, pode ser especificado com detalhe o papel do coordenador e dos *peers* na rede *overlay*. Pode visualizar-se também como irá ser administrada a rede *overlay*, assim como a comunicação existente entre o coordenador e os *peers*. Pode verificar-se ainda a diferença, em termos funcionais, do *peer* de acesso e do *peer* de encaminhamento. Assim, enquanto que o *peer* de encaminhamento serve só como intermediário e disponibiliza o encaminhamento a nível aplicacional, o *peer* de acesso, disponibiliza a *interface* para o *utilizador*, armazena metadados acerca do mesmo e encaminha os pedidos para o coordenador. O coordenador, por sua vez, tem a tarefa de manter o funcionamento correto dos *peers*, calculando e enviando as rotas de nível aplicacional para os *peers* de acesso que iniciarão o processo de partilha, além de gerir os pedidos dos utilizadores, os métodos de engenharia de tráfego e selecionar o modo de transferência a ser aplicado em cada *download*.

3.2.2 *Modos de Transferência do Sistema de Partilha*

O sistema de distribuição de conteúdos irá funcionar de acordo com dois modos de transferência: cliente-servidor e *P2P*. Estes modos irão ser determinados pelo coordenador, dependendo do pedido do utilizador. Caso o utilizador efetue uma transferência e o conteúdo em questã exista apenas em 1 *peer*, ou o seu tamanho seja pequeno demais para ser utilizado por mais do que 1 *peer* na transferência, o coordenador seleciona o modo cliente-servidor, no qual apenas 1 *peer* transfere o conteúdo para o cliente. Caso contrário, todos os *peers* que contêm o conteúdo, transferem diferentes conjuntos de fragmentos para o cliente, de modo a serem posteriormente colecionados e ordenados, recriando o conteúdo em questão.

3.2.3 *Mecanismos Colaborativos com o ISP*

De forma a que o sistema de partilha de conteúdos suporte mecanismos de colaboração com o *ISP*, foi armazenada a topologia de rede física e *overlay*, onde o sistema atua. Assim, a partir do coordenador da rede *overlay*, o sistema de partilha de conteúdos, ganha a consciência da topologia de rede física, em conjunto com os caminhos mínimos entre dois *routers*, assim como a consciência dos conteúdos armazenados por cada *peer*, armazenadas na rede *overlay*. Deste modo, foram armazenados os *links* percorridos no caminho mínimo do nível da rede do *ISP*, assim como os *routers*, o somatório do número de saltos e o somatório dos custos de encaminhamento, resultantes do percurso mínimo percorrido.

Para que o fornecedor de rede consiga controlar o tráfego *P2P* do sistema de partilha de conteúdos, inicialmente, foram criadas duas estratégias de encaminhamento aplicacional suportadas pelo sistema de rede *overlay*, sendo estas: a estratégia de minimização por custos de encaminhamento e a estratégia de minimização por número de saltos. De seguida, irão ser efetuados tópicos que visam explicar as estratégias abordadas, assim como a proteção de *links*/*routers* e a priorização de *links*/*routers*. Para estes serem descritos com detalhe, vão ser demonstradas figuras exemplificativas do seu funcionamento, a partir de um grafo representante da topologia de rede do *ISP* e de um grafo representante da topologia de rede *overlay*. No grafo da topologia física, os nós representam os nomes dos *routers* e as arestas representam os custos dos *links* entre eles. No grafo da topologia *overlay*, os nós representam os nomes dos *peers* e as arestas representam o nº de saltos do caminho mínimo a nível de rede. Por fim, vão também ser explicadas duas funcionalidades adicionais, que permitem ao *ISP*, efetuar o planeamento de decisões de tráfego e inserir um limite de circulação de conteúdos no sistema de rede *overlay*. Por fim, irá ser abordado com maior detalhe o modo de transferência do sistema.

3.2.3.1 *Estratégias de Encaminhamento Aplicacional*

A Figura 3.11 demonstra um exemplo da aplicação das duas estratégias abordadadas anteriormente, segundo a rota aplicacional com origem em *"P1"* e destino em *"P4"*. Inicialmente, é calculada a rota mínima no nível de rede, correspondente ao grafo da rede física, no qual se pode verificar que o percurso mínimo escolhido de *"R1"* para o *"R4"* (correspondente ao caminho entre *"P1"* e *"P4"*) foi destacado a azul, visto ser o caminho com menor custo de encaminhamento agregado. Porém, visto que a estratégia de minimização de saltos é a optada pelo sistema por defeito, a rota escolhida não irá ser a que tem menor número custo de encaminhamento agregado, mas sim, a do caminho com menor número de saltos. Assim, caso o *ISP* não modificasse a estratégia de encaminhamento aplicacional utilizada pelo sistema, a rota que iria ser percorrida, seria, o caminho representado a verde, visto que o número de saltos é de 2 (caminho físico: L1-L2, caminho aplicacional: P1-P2-P4). Caso o

ISP defini-se a estratégia de minimização de custos de encaminhamento, como estratégia a utilizar no sistema, a rota que iria ser percorrida, seria a que está destacada com a cor azul (rota física: R1-R6-R7-R4, rota aplicacional: P1-P4), sendo percorrido o caminho mínimo calculado no nível de rede.

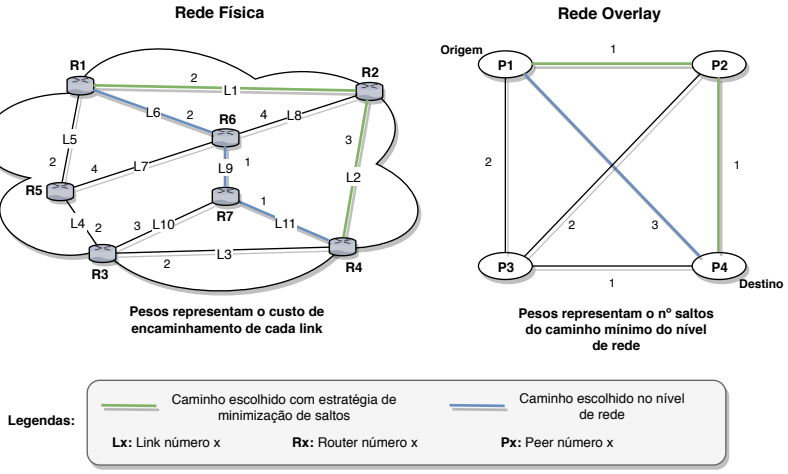


Figura 3.11: Exemplo da estratégia de minimização de número de saltos para a rota aplicacional com origem em P1 e destino P4

3.2.3.2 *Proteção de Links/Routers*

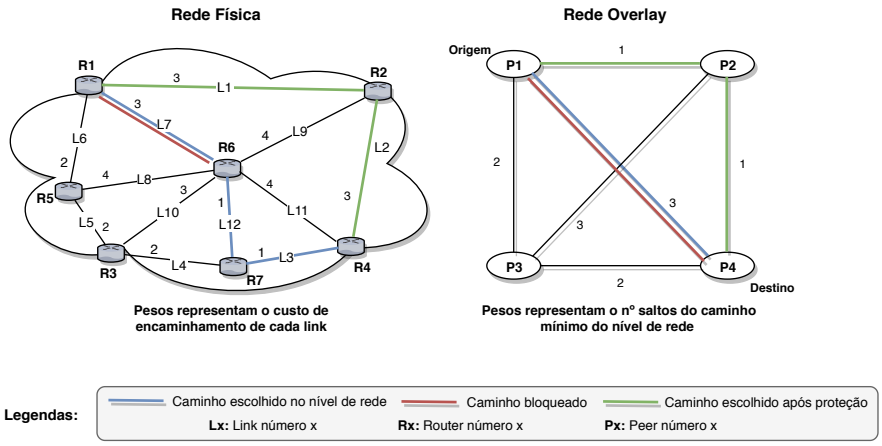


Figura 3.12: Exemplo de proteção do link L7 para a rota com origem em P1 e destino P4

Dado que os sistemas *overlay* geram grandes quantidades de tráfego *P2P*, de modo a que o *ISP* consiga impedir que o tráfego aceda a certos pontos da rede física, foi criado um mecanismo de proteção orientado a *links/routers*. Este mecanismo, pode ser utilizado, por exemplo, para não deixar degradar a qualidade de outro tráfego com maior prioridade, colocando assim o tráfego *P2P* em segundo plano.

A Figura 3.12 ilustra um exemplo específico da aplicação deste mecanismo de proteção, aplicado ao *link "L7"*, para a rota aplicacional com origem em *"P1"* e destino *"P4"*. Na

rede *overlay* e na rede física, são utilizados números para distinguir, respetivamente, os *peers* e os *routers*, sendo que quando um número de um *peer* é igual ao número de um *router*, significa que o *peer* corresponde ao *router* em questão. A rota percorrida após a aplicação da proteção foi assinalada a verde, a rota protegido foi assinalada a vermelho e a rota do caminho mínimo a nível de rede foi representada a azul. Inicialmente, foi calculado o caminho mínimo a nível de rede do *ISP*, sendo que o caminho com menor custo de encaminhamento agregado foi assinalado a azul (rota física: R1-R6-R7-R4, rota aplicacional: P1-P4). Posteriormente, é aplicada uma proteção ao *link "L7"*. Assim, em qualquer uma das estratégias suportadas, o sistema tentará procurar um caminho que não contenha o *link "L7"* na sua travessia. Por fim, o sistema tenta calcular outro caminho existente entre *"R1"* e *"R4"*, encontrando o caminho assinalado a verde (rota física: R1-R2-R4, rota aplicacional: P1-P2-P4). Em ambas as estratégias, este é o caminho escolhido, visto que é o que tem menor número de saltos e menor custo de encaminhamento agregado, dado que o *link "L7"* foi protegido.

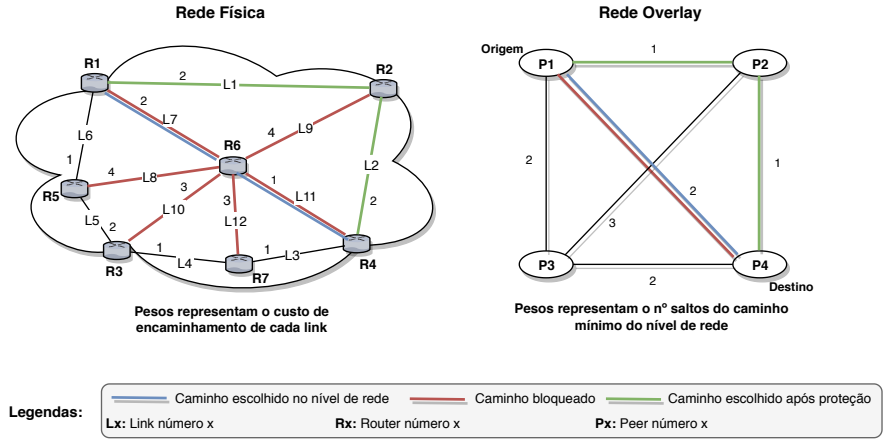


Figura 3.13: Exemplo de proteção do router R6 para a rota aplicacional com origem em P1 e destino P4

A Figura 3.13 ilustra um exemplo específico da aplicação deste mecanismo de proteção, aplicado ao *router "R6"*, para a rota aplicacional com origem em *"P1"* e destino *"P4"*. Inicialmente, é calculado o caminho com menor custo de encaminhamento na rede física, sendo determinado o caminho assinalado a azul. Posteriormente, é efetuada uma proteção ao *router "R6"*, ou seja, a todos os *links* pertencentes ao próprio. Deste modo, em qualquer uma das estratégias suportadas, o sistema tentará procurar um caminho, que não tenha nenhum dos *links* que foram protegidos (*L7, L9, L11, L8, L10, L12*), na sua travessia. Por fim, o sistema tenta calcular outro caminho existente entre *"R1"* e *"R4"*, encontrando o caminho assinalado a verde. Em ambas as estratégias, este é o caminho escolhido, visto que é que tem menor número de saltos e menor custo de encaminhamento agregado (rota física:

R1-R2-R4, rota aplicacional: P1-P2-P4), dado que os *links* pertencentes ao *router R6* foram protegidos.

3.2.3.3 *Priorização de Links/Routers*

Assim como a proteção de *links/routers*, a estratégia de priorização é uma ferramenta de controlo de tráfego *P2P* que pode ser utilizada pelo *ISP*, em momentos críticos que possam ser vantajosos. Este mecanismo foi criado, para que o *ISP* possa priorizar o tráfego *P2P* em certos pontos de rede, de modo a balancear a carga do sistema e/ou a aproveitar *links* com pouca carga, de modo a que sejam mais utilizados.

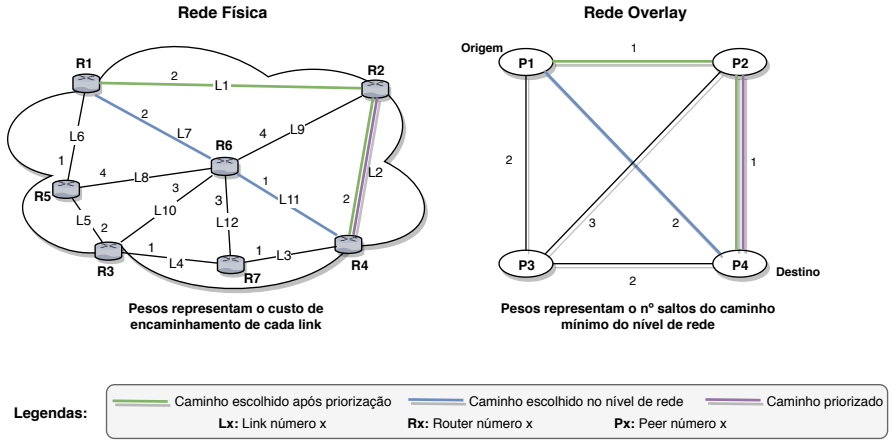


Figura 3.14: Exemplo de priorização do link L2 para a rota aplicacional com origem em P1 e destino P4

A Figura 3.14 ilustra um exemplo específico da aplicação deste mecanismo de priorização, aplicado ao *link "L2"*, para a rota aplicacional com origem em *"P1"* e destino *"P4"*. Na rede *overlay* e na rede física, são utilizados números para distinguir, respetivamente, os *peers* e os *routers*, sendo que quando um número de um *peer* é igual ao número de um *router*, significa que o *peer* corresponde ao *router* em questão. A rota percorrida após a aplicação da priorização foi assinalada a verde, a rota priorizada foi assinalada a roxo e a rota do caminho mínimo percorrido no nível de rede foi representada a azul. Inicialmente, foi calculado o caminho mínimo a nível de rede do *ISP*, sendo que o caminho com menor custo de encaminhamento agregado foi assinalado a azul. Posteriormente, é aplicada uma priorização ao *link "L2"*. Assim, em qualquer uma das estratégias suportadas, o sistema tentará procurar um caminho, no qual se verifique a existência do *link "L2"* na sua travessia. De seguida, o sistema tenta calcular outro caminho existente entre *"R1"* e *"R4"*, encontrando o caminho assinalado a verde. Em ambas as estratégias, este é o caminho escolhido, visto que é o que tem menor número de saltos e menor custo de encaminhamento agregado, além de conter o *link* que foi priorizado (rota física: R1-R2-R4, rota aplicacional: P1-P2-P4).

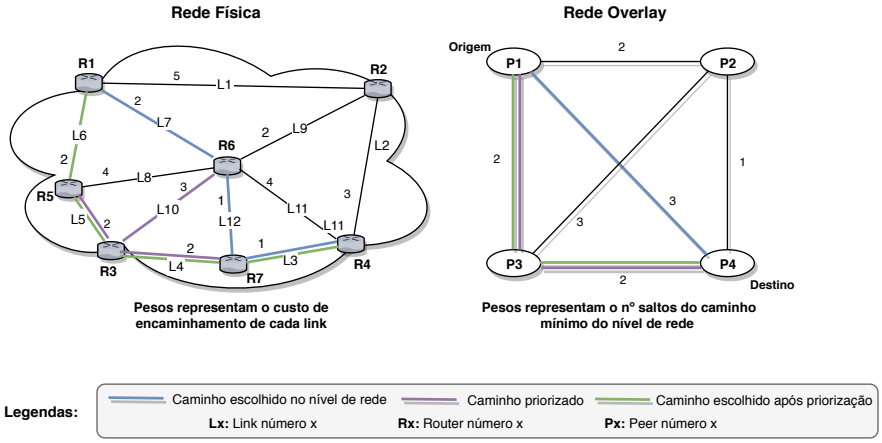


Figura 3.15: Exemplo de priorização do router R3 para a rota aplicacional com origem em P1 e destino P4

A Figura 3.15 ilustra um exemplo específico da aplicação deste mecanismo de priorização, aplicado ao *router "R3"*, para a rota aplicacional com origem em *"P1"* e destino *"P4"*. Inicialmente, é calculado o caminho com menor custo de encaminhamento na rede física, sendo determinado o caminho assinalado a azul. Posteriormente, é efetuada uma priorização ao *router "R3"*, ou seja, a todos os *links* pertencentes ao próprio. Deste modo, em qualquer uma das estratégias suportadas, o sistema tentará procurar um caminho, no qual se verifique a existência de um ou mais dos *links* pertencentes ao *router "R3"* (*L4, L5, L10*), na sua travessia. De seguida, o sistema tenta calcular outro caminho existente entre *"R1"* e *"R4"*, encontrando o caminho assinalado a verde. Em ambas as estratégias, este é o caminho escolhido, visto que é que tem menor número de saltos e menor custo de encaminhamento agregado (rota física: R1-R5-R3-R7-R4, rota aplicacional: P1-P3-P4), dado que contém dois *links* priorizados pelo *ISP* (L5 e L4).

3.2.3.4 *Planeamento de Decisões do ISP*

De modo a estabilizar o tráfego associado ao sistema de rede *overlay* e/ou que percorre a rede física, foi criada uma possibilidade para o *ISP* definir um planeamento de proteção/-priorização de *links/routers* entre datas. Deste modo, quando o *ISP* analisa períodos críticos, nos quais o sistema está sobrecarregado com pedidos de utilizadores ou um ponto físico na rede do *ISP* está congestionado, é atribuído ao *ISP* a possibilidade de resolver esta situação. A solução consiste em planear horários de proteção ou priorização de *routers/links* em certos momentos do dia-a-dia críticos. Esta medida é importante, pois fornece ao *ISP* uma maior flexibilidade na gestão da rede, na qual este apenas necessita definir o *router/link* a ser protegido ou priorizado, na tentativa de resolver o problema.

3.2.3.5 *Limite de Circulação de Conteúdos*

Outra medida que permite ao *ISP* controlar a sobrecarga de pedidos de transferência na rede *overlay* é o mecanismo de limite de circulação de conteúdos. Com esta medida, o *ISP* consegue definir um limite de circulação no coordenador, ativado no caso da soma dos tamanhos dos conteúdos em circulação nesse instante seja superior ao respetivo valor do limite. Caso essa condição seja verdadeira, os clientes que queiram efetuar transferências, posteriormente, serão colocados numa fila de espera, até que oa soma do tamanho das transferências em curso seja menores que o valor do limite. Com este mecanismo pretende-se que o *ISP* possa limitar o tráfego da rede *overlay* em situações críticas em que este considere necessário, onde a proteção ou priorização de *links/routers* não seja adequada, devido a um excesso de pedidos de *download* por toda a rede.

3.3 SUMÁRIO

Este capítulo abordou a arquitetura do sistema de rede *overlay*, assim como a descrição das entidades representantes do mesmo e especificação de mecanismos de colaboração com o *ISP* que irão ser implementados. Inicialmente, foi descrita a arquitetura do sistema, sendo apresentada posteriormente, uma visão modular das entidades compostas pelo próprio. Por fim, foram introduzidos os mecanismos de colaboração com o *ISP*, assim como a sua explicação, por meio de figuras exemplificativas.

4

IMPLEMENTAÇÃO

Neste capítulo são descritas as tecnologias utilizadas na implementação da rede *overlay* para partilha de conteúdos controlada pelo *ISP*, assim como a apresentação da visão global do sistema em si. Na primeira secção (4.1), são abordadas as linguagens de programação, bases de dados e outras ferramentas utilizadas. Na secção 4.2, é feita a descrição da implementação com detalhe, abordando as aplicações que foram implementadas e os mecanismos colaborativos com o *ISP*. Posteriormente, na secção 4.3, é explicado com detalhe o funcionamento da *interface desktop* implementada para os clientes e da página *web* administrativa, dirigida para o administrador do *ISP*. Por fim, na secção 4.4, é efetuado o sumário do capítulo.

4.1 TECNOLOGIAS UTILIZADAS

A rede de partilha de conteúdos abordada na tese, foi implementada com a utilização da linguagem de programação *Java*. Para manter o estado da aplicação, também foram utilizadas bases de dados não relacionais, *MongoDB* [45] e *Neo4j* [46], respetivamente para armazenar os detalhes sobre os ficheiros partilhados na rede *overlay* e para armazenar as topologias física/*overlay*. Para os clientes da rede *overlay*, foi utilizado o *JavaFX* [47] para implementar uma *interface desktop*. Para o *ISP* foi criada uma página *web* de administração da rede, onde foi utilizada a Representational State Transfer (REST) Application Program Interface (API) de *Java*, denominada *Spark Framework* [48], com a função de transformar o *input* do *ISP* na página *web* em comandos para o coordenador da rede *overlay* exercer. Posteriormente, para a criação da página *web* em si, foram utilizadas as linguagens *HTML* Cascading Style Sheets (CSS), em conjunto com *Javascript* [49] e algumas das suas *frameworks*, como: *Neovis* [50], *JQuery* [51] e *Bootstrap* [52]. Por fim, foi utilizada como ferramenta de testes, a aplicação *CORE* [53], com a função de emular um cenário de uma topologia física administrada por um *ISP* e executar os vários componentes desenvolvidos do sistema proposto.

4.2 Rede Overlay para Partilha de Conteúdos

A rede *overlay* implementada, como o nome indica, é um sistema de rede de partilha de conteúdos onde é utilizado o poder computacional dos *peers* aderentes, para aumentar a escalabiliade do sistema. Porém, como já foi abordado anteriormente, ao contrário de outros sistemas semelhantes, esta rede é administrada por um *ISP*. Para isso, existe um nó central denominado coordenador, que efetua o controlo da rede *overlay* com as configurações pedidas pelo *ISP*.

Seguidamente, vão ser abordadas: a vista geral do sistema e, posteriormente as aplicações implementadas.

4.2.1 *Vista Geral do Sistema*

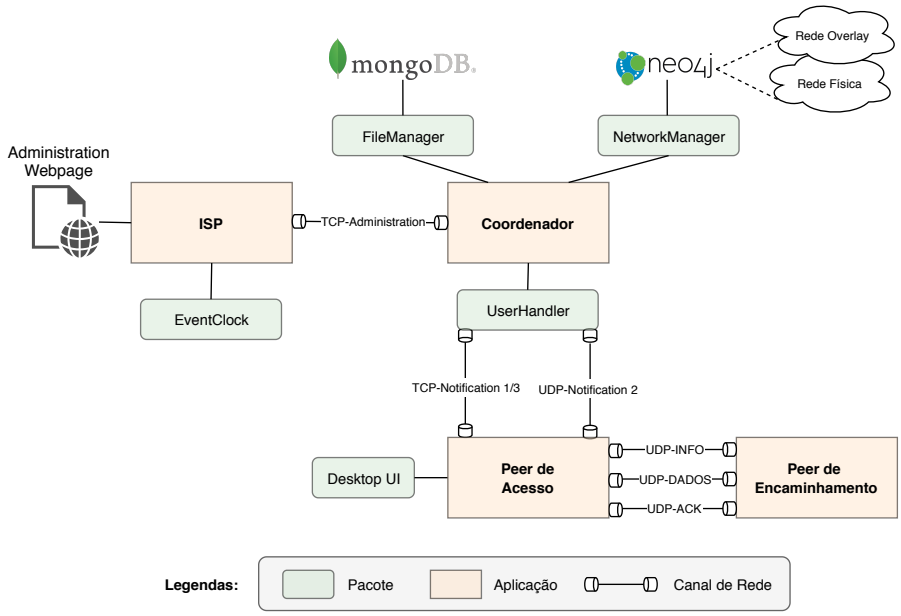


Figura 4.1: Vista geral do sistema de rede

A Figura 4.1 ilustra a vista geral do sistema de rede *overlay* de partilha. Nesta, pode visualizar-se a constituição do sistema, definido por quatro principais aplicações que são interligadas por canais de rede, em constante comunicação entre elas: o *ISP*, o coordenador, o *peer* de acesso (cujo nome foi denominado *AccessPeer*) e o *peer* de encaminhamento (denominado *RoutingPeer*). O *ISP* é a entidade administrativa da rede física, na qual a rede *overlay* irá atuar, e como tal, estará em constante comunicação com o coordenador a partir de um canal de rede *TCP*. Como se observa na Figura 4.1, o administrador do *ISP* terá acesso a uma página *web* de administração, onde irá efetuar as configurações da rede *overlay*, que posteriormente irão ser enviadas para o coordenador executar. O *peer* de encaminhamento

é a entidade responsável pela distribuição de pacotes a nível aplicacional, executando apenas a função de encaminhamento de pacotes para outros *peers*. É de notar que a quantidade de *peers* de encaminhamento está relacionada com o aumento da flexibilidade de controlo do tráfego da rede *overlay* pelo *ISP*. Por outro lado, o *peer* de acesso é o que proporciona ao utilizador a *interface desktop* gráfica para os clientes do sistema de partilha de ficheiros. Este, está conectado (por dois canais de rede distintos) ao pacote *UserHandler* (incluído no coordenador), responsável por efetuar a gestão de pedidos dos *peers* de acesso. Existe para isso, um canal de rede *TCP* que trata o encaminhamento de pedidos do utilizador e metadados do *download* (*PDU* de notificação de classe 1/3 respetivamente) e outro canal de rede *UDP*, para notificar os *peers* de acesso que têm que começar o encaminhamento aplicacional (*PDU* de notificação de classe 2). A comunicação entre a entidade *peer* de acesso e *peer* de encaminhamento, é efetuada a partir de canais de rede *UDP* distintos, sendo que, no primeiro são encaminhados pacotes com informação sobre o conteúdo a ser enviado, no segundo os dados do conteúdo em si e no último os pacotes de confirmação. Por fim, a maior e mais complexa entidade, onde foram elaborados os métodos colaborativos com o *ISP* é o coordenador. Esta, inclui distintos componentes (*threads*) que ajudam na gestão da rede, destacando-se, o *UserHandler* , o *NetworkManager* e o *FileManager*. O *UserHandler* é uma das componentes mais importantes, sendo responsável por efetuar a gestão dos clientes da rede, controlo de filas de espera e limite de circulação de ficheiros. O *NetworkManager* armazena a topologia de rede física e o estado de rede da topologia *overlay*, a partir da comunicação com a base de dados não relacional, *Neo4j*. É nesta componente, que os algoritmos de decisão de tráfego aplicacional são efetuados, daí poder ser considerada a mais importante, visto que o foco do sistema proposto incide sobre os métodos de colaboração com o *ISP*. Por fim o *FileManager* é a *thread* que comunica com a base de dados (*MongoDB*), na qual, são armazenados os metadados dos conteúdos presentes no sistema da rede *overlay*.

4.2.2 *Representação das Topologias de Rede no Neo4j*

Como explicado anteriormente, o *Neo4j* é a base de dados responsável por armazenar as informações da topologia física e topologia *overlay*. Utilizou-se, devido à estrutura da base de dados ser orientada a grafos, sendo ótimo para represar topologias de rede. Além disso, é possível aplicar algoritmos de minimização de custos, como o *Dijkstra*, tornando-se ainda mais valioso para a implementação do sistema de rede proposto. A estrutura desta base de dados consiste em grafos que são compostos por nós, relacionamentos (arestas) e propriedades (pares "nome-valor"que representam qualidades de um nó ou relacionamento).

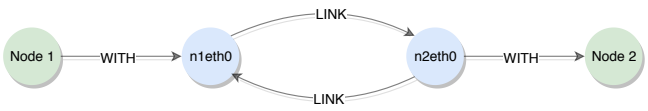


Figura 4.2: Representação da topologia física no Neo4j

A Figura 4.2 ilustra o modo como foi armazenada a topologia física. Foram criados dois tipos de entidades: *Node* e *Interface*. A primeira entidade representa um *router* ou outro dispositivo de rede, localizado na topologia física administrada pelo *ISP*, tendo sido armazenadas nas suas propriedades o seu *ID* e o nome. A segunda entidade representa uma *interface* pertencente a um dispositivo de rede, tendo sido armazenadas nas suas propriedades o seu *IP* e o nome. Foram estabelecidos os relacionamentos *WITH* e *LINK*. O primeiro representa o relacionamento entre os dispositivos de rede e as suas *interfaces* e o segundo a ligação entre duas *interfaces* pertencentes a dispositivos de rede distintas. No relacionamento *LINK* foi armazenado o respetivo custo de encaminhamento, para posteriormente ser possível efetuar o algoritmo de *Dijkstra*.

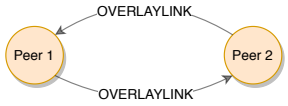


Figura 4.3: Representação da topologia overlay no Neo4j

A Figura 4.3 ilustra o modo como foi armazenada a topologia *overlay*. Nesta apenas existe uma entidade e um relacionamento, respetivamente o *peer* e *OVERLAYLINK*. Na entidade *peer*, foram armazenados: o *IP*, nome, *ID* do *router* associado ao *peer* e o seu próprio *ID*. No relacionamento representante do caminho entre dois *peers*, foram armazenados: o próprio *ID*, lista de custos de encaminhamento, somatório do número de saltos, lista de *IPs* das interfaces, lista de nomes de *Nodes* e somatório dos custos de encaminhamento.

4.2.3 *Modos de Transferência Suportados*

O sistema de partilha de conteúdos suporta dois modos de transferência: o modo cliente-servidor e o modo *P2P*. A escolha do modo, é determinada tendo em conta duas condições, o tamanho do conteúdo em questão e o número de *peers* de acesso que contém o conteúdo e que permitiram a sua utilização pela rede *overlay*, denominados *seeders*. O modo cliente-servidor consiste no *download* integral de um conteúdo, efetuado entre dois *peers* de acesso. Desta forma, o mesmo é escolhido, quando o tamanho do conteúdo é inferior ao tamanho máximo estipulado para os pacotes de dados (16 KB), ou quando apenas existe um *seeder* para o conteúdo em questão. Por conseguinte, o modo *P2P* consiste no *download* em paralelo de um conteúdo, dividido em fragmentos, consoante o número de *seeders* disponíveis. Este modo é acionado pelo sistema, quando o tamanho do conteúdo é superior ao tamanho máximo estipulado para os pacotes de dados utilizados e também quando o número de *seeders* que contêm o conteúdo em questão, é superior a 1.

4.2.4 *Mecanismos de Colaboração com o ISP*

Como foi referido no capítulo anterior, foram implementados mecanismos de colaboração entre a rede *overlay* e o *ISP*. Como previsto, foram implementados os seguintes: escolha da estratégia de encaminhamento aplicacional a ser utilizada pelo sistema (minimização do número de saltos ou minimização de custos de encaminhamento) , proteção/priorização de *links/routers*, planeamento de decisões por datas e limite de circulação de conteúdos. Todos estes mecanismos abordados são controlados pelo *ISP* a partir da utilização da página *web* de configurações administrativas.

4.2.4.1 *Estratégias de Encaminhamento Aplicacional*

As estratégias de encaminhamento suportadas pela rede *overlay* são, nomeadamente, a minimização de custos de encaminhamento e a minimização do número de saltos (na camada de rede) entre *peers* da *overlay* (por defeito é a estratégia escolhida pelo sistema *overlay*). O *ISP* pode escolher qual a estratégia que considerar mais apropriado na página *web*, que depois irá ser enviado ao coordenador, para o pacote *UserHandler* atualizar a estratégia, modificando posteriormente as decisões de tráfego a nível aplicacional.

4.2.4.2 *Proteção de Links/Routers*

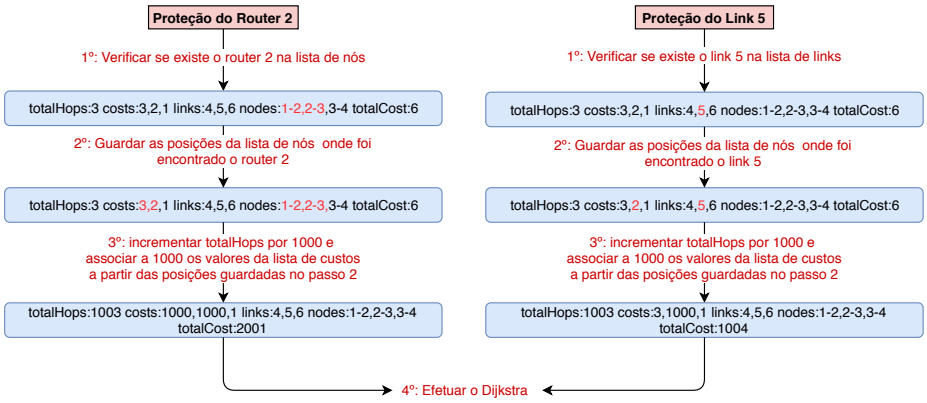


Figura 4.4: Ilustração do algoritmo de proteção de links/routers

A proteção de *links/routers* foi implementada no pacote *NetworkManager*, presente no coordenador, em conjunto com o pacote *UserHandler*, de modo a interligar a consciencialização da rede física e *overlay*, com o processo de encaminhamento de pedidos dos utilizadores finais. Para tal, foi implementado um algoritmo de proteção com uso posterior do algoritmo de minimização de custos, *Dijkstra*. Como foi ilustrado na Figura 4.3, foram armazenados: lista de custos de encaminhamento de cada *link*, somatório da lista de custos de encaminhamentos (custo total por minimização de encaminhamento), somatório do número de saltos (custo total por minimização de número saltos), lista de nomes de cada *link* e lista de

nomes de cada *router* presentes no caminho da topologia física correspondente ao caminho na rede *overlay* entre cada par de *peers*. Recorrendo à utilização desses valores armazenados no grafo da topologia *overlay* representado no *Neo4j*, o algoritmo de proteção foi elaborado a partir dos seguintes passos descritos na Figura 4.4:

- **1º Passo:** para cada par de *peers*, verificar se existe o *link/router* na lista de *links/routers* presente em cada relacionamento;
- **2º Passo:** caso o 1º passo tenha resultados válidos, identificar os relacionamentos que compreendam essa condição e guardar as posições da lista de *links/routers* onde a condição se tenha verificado;
- **3º Passo:** para o caso da estratégia de encaminhamento aplicacional utilizar minimização por número de saltos, incrementar o valor do somatório do número de saltos por 1000 (valor hipotético para infinito), nos relacionamentos identificados no passo 2. Caso a estratégia de encaminhamento aplicacional utilize minimização por custos de encaminhamento, alterar a lista de custos de encaminhamento a partir das posições guardadas no passo 2, pelo valor 1000, para todos os relacionamentos identificados no passo 2. Antes de efetuar as alterações, é necessário guardar os valores iniciais das propriedades dos relacionamentos, para, no caso do *ISP* remover a proteção do *link/router*, ser efetuada a substituição dos novos valores pelos iniciais, mantendo a coerência;

Após estes passos, quando é efetuado um pedido de *download* pelos utilizadores, é efetuado o *dijkstra* na rede *overlay*, tendo em conta o tipo de estratégia de encaminhamento a ser efetuado. Visto que os valores dos *links* a proteger são modificados para 1000, o algoritmo de minimização de custos, na maior parte dos casos, irá calcular o caminho mínimo, de modo a tentar evitar os *links* progetidos. A Figura 4.4 representa um exemplo ilustrativo deste algoritmo, para o caso de proteção de um *link* e de um *router*.

4.2.4.3 *Priorização de Links/Routers*

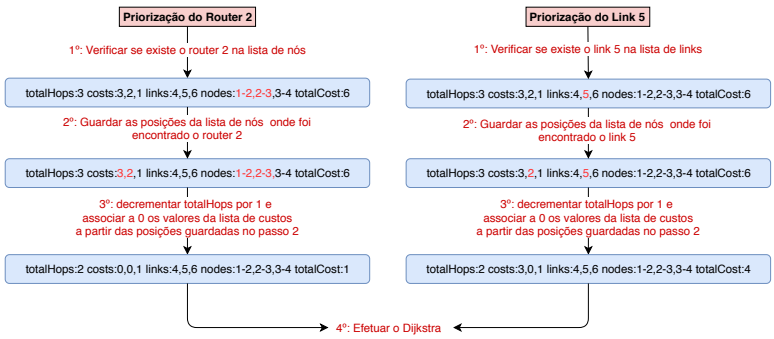


Figura 4.5: Ilustração do algoritmo de priorização de links/routers

De forma semelhante à proteção de *links/routers*, o algoritmo de priorização foi implementado no pacote *NetworkManager* em conjunção com o *UserHandler*, com uso posterior do algoritmo de minimização de custos, *Dijkstra*. Tendo em conta novamente as propriedades dos relacionamentos entre *peers* na figura 4.3, o algoritmo de priorização consiste nos seguintes passos:

- **1º Passo:** - Para cada par de *peers*, verificar se existe o *link/router* na lista de *links/routers*, para todos os seus relacionamentos;
- **2º Passo:** - Caso o 1º passo tenha resultados válidos, identificar os relacionamentos que compreendam essa condição e guardar as posições da lista de *links/routers* onde a condição se tenha verificado;
- **3º Passo:** - Para o caso da estratégia de encaminhamento aplicacional utilizar minimização por número de saltos, basta decrementar o valor do somatório do número de saltos por 1, nos relacionamentos identificados no passo 2. Caso a estratégia de encaminhamento aplicacional utilize minimização por custos de encaminhamento, alterar a lista de custos de encaminhamento a partir das posições guardadas no passo 2 pelo valor 0. Antes de efetuar as alterações, é necessário guardar os valores iniciais das propriedades dos relacionamentos, para que, quando o *ISP* remover a proteção do *link/router*, seja efetuada a substituição dos novos valores pelos iniciais, mantendo a coerência;

Novamente, quando é efetuado um pedido de *download* pelos utilizadores, é efetuado o *dijkstra* na rede *overlay*, tendo em consideração o tipo de estratégia de encaminhamento a ser efetuada. Visto que os valores dos *links* a priorizar foram modificados para 0, o algoritmo de minimização de custos, em muitos casos, irá calcular o caminho mínimo percorrendo um dos *links* priorizados. A Figura 4.5 representa um exemplo ilustrativo deste algoritmo para o caso de priorização de um *link* e de um *router*.

4.2.4.4 *Planeamento de Decisões por Datas*

O planeamento de decisões por datas foi implementado no pacote *EventClock*, presente no *ISP*. Este método, consiste na possibilidade do *ISP* agendar uma proteção/priorização de *links/routers* entre duas datas. Para tal, é efetuada de x em x segundos uma verificação do tempo atual, comparativamente à data inicial escolhida pelo *ISP*, para que quando a condição se verifique, seja enviada a decisão para o coordenador. Caso a data atual seja maior ou igual à data final escolhida pelo *ISP*, é enviado o cancelamento da decisão para o coordenador, de modo a ser efetuado o cancelamento da proteção/priorização no sistema de rede *overlay*.

4.2.4.5 *Limite de Circulação de Conteúdos*

O mecanismo de limite de circulação de conteúdos foi implementado diretamente na aplicação do coordenador, no pacote *UserHandler*. Foi utilizada uma condição de verificação, partilhada por todos os *peers* de *acesso* (clientes da rede *overlay*), que verifica se a cada pedido de *download*, o tamanho total dos ficheiros em circulação na rede (tamanho agredado dos *downloads* a serem efetuados) é menor ou igual ao limite máximo proposto pelo *ISP*. Caso essa condição não se verifique, é enviada uma mensagem informativa aos novos clientes que querem efetuar o *download*, transmitindo-lhes que estão presentes numa fila de espera e qual a sua respetiva posição na fila. Os pedidos pendentes serão então progressivamente atendidos à medida que outras transferências se finalizem.

4.3 Componente Gráfica do Sistema de Rede

A componente gráfica de interação com o utilizador é composta por duas distintas aplicações: aplicação *desktop* e aplicação *web*. A primeira, é a aplicação que fica em contacto com os utilizadores da rede *overlay* de partilha de conteúdos, tendo-se optado pela utilização de um ambiente *desktop* para interagir com o sistema de rede implementado. A segunda, designa um ambiente de interação com o *ISP*, propositadamente para configuração dos diversos mecanismos projetados, a partir da utilização de uma página *web*.

4.3.1 *Aplicação Desktop*

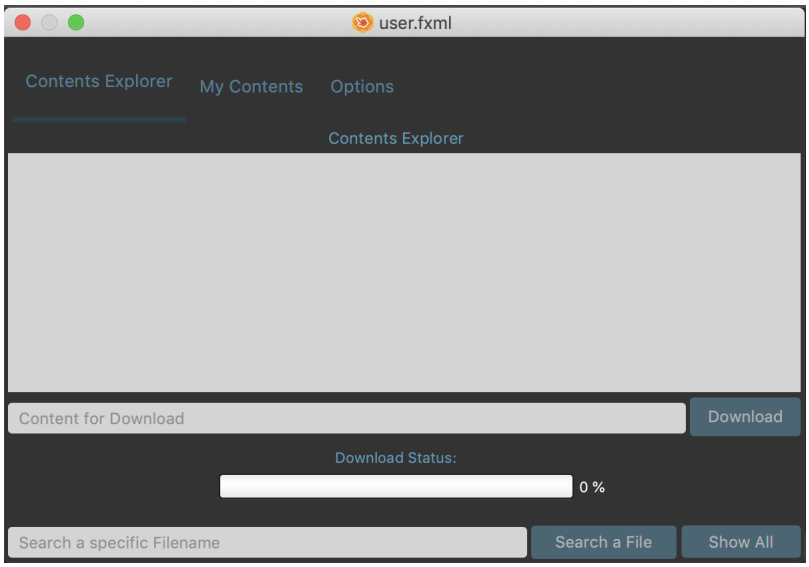


Figura 4.6: Vista principal da aplicação desktop (Contents Explorer)

A aplicação *desktop* é a componente de interação entre o utilizador (*peer* de acesso) e o sistema da rede *overlay*. A Figura 4.6 representa a visão geral desta aplicação, divida em quatro vistas separadas por separadores superiores: *Contents Explorer*, *My Contents* e *Options*.

4.3.1.1 *Contents Explorer*

A vista inicial da aplicação, visível quando o utilizador começa o programa ou carrega na aba superior *Contents Explorer*, representa o ambiente gráfico de *downloads* do utilizador (Figura 4.6). Nesta, os conteúdos do sistema (armazenados na base de dados *MongoDB*), são apresentados na vista de texto correspondente ao título *"Contents Explorer"*, organizados por: nome do conteúdo, nº de *seeders* com o conteúdo e tamanho do conteúdo. A mesma, pode ser modificada pelos botões: *Search a File* e *Show All*. O primeiro, permite aplicar um filtro por nome, consoante o que for escrito pelo utilizador na vista de texto *"Search a specific Filename"*. O segundo botão, mostra todos os conteúdos existentes na base de dados (sendo que pode ser utilizado para atualizar os dados). Para efetuar um *download*, o utilizador escolhe um conteúdo da lista que aparece na vista de texto *"Contents Explorer"*, que posteriormente irá aparecer na vista de texto *"Content for Download"*, onde o utilizador pode confirmar o conteúdo selecionado, e finalizar clicando no botão *Download*. Após o clique no botão de *download*, este fica bloqueado até o *download* ser acabado, e posteriormente, irá aparecer em baixo, o estado do *download* representado por texto e por uma barra de progresso com a atualização temporal do mesmo.

4.3.1.2 *My Contents*

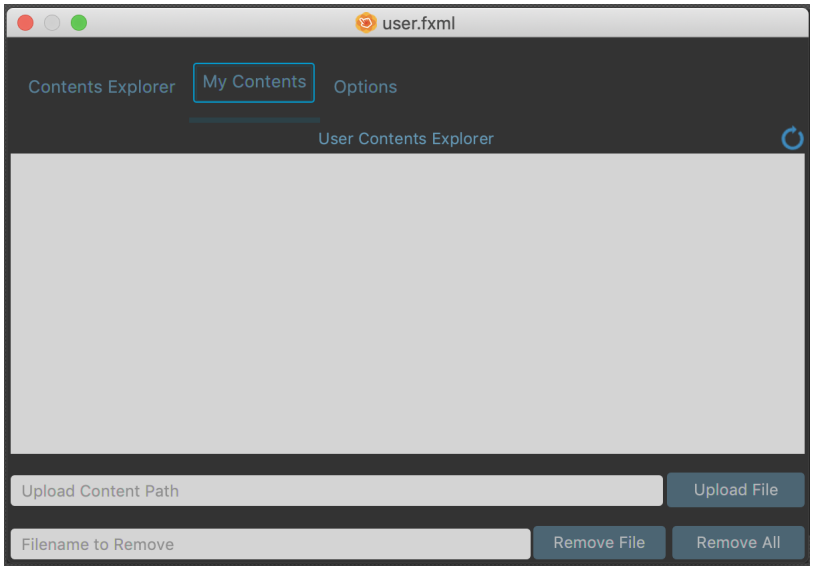


Figura 4.7: Segunda vista da aplicação desktop (My Contents)

A segunda vista da aplicação, visível quando o utilizador clica na aba superior *My Contents*, representa o ambiente gráfico de gestão dos conteúdos que correspondem ao próprio (Figura 4.7). Nesta, os conteúdos pertencentes ao utilizador aparecem na vista de texto correspondente ao titulo (*"User Contents Explorer"*), que pode ser atualizada ao clicar no botão de atualização (representado por uma imagem) e localizado no canto superior direito em cima da vista em questão. Adicionalmente, o utilizador pode fazer *upload* dos metadados de um conteúdo, permitindo a utilização do mesmo ao sistema de partilha, ao escrever a localização do conteúdo na vista de texto *"Upload Content Path"*, e clicando no botão *Upload File* para finalizar a operação. Por fim, tem a possibilidade de remover todos os seus ficheiros, clicando no botão *Remove All* ou a partir da escolha de um ficheiro na vista de texto correspondente ao título *"User Contents Explorer"*, selecionar apenas um ficheiro e proceder à sua remoção.

4.3.1.3 *Options*

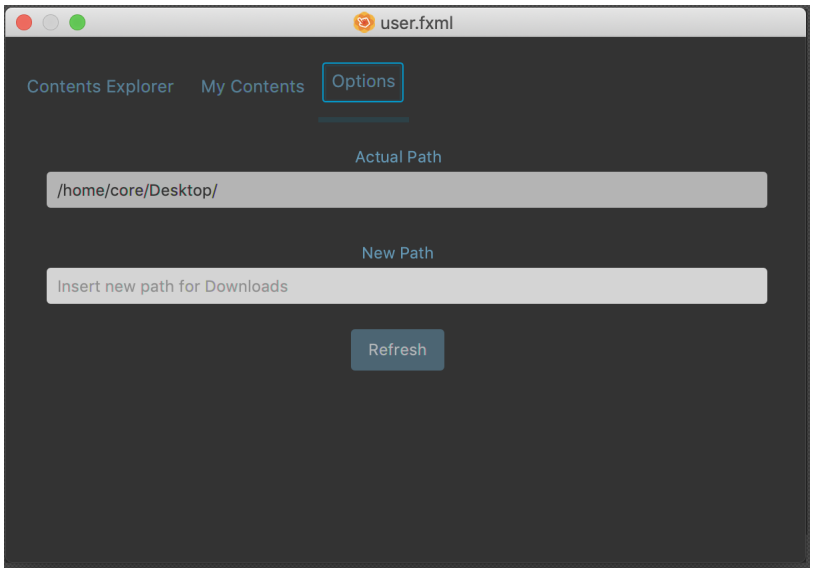


Figura 4.8: Vista final da aplicação desktop (Options)

A terceira e última vista da aplicação, visível quando o utilizador clica na aba superior *Options*, representa o ambiente gráfico de seleção da localização do *download* do conteúdo (Figura 4.8). É a vista mais simples, compreendida por uma vista de texto imutável com a localização de *download* e uma vista com o texto *Insert new path for Downloads*, que permite ao utilizador escrever uma nova localização e carregar no botão *Refresh* para a atualizar.

4.3.2 *Aplicação Web*

A aplicação *web* é a componente administrativa visual que interage com o administrador de rede (*ISP*). Nesta, são efetuadas e tomadas decisões críticas na tentativa de controlar o tráfego, e ao mesmo tempo tentar maximizar a qualidade de serviço do sistema de rede *overlay* de partilha de conteúdos implementado. A Figura 4.9 representa a visão geral da página *web* composta por quatro vistas separadas pelas abas laterais localizadas à esquerda: *Tecnical Configuration*, *Content Database*, *Download Routes* e *Overlay Topology*.

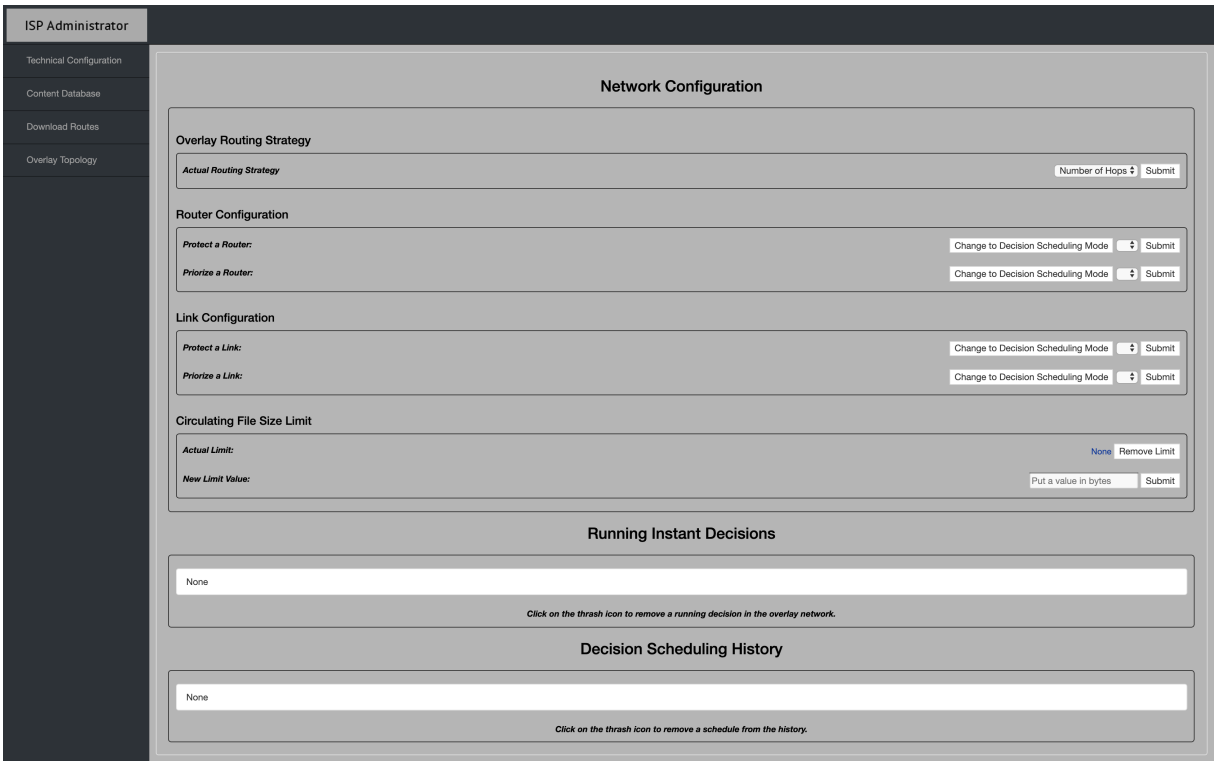


Figura 4.9: Vista principal da página web de configurações do ISP (Technical Configuration)

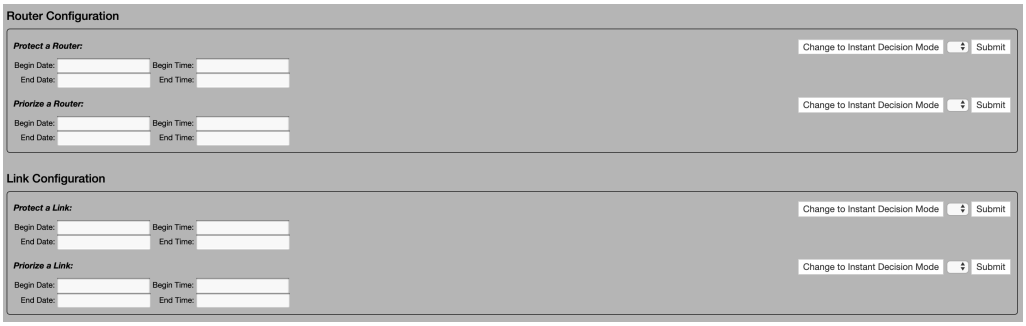


Figura 4.10: Vista principal da página web com a opção de agendamento de decisões do ISP ativada (Technical Configuration)

4.3.2.1 *Technical Configuration*

A vista inicial da página *web*, visível quando o administrador carrega na aba lateral *Technical Configuration* ou quando a página é carregada pela primeira vez, representa o ambiente gráfico de controlo de decisões a aplicar na rede *overlay* (Figura 4.9). Neste ambiente de configurações técnicas, o *ISP* tem a possibilidade de: escolher o tipo de algoritmo de encaminhamento aplicacional (estratégia de minimização por número de saltos ou estratégia de minimização de custos de encaminhamento), adicionar/remover proteção/priorização de *router* presente na rede física, adicionar/remover proteção/priorização de *link* presente na rede física e adicionar/alterar/remover limite máximo correspondente ao somatório do tamanho de todos os ficheiros em circulação na rede *overlay*.

Para alterar a estratégia de encaminhamento aplicacional, o administrador apenas necessita escolher uma das estratégias existentes na lista de texto situada à direita no campo relativo ao título *"Overlay Routing Strategy"* e pressionar o botão *Submit*.

No caso do administrador pretender adicionar instantaneamente uma proteção/priorização de *router*, simplesmente necessita de navegar até ao campo relativo ao título *"Router Configuration"* e, escolher o *router* na lista de texto situada à direita do campo de texto *"Protect a Router"* e *"Priorize a Router"*, respetivamente. Para finalizar, o administrador apenas necessita de pressionar o botão *Submit*. Por outro lado, caso este pretenda efetuar um planeamento de proteção/priorização de *router*, terá que pressionar o botão de *Change to Decision Scheduling Mode*, tal como indica a Figura 4.10. Após esse passo, irão aparecer quatro campos agrupados em pares, que representam a data inicial em conjunto com o tempo inicial e a data final em conjunto com o tempo final, para o administrador escolher e, posteriormente, efetuar o mesmo passo de escolha de *router*, descrito anteriormente. Relativamente ao caso de se adicionar proteção/priorização de *link*, a explicação é a mesma que nos casos descritos anteriormente relativos ao *router*, porém este processo será efetuado dentro do campo relativo ao título *Link Configuration*.

Para adicionar um limite de ficheiros em circulação, o administrador tem que inserir o novo valor na caixa de texto *"Put a value in bytes"*, relativamente ao campo com o título *"Circulating File Size Limit"*, pressionando o botão *Submit* à frente, para finalizar a decisão. O limite irá aparecer à direita do texto *"Actual Limit"* e, caso o administrador pretenda remover completamente o mesmo, apenas é necessário o clique no botão *Remove Limit*.

Sempre que são efetuadas decisões instantâneas de proteção/priorização pelo *ISP*, estas são armazenadas em múltiplos itens com informações sobre: a hora em que foi efetuada a decisão instantânea, o tipo de decisão efetuada (proteção/priorização), o alvo em questão (*router/link*) e um ícone de um caixote do lixo no final. Os itens são armazenados na lista relativa ao título *"Running Instant Decisions"*, presente na Figura 4.9. Adicionalmente, o administrador pode remover decisões presentes na lista em causa, pressionando o ícone do caixote do lixo. No caso das decisões planeadas pelo *ISP*, estas são armazenadas em

itens com informações sobre: a hora em que foi efetuada a decisão de planeamento, o tipo de decisão efetuado (proteção/priorização), o alvo em questão (*router/link*) e a duração do agendamento. Os itens são armazenados na lista relativa ao título *"Decision Scheduling History"* (Figura 4.9). É de salientar que as decisões planeadas não podem ser removidas instantâneamente. Por esse motivo, é apresentado um histórico de planeamento de decisões, com a possibilidade de remoção de um *item*, que não alterará o funcionamento do planeamento removido em questão.

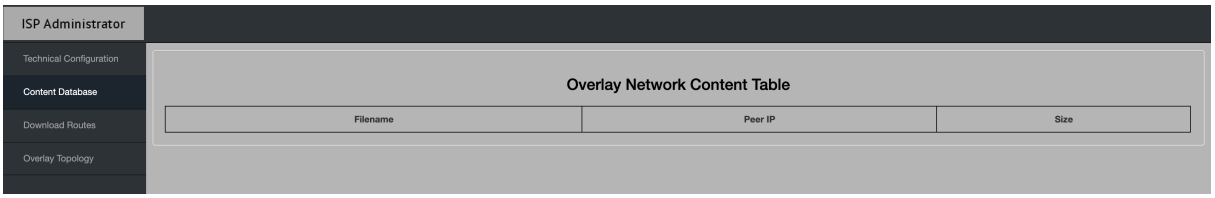


Figura 4.11: Segunda vista da página web, que representa os conteúdos presentes na rede overlay, organizados por uma tabela (Content Database)

4.3.2.2 *Content Database*

A segunda vista da aplicação *web*, visível quando o administrador clica na aba lateral *Content Database*, representa uma vista em tabela com a informação dos conteúdos que circulam na rede *overlay* (Figura 4.11). A tabela contém informações sobre o nome, *IP* e tamanho de cada conteúdo. A sua atualização é efetuada quando o utilizador clica na aba lateral *Content Database*.

4.3.2.3 *Download Routes*

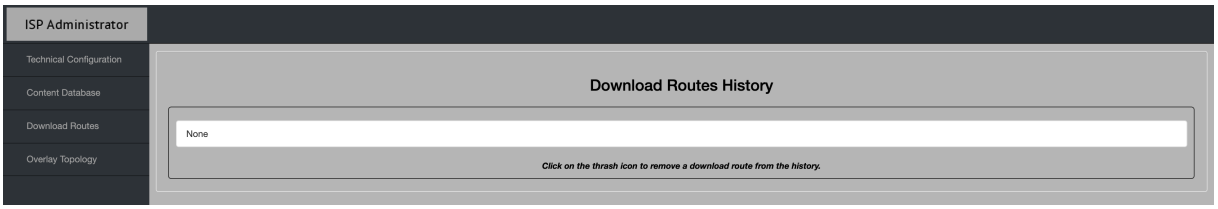


Figura 4.12: Terceira vista da página web, que representa uma lista com as rotas de download efetuadas no sistema

A terceira vista da aplicação *web*, visível quando o administrador carrega na aba lateral *Download Routes*, representa um ambiente de visualização simples com informações sobre os pedidos de *download* pelos clientes (Figura 4.12). Este ambiente contém uma lista constituída por itens, que representam um histórico de pedidos de *download* efetuados no sistema de rede. Cada item, tem informações sobre: a data do pedido de *download*, *ID* do *peer* de

acesso que efetuou o pedido, modo de transferência utilizado, rota aplicacional e física percorrida no *download*. Adicionalmente, visto que se trata de um histórico, foi adicionado o ícone em forma de caixote do lixo, para o caso do administrador querer remover itens e minimizar o espaço ocupado pela lista. A atualização do histórico será efetuada sempre que o utilizador clicar na aba lateral *Download Routes*.

4.3.2.4 *Overlay Topology*

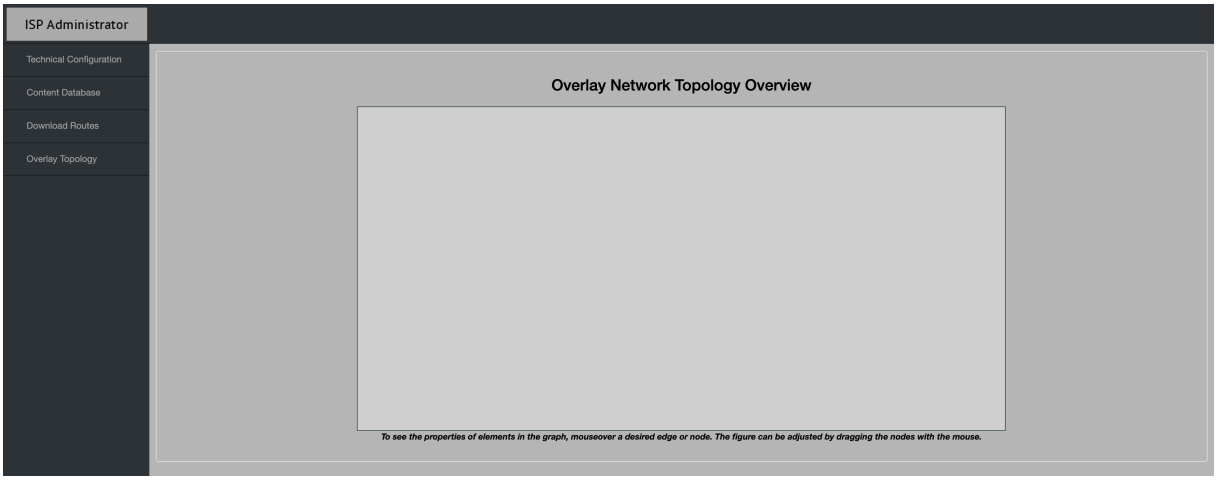


Figura 4.13: Quarta vista da página web, que representa uma ilustração da rede overlay representada em grafo (Overlay Topology)

Finalmente, tal como se pode observar na Figura 4.13, a quarta e última vista da aplicação *web*, visível quando o administrador carrega na aba lateral *Overlay Topology*, ilustra um ambiente gráfico com a representação da topologia *overlay* e das suas propriedades a partir de uma estrutura baseada em grafo. O ambiente gráfico em causa, ilustra a representação gráfica da rede *overlay* produzida e armazenada no *Neo4j*. Neste grafo, é possível visualizar todas as propriedades que existem em cada vértice e aresta, focando com o cursor do rato a entidade desejada. Além disso, também é possível remodelar o grafo, arrastando as entidades individualmente, com o cursor do rato. O grafo que representa a rede *overlay* é atualizado sempre que o utilizador clicar na aba lateral *Overlay Topology*.

4.4 SUMÁRIO

Este capítulo abordou a implementação do sistema de rede *overlay* para partilha de conteúdos, em conjunto com mecanismos de colaboração com o *ISP*. Inicialmente, foram referidas as tecnologias utilizadas na implementação do sistema em si e na criação de *interfaces* gráficas orientadas ao utilizador e ao *ISP*. Posteriormente, foram abordadas as aplicações desenvolvidas, assim como uma visão técnica representativa do sistema implementado. Por fim,

foram demonstradas as funcionalidades das *interfaces* gráficas referidas anteriormente. No capítulo seguinte, irão ser desenvolvidos testes com o propósito de analisar os mecanismos de colaboração com o *ISP* implementados.

5

TESTES E ANÁLISE DE RESULTADOS

Após a definição da arquitetura e implementação do protótipo relativo ao sistema de rede *overlay* controlado pelo *ISP* para partilha de conteúdos, é necessário neste capítulo testar, analisar e discutir os resultados, relativamente à veracidade das funcionalidades do sistema em si e retirar as respetivas inferências. Na análise de testes vão ser incluídas as funcionalidades relativas à administração da rede *overlay* pelo *ISP*, explicadas na subsecção 4.2.4 do capítulo anterior, dado que é o foco principal do trabalho. Inicialmente, na secção 5.1, é abordado o ambiente de testes e ferramentas auxiliares utilizadas. Posteriormente, na secção 5.2, são explicados com detalhe as métricas utilizadas para calcular a veracidade dos mecanismos implementados. Na secção 5.3, são apresentados os cenários de teste, em conjunto com a análise de resultados. Por fim, na secção 5.4, é apresentado um sumário dos resultados obtidos.

5.1 APRESENTAÇÃO DOS TESTES

Esta secção tem como objetivo descrever o ambiente de testes e ferramentas auxiliares utilizadas, em conjunto com cinco cenários que foram efetuados para comparar os mecanismos de controlo de tráfego presente na rede *overlay*.

5.1.1 *Ambiente de Testes*

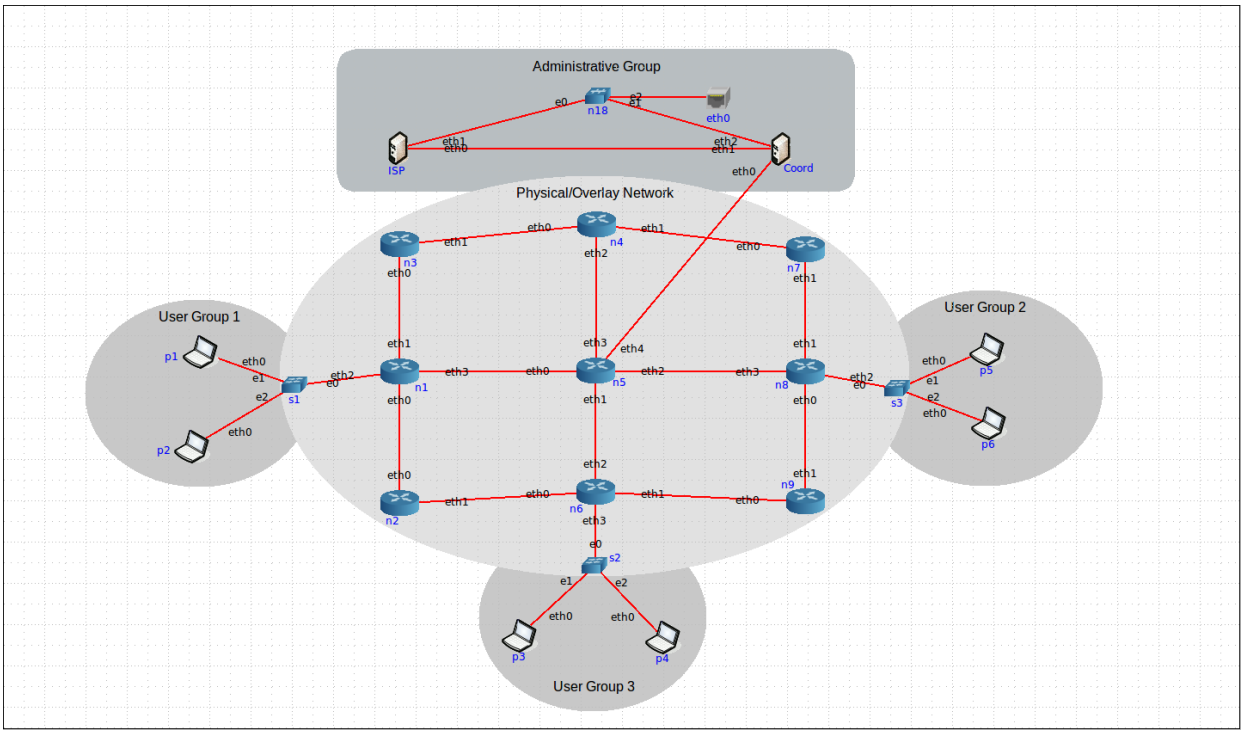


Figura 5.1: Topologia de rede física utilizada para testes

A fim de testar o sistema desenvolvido, utilizou-se uma ferramenta de emulação de rede, denominada *CORE* [53]. Esta, foi utilizada para emular a rede física subjacente à rede *overlay*, composta por *routers*, *links* que interligam os *routers*, servidores que representam o coordenador e o *ISP* e, uma entidade denominada *RJ45*, que representa um conector de rede, utilizado para aceder à rede local do computador anfitrião. Além disso, o *CORE* foi utilizado, visto ser uma ferramenta capaz de simular, não só redes, mas também computadores, permitindo assim a execução das aplicações desenvolvidas neste projeto, dentro do cenário emulado.

O protocolo de encaminhamento da rede física utilizado para os testes foi o Open Shortest Path First (OSPF). Este protocolo baseia-se no algoritmo *Dijkstra* (também utilizado na implementação da rede *overlay*) para determinar o caminho mais curto entre uma origem e um destino. Por omissão, o *CORE* atribui o custo de 10 a cada *link*, no entanto, este valor irá ser modificado de acordo com os objetivos dos vários testes a realizar.

A topologia de testes que vai ser utilizada nos diferentes cenários é ilustrada pela Figura 5.1. Esta, integra um conjunto de *routers* de *core* da rede, juntamente com seis *peers* de acesso distribuídos em três grupos de dois, de modo a serem situados em diferentes localizações. Além disso, a topologia de rede também integra o coordenador da rede *overlay* e o *ISP*,

representados por servidores. Estes, estão conectados à ferramenta *RJ45*, de modo a conseguirem conectar-se com o anfitrião da máquina, no qual serão executados os servidores, *Neo4J Server* e *MongoDB Server*.

5.1.2 *Ferramentas Auxiliares*

Para analisar os dados sobre o tráfego que circula entre os *routers* da topologia de rede apresentada anteriormente na Figura 5.1, foi utilizada a ferramenta *vcmd* [54] em conjunto com a ferramenta *vnstat* [55]. O *vcmd* permite a conexão com os nós de rede emulados pelo *CORE*. A partir da sua utilização, é possível executar o programa *vnstat* que efetua a monitorização do tráfego em circulação nos *routers* virtuais.

5.2 METODOLOGIA

De forma a preparar o ambiente de testes, foram executados via terminal, na máquina anfitriã, dois comandos para inicialização das bases de dados: *MongoDB Server* e*Neo4j Server*. O primeiro, responsável pelo armazenamento dos metadados dos conteúdos partilhados na rede *overlay*, e o segundo, responsável pelo armazenamento da topologia física/*overlay*. Os comandos executados no terminal para inicializar as bases de dados, foram: o comando *mongod* e *neo4j console*. Após este processo, foi aberto um ambiente emulado do sistema operativo *Xubuntu 12.04 LTS*, a partir da *VirtualBox* [56], onde foi executado o *CORE* (versão 4.6).

```
{
 "topology":{
   "nome_no1 id_no1":{
     "id": "valor",
     "interfaces":{
       "nome_interface1": ip_interface1",
       .
       .
     },
     "links":[
       "ip_interface_origem1 ip_interface_destino1",
       .
       .
     ],
     "routingCosts":[
       "custo ospf do primeiro link",
       .
       .
     ]
   },
   "nome_no2 id_no2":{
     .
```

```
        .
    }
  }
}
```

Figura 5.2: Exemplo da organização da topologia física armazenada em ficheiro

Após o processo de inicialização, foi executado o *script* "setScriptsOnCore"(Figura A.2 do Anexo A) com os argumentos: "ID da sessão do CORE"e "nome do cenário". Este *script*, colocou *scripts* de execução das aplicações implementadas (*Coordenador, ISP, Accesspeer* e *RoutingPeer*), nos ambientes virtuais correspondentes ao *ID* da sessão e aos nomes dos nós da topologia do *CORE* (Figura 5.1). Após este passo, foi executada a simulação dentro do *CORE* e foram abertos os terminais correspondentes ao coordenador, *ISP*, e *peers* de acesso/encaminhamento necessários para cada cenário. Porém, para executar as aplicações implementadas, é ainda necessário passar-lhes como argumento alguns parâmetros de inicialização:

- Para executar a aplicação do coordenador é necessário passar como argumento a localização de um ficheiro com extensão JavaScript Object Notation (JSON), que armazena as propriedades da topologia física (exceto os servidores e os *peers*), cuja organização consta na Figura 5.2, para a topologia física ser inicializada no *Neo4J*.
- Na aplicação *AccessPeer* é necessário passar três argumentos: o *ID* do *router* associado ao *peer* na rede física, o *ID* do peer na rede *overlay* e o nome do cenário.
- Na aplicação *RoutingPeer* é necessário passar dois argumentos: *ID* do *router* na rede física e *ID* do *router* na rede *overlay*.
- Na aplicação do *ISP*, é necessário abrir um terminal para executar a *REST API* e outro para abrir a página *web*.

Numa fase posterior, para efetuar os testes, foi criado um *script* de captura de tráfego transmitido em tempo real (Figura A.3 do Anexo A), para cada um dos *links* da topologia de rede física do *CORE*, no qual foi utilizado a ferramenta *vcmd* em conjunto com a *vnstat*. De seguida, foi utilizado outro *script*, denominado *calculateMean.sh*, para gerar a média de tráfego transmitido em todos os *links* de cada *router* (Figura A.1 do Anexo A). Finalmente, de modo a concluir a análise, foi utilizado um *script* em *python* (generateGraph.py) para gerar gráficos de barras para cada cenário, em função do tráfego médio transmitido em todos os *links* da topologia de rede.

Foram também armazenadas as rotas percorridas peloas transferências efetuadas na rede física/*overlay*, recorrendo à utilização da página *web* do *ISP*. Posteriormente, foram recolhidos os tempos de *download*, gerados a partir da aplicação *desktop* para os *peers* de acesso, com a utilização do método *System.currentTimeMillis()*, que retorna o tempo corrente em milissegundos. Este, foi aplicado antes e depois de cada *download* começar, tendo sido registada a sua duração, convertida em segundos e arredondada a duas casas decimais, sendo posteriormente armazenada num ficheiro com o nome e o número do cenário.

Os métodos abordados anteriormente, foram utilizados para todos os cenários que irão ser descritos, exceto para o cenário 1 e o cenário 5. Nestes, apenas foram comprovadas as funcionalidades do modo de transferência e limite de circulação do sistema de rede *overlay*, através de capturas da aplicação *desktop* e da página *web* do *ISP*. Deve-se salientar que os testes foram executados numa escala menor que o idealizado, devido à carga computacional exercida em tempo real, quer pela máquina virtual, quer pelos servidores utilizados (*Neo4J Server* e *MongoDB Server*).

5.3 Cenários de Teste e Resultados

5.3.1 *Cenário 1 - Modos de Transferência*

5.3.1.1 *Descrição*

Neste cenário, pretende-se demonstrar os dois modos de transferência suportados pelo sistema referidos no Capítulo 4. Como se pode visualizar na Figura A.4 do Anexo A, são criados três *peers* de acesso: *AccessPeer1*, *AccessPeer3*, *AccessPeer5*. Estes, são executados nos nós *p1*, *p3* e *p5* da topologia de rede física e que estão interligados respetivamente aos *routers n1*, *n6* e *n8* (Figura 5.1 do Anexo A). Através da aplicação *Desktop*, é adicionado o mesmo conteúdo nos *peers* de acesso, *AccessPeer1* e *AccessPeer5*, sendo também adicionado outro conteúdo no *AccessPeer5*. Finalmente, são iniciadas duas transferências a cada um dos conteúdos por parte do *AccessPeer3*, de forma a ser testado o modo de transferência cliente-servidor e *P2P*, respetivamente.

5.3.1.2 *Resultados*

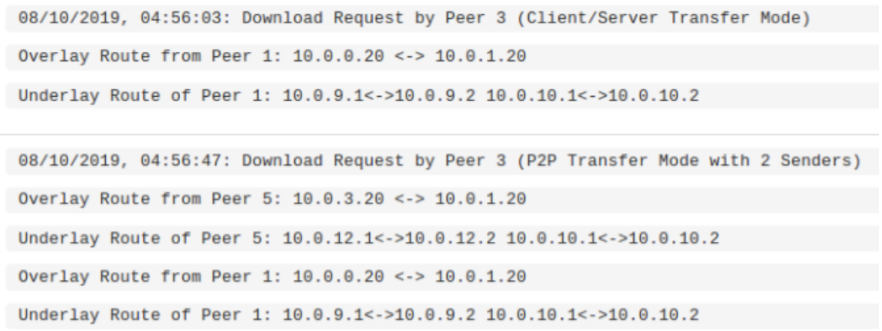


Figura 5.3: Vista da página web do ISP, com a lista de downloads efetuados para o cenário 1

A partir dos resultados obtidos na Figura 5.3, é possível analisar que na primeira transferência é utilizado o modo cliente-servidor e, também apresentada a rota tomada na rede *overlay/underlay*, transmitida pelo *peer* de acesso remetente de *ID* 1 (*p1*). Enquanto que, na segunda transferência se verifica a escolha do modo de transferência *P2P* com dois remetentes, sendo que, é também apresentada a respetiva rota tomada na rede *overlay* e *underlay*, a partir dos *seeders* com *ID* igual a 1 e 5.

5.3.2 *Cenário 2 - Estratégia de Encaminhamento da Rede Overlay*

5.3.2.1 *Descrição*

Este cenário simula um caso hipotético, no qual o administrador de rede posicionaria de modo estratégico um *peer* de encaminhamento num determinado *router*, com a finalidade de modificar a rota do tráfego *P2P*, quando selecionada a estratégia de encaminhamento de minimização de *links* da rede física. Isto, visto que a escolha da estratégia de encaminhamento aplicacional por minimização de custos, resultaria numa rota aplicacional com um trajeto idêntico ao do nível da rede do *ISP*.

5.3.2.2 *Resultados*

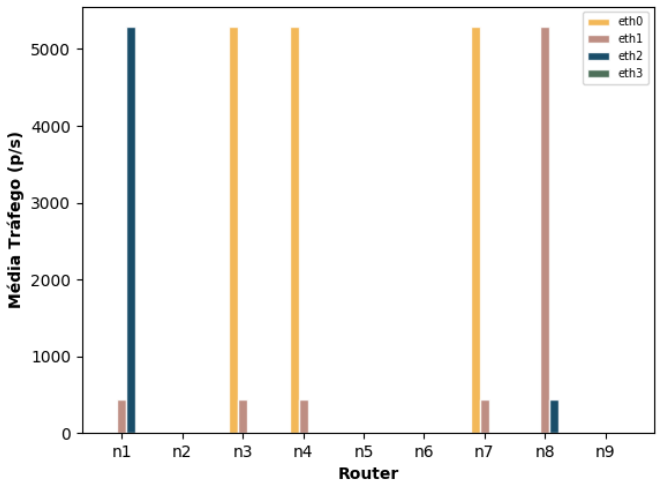


Figura 5.4: Gráfico de barras representativo do tráfego médio transmitido (em pacotes por segundo) no cenário 2, monitorizado antes do posicionamento do RoutingPeer11 (n5)

Com base nos resultados dos valores de tráfego médio transmitido, apresentados na Figura 5.4, conclui-se que, antes da aplicação do *peer* de encaminhamento, apenas se verifica tráfego nos *routers*: *n1*, *n3*, *n4*, *n7*, *n8* (Figura 5.1). A partir desta informação, prevê-se que o trajeto inicial tomado de *p1* para *p5*, assuma a rota da rede física *n8->n7->n4->n3->n1*, na qual o somatório do número de saltos admite o valor de 4 e o valor do somatório dos custos de encaminhamento o valor de 12. O tempo registado nesta transferência foi de 5.40 segundos.

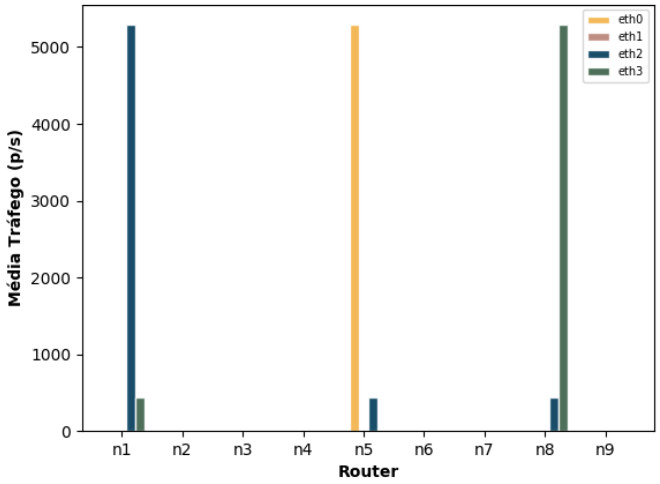


Figura 5.5: Gráfico de barras representativo do tráfego médio transmitido (em pacotes por segundo) no cenário 2, monitorizado depois do posicionamento do RoutingPeer11 (n5)

Após o posicionamento estratégico do *peer* de encaminhamento no *router n5*, e a partir da Figura 5.5, é possível verificar que o tráfego médio é apenas estabelecido nos *routers*, *n1, n5 e n8*. Isto é, a segunda transferência efetuada com o *peer* de encaminhamento adicional, teve a sua rota na rede física, redireccionada para: *n8->n5>->n1*, na qual o valor da soma de número de saltos diminuiu de 4 para 2, porém o valor do somatório de custos de encaminhamento aumentou de 12 para 20. Nesta transferência o tempo registado foi de 5.53 segundos, tendo sido verificado apenas, um aumento de tempo de transferência aproximado em 2.4% $((5.53 - 5.4)/5.4$. Desta forma, o tráfego da rede *overlay* percorreu menos *links* da topologia do *ISP*, tendo pois um menor impacto na mesma, sendo que o tempo registado também não influencia significativamente o serviço de partilha do sistema.

Para além disso, foram analisadas as duas rotas na rede *underlay/overlay* das transferências, antes e depois da colocação do *peer* de encaminhamento, através da página *web* do *ISP*, presente na Figura A.16 em anexo, em conjunto com a topologia de rede, com divulgação dos endereços de *IP* dos *links* (Figura A.15 do Anexo A).

5.3.3 *Cenário 3 - Proteção de Links/Routers*

5.3.3.1 *Descrição*

Este cenário simula um possível caso real, no qual o *ISP* identifica uma rota física congestionada com tráfego *P2P* e quer proteger o caminho num determinado *link/router*, de modo a desviar parte do tráfego em determinados pontos críticos identificados pelo mesmo.

Identificado o caso de uso para este cenário, o próprio está dividido em dois testes, de forma a efetuar a comparação entre o modo de decisão instantânea e o modo de planeamento de decisão relativo à proteção de *links/routers*. Em ambos os modos foi alterado o valor dos custos *OSPF* para 15 nos *links* que interligam os *routers*, *n1-n2* e *n8-n9*, de modo a que a rota mínima escolhida no nível de rede excluísse imediatamente as rotas com os *links* alterados, sendo que, por defeito, os custos de todos os outros *links* é 10. Para garantir o modo de transferência *P2P* e obter um caso de congestionamento onde se possa recorrer à proteção de *routers* e de *links*, foi adicionado à rede *overlay* o mesmo conteúdo pelos *AccessPeer1* e *AccessPeer5*, com o propósito do *AccessPeer3* fazer o pedido de *download* e receber o conteúdo dividido por duas rotas distintas (*p1->p3* e *p5->p3*). Além disso, a estratégia de encaminhamento utilizada foi a de minimização do número de *links* percorridos na rede *overlay* (utilizada por defeito), sendo que neste cenário, as duas estratégias convergem sempre para a mesma rota.

No modo de decisão instantânea, inicialmente, foram inicializados os *peers p1, p3* e *p5* que correspondem aos *routers n1, n6* e *n8*, respetivamente. Em segundo lugar, foi efetuado o pedido de *download* por parte do *AccessPeer3*, de modo a que o *AccessPeer1* e o *AccessPeer5* enviassem o conteúdo dividido em fragmentos. Com a alteração dos custos *OSPF* nos *links*

referidos no inicio da secção, foi analisado que os *links* que interligam os *routers*, *n1-n5* e *n5-n8*, são os mais congestionados. Portanto, tomando o papel administrativo do *ISP*, foi posicionado o *RoutingPeer8* no *router n2*, e, através da página *web* de administração foi efetuada uma proteção ao *link* que interliga o *router n1* e *n5* (Figura A.8 do Anexo A). De seguida, foi efetuado o *download* do mesmo conteúdo pelo *AccessPeer3*. Finalmente, foi adicionado o *RoutingPeer15* correspondente ao nó *n9*, adicionando uma proteção ao *router n5* (Figura A.9 do Anexo A), ou seja, a todos os seus *links*, e efetuou-se novamente o *download* por parte do *AccessPeer3*.

No modo de planeamento de decisão, foi efetuada uma proteção ao *router n5* (a todos os *links* pertencentes ao próprio) a partir da página *web* do *ISP*, desde uma data inicial até uma data final, como apresentado na Figura A.10 em anexo. Finalmente, foram efetuados três *downloads* pelo *AccessPeer3*: um antes da data inicial, um após a data inicial e o último após a data final.

A topologia de rede *overlay* utilizada neste cenário, pode ser visualizada na Figura A.6 em anexo.

5.3.3.2 *Resultados*

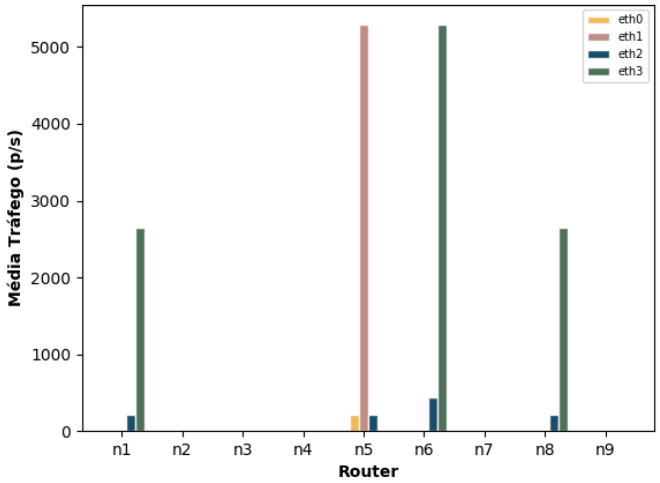


Figura 5.6: Gráfico de barras representativo do tráfego médio transmitido (em pacotes por segundo) no cenário 3, monitorizado antes da aplicação de proteção de link/router

Neste cenário, inicialmente foi efetuada a monitorização dos *links* na rede física, de modo a obter-se a rota inicial, sem a aplicação do mecanismo de proteção. Como tal, é possível visualizar na Figura 5.6 que o tráfego médio transmitido é apenas registado nos *routers*: *n1*, *n5*, *n6* e *n8*, destacando-se os *routers n5* e *n6*, por possuírem o valor mais alto de transmissão de tráfego. Este fenómeno, deve-se ao facto do tráfego na topologia de rede física percorrer em simultâneo o *router n5* e *n6*, sendo que as únicas rotas possíveis com destino a *p3*,

são, *n8->n5->n6*, através do *seeder p5* e *n1->n5->n6*, através do *seeder p1*. Em ambas as rotas o somatório do número de saltos assume o valor 2 e o somatório dos custos de encaminhamento o valor 20. O tempo registado por esta transferência foi de 6.47 segundos.

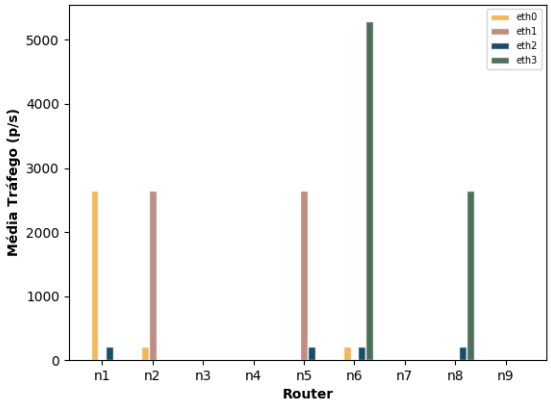


Figura 5.7: Gráfico de barras representativo do tráfego médio transmitido (em pacotes por segundo) no cenário 3.1, monitorizado depois da aplicação da proteção ao link n1-n5

A Figura 5.7 apresenta os valores médios de tráfego transmitido pelos *routers* da topologia de rede física após a aplicação da proteção ao *link n1-n5*. É facilmente verificável, que os mesmos, não coincidem com os valores médios de tráfego do caso inicial, sendo que o tráfego que anteriormente era transmitido de *n1* para *n5*, foi redirecionado para *n2*. Posto isto, é possível verificar que o percurso percorrido pelo tráfego foi: *n1->n2->n6*, através do *seeder p1*, e *n8->n5->n6*. Deste modo, o mecanismo de proteção efetuado pelo *ISP* permitiu redirecionar o tráfego que, anteriormente, percorria o *link n1-n5*. Em ambas as rotas, o somatório do número de saltos é igual a 2, porém o somatório do custo de encaminhamento da rota através do *seeder p1* apresenta o valor de 25, enquanto que o somatório do custo de encaminhamento da rota através do *p5* tem o custo de 20. O tempo registado nesta transferência foi de 6.53 segundos, sendo que apenas existe um aumento do tempo da transferência aproximado em 1%.

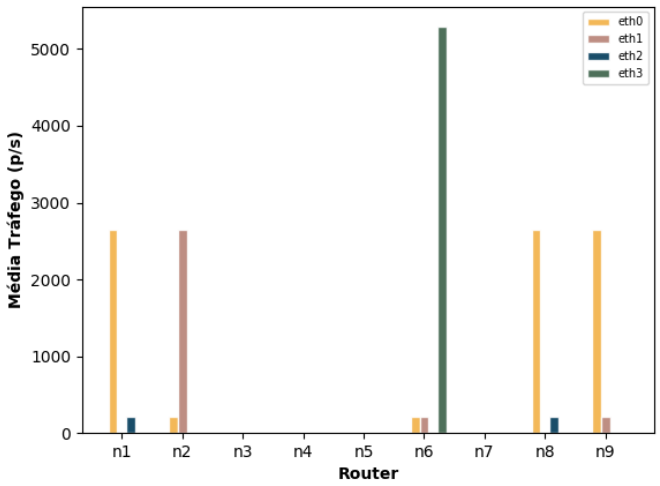


Figura 5.8: Gráfico de barras representativo do tráfego médio transmitido (em pacotes por segundo) no cenário 3.1, monitorizado depois da aplicação da proteção ao router n5

Relativamente ao caso de proteção do *router n5*, é apresentado um gráfico de barras, que representa o tráfego médio monitorizado pelos *routers* da topologia de rede física (Figura 5.8). Nesta figura, é possível confirmar a existência de um balanceamento que ocorreu no tráfego médio transmitido pelos *links* da rede física, além de que o tráfego que anteriormente era transmitido de *n1* para *n5*, foi redirecionado para *n2* e no caso de *n8* para *n5* foi redirecionado para n9. Posto isto, é possível verificar que as rotas que o tráfego percorreu foram: *n1->n2->n6*, através do *seeder p1* e *n8->n9->n6*, através do *seeder p5*. Deste modo, o mecanismo de proteção efetuado pelo *ISP*, permitiu que o tráfego fosse redirecionado em todos os *links* pertencentes ao *router n5*. embora o somatório do número de saltos nas duas rotas seja 2, mantendo um valor igual ao caso inicial, o somatório do custo de encaminhamento das duas rotas apresenta o valor 25, sofrendo um aumento de 25% $((50-40)/40$, em relação à transferência de teste efetuada inicialmente (sem proteções aplicadas). A transferência do conteúdo demorou 6.64 segundos, resultando num pequeno aumento temporal de 2.3% $((6.64-6.47)/6.47)$, tendo em conta o teste inicial.

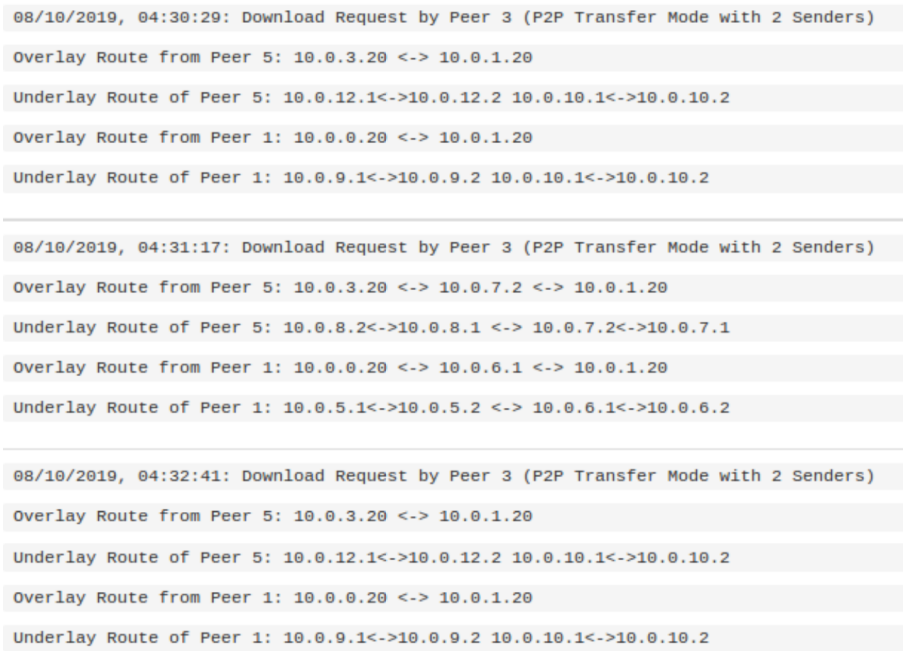


Figura 5.9: Vista da página web do ISP com informação sobre as rotas percorridas pela rede física/*overlay* nas transferências efetuadas no cenário 3.2

Por fim, a Figura 5.9 demonstra as rotas percorridas na rede física e *overlay*, a partir do planeamento de decisões de proteção efetuadas na página *web* do *ISP*. Esta, demonstra três transferências iniciadas antes da data inicial, entre a data inicial e a data final e, depois da data final. Posto isto, a partir da Figura 5.9, é possível concluir que o resultado obtido pelas rotas da topologia de rede física/*overlay*, antes e depois da data inicial é o mesmo (quando a proteção do *router n5* está desativada). Além disso, evidencia-se que as rotas da topologia de rede física, não contêm nenhum *link* pertencente ao *router n5*. Pois, como é possível visualizar na figura, as rotas com destino *p3*, são, *n1->n5->n6*, através do *seeder p1* e *n8->n5->n6*, através do *seeder p5*, onde é apresentada uma topologia com a descrição dos endereços de *IP* dos *links* na Figura A.15 em anexo. Finalmente, verifica-se que entre a data inicial e a data final (quando a proteção do *router n5* está ativa), ambas as rotas físicas são redireccionadas para os *links*: *n1->n2->n6*, através do *seeder p1* e *n8->n9->n6*, através do *seeder p5*, confirmando a veracidade da proteção aplicada.

Adicionalmente, é possível analisar as rotas na rede *underlay/overlay* das transferências, para cada caso de proteção referido, através da página *web* do *ISP*, presente na Figura A.17 em anexo, em conjunto com a topologia de rede com divulgação dos endereços de *IP* dos *links* (Figura A.15 do Anexo A).

5.3.4 *Cenário 4 - Priorização de Links/Routers*

5.3.4.1 *Descrição*

De modo semelhante ao cenário anterior, pretende-se simular uma situação presente no mundo real, na qual o *ISP* tenha determinado uma rota física pouco utilizada e existam outras rotas com elevado tráfego, nomeadamente *P2P*. Porém, para este tipo de casos, o sistema implementado, oferece ao *ISP* a possibilidade de priorização de um dado *router/link*, de modo a que o tráfego *P2P* da rede *overlay*, se possível, seja redireccionado para a rota com pouca utilização.

Este cenário, está divido em dois testes, de forma a comparar o modo de decisão instantâneo com o modo de planeamento de decisão entre datas para a priorização de *links/routers*. Em ambos os modos foi alterado o valor dos custos *OSPF* para 15, nos *links* que interligam os *routers*, *n1-n5* e *n5-n8*, de forma a que, a rota mínima escolhida no nível de rede excluísse imediatamente as rotas incluídas pelos *links* alterados, sendo que, por defeito, os custos de todos os outros *links* é 10. Para garantir o modo de transferência *P2P* e obter um caso de congestionamento onde se possa recorrer à priorização de *routers* e de *links*, foi adicionado à rede *overlay*, um conteúdo igual pelo *AccessPeer1* e *AccessPeer5*, com o propósito do *AccessPeer3* fazer o pedido de *download* e receber o conteúdo dividido por duas rotas distintas (*p5->p3* e *p1->p3*). Além disso, a estratégia de encaminhamento utilizada, por defeito, foi a estratégia de minimização do número de links percorridos na rede *overlay*, sendo que, neste cenário, as duas estratégias convergem sempre para a mesma rota. No modo de decisão instantânea, preliminarmente, foram inicializados os *peers p1, p3* e *p5* que correspondem aos *routers n1, n6* e *n8*, respetivamente. De seguida, foi efetuado o pedido de *download* por parte do *AccessPeer3*, de modo a que o *AccessPeer1* e o *AccessPeer5* enviassem o conteúdo dividido em fragmentos. Com a alteração dos custos *OSPF* nos *links* referidos no ínicio da secção, foi deduzido que os *links* que interligam os *routers*, *n1-n6* e *n6-n9*, são os mais congestionados. Assim sendo, foi adicionado um *RoutingPeer* no nó *n5* e através da página *web* do *ISP* foi aplicada uma priorização ao *link* entre o *router n1* e *n5* (Figura A.11 do Anexo A), sendo posteriormente efetuado o segundo *download* do conteúdo pelo *AccessPeer3*. Finalmente, foi adicionada uma priorização ao *router n5* na página de administração *web* (Figura A.12 do Anexo A), ou seja, a todos os seus *links*, e foi novamente efetuado o *download* por parte do *AccessPeer3*. No modo de planeamento de decisão, foi efetuada uma priorização ao *router n5*, ou seja, a todos os *links* pertencentes ao próprio, a partir da página *web* do *ISP*, desde uma data inicial até uma data final (Figura A.13 do Anexo A). O planeamento aplicado teve como parâmetros: a data inicial "8-10-2019, 04:14:00"e a data final "8-10-2019, 04:15:00". Finalmente, foram efetuados três *downloads* pelo *AccessPeer3*, um antes da data inicial, um após a data inicial e o último após a data final, de modo a verificar a veracidade do planeamento de proteção.

A topologia de rede *overlay* utilizada neste cenário, pode ser visualizada na Figura A.7 em anexo.

5.3.4.2 *Resultados*

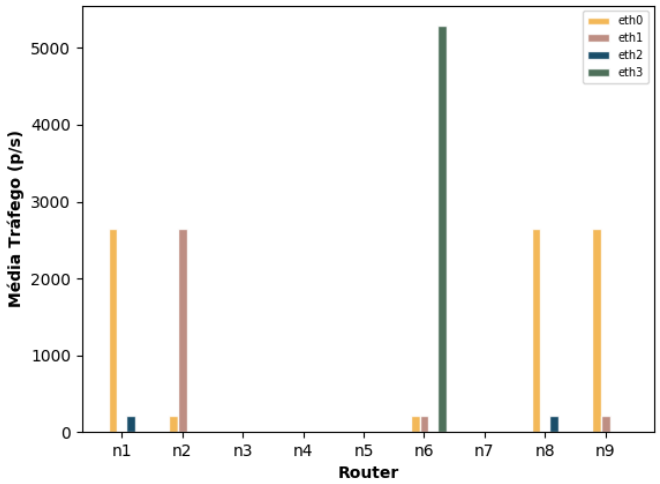


Figura 5.10: Gráfico de barras representativo do tráfego médio transmitido (em pacotes por segundo) no cenário 4.1, antes da aplicação de qualquer priorização

Neste cenário, inicialmente foi efetuada a monitorização dos *links* na rede física, de modo a ser obtida a rota inicial, sem qualquer priorização aplicada. Como tal, é possível visualizar na Figura 5.10 que o tráfego médio transmitido apenas é estabelecido nos *routers*, *n1*, *n2*, *n6*, *n8* e *n9*, de onde, se destaca o *router n6* como o *router* com maior tráfego médio transmitido, visto que recebe tráfego em simultâneo de *p2* e *p9*, com destino em *p3*. Logo, a partir desta análise, é possível verificar que a rota inicial efetuada pela transferência, percorre a rota na rede física, *n8->n9->n6*, através do *seeder p5* e *n1->n2->n6*, através do *seeder p1*. Em ambas as rotas o somatório do número de saltos assume o valor 2 e o somatório dos custos de encaminhamento o valor 20. O tempo registado por esta transferência foi de 6.44 segundos.

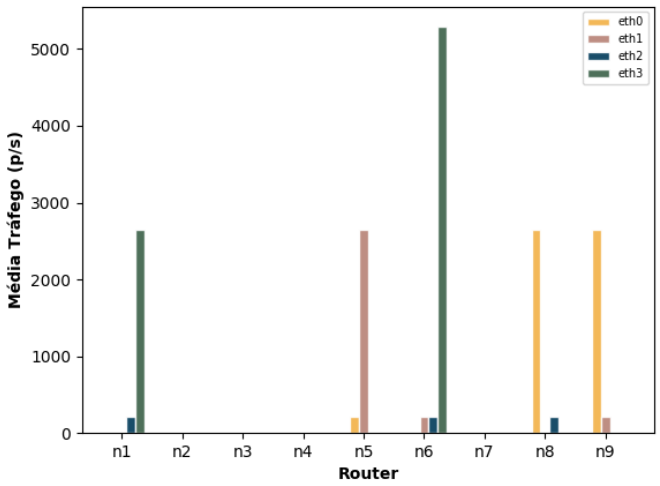


Figura 5.11: Gráfico de barras representativo do tráfego médio transmitido (em pacotes por segundo) no cenário 4.1, monitorizado depois da aplicação da priorização ao link n1-n5

Para o caso de priorização ao *link n1-n5*, explicado nas secções anteriores, a Figura 5.11 apresenta os valores médios de tráfego transmitido pelos *routers* da topologia de rede física. É facilmente verificável, que os mesmos, não coincidem com os valores médios de tráfego do caso inicial, destacando-se os *routers*, *n1*, *n5*, *n6*, *n8* e *n9*. Posto isto, considerando as possíveis rotas do tráfego, conclui-se que as únicas rotas na rede física com destino *p3*, são, *n1->n5->n6*, através do *seeder p1*, e *n8->n9->n6*, através do *seeder p5*. Deste modo, o mecanismo de priorização efetuado pelo *ISP*, permitiu que o tráfego fosse redirecionado, de forma a percorrer o *link n1-n5*. Em ambas as rotas, o somatório do número de saltos é igual a 2, porém o somatório do custo de encaminhamento da rota através do *seeder p1* apresenta o valor de 25, enquanto que o somatório do custo de encaminhamento da rota através do *p5* tem o custo de 20. O tempo registado nesta transferência foi de 6.54 segundos, sendo que a aplicação da priorização do *link n1-n5*, não teve praticamente efeitos negativos em termos de *QoS* para o sistema de rede *overlay* de transferência de conteúdos.

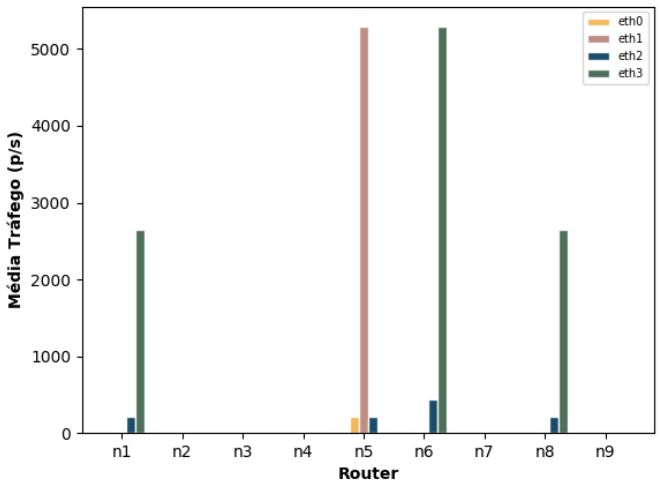


Figura 5.12: Gráfico de barras representativo do tráfego médio transmitido (em pacotes por segundo) no cenário 4.1, monitorizado depois da aplicação da priorização ao router n5

Relativamente ao caso de priorização do *router n5*, é apresentado um gráfico de barras, que representa o tráfego médio monitorizado pelos *routers* da topologia de rede física (Figura 5.12). Nesta figura, é possível confirmar a existência de um balanceamento que ocorreu no tráfego médio transmitido pelos *links* da rede física. Além disso, dado os *routers* destacados na transmissão de tráfego, as únicas rotas possíveis que a transferência possa ter tomado, são, *n1->n5->n6*, através do *seeder p1* e *n8->n5->n6*, através do *seeder p5*. Deste modo, o mecanismo de priorização efetuado pelo *ISP*, permitiu que o tráfego fosse redirecionado, de forma a percorrer os *links* do *router n5*. O somatório do número de saltos que se registou nas duas rotas foi 2, mantendo um valor igual ao caso inicial. Adicionalmente, o somatório do custo de encaminhamento das duas rotas apresenta o valor 25, sofrendo um aumento de 25% $((50-40)/40$, em relação à transferência de teste efetuada inicialmente (sem priorizações aplicadas). A transferência do conteúdo demorou 6.61 segundos, resultante num pequeno aumento temporal de 2.6% $((6.61-6.44)/6.61)$, tendo em conta o teste inicial.

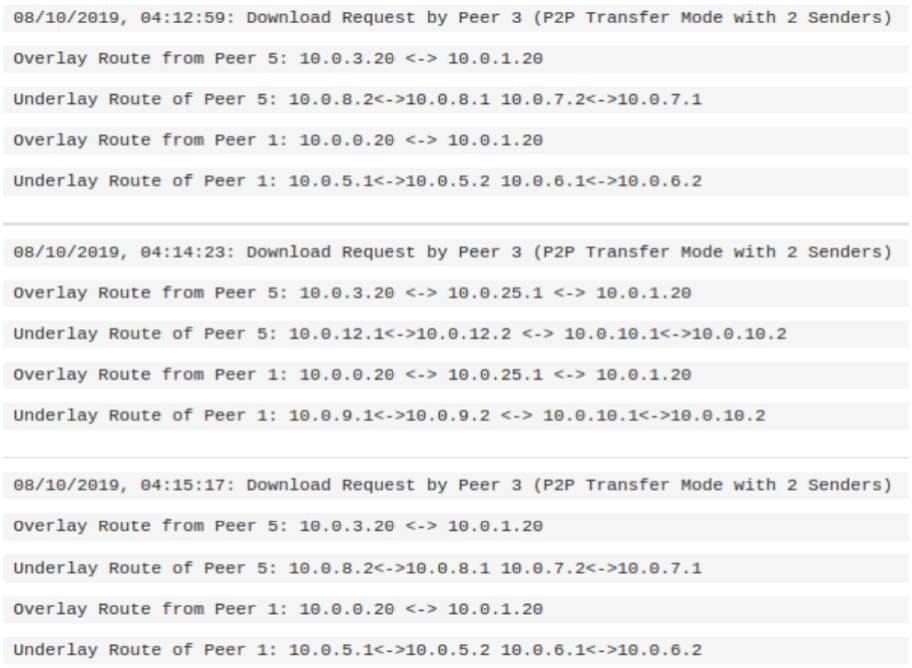


Figura 5.13: Vista da página web do ISP com informação sobre as rotas percorridas pela rede física/*overlay* nas transferências efetuadas no cenário 4.2

Por fim, a Figura 5.13 demonstra as rotas percorridas na rede física e *overlay*, a partir do planeamento de decisões de priorização efetuadas na página *web* do *ISP*. Esta, demonstra três transferências iniciadas: antes da data inicial, entre a data inicial e a data final, e depois da data final. Posto isto, a partir da Figura 5.13, é possível concluir que o resultado obtido pelas rotas da topologia de rede *física/overlay*, antes e depois da data inicial é o mesmo (quando a priorização do *router n5* está desativada). Além disso, é evidenciado que as rotas da topologia de rede física, não contêm nenhum *link* pertencente ao *router n5*. Isto, pois como é possível visualizar na figura, as rotas com destino *p3* são, *n1->n2->n6*, através do *seeder p1* e *n8->n9->n6* através do *seeder p5*, onde é apresentada uma topologia com a descrição dos endereços de *IP* dos *links* na Figura A.15 em anexo. Finalmente, é verificado que entre a data inicial e a data final (quando a priorização do *router n5* está ativada), ambas as rotas físicas são redireccionadas para os *links*, *n1->n5->n6*, através do *seeder p1* e *n8->n5->n6*, através do *seeder p5*.

Adicionalmente, é possível estudar as rotas na rede *underlay/overlay* das transferências, para cada caso de priorização referido, através da página *web* do *ISP*, presente no Anexo A.18, em conjunto com a topologia de rede com divulgação dos endereços de *IP* dos *links* (Figura A.15 do Anexo A).

5.3.5 *Cenário 5 - Limite de Circulação*

5.3.5.1 *Descrição*

Este cenário representa um caso de uso do sistema implementado, que permite ao *ISP* definir um limite de circulação de conteúdos, para quando forem determinadas situações críticas, nas quais, a utilização do sistema tenha que ser restrita. Para tal, neste cenário é demonstrada a possibilidade do sistema apresentar uma fila de espera ao utilizador, quando o somatório do tamanho de conteúdos em circulação na rede *overlay* ultrapassa um valor limite definido pelo *ISP*. Primeiramente, foram inicializados quatro *peers* nos nós *p1*, *p3*, *p4* e *p5*, interligados aos *routers n1*, *n6*, *n6* e *n8*, respetivamente (Figura 5.1). Em segundo, foi adicionado conteúdo com o tamanho aproximado de 9.64 MB de dimensão em numeração binária, pelo *AccessPeer1*. De seguida, a partir da página *web* do *ISP*, foi definido o limite de circulação para com o valor de 6000000 bytes (Figura A.14 do Anexo A), aproximadamente 5.72 MB. Por fim, iniciou-se o *download*, primeiro pelo *AccessPeer3* e depois pelo *AccessPeer4*, de modo a satisfazer as condições necessárias para a fila de espera ser ativada.

5.3.5.2 *Resultados*

De seguida, foram efetuadas duas transferências em modo sequencial: a primeira pelo *peer* de acesso 3 e a segunda pelo *peer* de acesso 4, interligados aos nós *p3* e *p4* apresentados na Figura 5.1. Em ambas, foi efetuada a transferência de um conteúdo com o tamanho aproximado de 9.64 MB, de modo a que a soma do tamanho das transferências em progresso (neste caso apenas uma, com o tamanho de 9.64 MB) seja maior que o valor do limite aplicado (5.72 MB).

Figura 5.14: Demonstração dos resultados do cenário 5 após a condição do limite ser cumprida na vista do utilizador

A partir da aplicação *desktop* para os *peers* de acesso, representado na Figura 5.14 em anexo, é apresentado o resultado deste cenário dividido em dois estados. O primeiro estado, demonstra a aplicação da fila de espera ao *peer* de acesso 4, na qual também é indicada a posição da fila, enquanto o *peer* de acesso 3 acaba de efetuar a transferência. Finalmente, no segundo caso, o *p3* acaba a transferência, sendo que a condição que mantém a fila de espera no *p4* não se verifica ($0 >= 5.62MB$), libertando-o da fila de espera e garantindo posteriormente a conclusão da transferência.

5.4 Sumário

Neste capítulo, inicialmente foi efetuada a apresentação dos testes, na qual foi especificado o ambiente de testes e as ferramentas auxiliares utilizadas. Posteriormente, apresentaram-

se as metodologias aplicadas a cada cenário de teste, nas quais, se descreveu a inicialização do sistema em si, em conjunto com os *scripts* implementados para efetuar os testes. De seguida, foram apresentados os cenários de teste, assim como a análise dos resultados relativos a cada caso, de modo a comprovar o funcionamento dos mecanismos implementados. Posteriormente, com a análise dos resultados, foram retiradas conclusões sobre a medição dos tempos de transferência de conteúdos, antes e após a utilização dos mecanismos implementados. Em suma, as funcionalidades implementadas, como: mecanismos de proteção/priorização de *links/routers*, suporte de duas estratégias de encaminhamento de tráfego aplicacional distintas, limitação de circulação de conteúdos na rede *overlay* e suporte de dois modos de transferência cliente-servidor/*P2P*, disponibilizam uma ferramenta eficiente de controlo do tráfego *P2P* ao *ISP*, não influenciando significativamente a qualidade do sistema de rede *overlay* para partilha de conteúdos.

6

CONCLUSÃO

Este capítulo é composto por duas secções que visam finalizar o estudo realizado nesta dissertação. Inicialmente, na secção 6.1, é efetuada uma conclusão, obtida depois da implementação do sistema de rede *overlay* proposto, em conjunto com as análises de testes aos mecanismos colaborativos com o *ISP*. Por fim, na secção 6.2, é feita uma descrição de uma possível continuação do protótipo implementado, como trabalho futiuro.

6.1 RESUMO

Com a realização desta dissertação foi criado um sistema de rede *overlay* para partilha de conteúdos, sob controlo do *ISP*. Este sistema, além de fornecer as funcionalidades básicas de partilha de ficheiros, com dois modos de funcionamento de transferência (cliente-servidor e *P2P*), foca-se principalmente no apoio à rede do *ISP*. No estado da arte, foram analisados artigos de redes *overlay* genéricas, de consciencialização da rede *underlay* de um sistema *P2P* e dos impactos para o *ISP*. De modo a complementar esta primeira análise, foram também abordadas soluções de interação entre as redes *overlay*/*P2P* e o *ISP*, com o propósito de estudar mecanismos existentes de colaboração com o fornecedor de rede. Por fim, foram analisados sistemas dedicados à distribuição de conteúdos.

Como tal, após a investigação efetuada, foi implementada uma rede *overlay* de partilha de conteúdos, na qual a principal prioridade, foi a criação de mecanismos que permitem controlar o tráfego *P2P* que percorre a rede física, de modo a beneficiar o fornecedor do serviço de *Internet*. Estes mecanismos oferecem métodos de proteção e priorização de tráfego *P2P* a *links*/*routers* determinados pelo *ISP*, além da funcionalidade de limitação de transferências em curso e dois modos de estratégia de encaminhamento aplicacional proporcionados pelo sistema de rede *overlay*. Comparativamente às soluções abordadas no estado da arte, o sistema implementado permite efetuar o controlo de tráfego *P2P* de modo simples, eficiente e flexível, a partir de uma página *web* de configurações administrativas. A principal vantagem do mesmo, é a disponibilização de diferentes ferramentas de engenharia de tráfego,

seleção da estratégia de encaminhamento aplicacional da rede *overlay* e aplicação de um limite de carga no sistema, não dependendo de uma arquitetura complexa.

A realização dos testes incluiu a avaliação dos mecanismos de colaboração com o *ISP*, em conjunto com as funcionalidades básicas de partilha de conteúdos do sistema de rede *overlay*, às quais foi atribuída uma menor prioridade. Os testes efetuados demonstraram resultados significativamente vantajosos para o *ISP*, sendo que os mecanismos implementados influenciaram de forma pouco significativa a qualidade de partilha de conteúdos do sistema. Como foi referido anteriormente, foram detectadas algumas limitações na fase de testes, nomeadamente na baixa carga aplicada a cada um dos cenários, devido à existência de limitações computacionais.

Esta dissertação demonstra assim a utilização de alguns mecanismos de colaboração com a rede física do *ISP*, através da utilização de uma rede *overlay* para partilha de conteúdos, capaz de corresponder a diferentes problemas de controlo de tráfego do fornecedor de rede. Assim, o trabalho consolidado nesta dissertação constitui-se como uma proposta inicial de um sistema de rede *overlay* com mecanismos de apoio à rede física controlada pelo *ISP*, sendo possível adicionar e automatizar alguns aspetos do mesmo, possibilitando ainda a sua continuação com algum trabalho futuro.

6.2 TRABALHO FUTURO

Como foi mencionado anteriormente, no que diz respeito à continuação deste estudo, podem ser criadas novas funcionalidades orientadas aos clientes da rede *overlay* de partilha de conteúdos e ao *ISP*. Além disso, pode ser melhorado o funcionamento do sistema, de forma a garantir integridade e segurança na sua comunicação e a oferecer um mecanismo de partilha de conteúdos legal.

Como tal, relativamente ao módulo de partilha de conteúdos criado para os clientes, poderia ser implementado um algoritmo de filtragem, com a finalidade de descartar os conteúdos que o sistema considerasse ilegais. De seguida, poderiam ser implementados mecanismos criptográficos de modo a tornar a comunicação *P2P* segura e técnicas de *hashing* de forma a garantir a integridade dos dados.

Relativamente ao módulo de administração do *ISP*, podem ser efetuadas melhorias visuais à página *web* criada. De forma a facilitar o processo de seleção de determinado mecanismo do sistema pelo *ISP*, poderia ser adicionada uma interface gráfica relativa à topologia de rede física. Posteriormente, poderia também ser adicionado um modo automático de priorização/proteção de *links/routers*, dependendo de determinados *thresholds* relacionados com a carga e a prioridade de tráfego, presentes em cada *link/router*.

Por fim, este trabalho apresenta uma solução de controlo de tráfego *P2P*, que pode ser futuramente explorada, de modo a originar uma aplicação paga, fornecida pelo *ISP* aos seus utilizadores.

BIBLIOGRAFIA

[1] Shusuke Yamazaki, Hideki Tode, and Koso Murakami. Cat: A cost-aware bittorrent. *IEICE transactions on communications*, 91(12):3831–3841, 2008.

[2] Gyorgy Dan, Tobias Hoßfeld, Simon Oechsner, Piotr Cholda, Rafal Stankiewicz, Ioanna Papafili, and George D Stamoulis. Interaction patterns between p2p content distribution systems and isps. *IEEE Communications Magazine*, 49(5):222–230, 2011.

[3] Tobias Hoßfeld, David Hausheer, Fabio Victora Hecht, Frank Lehrieder, Simon Oechsner, Ioanna Papafili, Peter Racz, Sergios Soursos, Dirk Staehle, George D Stamoulis, et al. An economic traffic management approach to enable the triplewin for users, isps, and overlay providers. In *Future Internet Assembly*, pages 24–34, 2009.

[4] Haiyong Xie, Y Richard Yang, Arvind Krishnamurthy, Yanbin Grace Liu, and Abraham Silberschatz. P4p: Provider portal for applications. In *ACM SIGCOMM Computer Communication Review*, volume 38, pages 351–362. ACM, 08 2008.

[5] Zhihui Lu, Ye Wang, Yang Richard Yang, et al. An analysis and comparison of cdn-p2p-hybrid content delivery system and model. *JCM*, 7(3):232–245, 2012.

[6] Osama Abboud, Aleksandra Kovacevic, Kalman Graffi, Konstantin Pussep, and Ralf Steinmetz. Underlay awareness in p2p systems: Techniques and challenges. In *Parallel & Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on*, pages 1–8. IEEE, 2009.

[7] Johan A Pouwelse, Pawel Garbacki, Jun Wang, Arno Bakker, Jie Yang, Alexandru Iosup, Dick HJ Epema, Marcel Reinders, Maarten R Van Steen, and Henk J Sips. Tribler: a social-based peer-to-peer system. *Concurrency and computation: Practice and experience*, 20(2):127–138, 2008.

[8] Bram Cohen. Incentives build robustness in bittorrent. *Workshop on Economics of Peer-toPeer systems*, 6, 06 2003.

[9] Salman Baset and Henning Schulzrinne. An analysis of the skype peer-to-peer internet telephony protocol. *Proceedings - IEEE INFOCOM*, 01 2005.

[10] Choon Hoong Ding, Sarana Nutanong, and Rajkumar Buyya. Peer-to-peer networks for content sharing. In *Peer-to-Peer Computing: the Evolution of a Disruptive Technology*, pages 28–65. IGI Global, 2005.

[11] NM Mosharaf Kabir Chowdhury and Raouf Boutaba. A survey of network virtualization. *Computer Networks*, 54(5):862–876, 2010.

[12] D. Doval and D. O'Mahony. Overlay networks: A scalable alternative for p2p. *IEEE Internet Computing*, 7:79–82, 07 2003.

[13] Jaime Galán-Jiménez and Alfonso Gazo-Cervero. Overlay networks: Overview, applications and challenges. *IJCSNS*, 10(12):40, 2010.

[14] Eng Keong Lua, Jon Crowcroft, Marcelo Pias, Ravi Sharma, and Steven Lim. A survey and comparison of peer-to-peer overlay network schemes. *IEEE Communications Surveys & Tutorials*, 7(2):72–93, 2005.

[15] Sameh El-Ansary and Seif Haridi. An overview of structured p2p overlay networks. *Handbook on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks*, 08 2005.

[16] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Shenker. A scalable content-addressable network. *ACM SIGCOMM Computer Communication Review*, 31, 09 2001.

[17] Ion Stoica, Robert Morris, David Liben-Nowell, David R Karger, M Frans Kaashoek, Frank Dabek, and Hari Balakrishnan. Chord: a scalable peer-to-peer lookup protocol for internet applications. *IEEE/ACM Transactions on Networking (TON)*, 11(1):17–32, 2003.

[18] Antony Rowstron and Peter Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *IFIP/ACM International Conference on Distributed Systems Platforms and Open Distributed Processing*, pages 329–350. Springer, 2001.

[19] João Leitão, José Orlando Pereira, and Luis Rodrigues. On the structure of unstructured overlay networks. In *38th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*. IEEE, IEEE, 06 2008.

[20] João Leitão, N. Carvalho, José Orlando Pereira, Rui Oliveira, and Luis Rodrigues. *On Adding Structure to Unstructured Overlay Networks*, pages 327–365. Springer, 2010.

[21] Ian Clarke, Oskar Sandberg, Brandon Wiley, and Theodore W Hong. Freenet: A distributed anonymous information storage and retrieval system. In *Designing privacy enhancing technologies*, pages 46–66. Springer, 2001.

[22] Jian Liang, Rakesh Kumar, and Keith W Ross. The fasttrack overlay: A measurement study. *Computer Networks*, 50(6):842–858, 2006.

[23] Marling Engle and Javed I Khan. Vulnerabilities of p2p systems and a critical look at their solutions. *Kent State University, Tech. Rep*, 2006.

[24] Khalied Shredeh. Analysis of attacks and security issues on the peer-to-peer networks. *International Journal of Computer Applications*, 138(2), 2016.

[25] Vinay Aggarwal, Anja Feldmann, and R Karrer. An internet coordinate system to enable collaboration between isps and p2p systems. In *Proceedings of the 11th International ICIN Conference*, 2007.

[26] Pedro Sousa. Context aware programmable trackers for the next generation internet. In *The Internet of the Future*, EUNICE 2009, pages 78–87. Springer, 2009.

[27] Thomas Karagiannis, Pablo Rodriguez, and Konstantina Papagiannaki. Should internet service providers fear peer-assisted content distribution? In *Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement*, pages 63–76, 01 2005.

[28] Amir H Rasti, Daniel Stutzbach, and Reza Rejaie. On the long-term evolution of the two-tier gnutella overlay. In *Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications*, pages 1–6. IEEE, 04 2006.

[29] Haiyong Xie, Yang Richard Yang, and Avi Silberschatz. Towards an isp-compliant, peer-friendly design for peer-to-peer networks. In *International Conference on Research in Networking*, pages 375–384. Springer, 2008.

[30] Jie Dai, Fangming Liu, and Bo Li. The disparity between p2p overlays and isp underlays: Issues, existing solutions, and challenges. *IEEE network*, 24(6):36–41, 2010.

[31] Guobin Shen, Ye Wang, Yongqiang Xiong, Ben Y Zhao, and Zhi-Li Zhang. Hptp: Relieving the tension between isps and p2p. In *IPTPS*, 2007.

[32] Weibin Zhao, David Olshefski, and Henning Schulzrinne. Internet quality of service: an overview. 03 2000.

[33] Xipeng Xiao, Alan Hannan, Brook Bailey, and Lionel M Ni. Traffic engineering with mpls in the internet. *IEEE network*, 14(2):28–33, 2000.

[34] David R Choffnes and Fabián E Bustamante. Taming the torrent: a practical approach to reducing cross-isp traffic in peer-to-peer systems. In *ACM SIGCOMM Computer Communication Review*, volume 38, pages 363–374. ACM, 2008.

[35] Vinay Aggarwal, Anja Feldmann, and Christian Scheideler. Can isps and p2p users cooperate for improved performance? *ACM SIGCOMM Computer Communication Review*, 37(3):29–40, 2007.

[36] R. Buyya, A. K. Pathan, J. Broberg, and Z. Tari. A case for peering of content delivery networks. *IEEE Distributed Systems Online*, 7(10):3–3, 10 2006.

[37] Nasreen Anjum, Dmytro Karamshuk, Mohammad Shikh-Bahaei, and Nishanth Sastry. Survey on peer-assisted content delivery networks. *Computer Networks*, 116:79–95, 2017.

[38] Erik Nygren, Ramesh K Sitaraman, and Jennifer Sun. The akamai network: a platform for high-performance internet applications. *ACM SIGOPS Operating Systems Review*, 44(3):2–19, 2010.

[39] Limelight Networks. Disponível em https://www.limelight.com. Acedido em Sep. 9, 2019.

[40] Netflix Open Connect. Disponível em https://openconnect.netflix.com/en_gb/. Acedido em Sep. 9, 2019.

[41] Gunnar Kreitz and Fredrik Niemela. Spotify – large scale, low latency, p2p music-on-demand streaming. In *Peer-to-Peer Computing (P2P), 2010 IEEE Tenth International Conference on*, pages 1–10. IEEE, 2010.

[42] Henrique Castro and Artur Alves. A p2p content delivery system for alternative business models — harnessing internet's full potential. *Proceedings of the 2011 7th International Conference on Next Generation Web Services Practices, NWeSP 2011*, 10 2011.

[43] Yann Nicolas, Daniel Wolff, Dario Rossi, and Alessandro Finamore. I tube, youtube, p2ptube: Assessing isp benefits of peer-assisted caching of youtube content. In *Peer-to-Peer Computing (P2P), 2013 IEEE Thirteenth International Conference on*, pages 1–2. IEEE, 2013.

[44] Jie Wu, BiSheng Liu, ShiYong Zhang, et al. Peercdn: A novel p2p network assisted streaming content delivery network scheme. In *Computer and Information Technology, 2008. CIT 2008. 8th IEEE International Conference on*, pages 601–606. IEEE, 2008.

[45] MongoDB. Disponível em https://www.mongodb.com/. Acedido em Sep. 9, 2019.

[46] Neo4j. Disponível em https://neo4j.com/. Acedido em Sep. 9, 2019.

[47] JavaFX. Disponível em https://openjfx.io/. Acedido em Sep. 9, 2019.

[48] Spark Framework. Disponível em http://sparkjava.com/. Acedido em Sep. 9, 2019.

[49] Javascript. Disponível em https://www.javascript.com/. Acedido em Sep. 9, 2019.

[50] Neovis. Disponível em https://github.com/neo4j--contrib/neovis.js/. Acedido em Sep. 9, 2019.

[51] jQuery. Disponível em https://jquery.com/. Acedido em Sep. 9, 2019.

[52] Bootstrap. Disponível em https://getbootstrap.com/. Acedido em Sep. 9, 2019.

[53] CORE. Disponível em https://www.nrl.navy.mil/itd/ncs/products/core. Acedido em Sep. 9, 2019.

[54] CORE. Disponível em https://downloads.pf.itd.nrl.navy.mil/docs/core/core--html/devguide.html. Acedido em Sep. 29, 2019.

[55] vnStat. Disponível em https://humdi.net/vnstat/. Acedido em Sep. 29, 2019.

[56] VirtualBox. Disponível em https://www.virtualbox.org. Acedido em Sep. 29, 2019.

A

MATERIAL DE SUPORTE

```
for file in path_cenario2 do
  aplica_media $file >> media_nome_cenario2.txt
done

for file in path_cenario3 do
  aplica_media $file >> media_nome_cenario3.txt
done

for file in path_cenario4 do
  aplica_media $file >> media_nome_cenario4.txt
done
```

Figura A.1: Pseudocódigo do script encarregue de aplicar a média de tráfego nos cenários de teste (meanCalculator.sh)

```
# arg = argumento passado a este script
# arg1 = id da sessao do core; arg2 = nome do cenario
# arg3 = local do ficheiro json com dados da topologia de rede fisica
# uso do ">" para encaminhar os dados num ficheiro
# criar no local da emulacao do coord presente no CORE um script
#  que inicia a app do coordenador, args: local_ficheiroJson
scritp_Coordinator arg3 > path_CORE_SESSION.arg1/Coord.conf/Coordinator

# criar no local da emulacao do ISP presente no CORE um script
# que inicia a rest api da pagina WEB do ISP
script_REST > path_CORE_SESSION.arg1/ISP.conf/REST

# criar no local da emulacao do ISP presente no CORE uma copia
# da aplicacao WEB que ira ser visualizada no firefox
cp -rf /home/core/Desktop/p2pcontentnetwork/ispWEB
path_CORE_SESSION.arg1/ISP.conf/ispWEB

# criar no local da emulacao do p1..p6 presente no CORE um
# script que inicia a aplicacao do peer de acesso, args:
# ID do router associado, ID peer na rede overlay, nome_cenario
script_AccessPeer 1 1 arg2 > path_CORE_SESSION.arg1/p1.conf/AccessPeer1
script_AccessPeer 1 2 arg2 > path_CORE_SESSION.arg1/p2.conf/AccessPeer2
```

```
script_AccessPeer 6 3 arg2 > path_CORE_SESSION.arg1/p3.conf/AccessPeer3
script_AccessPeer 6 4 arg2 > path_CORE_SESSION.arg1/p4.conf/AccessPeer4
script_AccessPeer 8 5 arg2 > path_CORE_SESSION.arg1/p5.conf/AccessPeer5
script_AccessPeer 8 6 arg2 > path_CORE_SESSION.arg1/p6.conf/AccessPeer6

# criar no local da emulacao do n1..n9 presente no CORE um
# script que inicia a aplicacao do peer de routing, args:
# ID do router associado, ID peer na rede overlay
script_RoutingPeer 1 7 > path_CORE_SESSION.arg1/n1.conf/RoutingPeer7
script_RoutingPeer 2 8 > path_CORE_SESSION.arg1/n2.conf/RoutingPeer8
script_RoutingPeer 3 9 > path_CORE_SESSION.arg1/n3.conf/RoutingPeer9
script_RoutingPeer 4 10 > path_CORE_SESSION.arg1/n4.conf/RoutingPeer10
script_RoutingPeer 5 11 > path_CORE_SESSION.arg1/n5.conf/RoutingPeer11
script_RoutingPeer 6 12 > path_CORE_SESSION.arg1/n6.conf/RoutingPeer12
script_RoutingPeer 7 13 > path_CORE_SESSION.arg1/n7.conf/RoutingPeer13
script_RoutingPeer 8 14 > path_CORE_SESSION.arg1/n8.conf/RoutingPeer14
script_RoutingPeer 9 15 > path_CORE_SESSION.arg1/n9.conf/RoutingPeer15
```

Figura A.2: Pseudocódigo do script de inicialização do sistema de rede overlay implementado (setScriptsOnCore.sh)

```
#!/bin/bash
# $1 - primeiro argumento ao script, que representa o local
# onde o ficheiro vai ser armazenado
# Caso o local nao exista - e criada uma diretoria
mkdir -p $1

# uso do vcmd para aceder ao ambiente emulado correspondente
# ao nome do no e ID da sessao do core

# uso do vnstat com as flags -i "nome_interface" e -l, que define
# a opcao de monitorizacao de uma interface especifica em tempo real

# uso do ">" para guardar os dados num ficheiro
# uso do "&" para o processo ser executado em background
#N1
vcmd -c /tmp/pycore.*/n1 -- vnstat -i eth0 -l > $1/n1_eth0.txt &
vcmd -c /tmp/pycore.*/n1 -- vnstat -i eth1 -l > $1/n1_eth1.txt &
vcmd -c /tmp/pycore.*/n1 -- vnstat -i eth2 -l > $1/n1_eth2.txt &
vcmd -c /tmp/pycore.*/n1 -- vnstat -i eth3 -l > $1/n1_eth3.txt &

#N2
vcmd -c /tmp/pycore.*/n2 -- vnstat -i eth0 -l > $1/n2_eth0.txt &
vcmd -c /tmp/pycore.*/n2 -- vnstat -i eth1 -l > $1/n2_eth1.txt &

#N3
vcmd -c /tmp/pycore.*/n3 -- vnstat -i eth0 -l > $1/n3_eth0.txt &
vcmd -c /tmp/pycore.*/n3 -- vnstat -i eth1 -l > $1/n3_eth1.txt &

#N4
vcmd -c /tmp/pycore.*/n4 -- vnstat -i eth0 -l > $1/n4_eth0.txt &
```

```
vcmd -c /tmp/pycore.*/n4 -- vnstat -i eth1 -l > $1/n4_eth1.txt &
vcmd -c /tmp/pycore.*/n4 -- vnstat -i eth2 -l > $1/n4_eth2.txt &

#N5
vcmd -c /tmp/pycore.*/n5 -- vnstat -i eth0 -l > $1/n5_eth0.txt &
vcmd -c /tmp/pycore.*/n5 -- vnstat -i eth1 -l > $1/n5_eth1.txt &
vcmd -c /tmp/pycore.*/n5 -- vnstat -i eth2 -l > $1/n5_eth2.txt &
vcmd -c /tmp/pycore.*/n5 -- vnstat -i eth3 -l > $1/n5_eth3.txt &

#N6
vcmd -c /tmp/pycore.*/n6 -- vnstat -i eth0 -l > $1/n6_eth0.txt &
vcmd -c /tmp/pycore.*/n6 -- vnstat -i eth1 -l > $1/n6_eth1.txt &
vcmd -c /tmp/pycore.*/n6 -- vnstat -i eth2 -l > $1/n6_eth2.txt &
vcmd -c /tmp/pycore.*/n6 -- vnstat -i eth3 -l > $1/n6_eth3.txt &

#N7
vcmd -c /tmp/pycore.*/n7 -- vnstat -i eth0 -l > $1/n7_eth0.txt &
vcmd -c /tmp/pycore.*/n7 -- vnstat -i eth1 -l > $1/n7_eth1.txt &

#N8
vcmd -c /tmp/pycore.*/n8 -- vnstat -i eth0 -l > $1/n8_eth0.txt &
vcmd -c /tmp/pycore.*/n8 -- vnstat -i eth1 -l > $1/n8_eth1.txt &
vcmd -c /tmp/pycore.*/n8 -- vnstat -i eth2 -l > $1/n8_eth2.txt &
vcmd -c /tmp/pycore.*/n8 -- vnstat -i eth3 -l > $1/n8_eth3.txt &

#N9
vcmd -c /tmp/pycore.*/n9 -- vnstat -i eth0 -l > $1/n9_eth0.txt &
vcmd -c /tmp/pycore.*/n9 -- vnstat -i eth1 -l > $1/n9_eth1.txt &
```

Figura A.3: Script utilizado para efetuar a captura de tráfego da topologia física do CORE

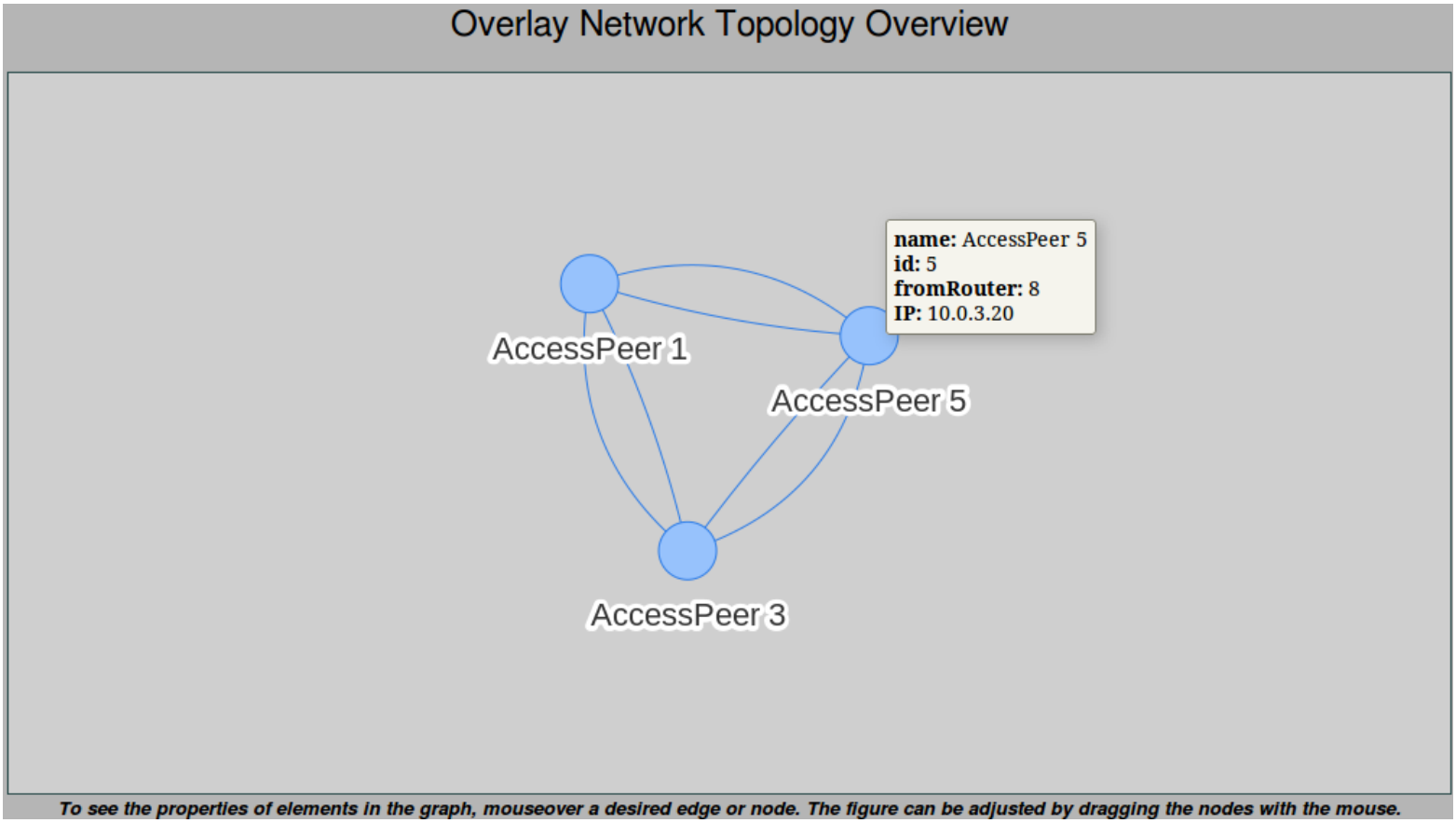


Figura A.4: Vista da rede overlay do cenário 1 a partir da aplicação WEB do ISP

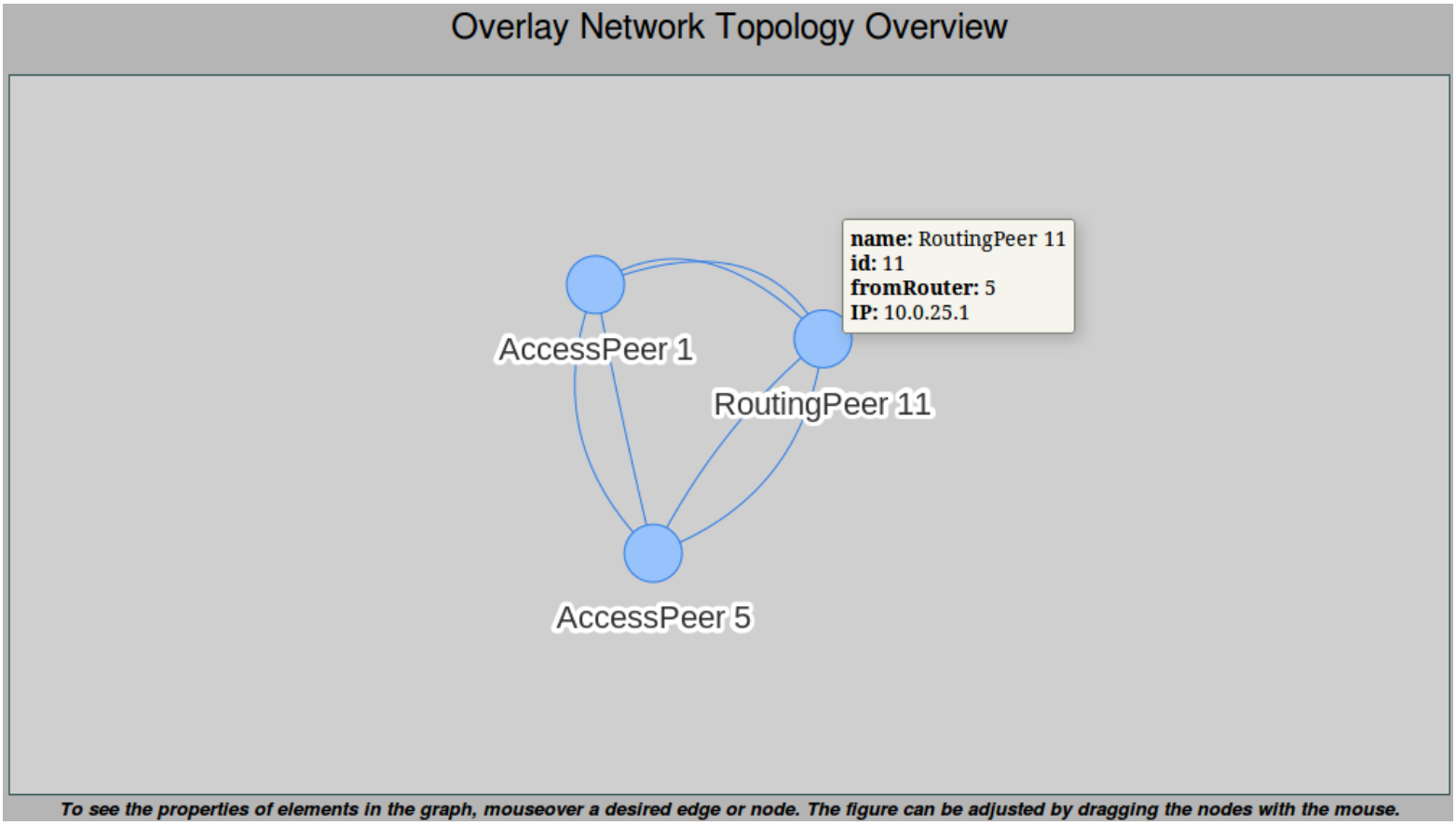


Figura A.5: Vista da rede overlay do cenário 2 a partir da aplicação WEB do ISP

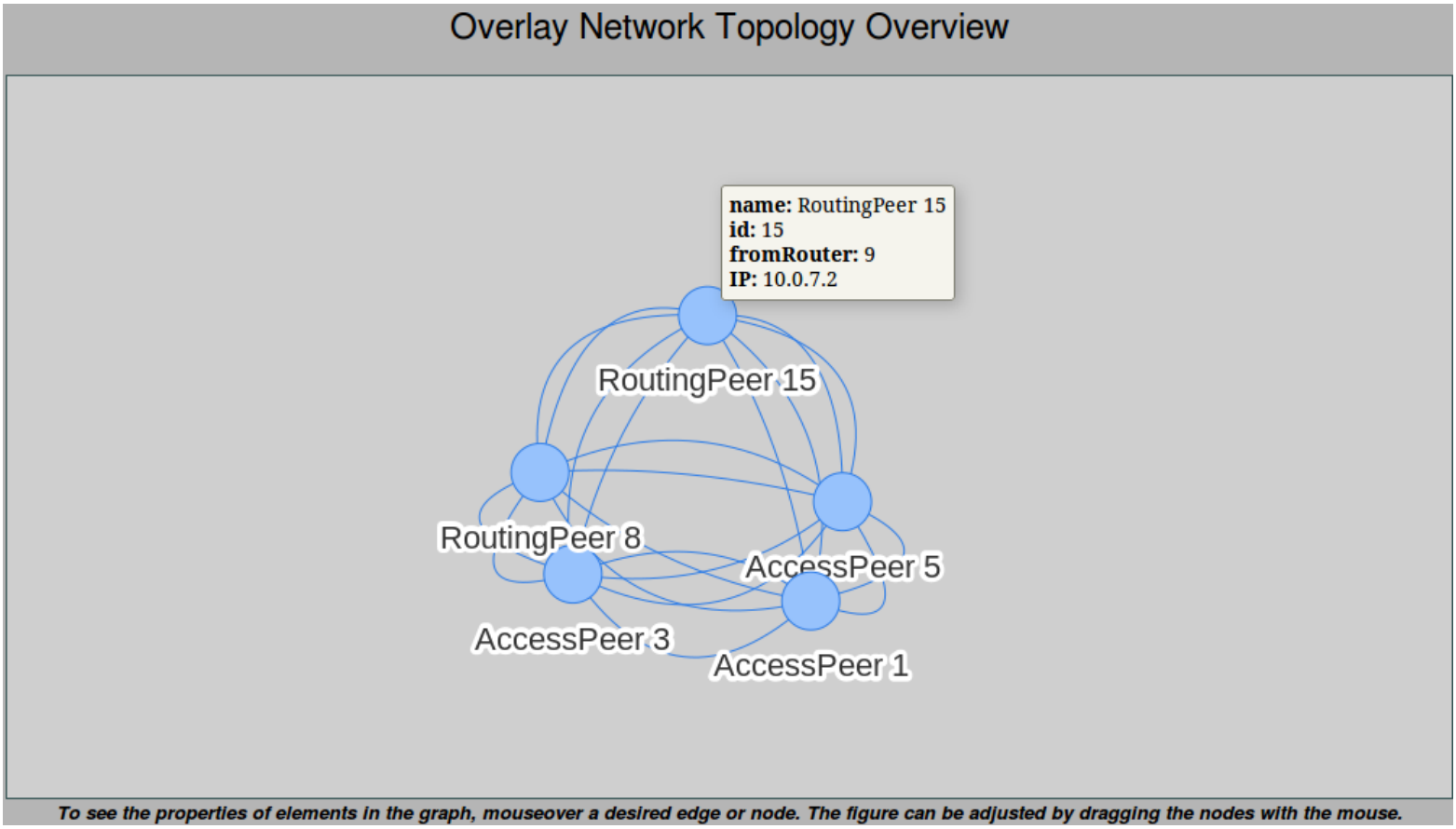


Figura A.6: Vista da rede overlay do cenário 3 a partir da aplicação WEB do ISP

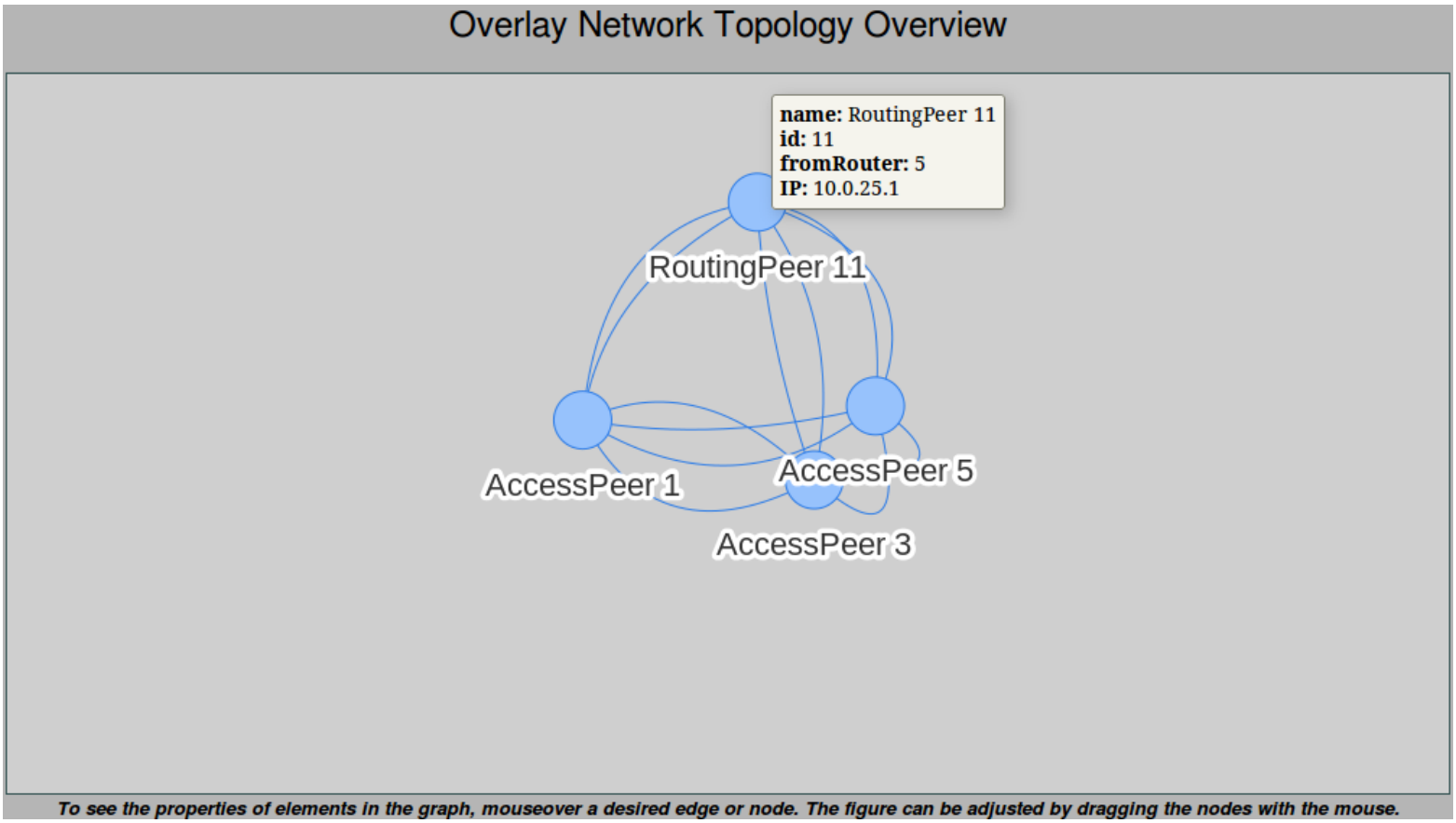


Figura A.7: Vista da rede overlay do cenário 4 a partir da aplicação WEB do ISP

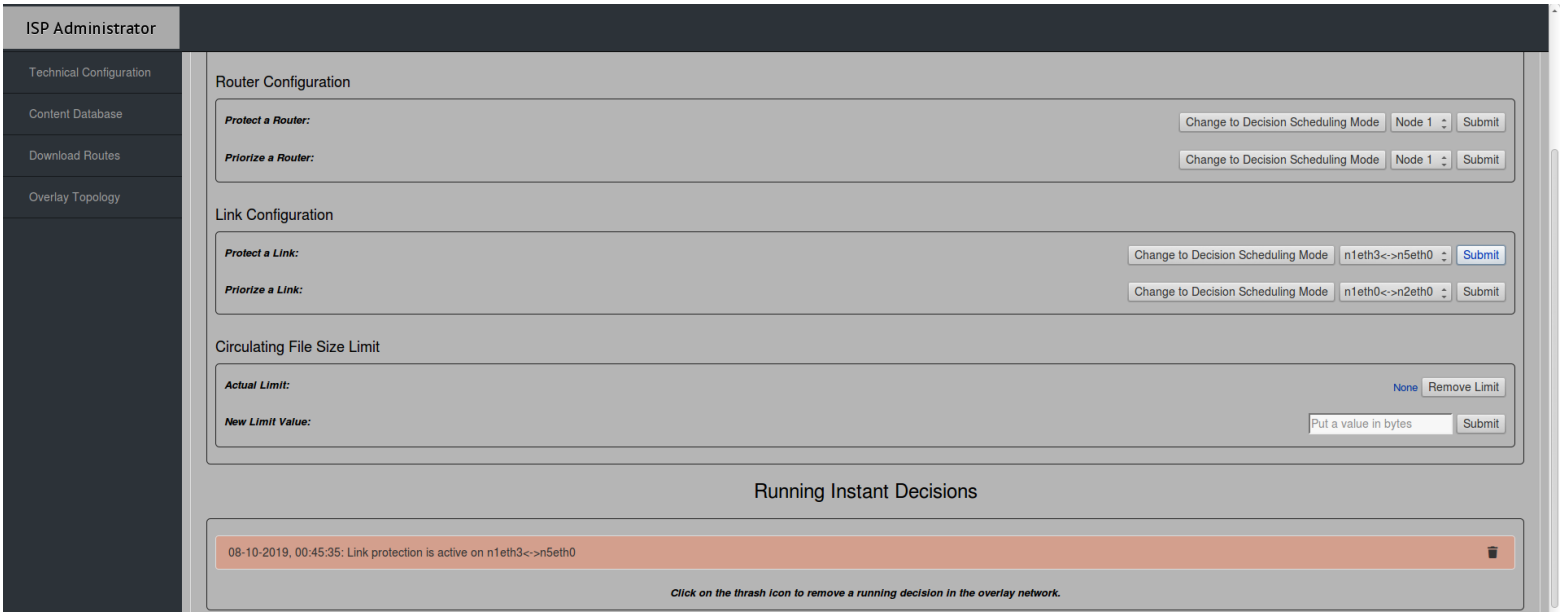


Figura A.8: Decisão instantânea de proteção aplicada ao link n1-n5 a partir da página WEB do ISP

Figura A.9: Decisão instantânea de proteção aplicada ao router n5 a partir da página WEB do ISP

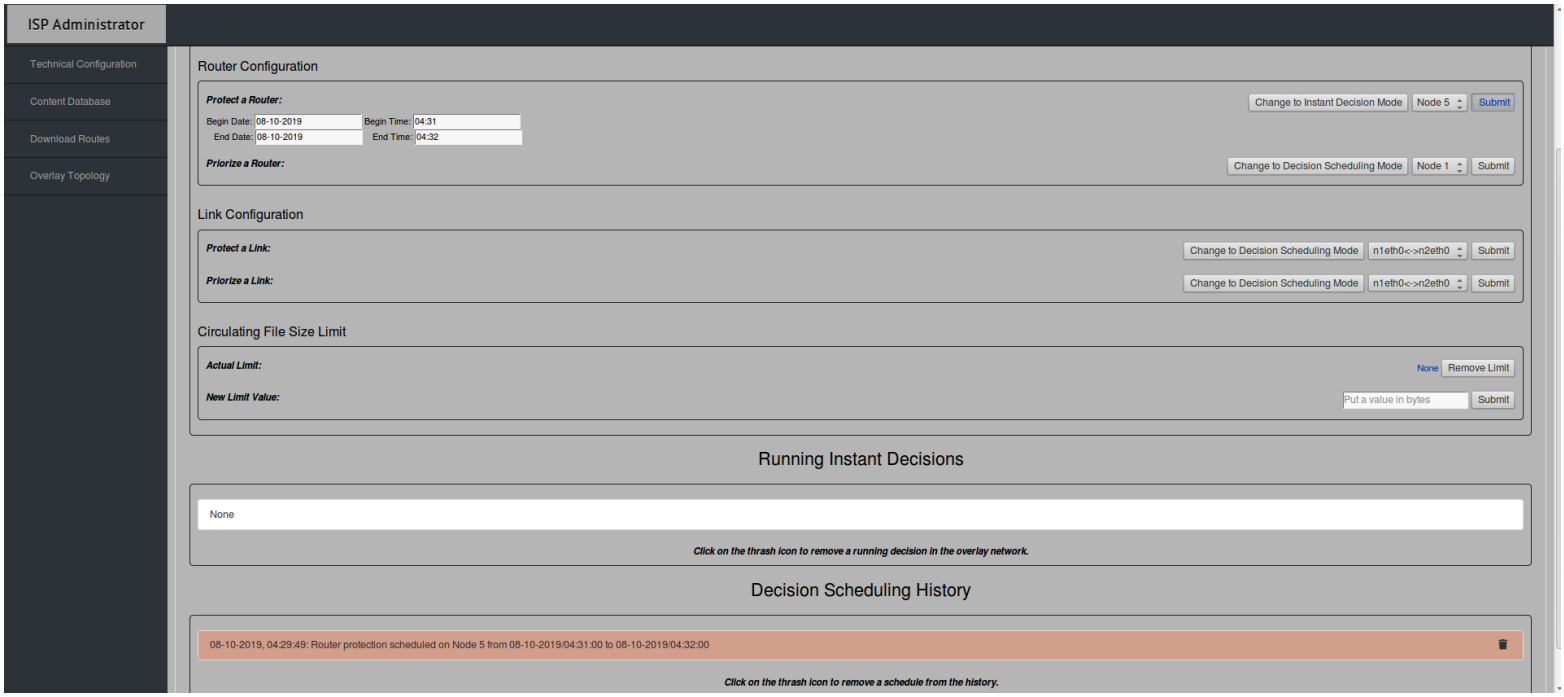


Figura A.10: Planeamento de proteção do router n5 a partir da página web do ISP

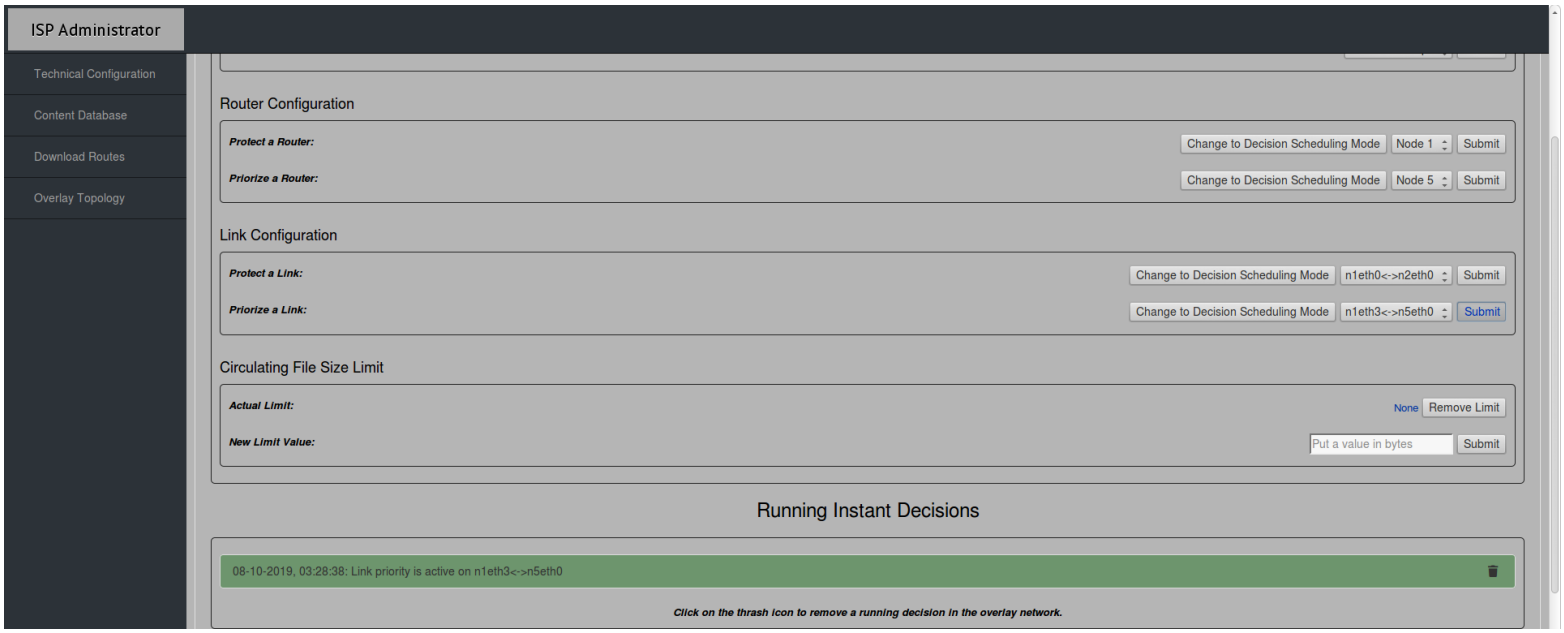


Figura A.11: Decisão instantânea de priorização do link n1-n5 a partir da página WEB do ISP

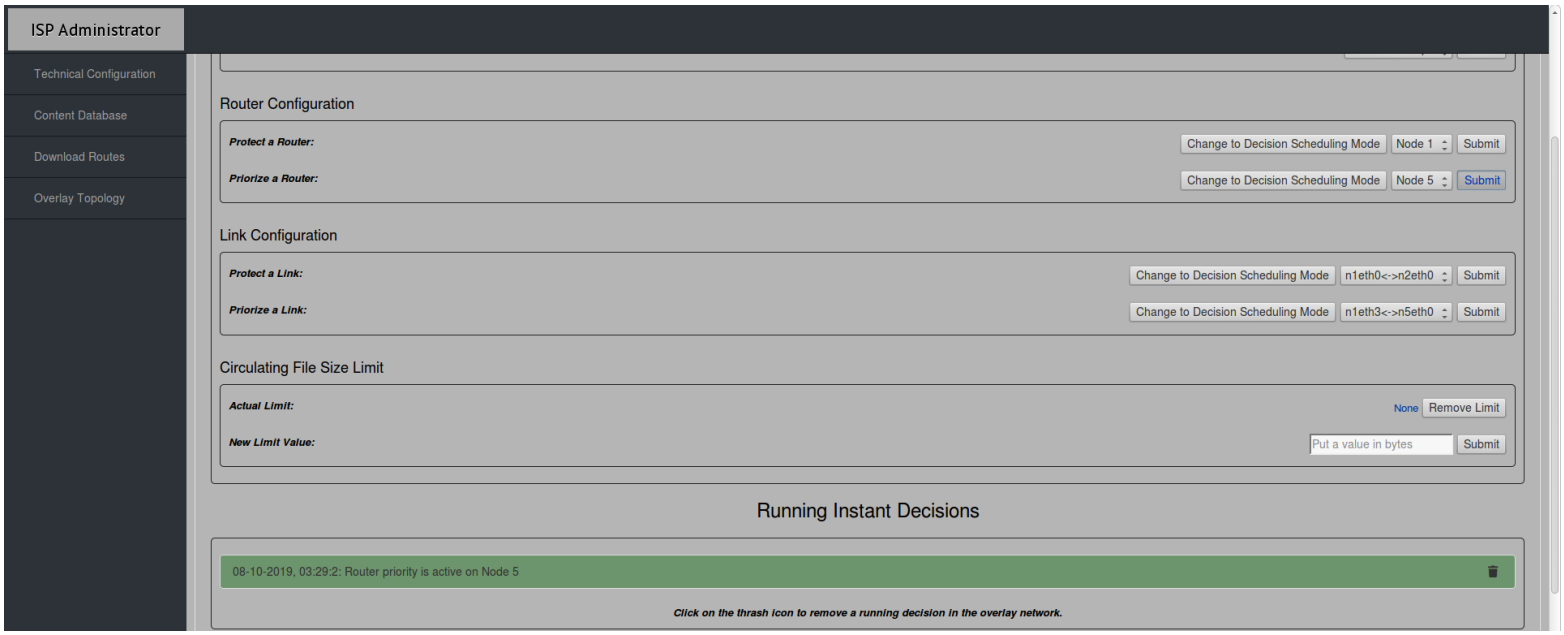


Figura A.12: Decisão instantânea de priorização do router n5 a partir da página WEB do ISP

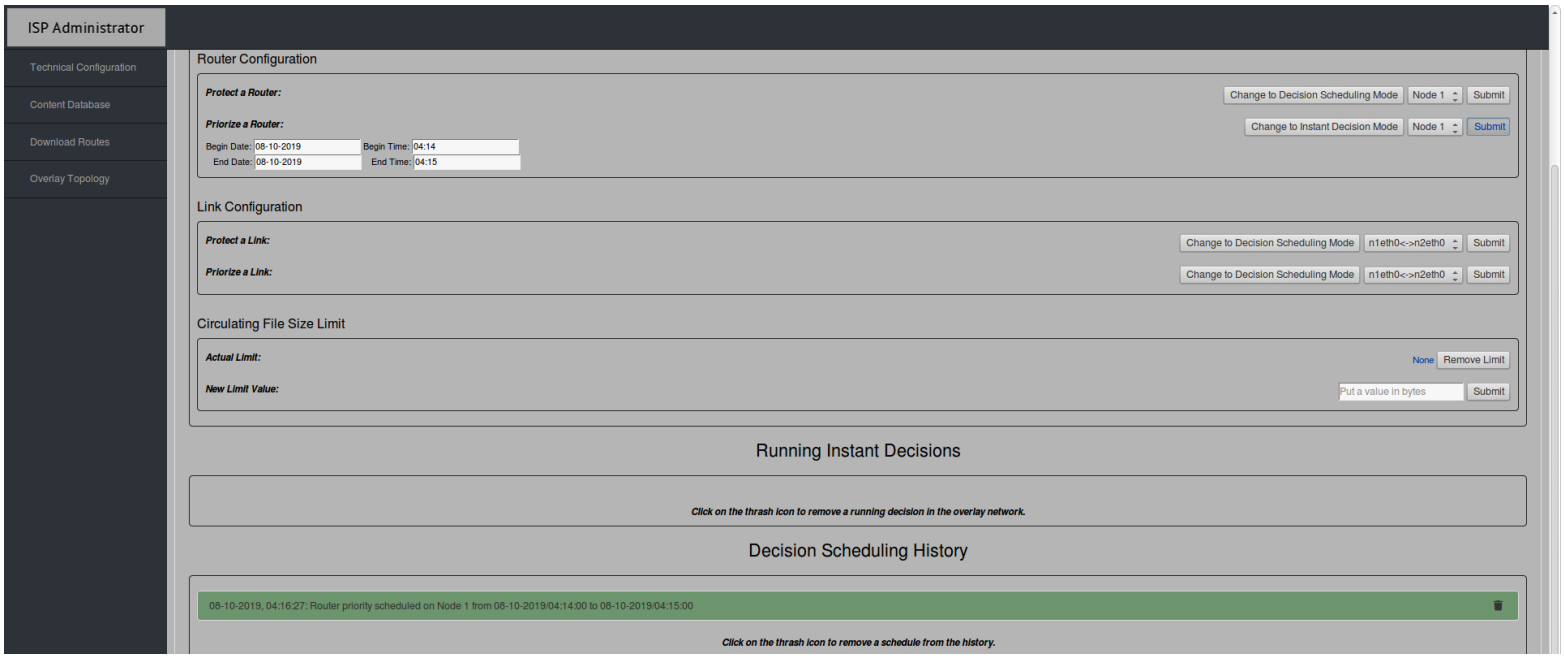


Figura A.13: Planeamento de priorização do router n5 a partir da página web do ISP

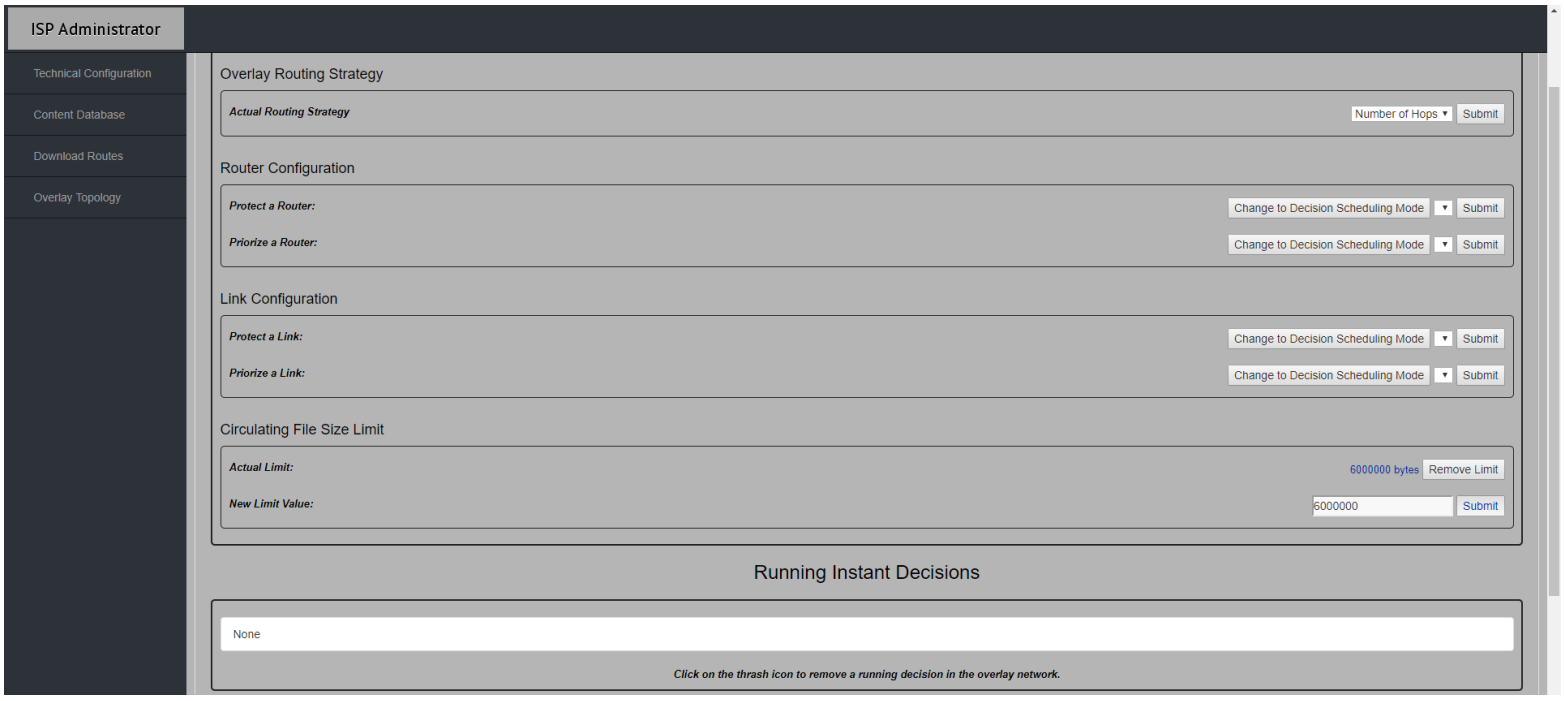


Figura A.14: Alteração do limite de circulação de conteúdos na página web do ISP

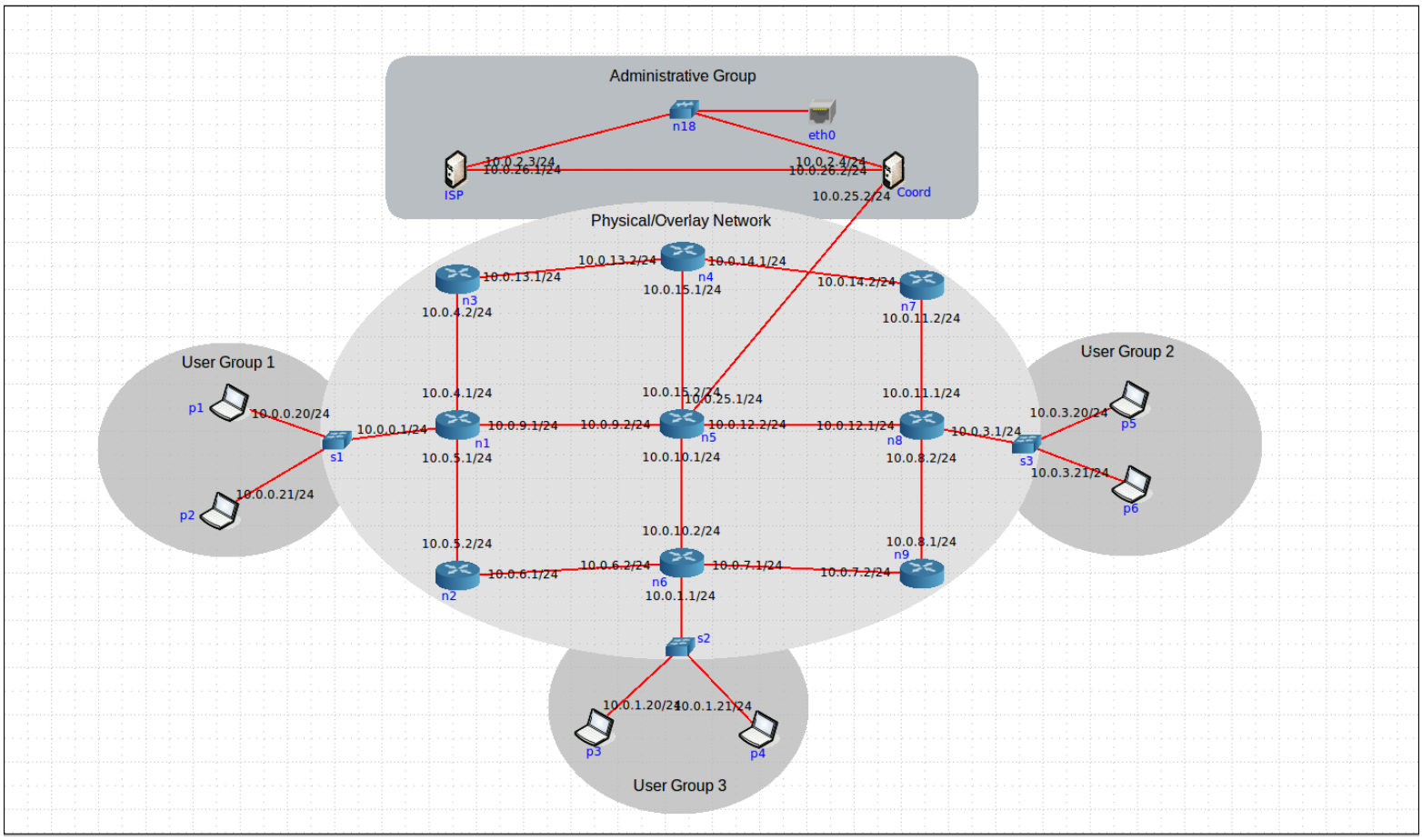


Figura A.15: Topologia de rede utilizada para os testes, com divulgação dos endereços de IP dos links

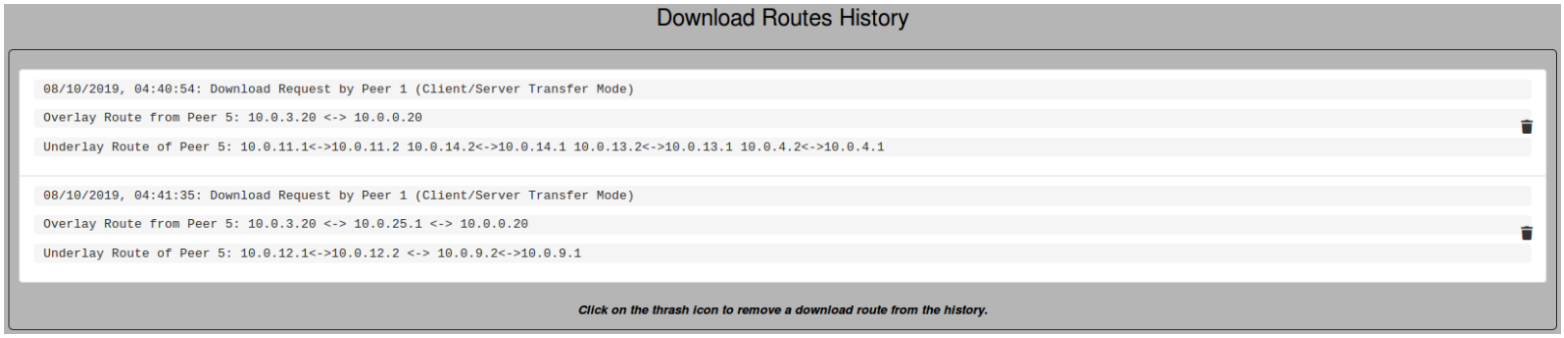


Figura A.16: Vista da página WEB do ISP com informação sobre as rotas percorridas pela rede física/overlay nas transferências efetuadas no cenário 2

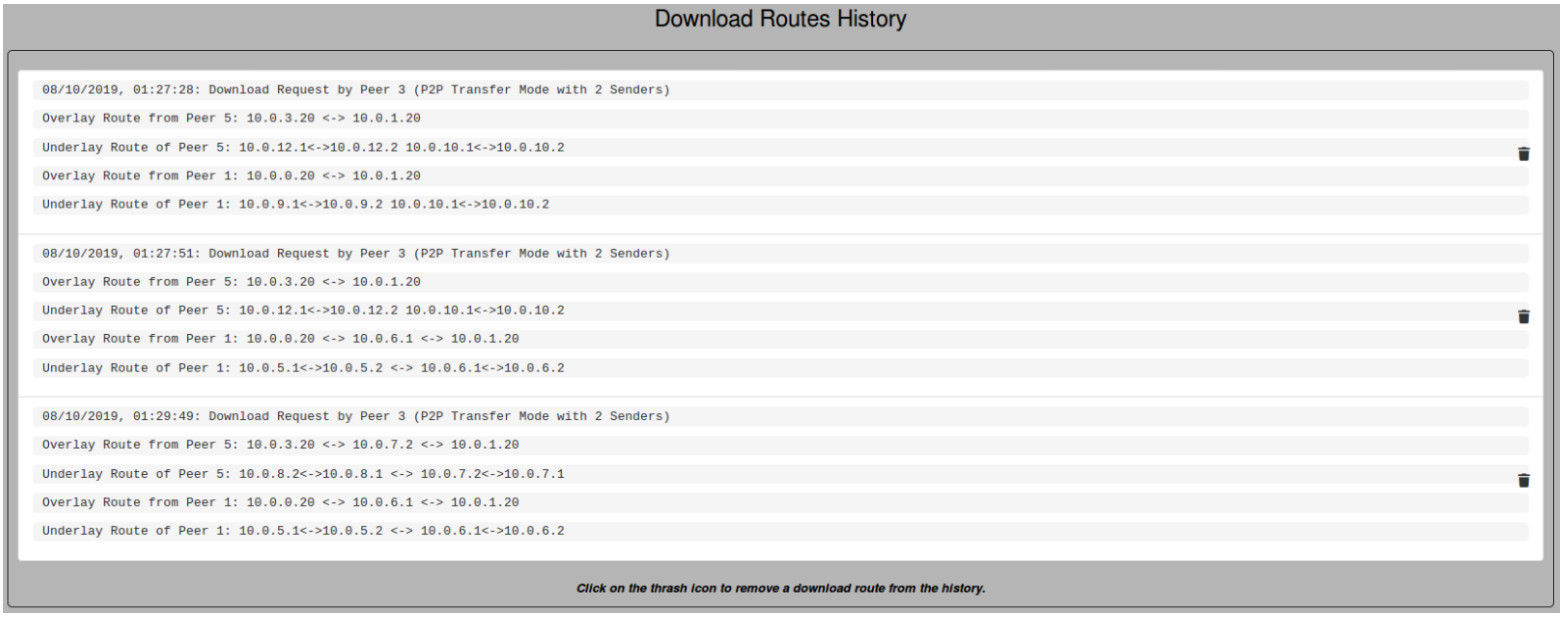


Figura A.17: Vista da página WEB do ISP com informação sobre as rotas percorridas pela rede física/overlay nas transferências efetuadas no cenário 3.1

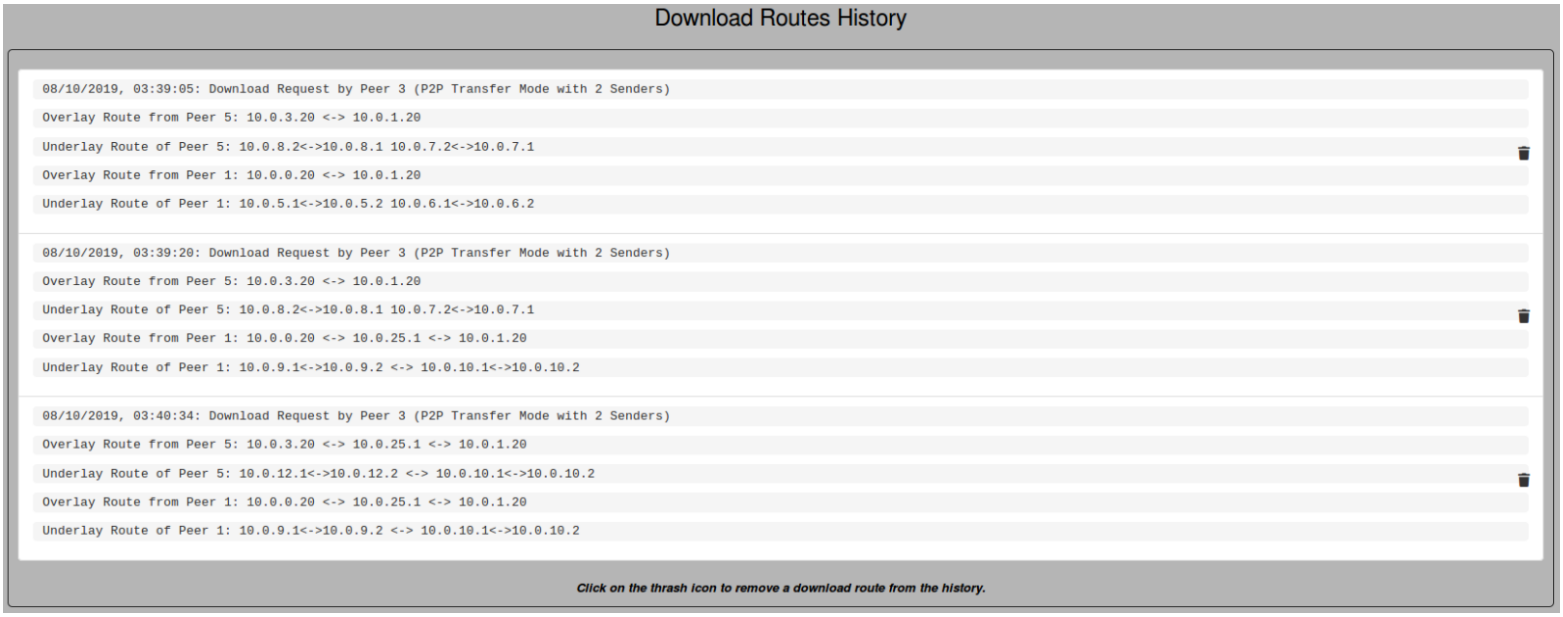


Figura A.18: Vista da página WEB do ISP com informação sobre as rotas percorridas pela rede física/overlay nas transferências efetuadas no cenário 4.1