UNIVERSIDADE D
COIMBRA

Rúben André Martins Gonçalves

Estudo de estratégias de configuração de uma rede MPLS com Fast-Reroute

Dissertação no âmbito do Mestrado Integrado em Engenharia Eletrotécnica e de Computadores, da especialização em Telecomunicações, orientada pela Professora Doutora Lúcia Maria dos Reis Albuquerque Martins e apresentada ao Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Ciências e Tecnologias da Universidade de Coimbra

Outubro de 2021

Faculdade de Ciências e Tecnologias da Universidade de Coimbra

Departamento de Engenharia Eletrotécnica e de Computadores

UNIVERSIDADE D
COIMBRA

Estudo de estratégias de configuração de uma rede MPLS com Fast-Reroute

Rúben André Martins Gonçalves

Júri:

Professor Doutor Carlos Alberto Henggeler de Carvalho Antunes (Presidente)

Professora Doutora Lúcia Maria dos Reis Albuquerque Martins (Orientadora)

Professora Doutora Rita Cristina Girão Coelho da Silva (Vogal)

Coimbra, Outubro de 2021

Agradecimentos

As primeiras palavras de agradecimento vão para a minha família, e em especial à minha mãe, pela paciência nesta longa jornada.

Um caloroso agradecimento à Professora Doutora Lúcia Martins, minha orientadora que propôs o presente trabalho, tudo aquilo que me ensinou. Agradeço ainda a boa disposição com a qual sempre se presenciou no laboratório. Lhe dirijo a minha grata admiração.

Agradeço também à minha namorada Maria, pelo apoio e carinho demonstrado ao longo deste último ano. Agradeço-lhe por ter permanecido sempre ao meu lado. Agradeço ainda aos meus amigos de Castelo Branco e colegas de faculdade, pelo companheirismo, pela sua amizade e os bons momentos vividos.

A todos, o meu muitíssimo obrigado.

O presente trabalho foi financiado no âmbito do Financiamento Plurianual de Unidades de I&D 2020-2023, da UI0308, Financiamento Base, com a referência UIDB/00308/2020, Instituto de Engenharia de Sistemas e Computadores de Coimbra (INESC Coimbra), BI Refª.UI0308-ResilienciaRedes.1/2020, com apoio financeiro da FCT/MCTES através de fundos nacionais (PIDDAC). Este trabalho teve ainda o apoio do projeto CENTRO-01-0145-FEDER-029312.

Resumo

O principal objetivo desta dissertação foi o estudo de estratégias de recuperação local para falhas em *links* com o objetivo de integrar posteriormente numa rede multicamada. Primeiramente estudaram-se alguns conceitos básicos de recuperação em redes, em particular em redes MPLS (*MultiProtocol Label Switching*) e redes WDM (*Wavelength Division Multiplexing*).

Foram implementadas duas estratégias para atribuição das capacidades envolvidas no restauro de *links* tendo para isso sido utilizada a linguagem *Java*.

Finalmente, a análise de desempenho dos algoritmos desenvolvidos foi feita através de simulação. As simulações neste trabalho foram efetuadas com o simulador Net2Plan 0.3.1 para análise, qualitativa e quantitativa, tendo permitido a comparação das diferentes estratégias testadas.

Palavras-chave: falhas, proteção local, restauro, MPLS, simulação

Abstract

The main objective of this dissertation was the study of recovery strategies for link failures in order to integrate it into a multilayer network. Firstly, some basic concepts of recovery in networks were studied, in particular in MPLS (MultiProtocol Label Switching) and WDM (Wavelength Division Multiplexing) networks.

Two strategies for capacity allocation for link restoration were developed using Java language.

Finally, the performance analysis of the developed algorithms was made through simulation. The simulations in this work were carried out with the Net2Plan 0.3.1 simulator for qualitative and quantitative analysis, allowing a comparison of the different tested strategies.

Keywords: faults, local protection, restoration, MPLS, simulation

Índice

Lista de Acrónimos e Siglas

BGP *Border Gateway Protocol*

ERO *Explicit Route Object*

FEC *Forwarding Equivalence Class*

FRR *Fast Reroute*

Gbps *Gigabits por segundo*

IDE *Integrated Development Environment*

IETF *Internet Engineering Task Force*

IS-IS *Intermediate System to Intermediate System*

IP *Internet Protocol*

LB *Largura de Banda*

LDP *Label Distribution Protocol*

LOS *Loss of Signal*

LSP *Label Swtiching Path*

LSR *Label Switching Router*

Mbps *Megabits por segundo*

MP *Merge Point*

MPLS *MultiProtocol Label Switching*

MTBF *Mean Time Between Failures*

MTTF *Mean Time To Failure*

MTTR *Mean Time To Repair*

OCh *Optical channel*

ODU *Optical channel Data Unit*

OMS *Optical Multiple Section*

OPU *Optical channel Payload Unit*

OSPF *Open Shortest Path First*

OTN *Optical Transport Network*

OTS *Optical Transmission Section*

OTU *Optical Transport Unit*

OXC *Optical Cross-Connect*

PLR *Point of Local Repair*

QoS *Quality of Service*

RESV *Reservation message*

RFC *Request for Comments*

RHE *Recovery head-end*

RSVP-TE *Resource Reservation Protocol-Traffic Engineering*

RTE *Recovery tail-end*

SDH *Synchronous Digital Hierarchy*

SRG *Shared Risk Group*

TE *Traffic Engineering*

VWP *Virtual Wavelength Path*

WDM *Wavelength Division Multiplexing*

WP *Wavelength path*

WR *Wavelength routing*

WT *Wavelength translating*

1. Introdução

Os serviços de comunicação têm um papel fundamental nas atividades económicas e sociais no dia-a-dia. Sobretudo para as empresas, uma perturbação nas comunicações pode suspender operações críticas e vitais para o seu negócio. Assim, é fulcral que haja mecanismos de recuperação em certos serviços, de forma que estes recuperem, em caso de falha, o mais rapidamente possível e de forma que estes tenham uma qualidade de serviço (*Quality of Service* - QoS) satisfatória.

1.1. Objetivos da tese

O objetivo desta dissertação foi o estudo de algoritmos que permitam o restauro em caso de falha de *links* e sua adaptação a redes multicamada. Assim sendo, foi feita a implementação de duas estratégias para determinação dos recursos envolvidos na recuperação local dos arcos numa rede MPLS.

Assim foi necessário inicialmente um estudo aprofundado das diferentes formas de recuperação de falhas de *links* em diferentes tipos de redes, bem como do funcionamento de um simulador preparado para testar os algoritmos implementados. As últimas versões do simulador escolhido (Net2plan) já permitem o estudo da fiabilidade em redes multicamada, mas os algoritmos de restauro só foram possíveis de testar em versões mais antigas do simulador sendo necessário adaptá-los para as últimas versões, o que não foi possível fazer por falta de tempo.

Um dos algoritmos de cálculo de capacidade de *backup* conduz de facto a redução da capacidade reservada para restauro na rede, embora apenas em casos de falhas simples.

1.2. Conteúdo

Esta dissertação é composta por cinco capítulos que são apresentados de seguida.

Capítulo 1 – Introdução: Capítulo introdutório que descreve o tema abordado nesta dissertação e objetivos da tese;

Capítulo 2 – Estratégias de recuperação em redes MPLS: Neste capítulo são descritos alguns conceitos básicos de recuperação em redes MPLS, sendo também descrito o princípio de funcionamento das redes MPLS;

Capítulo 3 – Estratégias de atribuição de recursos para restauro: São descritos os algoritmos implementados em linguagem *Java*.

Capítulo 4 – Simulações realizadas e análise de resultados: Contém a análise de resultados obtidos através da simulação, assim como a comparação com os resultados dos algoritmos desenvolvidos;

Capítulo 5 – Conclusões: São apresentadas as conclusões desta dissertação bem como possíveis trabalhos futuros que poderão ser desenvolvidos, em particular para redes multicamada, tendo como base o simulador Net2Plan. Por fim é considerado o trabalho futuro que poderá ser estudado.

2. Estratégias de recuperação em redes MPLS

As redes de comunicação estão sujeitas a diversas falhas, causadas nomeadamente por desastres naturais (tais como sismos, incêndios, cheias), erros de *software* ou até mesmo sabotagem.

2.1. Conceitos básicos

Introduzem-se de seguida algumas definições básicas [Vasseur *et al.*, 2004]:

A **fiabilidade** (*reliability*) define-se pela probabilidade de um determinado elemento da rede (um nó ou um *link*), estar totalmente operacional durante um certo intervalo de tempo.

A **integridade** (*network integrity*) define-se pela capacidade de a rede fornecer uma determinada qualidade de serviço, quer em circunstâncias normais quer quando a rede está congestionada ou em situação de falha.

A **capacidade de sobrevivência** (*network survivability*) é definida pela capacidade de a rede, em situação de falha, recuperar tráfego sem haver consequências para os utilizadores fornecendo um nível de serviço aceitável.

A **disponibilidade** (*availability*) define-se como a probabilidade de um certo elemento da rede estar disponível num determinado instante no tempo.
Para a definição de disponibilidade é necessário conhecer-se:

- ***Mean Time between Failures*** (MTBF) consiste no tempo médio entre duas avarias consecutivas do mesmo elemento da rede;
- ***Mean Time To Failure*** (MTTF) consiste no tempo médio durante o qual o elemento da rede está operacional;
- ***Mean Time to Repair*** (MTTR) é a duração média necessária para reparar o elemento da rede em caso de avaria.

Estes dois últimos conceitos estão diretamente relacionados com a disponibilidade média de um determinado elemento da rede, que se define por: $\mathbf{A = 1 - \frac{MTTR}{MTBF}}$

Prevenir e mitigar o efeito de falhas (*outages* ou *faults*) ou avarias (*failures*):
A ocorrência de uma avaria leva a que o elemento da rede entre em falha.

Assim, para se prevenir as avarias pode-se, por exemplo [Vasseur *et al.* (2004)]:

- colocar os cabos numa região mais funda ou com um revestimento mais forte;
- aumentar a qualidade intrínseca dos elementos da rede;
- introduzir redundância através da colocação de novos elementos na rede.

Para mitigar os efeitos de uma falha, a solução passa pelo uso de esquemas de recuperação. Assim quando é detetada uma falha, há mecanismos que desviam o fluxo de dados para um outro caminho, sem presença de falhas. Numa situação normal, antes da ocorrência de uma falha, o tráfego segue pelo caminho ativo. Este caminho pode também designar-se frequentemente por caminho primário ou caminho de trabalho. Se é detetada uma falha, o mecanismo de recuperação é ativado, ou seja, o tráfego vai passar a seguir um outro caminho designado por **caminho de *backup*** (caminho de proteção ou ainda caminho de recuperação). No extremo (cabeça) do caminho de *backup* (*recovery head-end* - RHE), o tráfego é redirecionado em direção a este caminho (é feito o *switch-over*) até chegar ao outro extremo do caminho de *backup* (*recovery tail-end* - RTE), ou seja, a cauda do caminho de *backup*. Quando a situação de falha é resolvida o tráfego passa novamente para o caminho ativo para chegar a um determinado destino, tal como podemos verificar pela figura abaixo apresentada.

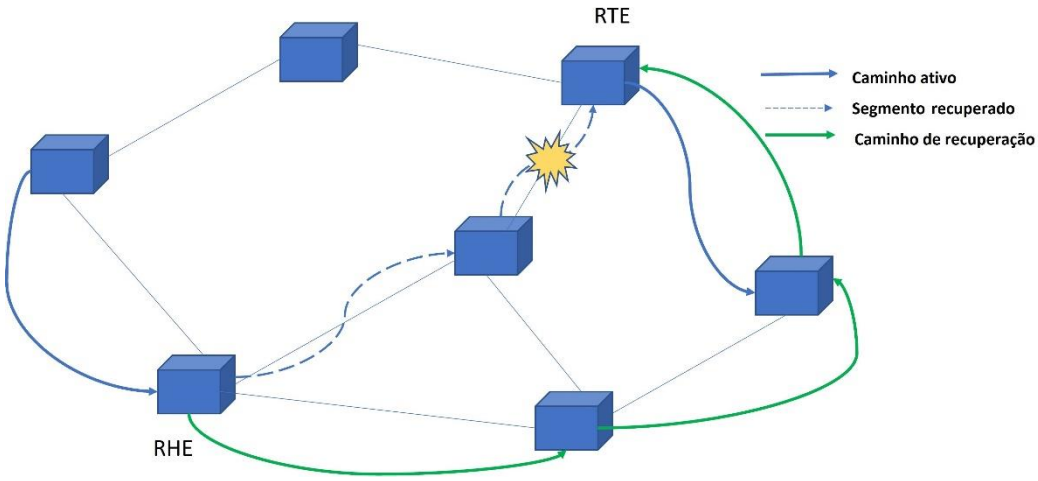

*Figura 1 - Noção de mecanismo de recuperação. [*adaptada de *Vasseur et al., (2004)].*

Desta forma torna-se importante a existência de caminhos alternativos na rede que possam funcionar como caminhos de *backup*. Aliado a este requisito, deve também existir largura de banda suficiente para ser disponibilizada a um caminho de *backup*.

2.2. Caracterização dos mecanismos de recuperação

Dependendo dos resultados pretendidos, um mecanismo de recuperação vai sempre ter vantagens e desvantagens.

Capacidade de *backup*: Dedicada *vs.* Partilhada

Um dos critérios a ter em conta é se a capacidade de *backup* é dedicada ou partilhada. No caso da capacidade de *backup* dedicada, um determinado recurso do caminho de *backup* só pode ser utilizado para transportar o tráfego de um determinado caminho ativo. Por outro lado, na capacidade de *backup* partilhada, os recursos do caminho de *backup* são partilhados por vários caminhos ativos. Esta última é mais eficiente dado que poupa largura de banda.

Caminhos de *backup*: Pré-calculado *vs.* Dinâmicos

Outro critério a ter em conta é o caminho de *backup* que pode ser pré-calculado ou dinâmico. A opção de calcular o caminho de antemão tem como vantagem uma recuperação rápida. O caminho de *backup* dinâmico significa que este não é planeado antecipadamente. Neste caso a computação do caminho é feita *on-the-fly* assim que a avaria é detetada. Em caso de avaria, o mecanismo de recuperação deve, dinamicamente, procurar possíveis caminhos de recuperação por toda rede, sendo este um processo relativamente mais lento.

Proteção *vs.* Restauro

Existem duas distinções a fazer no que diz respeito aos mecanismos de recuperação. Na **proteção**, os caminhos de *backup* são pré-calculados e sinalizados antes de ocorrer uma falha. Não há assim necessidade de uma sinalização adicional para estabelecer o caminho de *backup*, sendo esta solução mais rápida [Vasseur *et al.*, (2004)]:

- **Proteção 1 + 1 ("um mais um"):** Um caminho com proteção dedicada protege apenas um caminho ativo, todo o tráfego vai seguir em duplicado a partir do extremo RHE (*recovery head-end*) em ambos os caminhos (ativo e de *backup*). Esta variante tem uma elevada eficiência em termos de tempo de recuperação, mas em contrapartida tem a desvantagem de usar muita largura de banda.
- **Proteção 1 : 1 ("um para um"):** Nesta variante, um caminho com proteção dedicada protege apenas um segmento do caminho ativo. Numa situação sem

ocorrência de falha o tráfego é transportado no caminho ativo. Desta forma há possibilidade de tráfego não prioritário ser transportado no caminho de proteção. Quando ocorre uma falha no caminho ativo protegido, o tráfego passa deste para o caminho de proteção, sendo descartado o tráfego que estava a ser transportado antes por este caminho.

- **Proteção 1 : N ("um para N"):** Um determinado elemento de recuperação é escolhido para a proteção de até *N* elementos ativos. Quando não há falhas a afetar estes elementos, o elemento de *backup* pode ser usado para tráfego extra.
- **Proteção M : N ("M para N"):** Nesta variante, *M* elementos do caminho de *backup* vão proteger *N* entidades ativas. Quando a falha não afeta os elementos ativos, os *M* elementos de *backup* podem ser usados para tráfego extra.

A proteção, em relação ao restauro, tem a grande vantagem de o tempo de recuperação ser ainda mais pequeno. Por outro lado, no **restauro** os caminhos de recuperação não são previamente sinalizados. Esta é uma solução mais flexível em certos cenários e que em muitos casos não requer tanta capacidade de *backup*.

Recuperação global *vs.* local

Na **recuperação local**, só os elementos atingidos pela falha são contornados. Por outras palavras, os extremos RHE e RTE ficam o mais próximo possível do elemento que falhou e assim consegue-se que o tempo de recuperação seja pequeno. Este método é usualmente mais rápido do que a técnica de recuperação global. No entanto, tem a desvantagem de o tráfego poder passar pelo mesmo *link* mais que uma vez.

Na **recuperação global**, quando ocorre uma falha, o tráfego é redirecionado na origem para o caminho de *backup* até chegar ao destino. Os extremos RHE e RTE vão coincidir com a origem e destino (respetivamente) do caminho ativo.

Controlo dos mecanismos de recuperação: Centralizado *vs.* Descentralizado

- **Centralizado:** Este tipo de mecanismo de recuperação depende de um controlador central que vai determinar quais as ações de recuperação a serem realizadas. Este controlador determina quando e onde ocorreu a falha, uma vez que tem uma visão global do estado da rede, o que permite reconfigurar todos os elementos na rede incluídos no processo de recuperação. O mecanismo centralizado tem como principal vantagem a sua simplicidade. No aspeto da

capacidade, este mecanismo é também mais eficiente e relativamente à implementação de algoritmos mais fácil de implementar.

- **Descentralizado:** Este mecanismo opera sem a intervenção de um sistema de controlo central, possuindo sistemas de controlo que autonomamente iniciam as ações de recuperação. No entanto estes têm apenas a visão local do estado da rede e para colmatar esta desvantagem, comparativamente ao mecanismo de recuperação centralizado, os sistemas de controlo permitem a troca de mensagens para dar conhecimento de informação entre eles, de modo a coordenar as ações de recuperação. O mecanismo descentralizado tem uma elevada escalabilidade.

Modo revertível *vs.* não revertível

Por último é importante ter em atenção se o modo é revertível ou não revertível. No modo revertível, assim que a avaria é completamente reparada, há mecanismos que automaticamente fazem o *switch-back* do caminho de recuperação para o caminho ativo. Este modo pode levar a uma melhor eficiência de utilização da rede. Por outro lado, o modo não revertível evita efeitos temporários que podem ocorrer de uma operação de *switch-back*.

2.3. Recuperação multicamada

Anteriormente foi feita referência aos mecanismos de recuperação que podem ser implementados numa rede de uma só camada (*single-layer*). No entanto numa situação real, multicamada, é previsível um mecanismo de recuperação para o conjunto das camadas e não para cada uma em separado. Existem duas possíveis abordagens para a recuperação *multilayer*.

Abordagem sequencial

De certa forma, uma falha não é resolvida nas várias camadas exatamente ao mesmo tempo. Esta abordagem leva a que sejam impostas condições aos mecanismos de recuperação e a que estes tenham uma ordem cronológica na realização de tarefas, podendo este mecanismo ser implementado com a adição dum tempo de *hold-off*.

Abordagem integrada

Esta consiste na combinação de vários mecanismos que vão resultar num só esquema de recuperação *multilayer* todo ele integrado. Implica assim que este esquema tenha uma visão completa de todas as camadas e que tome a decisão de onde e quando é que cada camada pode tomar uma ação de recuperação que seja adequada por um esquema integrado de recuperação.

2.4. MPLS – *MultiProtocol Label Switching*

A tecnologia MPLS (*Multiprotocol Label Switching*) incorpora um conjunto de especificações (ou *standards*), que foram desenvolvidas pelo *Internet Engineering Task Force* (IETF) com o propósito de incluir o encaminhamento de pacotes integrando técnicas de engenharia de tráfego (*traffic engineering* – TE). A tecnologia MPLS garante que todos os pacotes, num determinado fluxo de dados, sigam o mesmo caminho sobre o *backbone* da rede, com qualidade de serviço (QoS), necessária para suportar serviços de voz e vídeo em tempo real, através da garantia de largura de banda.

2.4.1. Fundamentos do MPLS

O MPLS baseia-se uma técnica eficiente de comutação de etiquetas (*labels*) para fazer o envio e encaminhamento de pacotes através da rede. Esta tecnologia foi desenhada para funcionar em conjunto com o protocolo IP, e pode ser utilizada com qualquer outro protocolo da camada de dados. A etiqueta com tamanho fixo de 32 bits, é inserida no cabeçalho do pacote. O uso de etiquetas em MPLS permite o encaminhamento de pacotes, a garantia de qualidade de serviço bem como outras funções de gestão de tráfego. O MPLS é uma tecnologia *connection-oriented*, ou seja, orientada à ligação com alta escalabilidade e que permite balancear o tráfego na rede, melhorando a eficiência na utilização de recursos.

Engenharia de Tráfego – TE

Uma das principais vantagens na utilização do MPLS é a possibilidade de se fazer a gestão do tráfego o que permite otimizar o desempenho e a utilização de recursos da rede, evitando assim o congestionamento causado por uma utilização ineficiente da rede. As especificações para implementação da técnica de Engenharia de tráfego sobre MPLS são descritas pela norma RFC 2702 [Awduche *et al*. (1999)]. Esta técnica tem em

consideração a necessidade de um desempenho ótimo por parte da rede. A Engenharia de tráfego tem como objetivo fazer a reserva de tráfego de modo a maximizar a utilização da capacidade da rede, permitindo [Stallings, (2014)]:

- controlar de forma dinâmica os percursos do tráfego de modo a garantir o melhor caminho (de uma forma eficiente ou com menos custo) para os pacotes na rede chegarem ao destino, conforme os requisitos de QoS dos diferentes fluxos de tráfego;
- melhorar a utilização da capacidade da rede;
- garantir os requisitos de Qualidade de Serviço;
- ter maior integridade da rede de forma a minimizar erros e falhas.

2.4.2. Funcionamento do MPLS

Numa rede MPLS existem nós importantes denominados ***Label Switching Routers*** (LSR) como é apresentado na figura seguinte. Estes *routers* têm a função de comutar as etiquetas e são capazes de encaminhar pacotes da camada de rede. Quando um pacote entra na rede MPLS é classificado e é-lhe atribuída uma ***Forwarding Equivalence Class*** (FEC). Uma FEC consiste num grupo de pacotes IP que são encaminhados de igual forma, ou seja, sobre o mesmo caminho e sob o mesmo tratamento até ao destino. O caminho seguido através de um ou mais LSRs pelos pacotes de uma determinada FEC é denominado por ***Label Switching Path*** (LSP).

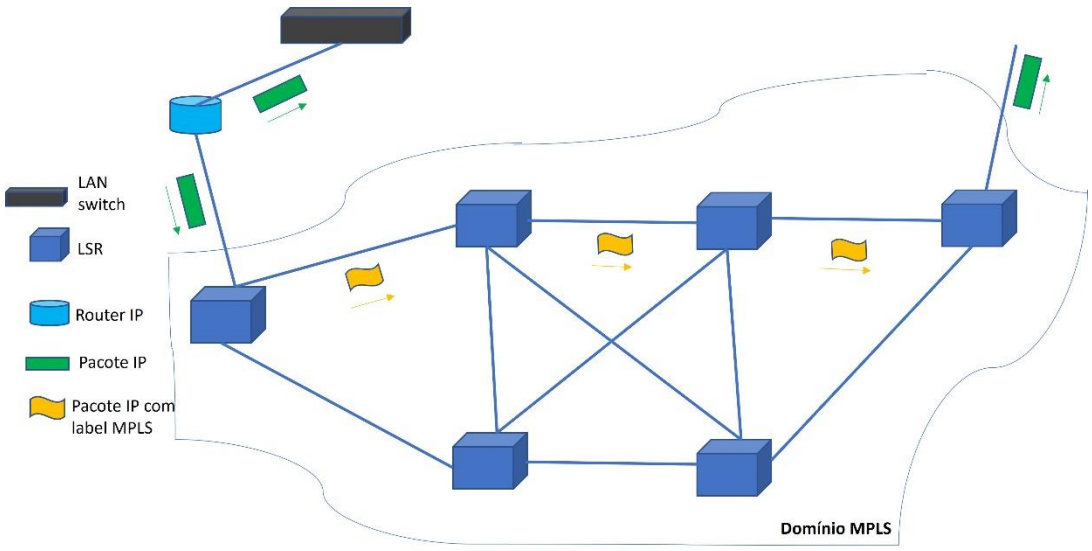

*Figura 2 - Funcionamento da rede MPLS [*adaptada de *Stallings, (2014)].*

Etiquetas

Uma das características intrínsecas do MPLS é o empilhamento de etiquetas. Este processo consiste na colocação de várias etiquetas num pacote, sendo que a primeira a ser processada é a que está no topo. Este mecanismo faz com que se possa criar túneis LSP. Assim o empilhamento de *labels* permite uma grande flexibilidade.

Relação entre FECs, LSPs e etiquetas

Seleção do tipo de encaminhamento no LSP: Existem duas opções para a seleção do LSP para uma determinada FEC: nó-a-nó (*hop-by-hop routing*) e encaminhamento explícito.

No encaminhamento nó-a-nó, cada LSR, independentemente, escolhe o próximo *hop* para cada FEC. Este tipo de encaminhamento não permite a gestão do tráfego. No entanto, no encaminhamento explícito, um único LSR, geralmente o *ingress* LSR ou o *egress* LSR (nó de entrada ou nó de saída, respetivamente), especifica vários LSRs no LSP para uma determinada FEC. Uma das vantagens deste tipo de encaminhamento é poder gerir-se a distribuição do tráfego na rede da forma mais conveniente.

Distribuição de etiquetas: *Label Distribution Protocol*

Em MPLS, existem vários protocolos que são usados para realizar a distribuição de *labels*. São exemplos o *Label Distribution Protocol* (LDP), *Resource Reservation Protocol-Traffic Engineering* (RSVP-TE) e ainda o multiprotocolo BGP (*Border Gateway Protocol*). Importa realçar o protocolo LDP porque surgiu com o principal objetivo de realizar a distribuição de etiquetas. Por conseguinte, estabelece LSPs em toda a rede através da troca de informação proveniente de protocolos de encaminhamento, tais como *Open Shortest Path First* (OSPF) e protocolo IS-IS (*Intermediate System to Intermediate System*), para atribuição e distribuição de etiquetas.

RSVP – TE:

O RSVP é um protocolo de sinalização, que é responsável pela realização da reserva de recursos e permite definir um caminho explícito de um túnel LSP. Este protocolo também é utilizado quando existem requisitos de QoS ou engenharia de tráfego (extensão ao RSVP para suporte TE). Na figura encontra-se o princípio do funcionamento do protocolo RSVP-TE, onde um *explicit route object* (ERO) define o caminho explícito que o LSP deve seguir e de seguida é enviada (em sentido oposto) a *reservation message* (RESV) contendo o objeto *LABEL*.

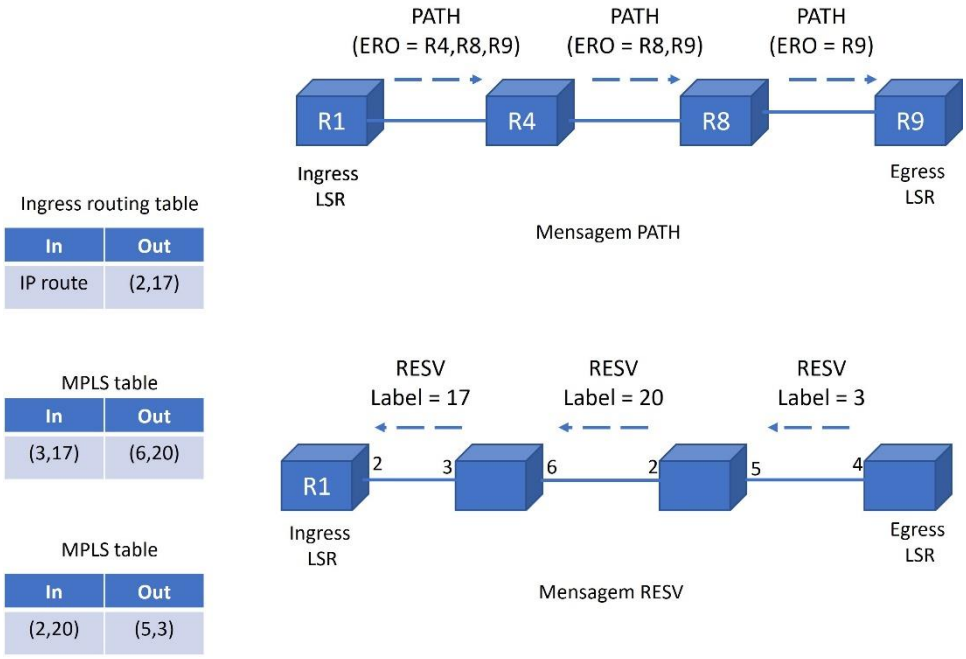

*Figura 3 - funcionamento do RSVP-TE [*adaptada de *Stallings (2014)].*

O pacote vai ser comutado desde o nó *ingress* até ao nó *egress*. Um nó *ingress* (na figura, R1) envia a mensagem *PATH* para definir um LSP consoante um determinado caminho explícito. Quando a mensagem chega ao destino, nó *egress*, este vai enviar uma mensagem RESV, fazendo o percurso inverso, distribuindo a etiqueta (*LABEL object*) ao caminho a que esta pertence [Stallings, (2004)].

Fast Reroute

O *Fast Reroute* (FRR) é uma técnica de proteção local que permite em caso de falha, que o tráfego seja redirecionado em 50 ms. Esta solução compreende dois esquemas distintos. No *One-to-one backup*, para cada LSP protegido é necessário estabelecer caminhos de proteção (*detour* LSP). No segundo esquema, o *Facility backup* protege recursos (nós ou *links*) recorrendo a *bypass tunnels*. Estes caminhos de proteção local têm origem no Ponto de Reparação Local (*Point of Local Repair* - PLR) e terminam no Ponto de Convergência (*Merge Point* – MP). A utilização do FRR permite solucionar de forma rápida cenários de falha, uma vez que encaminha, de forma imediata, o tráfego para o LSP que protege o LSP protegido [Pan *et al.* (2005)].

2.5. Redes óticas

2.5.1. Wavelenght Division Multiplexing - WDM

A multiplexagem por divisão do comprimento de onda (*Wavelenght Division Multiplexing* – WDM), permite uma maior capacidade de transmissão de uma fibra. Com a rede WDM, há a possibilidade de vários canais, cada um contendo um sinal ou informação, serem transmitidos em comprimentos de onda distintos na mesma fibra.

*Figura 4 - Multiplexagem por divisão do comprimento de onda [*adaptada de *Vasseur et al., (2004)].*

Redes óticas *meshed* (emalhadas)

No que diz respeito à arquitetura das redes óticas, para além de, por exemplo, redes em anel, existem também as redes do tipo *meshed*. Nestas redes, o componente mais importante é o *Optical Cross-Connect* (OXC) cuja principal função passa pelo *switch* do sinal entre vários portos de entrada e saída. Desta forma, torna-se mais fácil a comutação e encaminhamento. Existem várias classificações para os OXCs, entre elas opaco e transparente. Num OXC opaco o tráfego é convertido para o domínio elétrico e posteriormente é novamente convertido para domínio ótico. No entanto, estas conversões eletro-óticas e ótico-elétricas são dispendiosas.

Existem ainda mais dois tipos de OXC, *wavelength routing* (WR) e *wavelength translating* (WT). O primeiro é baseado na habilidade de converter o comprimento de onda do sinal entre portos de entrada e de saída. O segundo faz a transição do comprimento de onda para outro comprimento de onda de um sinal de entrada antes do sinal sair do porto de saída [Vasseur *et al.*, (2004)].

2.5.2. Optical Transport Network - OTN

Nesta secção é apresentada e discutida a arquitetura da rede OTN, sendo também feita uma enumeração das várias falhas e avarias possíveis de ocorrer em OTN e por fim são apresentados os mecanismos de recuperação neste tipo de redes.

Arquitetura da rede OTN

As camadas superiores do OTN são: *Optical channel Payload Unit* (OPU), *Optical channel Data Unit* (ODU) e *Optical channel Transport Unit* (OTU). A primeira camada, OPU, permite o suporte de vários tipos de sinais de cliente, tais como IP, Ethernet ou *Synchronous Digital Hierarchy* - SDH. A segunda camada, ODU, faz a monitorização de caminhos extremo-a-extremo. A última camada, OTU, faz a supervisão de sectores de regeneração.

As camadas inferiores do OTN são: *Optical Channel* (OCh), *Optical Multiple Section* (OMS) e *Optical Transmission Section* (OTS). Na primeira camada, inclui informação para desempenhar certas funções na gestão de falhas de canais óticos. Na segunda camada, OMS, inclui informação para desempenhar funções de funcionamento e manutenção de suporte ao sinal WDM multiplexado. Na última camada, OTS, inclui informação para desempenhar funções de suporte as secções de transporte óticas entre sectores de regeneração.

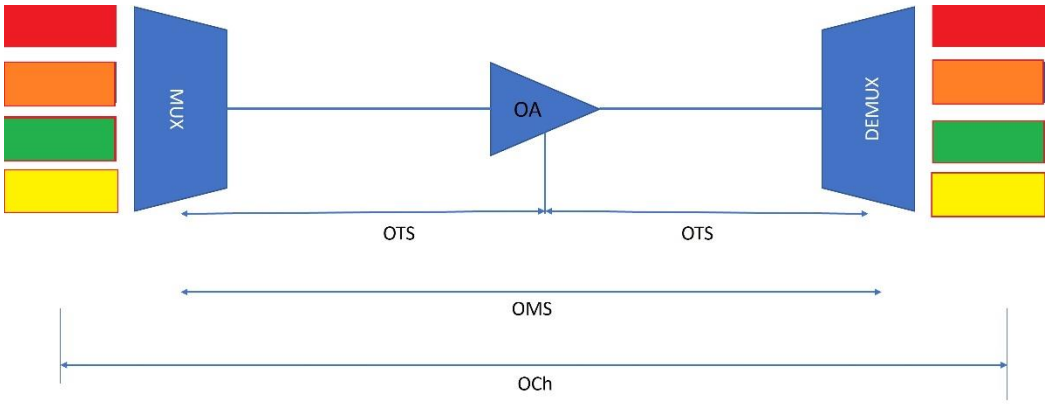

*Figura 5 - Canal ótico [*adaptada de *Vasseur et al. (2004)].*

A proteção pode ser dividida ao nível do canal ótico (único comprimento de onda) e ao nível do sinal multiplexado WDM, ou seja, da fibra toda.

Exemplos de falhas e avarias numa OTN

Depois de uma falha, ou avaria se for externa à rede, ser confirmada, é necessário tomar medidas para a rede OTN desempenhar determinadas ações em conformidade com a exigência da falha. Segundo os autores, alguns dos efeitos que podem ocorrer, são enumerados de seguida [Vasseur *et al.,* (2004)]:

- Perda de sinal (LOS);

- Degradação do Sinal (SDEG);
- Perda de tramas (LOF);
- *Alarm Indication Signal* (AIS);
- Indicação de perda do *Payload* (PMI).

Considerando por exemplo uma rede com uma fibra apta para transportar 200 comprimentos de onda em que cada cabo contém dentro 96 fibras, sendo estes agrupados em 5 cabos por conduta, verificou-se que em caso de falha devido a escavações e consequente corte do cabo, 5×96×200 = 96 mil comprimentos de onda seriam afetados, o que poderia gerar 96 mil alarmes por perda de sinal, LOS.

2.5.3. Recuperação em redes óticas

Como já foi dito no primeiro capítulo, os cortes nos cabos e as falhas de equipamentos são geralmente frequentes. Numa rede ótica, o esquema de recuperação pode operar ao nível do Canal Ótico, OCh, ou ao nível do OMS. Quando há a ocorrência de uma falha, ao nível do OCh, cada *lightpath* (ou caminho de luz) é passado para o seu *lightpath* de *backup* individualmente, ao passo que, ao nível do OMS, dá-se a multiplexação dos canais óticos transmitidos sobre uma única fibra passando posteriormente do caminho ativo para o caminho de *backup* noutras fibras [Vasseur *et al.* (2004)].

Noção de *Shared Risk Group* (SRG)

Quando se fala de recuperação em redes óticas introduz-se o conceito de *Shared Risk Group* (SRG). Os SRGs são elementos definidos na rede para representar riscos associados a um conjunto de falhas que podem ocorrer simultaneamente. Cada SRG tem associado um conjunto de *links* e nós que falham simultaneamente tendo também médias associadas, nomeadamente o MTTF (*Mean Time To Fail*) e o MTTR (*Mean Time To Repair*), ambos já explicados no primeiro capítulo. Assim sendo, a ocorrência de uma falha ou corte num cabo de fibra ótica irá afetar os caminhos ativo e de *backup*.

2.5.4. Mecanismos de recuperação em redes *meshed*

Os mecanismos de recuperação em redes emalhadas (*meshed*) podem ser divididos em esquemas de proteção e em esquemas de restauro. Outra distinção que se poderá fazer é o local onde é implementado o mecanismo de recuperação. Este poderá ser implementado ao nível da OMS ou da ODU. Neste último caso, cada canal ótico é

protegido individualmente entre os nós origem e destino. No caso da implementação ao nível da OMS, um conjunto de canais óticos multiplexados são protegidos entre os extremos da OMS, que eventualmente serão *optical cross-connects* (OXC). A proteção ao nível da ODU tem a vantagem de conseguir sobreviver a cenários de falha ou avaria nos nós, o que não acontece ao nível da OMS. Uma estratégia de reparação local ao nível dos *links* tem algumas desvantagens, sendo uma delas resultado do caminho de *backup* não ser o caminho mais curto [Vasseur *et al.* (2004)].

Proteção

Uma solução eficaz para recuperação de falhas em redes *meshed* é a utilização de um esquema de proteção. Na **proteção 1 + 1**, o mesmo sinal está a ser enviado pelos dois caminhos, e o destino só escolhe o que tem maior nível do sinal (o que estiver melhor). Na **proteção 1 : 1**, na altura da falha, o caminho de proteção, descarta o tráfego menos prioritário para passar a levar o tráfego protegido.

No início deste capítulo foram apresentados vários tipos de OXCs. Uma rede onde são instalados WR-OXCs (*wavelength routing*) é designada por *wavelength path* (WP), onde o caminho entre o OXC de origem e o OXC destino tem de transportar todos os *links* ao longo desse caminho, sendo que, o comprimento de onda tem obrigatoriamente de ser o mesmo. O segundo tipo, tal como foi dito previamente no início deste capítulo, são os WT-OXC que são instalados numa rede designada por *virtual wavelength path* (VWP). Este tipo de OXCs permite que que não haja restrição na continuidade de ter o mesmo comprimento de onda ao longo de todo o caminho.

Restauro

Os esquemas de restauro são superiores aos esquemas de proteção em termos de eficiência e capacidade. Por outro lado, o restauro, uma solução mais lenta, tem também a desvantagem de, em redes emalhadas, a implementação ser complexa e mais sofisticada. O restauro, orientado aos caminhos, tipicamente distribui os vários caminhos de *backup* sobre uma grande parte da rede, o que não acontece com o restauro ao nível dos *links*. Com este restauro, o processo de computação é mais simples e o número de ações de comutação (*switching*) é limitada comparativamente à anterior. Por fim, numa rede

meshed em que é usado um esquema de restauro, o tráfego extra (por exemplo tráfego não-prioritário) pode ser transportado pela capacidade de reserva que o esquema possui.

Proteção juntamente com restauro

Também é possível usar os dois esquemas simultaneamente. Poderá ser necessário recuperar tráfego prioritário e com isso pretende-se que seja recuperado com rapidez (fazendo uso da proteção 1+1) e por outro lado, no tráfego restante, poderá usar-se um esquema de restauro.

2.5.5. Disponibilidade

Para calcular a disponibilidade de uma rede ótica é necessário ter em conta os diferentes componentes da rede tais como OXCs que são compostos por várias partes ou componentes, sendo que cada componente tem os seus tempos MTBF e MTTR. Por outro lado, para ligar os nós na rede ótica WDM são necessários cabos físicos que formam uma ligação bidirecional, que também terão tempos MTBF e MTTR [Vasseur *et al.*, (2004)].

3. Estratégias de atribuição de recursos para restauro em redes MPLS

Após a ocorrência de uma avaria, é de extrema importância ter esquemas de recuperação sobretudo em redes de alto débito. Em [Ho *et al.* (2004)] é dada uma visão global sobre proteção partilhada em redes WDM e são apresentados problemas relacionados com o desenvolvimento e planeamento de esquemas de recuperação neste tipo de redes emalhadas. De seguida os autores discutem alguns problemas de encaminhamento para dois tipos de proteção em redes emalhadas – proteção ao caminho e proteção partilhada ao segmento. Posteriormente, são também discutidos em detalhe três algoritmos: *Iterative Two-Step-Approach* (ITSA), *Potential Backup Cost* (PBC) e o *Maximum Likelihood Relaxation* (MLR). O primeiro algoritmo, faz a computação dos caminhos ativo e de proteção, em que o primeiro é dado com o auxílio do algoritmo de *Dijkstra*. Contudo, estes caminhos podem não ser os ideais. O segundo algoritmo PBC propõe uma nova métrica para o custo do arco, com base na maior capacidade livre (capacidade que não está a ser utilizada pelos caminhos ativos) de entre todos os *links* que protegem esse arco. O par de caminhos de trabalho e de *backup* é o de mais baixo custo com base nessa métrica. O algoritmo MLR tenta estimar a melhor localização para o caminho ativo de tal modo a que a capacidade de *backup* desse caminho consiga ter capacidade partilhada extra. Por fim, os autores concluíram que o primeiro algoritmo é o mais eficiente em termos de probabilidade de bloqueio do tráfego e o segundo algoritmo é o mais eficiente em termos de rapidez computacional.

Há autores [Nelakuditi *et al.*, (2007)] que propõem um algoritmo de encaminhamento local FIR (*Failure Insensitive Routing*), baseado no *Interface Specific Forwarding*, que tornam as entradas das tabelas de encaminhamento "*interface-specific*" sem destabilizar a rede e sem criar ciclos no reencaminhamento. Sob execução do FIR, em caso de falha de um *link*, o nó (neste caso um router) adjacente suspende a notificação ao nível global que ocorreu a falha e de seguida desencadeia o encaminhamento local do fluxo de dados, que previamente iria passar pelo *link* que falhou. Esta estratégia foi proposta para ser uma alternativa aos protocolos convencionais OSPF/IS-IS. Os autores concluíram que o algoritmo de encaminhamento FIR oferece maior estabilidade e disponibilidade que o protocolo OSPF, não dependendo de nenhum fator externo (tamanho da rede ou tempos de convergência).

Outros autores [Lu *et al.*, (2018)] propõem um esquema de restauro com FRR aplicado a uma rede multicamada MPLS sobre WDM (designado por *mixed scheme*), nomeadamente com o pré-estabelecimento dos caminhos ativos e de *backup*, de modo a terem um tempo de recuperação mais reduzido que o habitual. Propõem também outro algoritmo de encaminhamento (MaxE2E) que teve como objetivo melhorar a utilização da rede através da distribuição balanceada de tráfego por diferentes comprimentos de onda. Estas estratégias foram comparadas com um esquema de proteção 1:1:1 e um algoritmo de encaminhamento convencional que conta o número de saltos (*hops*). Concluíram que ambos os algoritmos propostos distribuem os recursos na rede de forma mais eficiente para topologias com um grau do nó elevado.

Foram estudadas duas estratégias, propostas em [Alicherry *et al.*, (2007)], para reservar largura de banda para restauro local. Estas estratégias foram implementadas para a determinação dos recursos envolvidos no restauro local dos arcos da rede em situações de falha. A primeira estratégia, baseada no conceito de árvore abrangente mínima, é descrita na secção 3.1 e a segunda estratégia, mais elaborada, é descrita na secção 3.2.

Foram ainda estudados e implementados os algoritmos de base para este trabalho, os algoritmos de *Dijkstra* e de *Prim* para cálculo do caminho mais curto e da árvore abrangente mínima, respetivamente. Ambos os algoritmos recorrem a uma *heap* binária, de modo a resolver estes problemas de forma eficiente.

Todos os algoritmos foram desenvolvidos no ambiente de desenvolvimento integrado *Eclipse* em linguagem *Java*.

3.1. Estratégia 1: árvore abrangente mínima

A primeira estratégia é baseada num algoritmo de cálculo de uma árvore abrangente mínima. De todos os algoritmos existentes, escolheu-se o de *Prim*, que se descreve a seguir. A saída deste algoritmo é uma árvore *T* abrangente, sendo esta a árvore para proteção num dado grafo *G*.

3.1.1. Algoritmo de *Prim*

O algoritmo de *Prim* constrói uma árvore abrangente *T* a partir do conceito de corte selecionando o arco de custo mínimo em cada corte que considera na rede. O primeiro corte é constituído pelo conjunto de arcos que interliga o nó inicial na árvore com o resto da rede. A árvore abrangente vai sendo construída a partir dos cortes sucessivos definidos pela interligação dos nós que fazem parte da árvore com os nós que ainda não fazem. O resultado deste algoritmo é uma árvore abrangente *T* que contém todos os nós presentes no grafo *G* e *N-1* arcos.

De seguida, no Algoritmo 1, é apresentado o pseudo-código do algoritmo de *Prim* [Ahuja *et al.*, (1993)].

Considera-se a notação definida como se segue. O grafo G é constituído pelo conjunto N de nós v_i com $i = 1, \ldots, |N|$ e o conjunto E de arcos $e_{i,j}$ que interligam o nó v_i com o nó v_j, o custo do arco $e_{i,j}$ é definido por $c_{i,j}$. A(i) é a lista de arcos adjacentes ao nó v_i, ou seja, $A(i) = \{e_{i,j} \in E : v_j \in N\}$. A descrição da *heap* é feita adiante.

Algoritmo 1: Prim utilizando a *heap*

```
     Entrada: Grafo G
     Saída: T, Árvore abrangente mínima
1    inicialização
2        cria a heap (H)
3        para cada v_j ∈ N \ {v_1} faz
4            chave(v_j) = C + 1
             // em que C é o custo máximo dos arcos contidos em T
5        fim
6        chave(v_1) = 0
7        predecessor(v_1) = 0
8        para cada v_j ∈ N faz
9            insere (v_j, H)
10       fim
11       T = ∅
12       enquanto |T| < (|N| – 1) faz
13           encontra-min (v_i, H)
```

```
14                                  apaga-min (v_i, H)
15                                  T = T ∪ (predecessor (v_i), v_i, chave (v_i))
16                                  para cada e_i,j ∈ A(i) com v_j ∈ H faz
17                                      se chave(v_j) > cij então
18                                          chave(v_j) = c_ij
19                                          predecessor (v_j) = v_i
20                                          decrementa-chave (c_ij, v_j, H)
21                                      fim
22                                  fim
23                              fim
24                          fim
```

3.1.2. Atribuição de LB de *backup*

Todos os arcos contidos em T, são arcos para proteção, i.e., a capacidade total do arco $u(e_{i,j})$ vai ser reservada para *backup*, i.e., $p(e_{i,j}) = u(e_{i,j})$ e a capacidade de trabalho do arco $e_{i,j}$, $w(e_{i,j}) = 0$. Assim sendo, não é necessário reservar capacidade de *backup* para os arcos pertencentes à árvore.

Para todos os arcos que não estão contidos em T, a capacidade de trabalho é máxima ($w(e_i) = u(e_i)$ e consequentemente $p(e_i) = 0$), sendo necessário reservar capacidade para restauro destes arcos na árvore.

3.2. Estratégia 2: *2-Edge*

A segunda estratégia implementada, é designada [Alicherry *et al.*, (2007)] por *2-Edge*. A árvore T que sai da primeira estratégia, vai ser usada nesta segunda estratégia para determinação das capacidades que vão sendo atribuídas a cada arco.

Para todos os arcos do grafo, que não pertencem a T, vai ser analisado arco a arco se algum dos extremos é *2-edge*. Se algum dos extremos for *2-edge*, esse arco não vai ser tido em consideração para ter capacidade de *backup* e a sua capacidade de trabalho vai ser máxima ($w(e_{i,j}) = u(e_{i,j})$ e consequentemente $p(e_{i,j}) = 0$).

Diz-se que v_i e v_j não são 2-*edge connected* (ou seja, ligados por dois arcos) em F, se ao remover qualquer *link* em F não se desligar v_i de v_j.

Caso ambos os extremos do arco $e_{i,j}$ não sejam *2-edge*, ou seja, ambos são nós folha da árvore T, $e_{i,j}$ vai ser adicionado ao grafo F que é inicialmente igual à árvore T. Ao adicionar $e_{i,j}$ a F, é criado um ciclo. Para todos os arcos no ciclo C menos os arcos contidos no conjunto M, inicialmente vazio, vai existir para além da capacidade de trabalho, também capacidade de *backup*. Todos os arcos nesta primeira iteração são adicionados ao conjunto M, que representa assim o conjunto de arcos que vão ter capacidade de *backup* atribuída.

De seguida, é apresentado o pseudo-código da Estratégia 2 [Alicherry *et al.*, (2007)].

Estratégia 2

Entrada: *Grafo G e árvore T, do algoritmo de Prim*

Saída: *M, conjunto de arcos com a capacidade de backup atribuída*

```
1  inicialização
2      Sejam os arcos {e_{i,j}, ∈ E} ordenados por ordem não crescente de capacidade
       F = árvore T
       M = ∅
3      para todos os arcos e_{i,j} ∉ T faz
4          se v_i ou v_j não forem 2-edge connected em F
5              F = F ∪ e_{i,j}
               C, ciclo formado ao adicionar e_{i,j} em F
               para cada link e_{k,l} ∈ C – M faz
6                  p(e_{k,l}) = w(e_{i,j})
                   u(e_{k,l}) = w(e_{k,l}) + p(e_{k,l})
                   M = M ∪{ e_{k,l} }
7          senão
8              u(e_{ij}) = w(e_{i,j})
               p(e_{i,j})=0
9      fim
10 fim
```

É de notar que tanto a estratégia 1 como a estratégia 2 [Alicherry *et al.*, (2007)] não referem como se contabiliza a capacidade de trabalho em cada *link* e, por conseguinte, não levam em conta as matrizes de tráfego. Para além disso, a forma como contabilizam a capacidade de trabalho e a capacidade de *backup* não está totalmente correta o que levou a um esforço adicional de correção do segundo algoritmo. Assim, todo o dimensionamento da capacidade de trabalho e de *backup* usados neste trabalho, foi feito de raiz nesta dissertação de mestrado para as estratégias referidas anteriormente de modo a poderem ser testadas no simulador realista como o Net2plan.

3.3. Algoritmo complementar

3.3.1. Algoritmo de *Dijkstra*

Este algoritmo, apresentado por Edsger Dijkstra, encontra o caminho de menor custo entre um nó fonte (v_s) e todos os outros nós, sendo aplicável a redes cujos custos dos arcos sejam não negativos. É, assim, suposto existir pelo menos um caminho entre o nó fonte e qualquer outro nó. O algoritmo vai guardando uma chave para cada nó (v_i), que corresponde a um limiar superior do caminho de menor custo, até chegar ao nó fonte. Primeiramente, é atribuído ao nó fonte uma chave e predecessor nulos ($chave(v_s) = 0$, $pred(v_s) = 0$) e aos restantes nós é atribuída uma chave com valor a infinito ($chave(v_i) = \infty$). De seguida, no Algoritmo 2, é apresentado o pseudo-código do algoritmo de *Dijkstra* [Ahuja *et al.*, (1993)].

Algoritmo 2: Dijkstra utilizando a *heap*

Entrada: *source node* (v_s)

Saída: T, *Árvore dos caminhos de menor custo desde* v_s, *para todos os outros nós*

```
1  inicialização
2      cria a heap (H)
3      chave(v_j) = ∞ para todos os v_j ∈ N
4      chave(v_s) = 0 e predecessor(v_s) = 0
5      insere na heap (v_s, H)
6      enquanto a heap H ≠ ∅ faz
7          encontra-min (v_i, H)
8          apaga-min (v_i, H)
9          coloca na árvore (v_i, T)
10         para cada e_i,j ∈ A(i) faz
11             valor = chave(v_i) + c_ij
12             se chave (v_j) > valor então
13                 se chave(v_j) = ∞ então
14                     chave(v_j)=valor, predecessor(v_j)=v_i e insere(v_i, H)
15                 fim
16                 senão
17                     d(v_j) = valor, predecessor(v_j) = v_i e decrementa-chave(valor,
                       v_j, H)
18                 fim
19             fim
20         fim
21     fim
22 fim
```

3.3.2. Heap

Uma *heap* é uma estrutura de dados em árvore, que é utilizada para melhorar o desempenho de diversos algoritmos devido à sua capacidade de armazenar e ordenar, de uma forma eficiente. Neste trabalho, é utilizada uma *heap* binária que recorre a uma árvore binária e que vai sendo preenchida da esquerda para a direita. Todos os nós armazenados na *heap* têm um valor real associado designado por chave, possibilitando a

troca de posições. É utilizada uma *heap* mínima, sendo que a raiz é o nó que contém o valor de chave mínima (o valor mais pequeno).

Esta estrutura de dados permite executar diversas operações [Ahuja *et al.*, (1993)], nomeadamente:

- Cria-*heap* (H): cria uma *heap* vazia;
- Encontra-min (i, H): encontra e retorna o objeto *i* com chave mínima na *heap*;
- Insere (i, H): insere um novo objeto *i* com uma chave na *heap*;
- Decrementa-chave (valor, i, H): decrementa o valor da chave de um objeto *i*, substitui por *valor* e restaura a propriedade da *heap*;
- Apaga-min (i, H): apaga o objeto *i* com chave mínima na *heap*.

4. Simulações realizadas e análise de resultados

Neste capítulo é feita uma análise através do simulador Net2Plan 0.3.1 das estratégias desenvolvidas. Esta ferramenta foi desenvolvida em 2011, com o objetivo de ser utilizada no curso de Engenharia de Redes da Universidade Politécnica de Cartagena em Espanha. Neste momento existem vários simuladores disponíveis para planeamento de redes, nomeadamente OMNeT++, NS-3, entre outros. Atualmente a versão mais recente do Net2Plan é a versão 0.6.6 de Abril de 2020, mas para o uso de segmentos de proteção para proteção local foi utilizada a versão 0.3.1 de Novembro de 2015. Os algoritmos foram desenvolvidos na linguagem *Java.*

Nesta versão, o simulador tem disponível dois algoritmos:

- "*NRSim_AA_genericProtectionAlgorithm*" dedicado à *Proteção*;
- "*NRSim_AA_genericRestorationAlgorithm"* dedicado ao *Restauro*;

Nas experiências que se seguem foi testada proteção dedicada ao arco em que se usou o algoritmo "*NRSim_AA_genericProtectionAlgorithm*" utilizando segmentos de proteção aos arcos. Foram ainda testadas diferentes formas de restauro em diferentes situações de falhas de arcos, em que se utilizou o algoritmo "*NRSim_AA_genericRestorationAlgorithm*" com a opção *link restoration*.

4.1. Configuração da Rede 1

A primeira rede de teste, que será designada por Rede 1, foi configurada no simulador de raiz. Trata-se de uma rede com 5 nós e 8 arcos bidirecionais, sendo o grau médio do nó de 3.2. Em cada um dos fluxos de tráfego, ou em cada *demand*, o tráfego oferecido ponto-a-ponto é de 50 Mbps, sendo o tráfego oferecido total de 1000 Mbps.

Nas experiências que se descrevem a seguir foram consideradas duas formas de encaminhamento de tráfego: i) através dos caminhos mais curtos existentes na rede; ii) através dos caminhos mais curtos existentes na rede excluindo os arcos pertencentes à árvore abrangente dedicada à proteção, como adiante se descreve.

Todas as simulações, foram realizadas com um tempo de simulação e tempo de *warmup* iguais de modo a serem comparadas numa situação em que existiu um número

significativo de falhas. Estas falhas nos *links* podem ser falhas simples e falhas múltiplas (falhas duplas), sendo o MTTR de 12h e o MTTF de 8748h. Como o número de falhas não é exatamente o mesmo em todas as simulações, foram feitas 10 simulações para cada um dos casos simulados, e posteriormente, foram recolhidos e calculados a média e o desvio padrão do total de falhas, do tempo em que há falhas simples ou duplas, do tráfego perdido e da disponibilidade da rede.

4.1.1. Proteção dedicada ao arco – caso 1

No primeiro caso optou-se por implementar proteção dedicada ao arco para a Rede 1. Para os segmentos de proteção de cada arco escolheu-se o caminho mais curto entre os extremos desse arco e reservou-se LB nesse segmento igual à LB usada no arco que esse segmento protege. As capacidades totais e de trabalho de cada arco são apresentadas na figura seguinte, assim como os segmentos de proteção que protegem cada arco. Relembra-se que *u* define a capacidade total de arco e *w* define a capacidade de trabalho, sendo que a capacidade reservada para proteção é a diferença entre os dois valores. Os segmentos desenhados na Fig. 6 são bidirecionais e, por conseguinte, protegem os arcos em ambos os sentidos. Como já foi referido cada segmento de proteção é o caminho mais curto entre os extremos do arco, excluindo o próprio arco. A LB total instalada nesta configuração é de 3600 Mbps.

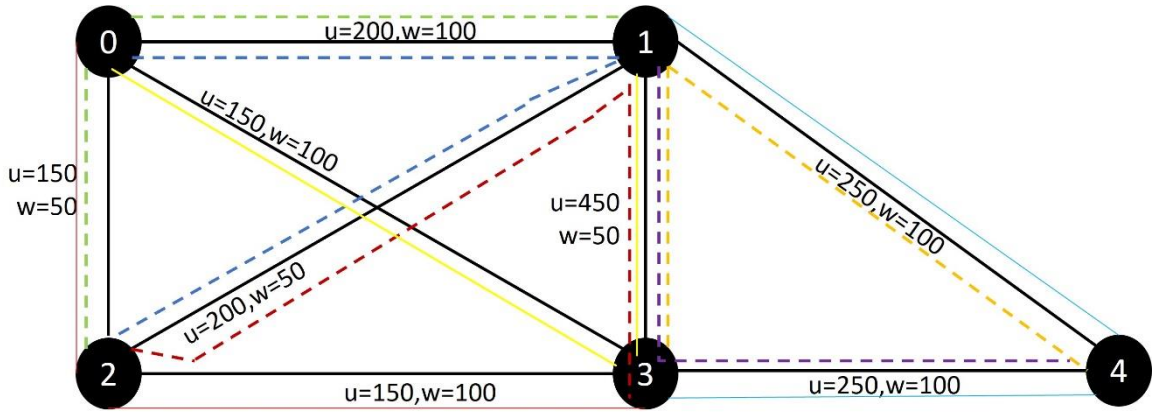

Figura 6 – Capacidades, em Mbps, e segmentos de proteção para cada arco (Proteção – caso 1).

Reação em caso de falha

O tráfego transportado foi, em média, de 999.993 Mbps num máximo de 1000 Mbps, tendo a disponibilidade média da rede sido de 0.999985. Não houve perda de tráfego no caso de falhas simples, contudo, houve uma ínfima percentagem de tráfego bloqueado em 4 das 20 *demands* de 50 Mbps em média, havendo tráfego perdido durante falhas múltiplas onde falharam em simultâneo o caminho ativo e de proteção dessas *demands*. É de notar que em algumas das falhas simultâneas não se perde tráfego.

Nesta simulação a média do total de falhas é de 1503,8 falhas. Em 98,909% do tempo a rede não teve qualquer falha, em 1,0856% do tempo a rede teve apenas falhas simples e em 0,0051% do tempo, a rede teve duas falhas em simultâneo.

Tabela 1 - Distribuição de falhas na rede: Proteção - caso 1 (Média/Desvio Padrão)

Total de falhas	% de tempo sem falhas	% de tempo com 1 falha	% de tempo com 2 falhas	Tráfego Transportado (Mbps)	Disponibilidade da rede
1503,8 / 37,6771	98,909 / 0,04055	1,0856 / 0,0413	0,0051 / 0,0019	999.9932 / 0.0112	0.999985 / 0,000007

No pior caso, numa das simulações, esta configuração de proteção ao arco conduz à perda de tráfego de 400 Mbps em situações de 2 falhas em simultâneo. Um exemplo onde se perde tráfego, é nas falhas simultâneas dos arcos (0,1) e (1,3).

4.1.2. Proteção dedicada ao arco: Estratégia 1 - caso 2

Neste caso estudou-se uma estratégia de proteção ao arco baseada na estratégia 1 descrita no capítulo anterior. Assim, existe na rede uma árvore abrangente apenas dedicada à proteção. Os arcos dessa árvore foram desenhados na Fig. 6 com um tracejado mais carregado. Os caminhos de proteção são o caminho mais curto na árvore, entre os extremos do arco que falha. O caminho ativo é o mais curto no resto da rede (excluindo a árvore de proteção). A LB total instalada desta configuração é de 6000 Mbps, sendo a que tem mais LB instalada de todas as estratégias aqui discutidas. É assim 66,6% superior à capacidade instalada no caso anterior.

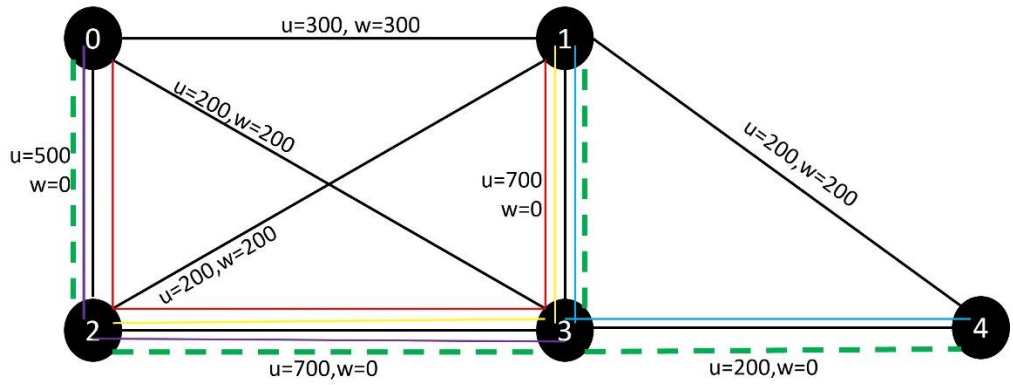

Figura 7- Árvore de proteção e capacidades nos arcos, em Mbps (Proteção - caso 2).

É de notar que a Estratégia 1 não está pensada para proteção dedicada, mas sim para restauro, no entanto esta estratégia de restauro não foi possível ser testada neste simulador. Assim sendo, as capacidades que deviam estar envolvidas, na situação de restauro, não são as que estão na Fig. 7, mas sim as capacidades que estão na Fig. 8, sendo que os resultados de simulação iriam ser semelhantes aos obtidos para o caso da proteção que está a ser considerada. É importante referir que a diferença entre proteção e restauro não pode ser avaliada neste simulador. Assim sendo, a capacidade a ser instalada seria de 4000 Mbps (inferior em 33,3% ao que foi testado).

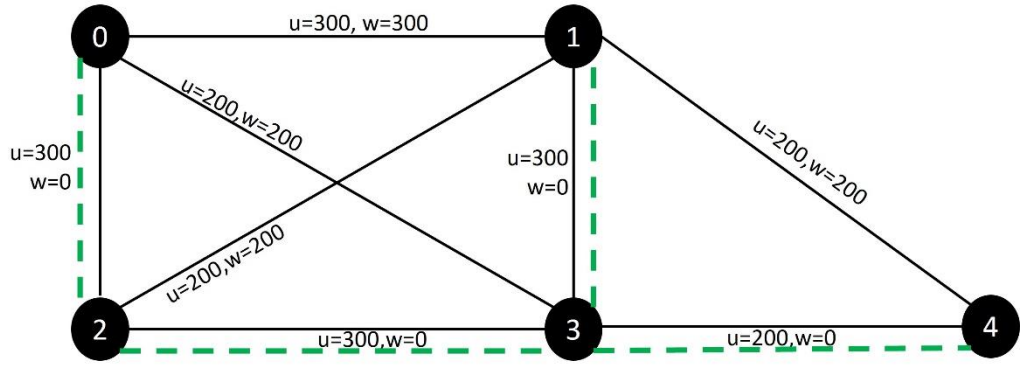

Figura 8 - Árvore de proteção e capacidades nos arcos, em Mbps (Restauro – caso 2).

Reação em caso de falha

O que se pretende é ter um caminho com capacidade de tráfego suficiente para passar todo o tráfego que está a ser oferecido e que iria passar pelo arco que falhou. O

tráfego transportado foi, em média, de 999.991 Mbps num máximo de 1000 Mbps. A disponibilidade da rede foi, em média, de 0.99998. Não houve perda de tráfego no caso das falhas simples, contudo, houve alguma percentagem de tráfego bloqueado em 6 das 20 *demands* de 50 Mbps em média, havendo tráfego perdido durante falhas múltiplas onde falharam em simultâneo o caminho ativo e de proteção.

Tabela 2 - Distribuição de falhas na rede: caso 2 (Média / Desvio Padrão)

Total de falhas	% de tempo sem falhas	% de tempo com 1 falha	% de tempo com 2 falhas	Tráfego Transportado (Mbps)	Disponibilidade da rede
1519 / 34,7505	98,909 / 0,0324	1,085 / 0,0318	0,0053 / 0,0022	999,991 / 0,0065	0.99998 / 0,000015

No pior caso, numa das simulações, esta configuração conduz à perda, no máximo, de 600 Mbps em situações de múltipla falha. Assim sendo, esta estratégia é menos eficiente que a anterior em termos de LB e de tráfego transportado.

4.1.3. *Link restoration* – caso 3

Neste caso simulou-se uma estratégia de restauro utilizando o algoritmo de provisionamento referido anteriormente com a opção "*link restoration*", em que os caminhos ativos são os caminhos mais curtos e os segmentos de proteção ao *link* são os caminhos mais curtos que interligam as extremidades do *link* com capacidade para proteger o tráfego em causa. As capacidades totais e de trabalho de cada arco são apresentadas na figura seguinte. Relativamente a esta configuração, o algoritmo de provisionamento não utiliza segmentos de proteção por ser um algoritmo de restauro.

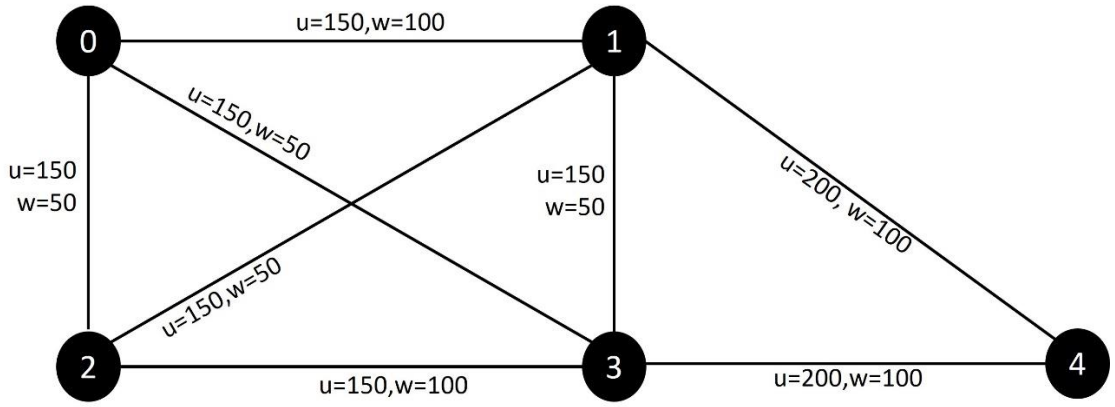

Figura 9 - Capacidade dos arcos, em Mbps (Restauro - caso 3).

O encaminhamento de tráfego (ativo e de proteção) vai ser igual ao do primeiro caso. A LB total instalada nesta configuração é de 2600 Mbps, sendo inferior em 27,8% quando comparado com o primeiro caso de proteção dedicada e inferior em 56,7% ao segundo caso.

Reação em caso de falha

O tráfego transportado foi, em média, de 999.998 Mbps num máximo de 1000, tendo uma disponibilidade, em média, de 0.99997. Não houve, novamente, perda de tráfego no caso de falhas simples, contudo, houve alguma percentagem de tráfego bloqueado em 4 das 20 *demands* de 50 Mbps em média, havendo tráfego perdido durante falhas múltiplas onde falham em simultâneo o caminho ativo e de proteção dessas *demands*. É de notar, mais uma vez, que em algumas situações com 2 falhas em simultâneo não se perde tráfego.

Nesta simulação a média do total de falhas é de 1511.2 falhas. Em 98,908% do tempo a rede não teve qualquer falha, em 1,0861% do tempo a rede teve apenas falhas simples e em 0,0054% do tempo, a rede teve duas falhas em simultâneo.

Tabela 3 - Distribuição das falhas na rede: Restauro – caso 3 (Média / Desvio Padrão)

Total de falhas	% de tempo sem falhas	% de tempo com 1 falha	% de tempo com 2 falhas	Tráfego Transportado (Mbps)	Disponibilidade da rede
1511,2 / 24,6487	98,908 / 0,0335	1,0861 / 0,0327	0,0054 / 0,0026	999.9977 / 0,0008	0.99997 / 0,000011

Na pior situação, em que houve duas falhas simultâneas, perde-se no máximo, 400 Mbps do tráfego. Contudo, sendo um algoritmo de restauro, esta configuração precisa de menos LB para proteção dos caminhos. Pode-se concluir que esta estratégia requer menos LB do que a primeira estratégia de proteção, sendo idêntica em termos do tráfego transportado.

4.1.4. *Link restoration*: Estratégia 2 – caso 4

Neste caso simulou-se novamente uma estratégia de restauro utilizando o algoritmo de provisionamento "*link restoration*". As capacidades totais e de trabalho de

cada arco são apresentadas na figura seguinte e segue a segunda estratégia descrita no capítulo anterior para determinação da capacidade de *backup* dos arcos. Tal como no caso anterior, não existem segmentos de proteção. A LB total que foi instalada nesta configuração é de 2400 Mbps, inferior em 7,7% em relação ao caso anterior de restauro, inferior em 33,3% em relação ao primeiro caso de proteção dedicada e, por fim, inferior em 60% quando comparado com o segundo caso.

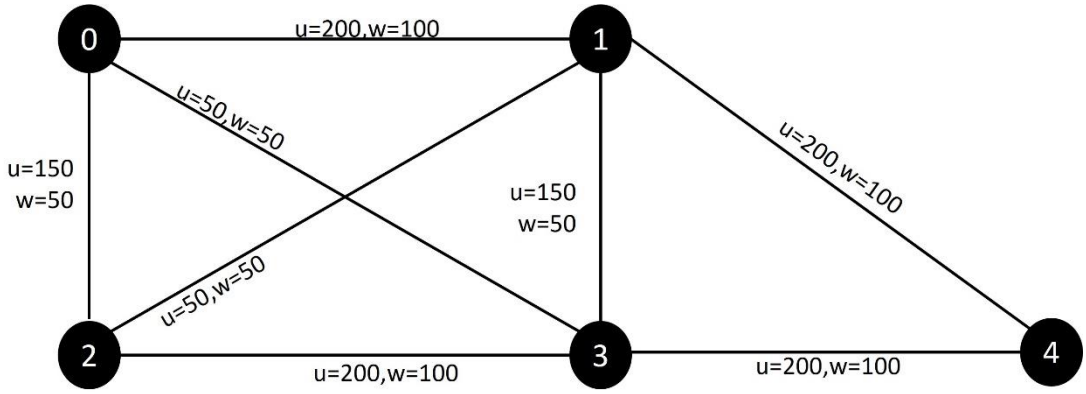

Figura 10- Capacidade dos arcos, em Mbps (Restauro - caso 4).

Reação em caso de falha

O tráfego transportado foi, em média, de 999.993 Mbps num máximo de 1000 Mbps, tendo a disponibilidade da rede sido de 0.99996. Não houve perda de tráfego no caso de falhas simples, havendo uma ínfima percentagem de tráfego bloqueado em 2 das 20 *demands* de 50 Mbps em média, havendo tráfego perdido durante falhas múltiplas onde falham, em simultâneo, o caminho ativo e de proteção destas *demands*. Novamente, há algumas falhas múltiplas em que não se perde tráfego.

Nesta simulação a média do total de falhas é de 1510,6 falhas. Em 98,895% do tempo a rede não teve qualquer falha, em 1,0997% do tempo a rede teve apenas falhas simples e em 0,0051% do tempo, a rede teve duas falhas em simultâneo.

Tabela 4 - Distribuição de falhas: Restauro – caso 4 (Média / Desvio Padrão)

Total de falhas	% de tempo sem falhas	% de tempo com 1 falha	% de tempo com 2 falhas	Tráfego Transportado (Mbps)	Disponibilidade da rede
1510,6 / 53,3558	98,895 / 0,0571	1,0997 / 0,0572	0,0051 / 0,0020	999,9934 / 0,0029	0.99996 / 0,000016

Esta configuração tem um desempenho muito idêntico ao anterior sendo que, no pior dos casos, também vai perder, no máximo, 400 Mbps do tráfego. Comparativamente às estratégias anteriores e no que diz respeito ao tráfego transportado, observa-se que é muito idêntica. Assim sendo, conclui-se que esta estratégia é a mais eficiente de todas em termos da LB instalada na rede.

4.2. Configuração da Rede 2

A segunda rede de teste (NSFNet_N14_E42_complete), que será designada por Rede 2, é uma rede que tenta mimetizar a rede entre universidades americanas. Esta rede foi configurada no simulador ajustando todos os valores dos fluxos de tráfego para o maior valor de entre os dois valores de tráfego existentes entre os mesmos nós da rede e para valores arredondados às unidades, de modo a facilitar todo o processo de dimensionamento das capacidades nesta segunda rede. Alteraram-se também diversos caminhos de modo a ter os caminhos mais curtos entre os nós da rede. Trata-se de uma rede com 14 nós e 21 arcos bidirecionais, tendo a rede um grau do nó médio igual a 3. O tráfego oferecido total é de 4084 Mbps para todas as situações. Os fluxos de tráfego em ambos os sentidos entre os mesmos nós da rede e as distâncias dos arcos, são apresentados na figura seguinte.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0		30	12	31	33	18	24	26	19	22	17	22	17	23
1	1100		13	30	16	19	26	18	15	12	19	19	11	17
2	1600	600		31	17	26	11	25	12	26	23	16	17	13
3	-	1000	-		15	15	15	20	22	21	17	11	10	25
4	-	-	-	600		15	35	23	27	25	20	23	21	21
5	-	-	2000	-	1100		24	22	19	17	22	16	22	25
6	-	-	-	-	800	-		25	13	26	26	14	31	21
7	2800	-	-	-	-	-	700		27	37	33	27	30	37
8	-	-	-	-	-	-	-	700		21	30	36	32	15
9	-	-	-	-	-	1200	-	-	900		29	26	21	28
10	-	-	-	2400	-	-	-	-	-	-		42	29	19
11	-	-	-	-	-	-	-	-	500	-	800		37	34
12	-	-	-	-	-	-	-	-	500	-	800	-		22
13	-	-	-	-	-	2000	-	-	-	-	-	300	300	

Figura 11 - Fluxos de tráfego em Mbps (parte triangular superior) e distância dos arcos em quilómetros (parte triangular inferior) para a Rede 2.

Nas experiências que se descrevem a seguir foi considerada somente uma forma de encaminhamento de tráfego – através dos caminhos mais curtos existentes na rede. Os fluxos de tráfego, em ambos os sentidos são iguais, e são apresentados na figura seguinte.

4.2.1. *Link restoration*: Estratégia 1 – caso 1

Nesta rede não se pode aplicar a Estratégia 1 por não existirem arcos suficientes nesta topologia.

4.2.2. *Link restoration* – caso 2

Neste caso simulou-se uma estratégia de restauro utilizando o algoritmo de provisionamento referido no início deste capítulo com a opção "*link restoration*", em que os caminhos ativos são os caminhos mais curtos e os segmentos de proteção ao *link* são os caminhos mais curtos que interligam as extremidades do *link* com capacidade para proteger o tráfego em causa. As capacidades totais e de trabalho em cada arco são apresentadas na figura seguinte.

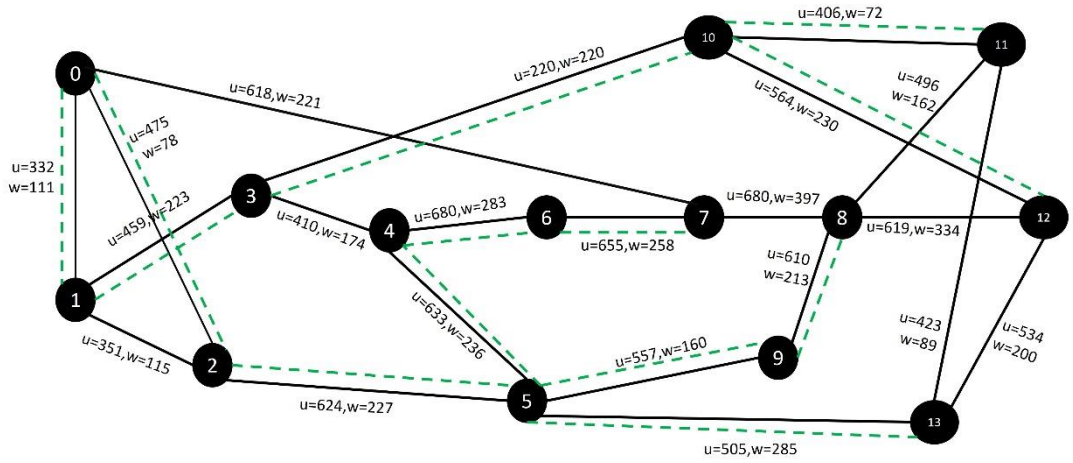

Figura 12 - Topologia da Rede 2: árvore de proteção e capacidades envolvidas para o Restauro - caso 2 (adaptada de Ramaswami e Sivarajan, 1996).

A LB total instalada nesta configuração é de 21702 Mbps, ou 21.7Gbps, tendo sido ajustada por simulação.

Reação em caso de falha

O tráfego transportado foi, em média, de 4083,97 Mbps num máximo de 4084 Mbps, tendo a disponibilidade, em média, sido de 0.9998. Não houve perda de tráfego quando ocorrem falhas simples, contudo, existe uma pequena percentagem de tráfego bloqueado em algumas das *demands*, havendo tráfego perdido quando ocorrem duas falhas em simultâneo, falhando o caminho ativo e o caminho de proteção.

É de notar que a disponibilidade nesta rede é inferior ao mesmo caso na Rede 1, dado que o tamanho da rede leva à ocorrência de mais falhas simultâneas para as quais a rede não está preparada.

Nesta simulação a média do total de falhas é de 3992,2 falhas. Em 97,161% do tempo a rede não teve qualquer falha, em 2,80% do tempo teve apenas falhas simples e em 0,005% do tempo, a rede teve duas falhas em simultâneo. De notar que houve, em algumas simulações, uma ínfima percentagem de tempo (0,001%) que houve três falhas em simultâneo, mas como não ocorreram em todos os 10 casos, não foram contabilizadas.

Tabela 5 – Distribuição de falhas: Restauro – caso 2 (Média / Desvio padrão)

Total de falhas	% de tempo sem falhas	% de tempo com 1 falha	% de tempo com 2 falhas	Tráfego Transportado (Mbps)	Disponibilidade da rede
3992,2 / 42,5295	97,161 / 0,0562	2,7999 / 0,0552	0,0053 / 0,0022	4083,9695 / 0,0095	0.99978 / 0,000006

Na pior situação, em que houve falhas múltiplas, perdeu-se no máximo 689 Mbps do tráfego.

4.2.3. *Link restoration*: Estratégia 2 – caso 3

Neste segundo caso simulou-se uma estratégia de restauro utilizando o algoritmo de provisionamento com a opção "*link restoration*", em que os caminhos ativos são os caminhos mais curtos e os segmentos de proteção ao *link* são os caminhos mais curtos que interligam as extremidades do *link* com capacidade para proteger o tráfego em causa. As capacidades totais e de trabalho em cada arco são apresentadas na figura seguinte.

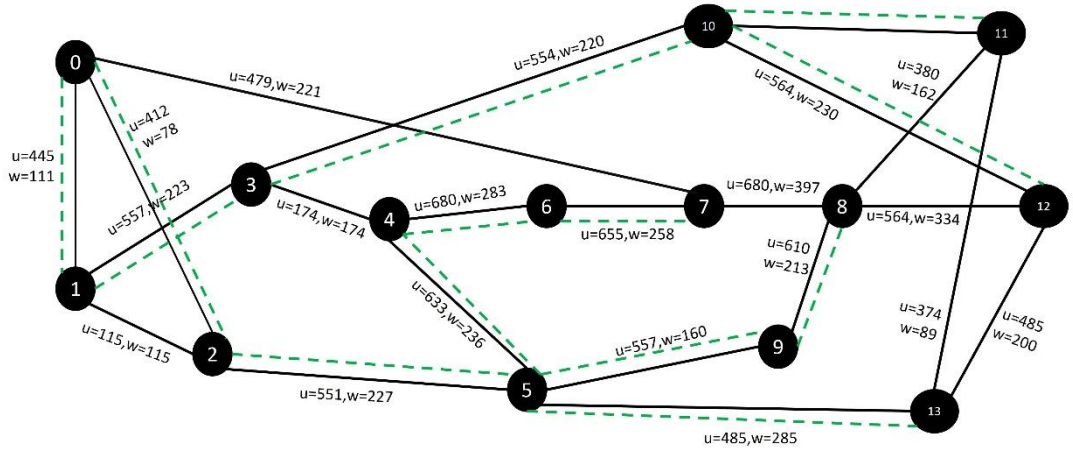

Figura 13 - Topologia da Rede 2: árvore de proteção e capacidades envolvidas para o Restauro - caso 3 (adaptada de Ramaswami e Sivarajan, 1996).

A LB total instalada nesta configuração é de 20414 Mbps (20.4 Gbps), inferior em 6,32% ao caso anterior. O tráfego oferecido total é, novamente, 4084 Mbps.

Reação em caso de falha

O tráfego transportado foi, em média, de 4083,47 Mbps num máximo de 4084 Mbps, tendo a disponibilidade, em média, sido de 0.993. Houve perda de tráfego numa situação de falha simples, nomeadamente quando falham os arcos (3,10) e (1,3). Quando,

por exemplo, falha o arco (3,10) perde-se tráfego indevidamente das *demands* do nó 10 para o nó 4 e do nó 10 para o nó 6. Julga-se que este comportamento se deve a uma falha no simulador que se deve ao facto de os caminhos de proteção serem demasiado compridos.

Nesta simulação a média do total de falhas é de 4031,1 falhas. Em 97,15% do tempo a rede não teve qualquer falha, em 2,809% do tempo teve apenas falhas simples e em 0,039% do tempo, a rede teve duas falhas em simultâneo. De notar que houve, em algumas simulações, uma ínfima percentagem de tempo (0,001%) em que houve três falhas em simultâneo, não sendo contabilizadas.

Tabela 6 – Distribuição de falhas: Restauro – caso 3 (Média / Desvio padrão)

Total de falhas	% de tempo sem falhas	% de tempo com 1 falha	% de tempo com 2 falhas	Tráfego Transportado (Mbps)	Disponibilidade da rede
4031,1 / 48,7636	97,1515 / 0,0389	2,8086 / 0,03869	0,0394 / 0,0046	4083,4713 / 0,01912	0.99303 / 0,00021

Na pior situação, em que houve falhas múltiplas, perdeu-se no máximo 796 Mbps do tráfego. Esta estratégia é menos eficiente que a anterior no que diz respeito ao tráfego perdido. No entanto, requer menos LB relativamente à anterior.

Como nota final acrescenta-se que apesar de não se ter chegado à mesma disponibilidade do caso anterior, por mau funcionamento do simulador nesta rede em particular, esta estratégia precisa de menos LB sendo por isso uma estratégia mais eficiente do que aquela que é baseada simplesmente nos caminhos mais curtos.

5. Conclusão

Nesta dissertação de mestrado foi abordado o tema da recuperação em redes, nomeadamente em redes MPLS. O estudo destas estratégias de recuperação teve em vista a sua integração em redes multicamada. Por falta de tempo e de limitações do simulador apenas foi possível testar as estratégias estudadas numa única camada de rede, sendo que a sua aplicação em redes multicamada pode ser vista como trabalho futuro.

É importante também salientar que uma das diferenças entre Proteção e Restauro é a rapidez com que uma e outra atuam em caso de falha, mas essa diferença não foi avaliada dado que o simulador utilizado não está preparado para isso.

Uma das estratégias estudadas conduz de facto a menor LB para restauro, sendo por isso uma estratégia mais eficiente embora não esteja preparada para falhas múltiplas. O estudo de estratégias de restauro para falhas múltiplas é também um assunto para trabalho futuro.

O simulador utilizado tem potencial para este tipo de estudo, embora seja necessário investir no desenvolvimento de algoritmos para integração das estratégias a seguir, o que não foi feito no âmbito desta dissertação por falta de tempo. Em particular um dos problemas detetado numa das redes de teste poderia ser resolvido corrigindo o algoritmo usado no simulador.

6. Bibliografia

[Vasseur *et al.* (2004)] Vasseur, J.-P., Pickavet M. e Demeester, P. (2004), "Network Recovery – Protection and Restoration of Optical, SONET-SDH, IP, and MPLS". Morgan Kaufmann, Elsevier.

[Awduche *et al*. (1999)] Awduche, D., Malcolm, J., Agogbua, J., O'Dell, M., e McManus, J. (1999), "Requirements for Traffic Engineering Over MPLS", RFC 2702.

[Stallings (2004)] Stallings, W. (2004), "Data and Computer Communications". 10th Editions, Pearson Education – Prentice Hall.

[Pan *et al.* (2005)] Pan, P., Swallow, G., Atlas, A. (2005), "Fast Reroute extensions to RSVP-TE for LSP tunnels", IETF RFC 4090.

[Ho *et al*. (2004)] Ho, P-H., Mouftah, H. (2004), "Shared Protection in mesh WDM networks", IEEE Communications Magazine, vol 42, p.70-76, 2004.

[Nelakuditi *et al.* (2007)] Nelakuditi, S. *et al.* (2007), "Fast local rerouting for handling transient link failures", IEEE/ACM Trans. Network, vol 15, no 2, p.359-372.

[Lu *et al.* (2018)] Lu, Z. *et al.* (2018), "Combining Electronic Layer Protection and Pre-Planned Optical Restoration for Improved and Resource Efficient Reliability", Proceedings of ICTON.

[Alicherry *et al.* (2007)] Alicherry, M., Bhatia, R. (2007), "Simple Pre-Provisioning Scheme to Enable Fast Restoration", IEEE/ACM Transactions on Networking, vol 15, no 3, p. 400-412

[Ahuja *et al.* (1993)] Ahuja, R., Magnanti, T., Orlin, J. (1993), "Network Flows: Theory, Algorithms, and Applications", Prentice-Hall, Inc.

[Mariño *et al.* (2015)] Pavón-Mariño, P., Zaragoza, J.L. (2015), "Net2plan: an open source network planning tool for bridging the gap between academia and industry", IEEE Network, vol 29, no 5, p. 90-96.

Apêndice

Desenvolvimento de algoritmos para integração no Net2plan

Para integrar código no simulador:
Há inúmeras interfaces a incluir no ficheiro *.java* nomeadamente as mais importantes:

- com.net2plan.interfaces.networkDesign.IAlgorithm: algoritmos para o network design;
- com.net2plan.interfaces.simulation.IEventGenerator: módulo que gera eventos que são utilizados pelos algoritmos;
- com.net2plan.interfaces.networkDesign.IReport: para criar os reports.

Uma descrição completa está disponível no **Library API Javadoc**.

Dentro do IDE Eclipse (usar o Java 8 ou mais recente):
É necessário incluir algumas bibliotecas. Para isso faz-se dentro do Eclipse:
Project => Properties => Java Build Path => Libraries => Add External JARs...
Importar todos os ficheiros .JAR dentro da pasta **Net2plan-0.6.6\lib\.**

Quando o algoritmo estiver pronto, pode-se exportar o ficheiro .java e passar para .JAR. O ficheiro .JAR não necessita de estar na mesma pasta Net2plan-0.6.6. O ficheiro .JAR pode ser guardado em qualquer pasta.
Posteriormente esse ficheiro .JAR, é copiado para a pasta **Net2plan-0.6.6\workspace**\.

Para obter ajuda na implementação dos algoritmos para integração no simulador:

- Net2Plan - The open-source network planner - YouTube
- Overview (net2plan-javadoc 0.6.0.1 API)
- Overview (net2plan-javadoc 0.6.0.1 API)