UNIVERSIDADE NOVA DE LISBOA
Faculdade de Ciências e Tecnologia
Departamento de Engenharia Electrotécnica

ENGENHARIA DE TRÁFEGO LEVE AO NÍVEL DO INTER-DOMÍNIO

Por:
Edgar Miguel Felício Oliveira da Silva

Dissertação apresentada na Faculdade de Ciências e Tecnologia
da Universidade Nova de Lisboa para a obtenção do grau
de Mestre em Engenharia Electrotécnica e de Computadores

Orientador(es):
Doutor Luís Filipe Lourenço Bernardo
Doutor Paulo da Costa Luis da Fonseca Pinto

LISBOA
2010

Dedico à minha
Mãe e Tios

Resumo

Nesta dissertação são propostos algoritmos para efectuar Engenharia de Tráfego (ET) na Internet sobre a Arquitectura Dinâmica de Informação Topológica (*Dynamic Topological Information Arquitecture (DTIA)*). A arquitectura DTIA fornece uma solução para o encaminhamento inter-região baseado nos números de Sistema Autónomo (SA), ao contrário do protocolo utilizado na Internet para realizar encaminhamento inter-domínio, Border Gateway Protocol (BGP), que utiliza prefixos.

São exploradas as características da arquitectura DTIA para realizar engenharia de tráfego ao nível do inter-domínio. A arquitectura DTIA oferece encaminhamento multi-caminho e a capacidade de separar o controlo de tráfego e os avisos de rotas. É proposta uma abordagem muito leve para realizar engenharia de tráfego usando as características da arquitectura DTIA, a informação local disponível em cada SA e uma sinalização mínima que consiste num pacote de controlo simples.

São propostos três algoritmos, dois deles com aplicação de engenharia de tráfego salto-a-salto até à(s) origem(ns) e o terceiro percorrendo o caminho a partir da origem e actuando nos SAs considerados capazes de alterar o tráfego.

A implementação dos algoritmos propostos foi feita no simulador de redes ns-2 (*The Network Simulator 2*). Os resultados mostraram que os algoritmos melhoram o equilíbrio das ligações na rede, alguns deles com recurso a uma sinalização reduzida. Além disso, os algoritmos apresentados nesta dissertação reduzem o número de ligações congestionadas na rede.

Abstract

In this thesis, algorithms are proposed to perform traffic engineering in the Internet using the Dynamic Topological Information Arquitecture (DTIA). DTIA architecture provides a solution for inter-region routing based on Autonomous System (AS) numbers, unlike the Internet's inter-domain routing protocol Border Gateway Protocol (BGP), which uses prefixes.

In this thesis we explore DTIA's characteristics to perform inter-domain traffic engineering. DTIA provides multipath routing and the ability to separate traffic control from route dissemination. A very lightweight approach was proposed in the design of the TE protocols that uses only DTIA characteristics, locally available information to each AS and minimal signalling consisting in a simple control packet.

Three algorithms are proposed, two of them with application of traffic engineering hop-by-hop until the source(s) and the third covering the path from the source and acting within the ASs considered capable of disrupting traffic.

An implementation of the proposed algorithms was made in The Network Simulator 2 (ns-2). Results showed that the algorithms improve the traffic distribution, some of them using a reduced signalling. Furthermore, the algorithms presented in this thesis reduce the number of congested links in the network.

Acrónimos

BGP Border Gateway Protocol

C2F Cliente-Fornecedor

CAIDA Cooperative Association for Internet Data Analysis

CCE Caminhos com Comutação de Etiquetas

DTIA Dynamic Topological Information Arquitecture

eBGP External Border Gateway Protocol

EFSA Encaminhador de Fronteira do Sistema Autónomo

ET Engenharia de Tráfego

F2C Fornecedor-Cliente

F2F Fornecedor-Fornecedor

F2Fbkp Fornecedor-Fornecedor que permite *backup*

F2Fptt Fornecedor-Fornecedor que permite tráfego de trânsito

GMPLS Generalized Multiprotocol Label Switching

iBGP Internal Border Gateway Protocol

IGP Interior Gateway Protocol

IS-IS Intermediate System - Intermediate System

MPLS Multiprotocol Label Switching

MT Matriz de Tráfego

NAT Network Address Translation

ns-2 The Network Simulator 2

OSPF Open Shortest Path First

PCE Path Computation Element

PRI Provedor de Rede Internet

PSIs Provedores de Serviços de Internet

QdS Qualidade de Serviço

RSVP-TE Resource Reservation Protocol - Traffic Engineering

SA Sistema Autónomo

SVCL Vector de Caminhos Simulado Localmente

TCP Transport Control Protocol

Conteúdo

Lista de Tabelas

Lista de Figuras

Capítulo 1

Introdução

A Internet tem sentido um crescimento explosivo desde o seu nascimento. E está dividida em milhares de Sistemas Autónomos (SAs), e cada um consiste em redes de encaminhadores administrados por uma única organização. Por motivos de desempenho ou por razões de custo, os SAs muitas vezes necessitam de controlar os fluxos do tráfego de entrada inter-domínio [QB05].

Controlar o tráfego de entrada é uma tarefa difícil, visto que muitas vezes implica influenciar os SAs num caminho. O Border Gateway Protocol (BGP) é actualmente utilizado na Internet como uma solução para o encaminhamento inter-domínio. Provedores de Serviços de Internet (PSIs) também o usam para outros fins, tais como o equilíbrio do tráfego nas ligações e reduzir o custo do uso dessas ligações. Mas as técnicas actuais do BGP para esses fins são primitivas, e além disso, o seu efeito é muitas vezes difícil de prever. Além disso, a estrutura actual da Internet é massivamente *multi-homed*, onde os clientes estão ligados a mais do que um provedor, mas o BGP não consegue tirar pleno proveito da multiplicidade de ligações para comunicar com um determinado destino.

Nesta dissertação, são discutidos os problemas da aplicação da engenharia de tráfego na Internet actual com principal foco no inter-domínio. Tem sido desenvolvida uma arquitectura para encaminhamento inter-domínio chamada Dynamic Topological Information Arquitecture (DTIA) [ABP08, ABP09, AGA$^+$09], que apresenta um encaminhamento multi-caminho sem ciclos. Nesta arquitectura é possível um maior controlo sobre o tráfego, que pode ser alcançado sem alterar as configurações de encaminhamento e avisos de rotas. É assim uma nova maneira de abordar o problema da Engenharia de Tráfego (ET)

inter-domínio: definir a correcta distribuição do tráfego para realizar ET em vez de definir atributos para as rotas que levariam à distribuição do tráfego.

1.1 Hipótese

A partir das características da arquitectura DTIA para realizar engenharia de tráfego ao nível do inter-domínio, as seguintes hipóteses são formuladas: Será possível implementar um algoritmo capaz de reduzir o congestionamento na rede e levar a um maior equilíbrio do tráfego nas ligações directas de um SA que execute engenharia de tráfego com um tráfego de sinalização mínimo? Este algoritmo deve manter as vantagens do multi-caminho e a escalabilidade da rede.

1.2 Objectivos

O objectivo principal desta dissertação é aferir a viabilidade da implementação de um algoritmo de engenharia de tráfego capaz de estar em conformidade com a hipótese da secção 1.1.

As principais contribuições desta dissertação são a definição de algoritmos para a camada de engenharia de tráfego da arquitectura DTIA, e a implementação completa destes no simulador de redes 2 (*The Network Simulator 2 (ns-2)*).

1.3 Estrutura da Dissertação

Esta dissertação está estruturada em seis capítulos, cada um é enumerado nos parágrafos seguintes.

O capítulo 2 apresenta um conjunto de modelos para classificar a engenharia de tráfego. No final discutem-se os objectivos da engenharia de tráfego e como podem ser atingidos.

O capítulo 3 apresenta uma súmula da arquitectura DTIA - um protocolo de encaminhamento multi-região e acrescenta a camada de engenharia de tráfego e respectivos pressupostos.

O capítulo 4 apresenta os algoritmos de engenharia de tráfego desta dissertação e apresenta a implementação destes no simulador ns-2 através do auxílio de fluxogramas.

O capítulo 5 faz uma análise do desempenho da implementação no ns-2 dos algoritmos apresentados. Com base nos resultados, este capítulo tenta correlacionar os dados com o capítulo 4.

O capítulo 6 faz uma análise global desta dissertação, com base na hipótese. No final, este capítulo enumera alguns tópicos para trabalho futuro baseados nesta tese.

Capítulo 2

Trabalho Relacionado

2.1 Engenharia de Tráfego na Internet

A rede Internet é constituída por um número elevado de redes administradas por entidades independentes, designadas de Sistemas Autónomos (SAs). É necessário definir um processo que mantenha o desempenho destes sistemas, no que respeita ao atraso na entrega de pacotes, fiabilidade e gestão da congestão. O processo é conhecido por Engenharia de Tráfego (ET).

A engenharia de tráfego tem como função principal a gestão de recursos. Entre os recursos incluem-se a largura de banda, o espaço na memória (*buffer*) e os recursos computacionais. A optimização destes recursos é realizada tipicamente através do controlo do encaminhamento. Este é realizado pelo controlo da distribuição do tráfego pelos SAs da rede de uma forma mais eficaz.

Um dos maiores desafios da engenharia de tráfego na Internet, é a construção de mecanismos de controlo automático do fluxo de tráfego que não tenham um custo elevado ao nível das capacidades da rede e que consigam adaptar-se rapidamente às alterações significativas do estado da rede, mantendo a sua estabilidade.

Na secção seguinte, define-se uma taxonomia para a engenharia de tráfego representada na figura 2.1 [WHPH08]. Os vários modelos são classificados em:

- Tipo de tráfego, que pode ser ponto-a-ponto ou ponto-multi-ponto (*Unicast/Multicast*).

- Alcance da optimização de tráfego, que pode ser Inter-Domínio ou Intra-Domínio.
- Escala de tempo da manipulação do tráfego, com os valores desligado (*offline*) ou em linha (*online*).

Na secção 2.2 são apresentados dois métodos de controlo de tráfego: um baseado no protocolo de encaminhamento Border Gateway Protocol (BGP), e outro baseado no protocolo de comutação, Multiprotocol Label Switching (MPLS).

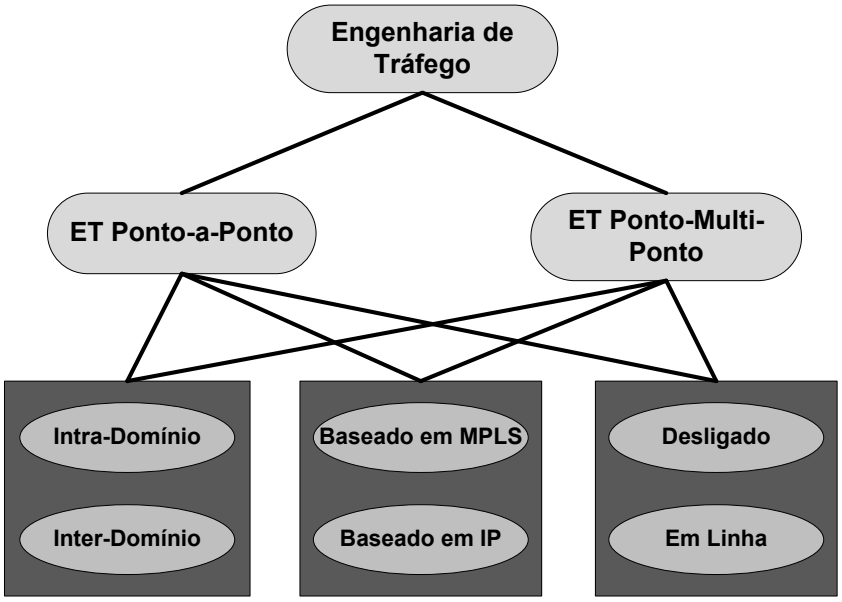

Figura 2.1: Classificação Hierárquica da Engenharia de Tráfego na Internet.

2.1.1 Tipo de tráfego

O tráfego existente na Internet é heterogéneo, incluindo tanto tráfego ponto-a-ponto como ponto-multi-ponto. Estes dois tipos de tráfego podem ser injectados simultaneamente na mesma rede, o que leva a que a engenharia de tráfego tenha de lidar com ambos em paralelo. O problema da distribuição óptima de tráfego ponto-multi-ponto é mais complicado comparando com o tráfego ponto-a-ponto, mas existe um objectivo comum: minimizar a

largura de banda consumida.

Convencionalmente é usado um protocolo de encaminhamento baseado no caminho mais curto, que nem sempre é a melhor solução a aplicar no caso de tráfego ponto-multi-ponto. Na literatura, a conservação da largura de banda no encaminhamento ponto-multi-ponto é indicado como o problema da árvore de Steiner [KLS03]. É, no entanto, importante referir que a tarefa da engenharia de tráfego ponto-multi-ponto não é necessariamente idêntica ao problema da clássica árvore de Steiner. Além da conservação de largura de banda existem outros objectivos na ET, tais como o equilíbrio do tráfego e a maximização da utilização das ligações.

Normalmente o objectivo varia de acordo com o agente em causa. Por exemplo, os utilizadores querem maximizar a utilização e o gestor de rede quer conservar a largura de banda. O objectivo global pode passar por uma ponderação de ambos.

2.1.2 Tráfego Inter-Domínio e Intra-Domínio

Em termos do alcance da optimização de tráfego, a engenharia de tráfego pode ser classificada em: Inter-Domínio, onde o objectivo é a optimização do tráfego entre os vários domínios, SAs; e em Intra-Domínio, onde a ET tem como tarefa a optimização do tráfego entre os encaminhadores de um só SA.

Com tráfego Inter-Domínio, o protocolo de encaminhamento usado é o BGP. SAs vizinhos trocam os seus caminhos, cada um contendo um vector com a sequência de SAs da origem até ao destino (i.e. o melhor caminho, não necessariamente o mais curto). Por razões de escalabilidade, o protocolo só tem conhecimento das ligações entre domínios. Não tem qualquer informação sobre as ligações dentro de cada um deles.

Com tráfego Intra-Domínio podem ser aplicados outros protocolos, que estão cientes das exigências da engenharia de tráfego e incluem características para o suportar. Existem várias alternativas que podem ser aplicadas num só domínio, cada uma tendo as suas vantagens e desvantagens (por exemplo: Vector de Distâncias (*Distance Vector (DV)*), Estado da Ligação (*Link State*)). Ambos os tipos de protocolo baseiam-se na descoberta do melhor caminho de acordo com uma determinada métrica de custo.

Por outro lado, quando um operador escolhe não utilizar o BGP tem de considerar as tarefas de tradução entre protocolos. Por exemplo, se for escolhido o protocolo de comutação MPLS para o Intra-Domínio, os encaminhadores de fronteira vão precisar de traduzir a informação de encaminhamento entre os protocolos MPLS e BGP. Consequentemente, os administradores escolhem frequentemente o BGP para o Intra-Domínio, para não terem de lidar com essa tradução.

Com tráfego Intra-Domínio, o administrador tem os mesmos problemas que no caso de Inter-Domínio. Mas tem todos os encaminhadores afectados pelas mesmas condições, o que faz com que o planeamento e distribuição de regras para a engenharia de tráfego sejam mais fáceis. O estudo mais pormenorizado da ET Inter-Domínio, é deixado para a secção 2.3.

2.1.3 Manipulação do tráfego, em linha e desligado

Relativamente à manipulação do tráfego e realização de operações, a engenharia de tráfego pode ser classificada como desligada (*offline*) ou em linha (*online*). Para definir estes conceitos é necessário introduzir o conceito de Matriz de Tráfego (MT)), originalmente associado à engenharia de tráfego Intra-Domínio, onde os pontos de entrada e saída do tráfego são fixos e constituem os índices da matriz. As principais diferenças entre ambos é a existência ou não da MT e quando é feita a decisão da manipulação do tráfego.

Tendo a MT para a topologia da rede, um Provedor de Rede Internet (PRI) (*Internet Network Provider*) pode realizar a engenharia de tráfego desligada. Um ponto importante é o período que define a duração média entre dois ciclos de ET consecutivos, designado de Ciclo de Provisão de Recursos (*Resource Provisioning Cycle*) [TAP+01]. Normalmente, um Ciclo de Provisão de Recursos para ET desligada é de uma semana ou de um mês.

O ponto mais fraco da engenharia de tráfego desligada é a falta de adaptabilidade na manipulação de tráfego consoante o tráfego ou a dinâmica da rede, não lidando correctamente com picos de tráfego ou falhas na rede sem intervenção humana.

Perante os problemas de adaptação à dinâmica do tráfego e das redes, é necessária a

existência de uma solução que possa funcionar em "tempo real" e independentemente de administradores. A solução é conhecida como engenharia de tráfego em linha.

Em alguns casos não é possível para um Fornecedor de Rede de Internet prever totalmente a MT. Assim sendo, é necessário recorrer à engenharia de tráfego em linha, que não precisa de nenhum conhecimento futuro da distribuição do tráfego. De forma a responder rapidamente às flutuações do tráfego, a ET em linha é executada em ciclos de horas, ou mesmo minutos.

Um interesse prático para os PRIs é que ao aplicar a engenharia de tráfego em linha a rede está a convergir sem intervenção humana. Geralmente, o tráfego é tratado de modo a ser possível evitar situações de congestão. Para este fim, a ET em linha certifica-se de que o tráfego está equilibrado o mais uniformemente possível na rede, para que qualquer tráfego novo seja facilmente acomodado. Contudo, podem existir fluxos que competem com outros e que causam instabilidade no tráfego, ou falhas nos serviços. Além disso, devido à incerteza do tráfego existente, a engenharia de tráfego em linha pode vir a demonstrar dificuldades em tratar futuras alterações de tráfego, tendo em conta o estado da rede. Para ultrapassar estas questões, um modo promissor é considerar ambos os tipos de ET, desligada e em linha, de modo a complementarem-se.

2.2 Controlo de Tráfego

Do ponto de vista da aplicação de mecanismos de controlo de tráfego, a engenharia de tráfego pode ser tipicamente classificada no multi-protocolo de troca de etiquetas (MPLS) e no protocolo por IP. Tipicamente é usado o BGP para o caso Inter-Domínio e para o Intra-Domínio os protocolos de estado de ligação (Open Shortest Path First (OSPF) e Intermediate System - Intermediate System (IS-IS)) ou também o BGP.

2.2.1 Multiprotocol Label Switching (MPLS)

O conceito de engenharia de tráfego foi inicialmente introduzido no protocolo (MPLS) [Awd99]. Este protocolo cria de uma forma inteligente, etiquetas dedicadas para os caminhos, conhecidos como Caminhos com Comutação de Etiquetas (CCE) (*Label Switched*

Path), para encaminhar pacotes IP encapsulados. Assim, a ET utilizando MPLS fornece um paradigma eficiente para a optimização de tráfego. A sua principal vantagem é a capacidade de encaminhamento explícito e de divisão arbitrária de tráfego, adaptada à optimização da distribuição de tráfego.

O MPLS é uma tecnologia poderosa para a engenharia de tráfego na Internet, porque permite que o tráfego seja enviado numa rota explícita arbitrária. Normalmente, os fluxos individuais são reunidos em feixes agregados de tráfego chamados de Classes de Transmissão Equivalente (*Forwarding Equivalence Classes*), que por sua vez são carregados em CCE entre o encaminhador de origem (*ingress router*) e de destino (*egress router*).

No entanto, quando os feixes agregados de tráfego são entregues por CCEs dedicados, a escalabilidade e a robustez podem-se tornar num problema. Primeiro, o número de CCEs pode tornar-se muito elevado em redes de grande dimensão. Por outro lado, são necessários mecanismos para a protecção de caminhos, para que seja possível o encaminhamento automático de tráfego por outros caminhos no caso da falha de alguma ligação. Isto pode tornar a criação de caminhos CCE muito complexa, tornando necessários mecanismos como o *Path Computation Element (PCE).*

O PCE é uma entidade capaz de calcular uma rota ou caminho da rede com base no grafo da rede, e aplicando regras durante o cálculo. Torna-se assim muito importante para a engenharia de tráfego em protocolos como MPLS e MPLS Generalizado (*Generalized Multiprotocol Label Switching (GMPLS)*) [FVA06]. A computação de caminhos em grandes redes, multi-domínios, multi-região, ou multi-camadas é complexa e podem ser necessários componentes especiais para a sua computação e cooperação entre os diferentes domínios. Por exemplo, um PCE seria capaz de calcular um caminho a partir dos CCEs recorrendo à base de dados da ET (contém a topologia e informação sobre os recursos existentes no domínio), considerando a largura de banda e outras restrições solicitadas no pedido para se executar ET.

Para efectuar sinalização, o MPLS e GMPLS podem recorrer por exemplo ao protocolo Resource Reservation Protocol - Traffic Engineering (RSVP-TE). Este protocolo suporta a instanciação explícita de CCEs, com ou sem reservas de recursos. Suporta também o redireccionamento suave de CCEs e a prevenção e detecção de *loops* [ABG+01].

No caso Intra-Domínio, o encaminhamento convencional baseado no protocolo, caminho mais curto (por exemplo, *Open Shortest Path First (OSPF)*) é substituído pelo protocolo de comutação MPLS utilizando encaminhamento explicito por túneis.

2.2.2 Border Gateway Protocol (BGP)

O BGP nasceu originalmente como um simples protocolo de vector de caminhos (*path-vector protocol*), mas evoluiu com a necessidade que os PRIs tiveram para controlar a selecção de rotas (para onde enviar os pacotes) e da propagação (como transferir as rotas). O BGP foi sendo modificado ao longo do tempo tendo-se acrescentado um conjunto de mecanismos que suportam políticas, aumentando assim a sua flexibilidade, e também a complexidade.

Habitualmente os Provedores de Serviços Internet ligam-se a múltiplos locais para reduzir o atraso da entrega de pacotes e melhorar a fiabilidade, aumentando o número disponível de rotas. Outro objectivo para os Provedores de Serviços de Internet (PSIs) é organizar o tráfego através de modificações das suas preferências, para ir de encontro ou maximizar certos critérios de desempenho (i.e. atingir qualidade e disponibilidade desejadas).

No BGP as rotas são representadas por prefixos de IP. Estes são trocados entre encaminhadores vizinhos, *peers*, usando mensagens do tipo *UPDATE*. Este mecanismo é geralmente conhecido como anúncio de rotas. O encaminhador que recebe a informação pode, ou não, introduzir essa informação na sua tabela de encaminhamento, dependendo do tamanho do caminho, da política da rede local e de outras regras.

Dois encaminhadores BGP precisam de estabelecer uma sessão *Transport Control Protocol (TCP)* para troca de rotas. Existem dois tipos de sessões: *Internal Border Gateway Protocol (iBGP)* e *External Border Gateway Protocol (eBGP)*. O primeiro é usado para aprender as rotas dentro de um SA, enquanto que o segundo é usado para aprender rotas entre domínios. As rotas trocadas nestas sessões são estruturadas com três parâmetros: o prefixo de IP do destino, o caminho e os seus atributos. O caminho descreve uma lista ordenada de SAs a percorrer até ao destino, enquanto os seus atributos são utilizados para decisões de encaminhamento.

Regras para a escolha de um caminho

De modo a discutir possíveis soluções para a engenharia de tráfego, é preciso compreender os critérios que o protocolo BGP segue para escolher uma rota [RL06]. A tabela 2.1 mostra a ordem de preferência. O primeiro critério a ser testado nas decisões de encaminhamento é a preferência local (*LOCAL_PREF*). Atribuindo um valor elevado ao anúncio de uma rota leva a que esta seja escolhida. Consequentemente o administrador do domínio tem uma influência directa em como é distribuído o tráfego. Como segundo parâmetro é considerado o caminho mais curto. Este atributo contém todos os SAs que compõem o caminho até ao prefixo IP anunciado. Quantos menos SAs houver, melhor o caminho. A ideia é reduzir o atraso total na transferência de um pacote.

Tabela 2.1: Ordem de preferência para a escolha de um caminho no BGP.

n°	Critério
1	Maior preferência local (*LOCAL_PREF*)
2	Caminho mais curto
3	Menor número no atributo de *ORIGIN*
4	Menor número no atributo *MULTI_EXIT_DISC*
5	Preferir rotas eBGP em relação a iBGP
6	Menor métrica para o próximo salto

Ultrapassados os dois primeiros critérios, se ainda não houver nenhuma decisão sobre qual o caminho a seguir, são considerados os critérios do menor valor no atributo de *ORIGIN* e do menor valor no atributo *MULTI_EXIT_DISC*. O atributo *ORIGIN* indica como uma rota é conhecida, tomando um de três valores, *eBGP*, *iBGP* ou *incompleto*. Estes valores são classificados do seguinte modo: *incompleto* < *eBGP* < *iBGP*. Ou seja, um caminho com o atributo *ORIGIN* com o valor *iBGP* é preferido a um caminho com o atributo *ORIGIN* com o valor *eBGP*. O atributo *MULTI_EXIT_DISC* é aplicado em cenários de multi-caminho. Destina-se a diferenciar os vários pontos de saída ou entrada para um SA vizinho. Os últimos dois critérios para o processo de decisão existem para casos excepcionais, onde todos os valores restantes são os mesmos para diferentes anúncios de caminhos.

Isto significa que a decisão final sobre se uma rota anunciada é aceite ou não, é deixada à responsabilidade do administrador local na maioria dos casos.

Relação entre encaminhadores e SAs

Um passo importante para a compreensão das necessidades da engenharia de tráfego é ter uma caracterização do tráfego na Internet. Um SA aplica políticas locais para seleccionar a melhor rota para cada destino e decide se divulga essa rota aos seus vizinhos, sem divulgar essas políticas ou a topologia interna dos SAs. Na prática, as políticas do BGP reflectem as relações comerciais entre os SAs vizinhos.

Dois modelos principais emergiram para classificar a relação entre dois SAs [SARK02]. O primeiro é uma relação cliente-fornecedor (*customer-provider*) e o segundo uma relação fornecedor-fornecedor (*peer-to-peer*). O cliente assina contratos com um ou mais fornecedores de serviço para ter acesso à Internet. O fornecedor aceita encaminhar todo o tráfego anunciado por esse SA e é pago por esse serviço. Geralmente, o cliente é a parte pequena no negócio em relação ao fornecedor.

Numa relação fornecedor-fornecedor (*peer-to-peer*), ambas as partes concordam em distribuir o tráfego provenientes de outras origens, compartilhando as suas taxas de tráfego. Entretanto os detalhes do contrato não estão na maioria dos casos disponíveis publicamente. Este tipo de relação é comum entre dois domínios geograficamente próximos ou com o mesmo poder no mercado.

2.3 Engenharia de Tráfego Inter-Domínio

A engenharia de tráfego inter-domínio é realizada tendo em consideração a informação de encaminhamento anunciada pelos outros domínios.

2.3.1 Visão Global

Uma prática usual é classificar os domínios numa de duas categorias: domínio de trânsito (*Transit Domain*) e domínio terminal (*Stub Domain*). Um domínio de trânsito oferece serviços para a circulação de tráfego (i.e. entrega de tráfego inter-domínio pela Internet). Por outro lado, os domínios terminais são considerados os domínios de último nível na hierarquia dos SAs.

Em geral, os dois tipos de domínio têm objectivos diferentes a nível da engenharia de tráfego inter-domínio. Os incentivos para os domínios de trânsito executarem ET inter-domínio são normalmente os de optimizar os recursos da rede e de maximizar as suas receitas financeiras. A principal preocupação dos domínios terminais (domínios que não são PRIs de nenhum outro SA) em relação à ET inter-domínio é de minimizar as despesas monetárias da subscrição de serviços de Internet dos seus PRIs.

A engenharia de tráfego inter-domínio pode-se classificar como engenharia de tráfego de saída (*outbound*) e engenharia de tráfego de entrada (*inbound*), dependendo de se focar no controlo respectivamente do tráfego de saída ou de entrada do SA. Nas secções seguintes são focados estes dois tipos de engenharia de tráfego.

2.3.2 Tráfego de saída

Actualmente são conhecidos alguns mecanismos de engenharia de tráfego de saída, destacando-se na tabela 2.2 os mais representativos.

Tabela 2.2: Mecanismos para a Engenharia de Tráfego de saída Inter-Domínio.

Mecanismo	Descrição	Técnicas de Implementação	Aplicável
Preferência local do BGP (Local_pref)	Para escolher directamente o encaminhador de saída, alterando para um valor elevado a preferência local do BGP	BGP	Domínio terminal/trânsito
Encaminhamento *hot potato*	Para escolher o encaminhador de saída com o menor valor do IGP[1]	BGP/IGP	Normalmente domínios de trânsito
Encaminhamento explícito - MPLS	Para escolher o encaminhador de saída estabelecendo caminhos explícitos pelos domínios	RSVP-TE, BGP/IGP-TE, PCE	Domínio terminal/trânsito

Preferência local(*Local_pref*)

O critério, 'preferência local', tem a prioridade mais elevada no processo de selecção de rotas no BGP. O valor atribuído a este critério indica a preferência que um encaminhador de fronteira tem em relação a outros candidatos para o melhor encaminhador de saída. Na figura 2.2 está representado um pequeno exemplo, onde se assume que o valor de preferência local para o prefixo 20.20.20.0/24 no encaminhador de fronteira 10.10.10.1 é maior do que para o 10.10.10.2. Nestas condições, o tráfego com destino SA 2 é encaminhado pelo encaminhador de saída 10.10.10.1.

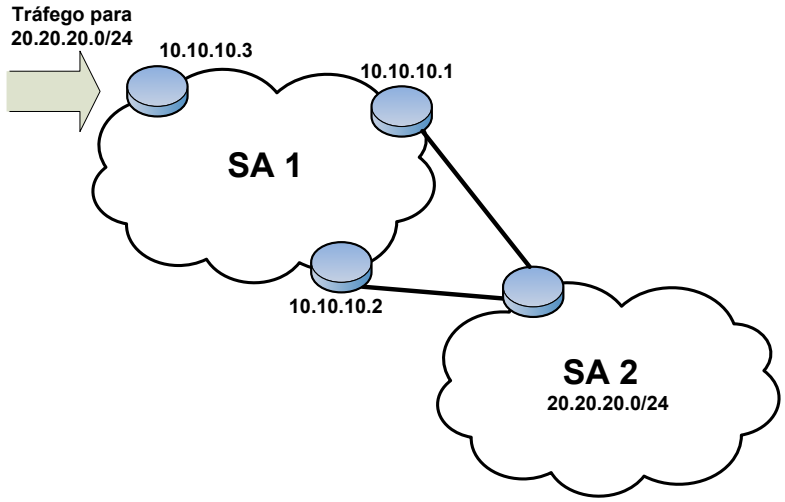

Figura 2.2: Exemplo de uma rede com dois SAs.

Encaminhamento *hot potato*

Neste caso, se existirem vários caminhos com valor igual no que toca aos critérios de selecção do BGP, tabela 2.1, é escolhido o caminho com a menor distância entre o ponto de entrada até ao ponto de saída a nível de *Interior Gateway Protocol (IGP)*. Este cenário é conhecido como *hot potato* ou *early-exit routing*, e é usualmente usado por grandes PRIs. Tem como objectivo enviar tráfego para os domínios abaixo (*downstream*) através

[1] *Interior Gateway Protocol* é um protocolo de encaminhamento para troca de informações de encaminhamento dentro de um sistema autónomo.

do núcleo da rede o mais rápido possível.

Ajustando os pesos das ligações IGP "à sua vontade", um PRI pode influenciar a selecção do encaminhador de saída dentro do domínio local. Retome-se o exemplo da figura 2.2, mas assumindo agora que todos os critérios de selecção de caminhos são iguais tanto para o 10.10.10.1 e 10.10.10.2. Se o valor do IGP do caminho mais curto A (entre 10.10.10.3 e 10.10.10.1) for menor que o do caminho mais curto C (entre 10.10.10.3 e 10.10.10.2), o ponto de saída escolhido é o 10.10.10.1 de acordo com o encaminhamento *hot potato*.

Encaminhamento explícito - MPLS

MPLS inter-domínio permite que um domínio force o tráfego a ser entregue através de caminhos explícitos até ao domínio de destino. Assim, os domínios podem estabelecer trajectos explícitos para os pontos de saída desejados. Actualmente, vários mecanismos que suportam MPLS inter-domínio foram propostos e implementados, tais como *Path Computation Element (PCE)* [FVA06].

2.3.3 Tráfego de entrada

Nesta secção dá-se uma noção dos mecanismos disponíveis para engenharia de tráfego de entrada. Há várias hipóteses conhecidas para implementar engenharia de tráfego no tráfego de entrada (e.g., *AS path prepending*). Na tabela 2.3 são enumerados vários mecanismos que podem ser utilizados para realizar ET de entrada.

Tabela 2.3: Mecanismos para a Engenharia de Tráfego de entrada Inter-Domínio.

Mecanismo	Descrição	Técnicas de Implementação	Aplicável
Avisos selectivos	Anunciar uma rota unicamente pelos pontos de entrada por onde se espera receber o tráfego	BGP	Domínio terminal/trânsito
Avisos mais específicos	Anunciar rotas com prefixos mais específicos para reduzir a quantidade de anúncios de rotas	BGP	Domínio terminal/trânsito
AS path prepending	Aumentar o número de saltos do caminho para reduzir a atractividade de uma rota	BGP	Domínio terminal/trânsito
Atribuição do valor *Multi-exit Discriminator* (MED)	Anunciar rotas preferidas com um valor MED menor	BGP	Domínio terminal/trânsito
Atributo de Comunidade	Sugerir a outros domínios como manipular os anúncios de rotas	BGP	Domínio terminal/trânsito
Tradução de endereços NAT	Modificar os cabeçalhos dos pacotes atribuindo o ponto de entrada desejado como origem dos pacotes	NAT	Normalmente Domínio terminal
BGP *overlay*	Comunicação directa entre dois domínios ignorando o BGP	Especificado pelo utilizador	Domínio terminal/trânsito

Avisos selectivos

Nesta abordagem, as rotas para um prefixo de destino só são divulgadas através de um conjunto escolhido de ligações de entrada. Vejamos a figura 2.3 como exemplo. Se SA400 quisesse receber o tráfego de SA300 pelo Encaminhador de Fronteira do Sistema Autónomo (EFSA) 40.40.0.1 com destino o SA401, este escolheria não divulgar rotas para o SA401 pelo EFSA 40.40.0.2. Contudo, a lacuna desta abordagem é que se o ponto de entrada escolhido falhar, não existe nenhuma rota alternativa.

Avisos mais específicos

Neste método, se existirem múltiplas rotas para o mesmo destino, aquele com o maior prefixo será seleccionado. Retomando a figura 2.3 e assumindo que o SA400 avisa o SA300 que o prefixo de destino 40.40.0.0/16 está acessível pelo EFSA 40.40.0.1, e que o seu subprefixo 40.40.40.0/24 está acessível pelo EFSA 40.40.0.2. O resultado, é que o tráfego com destino à gama do SA401 não usará o 40.40.0.1, porque o outro encaminhador de entrada

tem uma rota com um prefixo mais específico.

Comparando com o anúncio selectivo, este tipo de selecção de encaminhadores de entrada é mais robusta em caso de falha de ligação. Também é de realçar, que o anúncio de prefixos específicos causa problemas de escalabilidade [QUP+03], devido ao aumento das tabelas de encaminhamento do BGP, sendo esta a razão principal pelo qual não é comum a sua consideração para engenharia de tráfego inter-domínio.

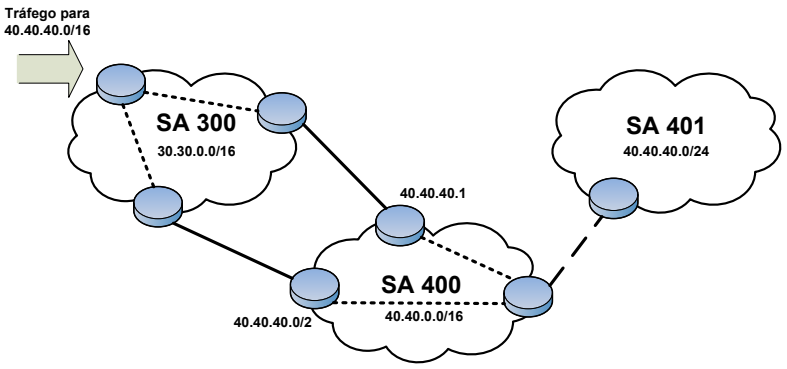

Figura 2.3: Exemplo de Engenharia de Tráfego de entrada.

AS path prepending

Um anúncio de rota é considerado menos atractivo pelos SAs, quando são adicionadas várias instâncias do mesmo SA ao atributo *AS path* aumentando assim o tamanho do caminho. Na figura 2.3, se o SA400 pretender receber o tráfego do SA300 com destino a SA401 pelo EFSA 40.40.0.1, este pode acrescentar várias vezes o seu número de SA no anúncio enviado pelo EFSA 40.40.0.2, fazendo deste modo com que o critério *AS path* neste EFSA seja mais favorável que no caso do EFSA 40.40.0.1. Este processo só é possível, se o AS300 não aplicar o primeiro critério de selecção de caminhos, preferência local (*Local_pref*), para escolher a rota que prefere utilizar.

Atribuição do valor *Multi-exit Discriminator (MED)*

Este mecanismo só é aplicável se dois SA adjacentes tiverem duas ou mais ligações directas entre eles, e ambos acordarem em implementar o discriminador multi-saída *Multi-exit*

Discriminator (MED). Nestas circunstâncias um domínio pode escolher o seu encaminhador de entrada atribuindo ao MED um valor baixo. Considerando o exemplo da figura 2.3, se o SA400 quiser receber o tráfego do SA300 pelo EFSA 30.30.30.1, pode anunciar uma rota BGP com um valor baixo no MED por este encaminhador.

O pré-requisito para a utilização do critério MED para a selecção do encaminhador de entrada, é que todos os atributos com maior prioridade na selecção de rotas BGP têm de ter valores iguais para as duas rotas (e.g. o SA400 deve definir o critério de preferência local e o comprimento do *AS path* internamente, para os dois encaminhadores de fronteira).

Atributo de Comunidade

Nesta abordagem, uma rota pode ser anunciada com o atributo de comunidade, indicando assim aos domínios acima na hierarquia, como estes podem manipular essa rota com determinadas acções. Por exemplo, o critério do caminho mais curto, pode ser incluído no atributo de comunidade de modo a que outros domínios executem o critério do caminho mais curto antes de enviar anúncios de rotas a domínios acima na hierarquia. [QTUB04]

Tradução de endereços NAT

Network Address Translation (NAT) é um método pelo qual os endereços IP são mapeados de um grupo para outro, de forma transparente para os utilizadores finais [SE01]. Assim, é alcançado um mecanismo que interliga endereços privados a um endereço externo globalmente exclusivo, sendo possível então o acesso a redes externas.

BGP *overlay*

Uma arquitectura de controlo de políticas definida sobre uma rede virtual (*Overlay Policy Control Architecture (OPCA)*) foi proposta para separar as políticas do encaminhamento, de modo a que um canal mais rápido possa ser utilizado para lidar com as alterações nas políticas de encaminhamento [AnCK03]. *Overlay Policy Control Architecture (OPCA)* consiste em vários componentes principais, incluindo agentes de políticas e bases de da-

dos, infraestruturas de medição, propagação de mensagens, etc. Os objectivos de *Overlay Policy Control Architecture (OPCA)* são os de resolver o problema de convergência do BGP, melhorando o tempo de resposta a falhas nos encaminhadores e o de equilibrar a carga do tráfego de entrada nos domínios *multihomed* (ligados a vários PRIs).

2.3.4 Cooperação

Tendo em conta a situação actual, em que os provedores de rede Internet podem ter políticas de encaminhamento diferentes ou mesmos contraditórias, não é surpreendente que existam problemas na optimização do desempenho da rede e na entrega de tráfego. Para resolver este problema, foi proposta a cooperação entre domínios adjacentes.

Como a maioria dos domínios da Internet são entidades reguladas individualmente e por vezes em concorrência uns com os outros, então é natural que estes executem engenharia de tráfego inter-domínio individualmente, sem considerar os seus vizinhos. Além disso, pesquisas recentes demonstram que quando domínios adjacentes executam ET inter-domínio de forma egoísta, não só o desempenho global da rede não é optimizado, como as estratégias de cada domínio para a ET inter-domínio pode afectar negativamente outros domínios [MWA04].

Ao existirem estratégias contraditórias, cada domínio irá agir sempre que ocorrerem alterações nas decisões de engenharia de tráfego dos domínios adjacentes, conduzindo assim a possíveis instabilidades no encaminhamento. Um método conveniente para alcançar um bom desempenho global da engenharia de tráfego, é incentivar os PRIs a negociarem entre eles, a fim de obter uma solução que beneficie todos. Esta solução é conhecida como ET baseada em cooperação [MWA05].

A engenharia de tráfego baseada em cooperação tem como princípio a negociação entre dois domínios adjacentes de modo a alcançarem um acordo em como encaminhar o tráfego entre as suas redes. Esse acordo, pode ser determinado através de métodos de optimização, tendo em consideração as topologias, objectivos da ET e as matrizes de tráfego dos dois domínios.

Alguns algoritmos foram propostos para a engenharia de tráfego inter-domínio por coo-

peração. Por exemplo, os autores de [QB05] referem uma solução utilizando túneis por IP para estabelecer caminhos explícitos entre os domínios de origem e destino, estabelecendo os pontos de entrada que recebem o tráfego. Esta abordagem só é considerada válida num cenário em que todos os domínios da rede cooperam. Já os autores de [LR07] propõem um algoritmo para a escolha de uma rota óptima, entre um grupo de domínios terminais cooperativos *multihomed*, para alcançar uma solução de ET global que evite oscilações provocadas por algum conflito entre os objectivos da ET dos domínios.

2.4 Sumário

A tabela 2.4 resume os prós e contras dos métodos discutidos anteriormente. Com base nos estudos realizados recentemente e as tendências dos métodos estudados, é possível extrair algumas ideias sobre o que deveria ser o futuro da engenharia de tráfego.

Primeiro é essencial definir os objectivos da engenharia de tráfego:

- Controlo e optimização do encaminhamento.
- Mecanismos de controlo automático.
- Manter a estabilidade da rede.
- Resposta eficaz a picos de tráfego.
- Boa resposta a falhas na rede.

A fim de atingir estes objectivos indica-se os seguintes processos:

- Gestão de recursos:
  - Largura de banda.
  - Espaço na memória (*buffer*).
  - Recursos computacionais.
- Condicionamento de tráfego.
- Gestão de filas.

- Programação.

Esta dissertação pretende encontrar soluções para a engenharia de tráfego a serem aplicadas no protocolo de encaminhamento Arquitectura Dinâmica de Informação Topológica (*Dynamic Topological Information Arquitecture (DTIA)*), apresentada no próximo capítulo.

Tabela 2.4: Sumário da taxonomia da engenharia de tráfego.

Método	Vantagens	Desvantagens	Aplicável
Em linha	• Realizado em 'tempo real', responde melhor ao dinamismo das redes. • Aplicável em ciclos de horas ou mesmo minutos.	• Dificuldade em tratar novas alterações da rede.	Intra/Inter-Domínio
Desligado	• Conhecimento prévio dos pontos de entrada e saída do tráfego.	• Falta de adaptabilidade ao dinamismo da rede. • Duração do Ciclo de Previsão de Recursos elevado (uma semana a um mês).	Intra/Inter-Domínio
MPLS	• Capacidade de encaminhamento explícito. • Agregação de fluxos individuais.	• Problemas de escalabilidade e robustez (número elevado de CCE).	Intra-Domínio
BGP	• Uso de atributos para encaminhamento. • Flexibilidade.	• Falta de qualidade de serviço. • Não é escalável[2].	Intra/Inter-Domínio

[2]Nas seguintes circunstâncias: SAs *multihomed*, encaminhamento multi-caminho e sem o Encaminhamento Inter-Domínio sem Classes (*Classless Inter Domain Routing (CIDR)*

Capítulo 3

Arquitectura Geral

3.1 Introdução

Neste capítulo é apresentada uma solução para o encaminhamento inter-domínio e as suas facilidades para aplicar a engenharia de tráfego. No capítulo anterior, percebeu-se que o protocolo utilizado em tráfego inter-domínio é o Border Gateway Protocol (BGP). A sua flexibilidade (por exemplo, com a utilização da manipulação de atributos) tem como contrapartida um aumento da complexidade (a manipulação dos atributos leva a alteração da distribuição de rotas), sendo difícil encontrar um equilíbrio. Administradores dos Sistemas Autónomos manipulam os atributos para alterarem os mecanismos de encaminhamento, mas também para outros fins, como a definição de Redes Virtuais Privadas baseadas em prefixos. Esta disparidade de funcionalidades pode tornar o sistema pouco adaptado a mudanças na topologia. Além disso, o BGP não tira vantagens de encaminhamento multi-caminho (*multipath*), ou mesmo de um cenário *multihomed*.

Amaral et al.'s [ABP08] propôs uma nova arquitectura para o encaminhamento inter-domínio, a Arquitectura Dinâmica de Informação Topológica (*Dynamic Topological Information Arquitecture (DTIA)*), que proporciona um protocolo de acessibilidade (*reachability*) simples. Nesta arquitectura, os encaminhadores constroem caminhos com base num mapa estático da rede e cooperam para aprender falhas nas ligações. Funcionalidades como encaminhamento e engenharia de tráfego podem ser implementadas sobre ela. A dissertação de mestrado de *Francisco Ganhão* [Gan09] estendeu o trabalho de *Amaral et al.'s* [ABP09] com o objectivo de melhorar a escalabilidade e convergência do encaminhamento, através de um encaminhamento multi-região. A dissertação apresentada neste

documento estende o trabalho de *Amaral et al.* com o objectivo de melhorar a Qualidade de Serviço (QdS) através da implementação da funcionalidade de engenharia de tráfego na arquitectura DTIA. Pretende-se reduzir a congestão de ligações, equilibrando a sua utilização.

Em primeiro lugar, é apresentada uma visão geral dos fundamentos do DTIA, seguido das camadas de acessibilidade e encaminhamento, terminando com a camada da engenharia de tráfego.

3.2 Arquitectura Dinâmica de Informação Topológica

A Arquitectura Dinâmica de Informação Topológica (*Dynamic Topological Information Arquitecture (DTIA)*) introduz uma nova abordagem para o encaminhamento inter-domínio, separando vários aspectos ao recorrer a um modelo por camadas: começando com a acessibilidade, seguido pelo encaminhamento e terminando com a engenharia de tráfego. Como referido na secção 3.1, esta separação de funções é uma vantagem em relação ao BGP. No entanto, para substituir as funcionalidades do BGP, a arquitectura DTIA usa um conjunto de regras para validar e diferenciar caminhos. Como o sistema trata caminhos de um modo diferente do BGP, i.e. de forma modular, está mais adaptada a redes com multi-caminho e *multihomed*.

3.2.1 Suposições do Modelo

Para a construção da arquitectura são feitas três suposições iniciais:

1. A continuidade do modelo actual de negócios baseado em Sistemas Autónomos e Provedores de Serviços de Internet;
2. O reconhecimento de uma estrutura hierárquica baseada em ligações do tipo cliente-fornecedor, e enriquecida com ligações do tipo fornecedor-fornecedor;
3. As ligações que formam a Internet são "estáveis" ao longo do tempo.

A primeira suposição é feita dada a realidade actual, que indica que dificilmente uma arquitectura baseada em modelos de negócio diferentes irá ser adoptada num futuro próximo

pelos administradores dos Sistemas Autónomos; contudo, o protocolo permite pequenas alterações.

A segunda suposição [AGA+09] é justificada pelo trabalho de *Gao's* [Gao00] que identifica uma estrutura hierárquica baseada em ligações cliente-fornecedor. Contudo, dados recentes retirados de Cooperative Association for Internet Data Analysis (CAIDA) [cai] mostram que a estrutura hierárquica era mais visível há uma década atrás. Nos dias que correm foram identificadas três tendências: ligações directas ignorando níveis hierárquicos (*tiers*); ligações entre fornecedores trocando grandes quantidades de tráfego; e acordos regionais a um nível *intermédio* da hierarquia. O BGP não consegue tirar vantagens desta quantidade de ligações em termos de multi-caminho e *backup*, visto que o protocolo só dá a conhecer o *melhor* caminho e as ligações de *backup* são ajustadas consoante a topologia da rede.

A terceira e última suposição é justificada pelo facto das ligações serem baseadas em relações comerciais. Mudanças na topologia da rede ocorrem de uma forma controlada. Para considerações em 'tempo real', não importa se existe uma relação ou não. O que é importante é saber se a ligação falhou ou não. Além disso, falhas intra-SA são mais prováveis que falhas inter-SA, reforçando a ideia que as ligações inter-domínio são estáveis[1].

3.2.2 Opções do Projecto

As suposições do modelo levaram a que fossem tomadas algumas opções no projecto:

1. Encaminhamento baseia-se no tipo de ligações entre SAs em vez de prefixos;
2. Um conjunto de regras implícitas na arquitectura DTIA impõem as políticas de encaminhamento;
3. Encaminhadores obtêm um mapa estático da rede e cooperam para conhecerem as falhas;
4. Mapas e cooperações são limitadas às regiões.

[1] Contudo, falhas intra-SA podem levar a falhas inter-SA.

A primeira opção é tomada considerando o tamanho das tabelas de encaminhamento do BGP. Trabalhar ao nível de prefixos amplia o tamanho das tabelas de encaminhamento, e é consensual que esse crescimento deve ser contido [MZF07]. Dado o número de SAs, a redução da tabela de encaminhamento é significativo se forem usados os números dos SAs em vez dos prefixos. Esta escolha é controversa, com algumas posições contra [Bon07] e outras a favor [SCE+05]. Esta escolha traz vantagens adicionais: a engenharia de tráfego permite obter o equilíbrio da carga das ligações entre os SAs, fornecendo uma solução mais eficiente baseada num único mapa em comparação com a solução dos prefixos; *multihoming* é reduzido à escolha de caminhos entre SAs, sem qualquer consequência para o tamanho da tabela de encaminhamento. Contudo, existem dois problemas: Primeiro, pacotes podem seguir caminhos diferentes com diferentes tempos de trânsito tornando-se necessário adaptar o algoritmo de controlo de congestão do TCP (O cálculo do *Round Trip Time* torna-se mais complexo, e a reacção do TCP tem de ser diferente caso receba pacotes desordenados) e segundo, deve existir uma lista com os vínculos entre SAs e prefixos.

Assume-se que existe um serviço que mapeia os prefixos com os SAs, suportando também *multihoming*. Com a arquitectura DTIA podem existir para um SA de destino vários caminhos diferentes, ou seja, diferentes SAs vizinhos para onde encaminhar o tráfego. É, então, necessário que seja possível mapear um conjunto de prefixos para um destino. Além disso, deve suportar mobilidade na atribuição de prefixos aos SAs, para lidar com os requisitos de mobilidade das redes militares.

No que diz respeito à segunda opção, foi mencionado na secção 2.2.2 quanto o BGP pode ser complexo. Os caminhos são exportados e escolhidos com base nos atributos que são definidos de acordo com os objectivos das políticas dos SAs. No BGP existe a liberdade absoluta na manipulação de atributos, logo, nas políticas também. Esta flexibilidade e a falta de coordenação entre SAs pode causar problemas de convergência. Soluções para estes problemas [MC05] podem ser: Aplicação de coordenação; restringir o uso de políticas; ou a detecção e correcção em 'tempo real'.

Na arquitectura DTIA é definido um conjunto de regras que determina se um caminho é válido ou não. Também é definido um nível de preferência para os caminhos com base nas características comerciais. Apesar da flexibilidade do sistema actual, a natureza das relações comerciais na Internet conduziu a um vasto uso das políticas de encaminhamento

baseado nas relações comerciais entre SAs [CR05]. Estas *políticas comuns* têm um papel importante na estabilidade do encaminhamento na Internet, uma vez que foi comprovado que a sua utilização leva a um conjunto de caminhos estáveis [GR01][Sob05]. O conjunto de regras abrange as *políticas comuns* mais dois casos: relações entre iguais (*sibling relations*) e ligações de reserva (*backup links*). Um conjunto estável de caminhos é formado e não existem ciclos no encaminhamento. Novas regras podem ser definidas enquanto certas características do sistema forem mantidas. Entretanto, a alteração das regras é considerada pelo DTIA uma grande mudança. Resumindo, parte da flexibilidade do BGP é perdida porque nem todas as políticas possíveis asseguram a convergência, logo, nem todas podem ser incluídas nas regras. Uma funcionalidade, como a engenharia de tráfego de entrada, que é actualmente obtida por uma complexa manipulação de atributos, como o *path prepending* [CL05] ou o uso de comunidades [DB08], é deixada de fora da arquitectura de encaminhamento para ser executada por cima desta a nível de engenharia de tráfego.

Para a terceira opção, os encaminhadores obtêm um mapa estático da rede e cooperam para conhecerem as falhas. Já assumimos que as ligações inter-domínio são estáveis devido à existência de relações comerciais. Um protocolo inter-domínio deve-se unicamente preocupar com a parte dinâmica da rede e não com a sua descoberta. Contrastando com o BGP, o algoritmo para a parte dinâmica deve ser leve e o algoritmo geral deve permitir realizar as funcionalidades da engenharia de tráfego. O BGP por outro lado, depende fortemente de mecanismos para a descoberta e gestão da rede.

A arquitectura DTIA assume que uma entidade central (ou replicada) entrega um mapa estático da rede a todos os encaminhadores. Não é garantido que o mapa seja o estado actual da rede, devido a falhas, mas todos os encaminhadores conhecem a mesma informação e reagem a isso dinamicamente. Em relação à parte estática, como não há necessidade de descobrir o mapa, os paradigmas tradicionais de encaminhamento não se aplicam (vector-distâncias, vector de caminhos, e estado de linha). Este conhecimento também simplifica a parte dinâmica em termos de troca de mensagens - a divulgação da informação de uma falha só deve "perturbar" os encaminhadores relevantes.

A quarta opção foca o encaminhamento em regiões. A maioria das preocupações do encaminhamento inter-domínio são referentes a caminhos locais ascendentes (e descen-

dentes). Acontecimentos globais reais no BGP estão relacionados com o procedimento de desempate entre prefixos. Ao não ter anúncios de prefixos ou processamento baseado em prefixos (como em [SCE+05], [YCB07]), ocorre uma maior simplificação nos eventos de encaminhamento e de propagação de falhas. Assim, as preocupações são o cálculo do mapa da rede e a gestão das falhas da rede. Para os conter é definido o conceito de 'região'. Uma região é simplesmente um conjunto de SAs sujeitos a algumas restrições. Para cada região é construído e entregue aos encaminhadores um mapa estático. Actualmente, RIPE tem uma base de dados [RIP] em fase inicial que pode ser usada para este fim (já é usado por alguns fornecedores para verificarem os anúncios de prefixos feitos pelos seus clientes). O seu conteúdo poderia ser utilizado para uma região *Europeia*. Uma definição mais concreta de regiões é dada na secção 3.2.3.

3.2.3 Definição de Região

O trabalho de *Amaral et al.* [ABP09] afirma que o número de SAs deve ser tal que o tempo necessário para realizar o cálculo dos caminhos seja realista. De acordo com o trabalho, 11,000 SAs é possível. Além disso, as regiões devem ter as seguintes características:

1. Um SA tem de ter caminhos para todos os SAs numa região (isto não é drástico, uma vez que tendo um fornecedor no primeiro nível da hierarquia resolve o problema);

2. Cada região tem de ter SAs conectados a todas as outras regiões e rotas de acordo com as regras apresentadas na secção 3.3 (esta característica assume que a região tem um fornecedor no nível-1, ou um SA como fornecedor no nível-1).

3.3 Camadas da Arquitectura

Esta secção explica a arquitectura do DTIA. Tradicionalmente, um algoritmo de encaminhamento é executado por duas operações distintas:

1. Mecanismo - define como as rotas são conhecidas (e.g. vector de distâncias) e define um algoritmo de selecção de rotas (escolha de uma ou mais rotas usando, por exemplo, o algoritmo de Dijkstra's - caminho mais curto).

2. Política - define as características de uma ligação (métrica ou atributos); tem consequências directas no algoritmo de selecção de rotas.

O BGP é inteiramente baseado em políticas de prefixos, o que significa que a componente política tem consequências directas na descoberta da rede para cada prefixo. Em oposição ao BGP, o DTIA divide as funções em três camadas:

- Acessibilidade (*Reachability*);
- Encaminhamento (*Routing*);
- Engenharia de tráfego (*Traffic engineering*).

Para uma melhor compreensão do funcionamento das duas primeiras camadas, é dada uma descrição dos níveis em que estas são aplicadas: nível de controlo e nível de envio. Ao nível de controlo a acessibilidade e o encaminhamento usam um mapa estático baseado na rede de SAs, para calcularem as rotas entre SAs. Ao nível de envio, são usados os números dos SAs para enviar pacotes.

Nível de Controlo

1) *Mapa estático ao nível de SAs*

O mapa estático da rede é a base das camadas de acessibilidade e de encaminhamento. Deve ser mantido por uma entidade com servidores possivelmente replicados.

Os vários estudos que tentaram deduzir mapas ao nível SAs da Internet [Gao00, OPW$^+$10, DKF$^+$07], mostram que a informação é incompleta, e por vezes imprecisa. No entanto, se existir a vontade por parte dos SAs em manter essas bases de dados, estes estudos poderiam ser a base para o mapa da rede. Um aspecto importante em relação aos mapas é a sua distribuição. A ideia do DTIA é fazer a distribuição por inundação. Este procedimento não é tão ineficiente quanto possa parecer devido à característica de "mundo pequeno" da Internet. Na inicialização da rede (fase que nunca deixará de existir), o mapa pode ser solicitado pelos SAs através das ligações aos seus fornecedores. O mapa seria então distribuído pelo primeiro nível da hierarquia de SAs (possivelmente por uma ligação directa) e um processo de distribuição seria utilizado para a sua divulgação aos restantes níveis de SAs. Sendo o *multihomed* uma das vantagens do DTIA, um SA pode receber a mesma versão do mapa várias vezes. Para que não o processe repetidamente, um algoritmo simples de verificação de versões pode ser executado pelos fornecedores sempre

que um mapa é recebido.

2) *Nível de acessibilidade - Validação de caminhos*

Após a recepção do mapa, os SAs analisam todos os caminhos possíveis para todos os SAs do mapa.

O primeiro passo, realizado na camada de acessibilidade [ABP09], é aplicar um conjunto de regras para remover caminhos inválidos. Um caminho válido é aquele que está em conformidade com as políticas. As políticas são aplicadas ao nível de SAs e não de prefixos. Como mencionado anteriormente, DTIA abrange as chamadas *políticas comuns* mais duas políticas extra. As *políticas comuns*, que incluem as ligações cliente-fornecedor e fornecedor-fornecedor, são suficientes para lidar com 99% das ligações existentes na Internet de hoje [Gao00, SCE+05]. Estas políticas excluem todas as rotas que incluam vales (fornecedor-cliente-fornecedor). As outras duas, são usadas para ligações do tipo *sibling* e *backup* (como sugerido no RFC 1998). O resultado é um mapa novo que contém todas as ligações válidas de um SA X para todos os outros SAs da rede. Este conjunto de caminhos é definido como $P_r(X)$. Contém para cada destino não só o melhor caminho mas todos os caminhos que cumprem as políticas. A diferença em relação ao BGP é que no DTIA o processo é local aos SAs e não há troca de mensagens de encaminhamento. Além disso, pode existir mais do que um caminho para o destino (muito devido ao *multihoming*), fornecendo uma base para o encaminhamento multi-caminho.

3) *Nível de encaminhamento - Classificação de caminhos*

Em termos gerais, diferentes protocolos de encaminhamento podem ser definidos. A arquitectura DTIA propõe um protocolo de encaminhamento com multi-caminhos, que atribui um qualificador a cada caminho válido. Este qualificador é baseado nas propriedades das ligações que formam o caminho. Por exemplo, um caminho pode ser do tipo cliente ou fornecedor, e é calculado de acordo com as relações estabelecidas para cada ligação que formam o caminho. É definida na secção 3.3.2 uma ordem de preferência para os qualificadores, determinando que caminhos são preferidos.

O conjunto de qualificadores de caminhos são calculados com base numa álgebra de en-

caminhamento [Sob05] e é estabelecida uma relação de preferência. São baseados nas mesmas políticas das regras de validação. As propriedades da álgebra de encaminhamento têm um papel fundamental na solidez do protocolo (como serão demonstradas na secção 3.3.2). Diferentes políticas podem ser usadas, desde que as propriedades da álgebra sejam mantidas. É de salientar que todos os SAs devem seguir as mesmas políticas. A secção 3.3.2 expressa os detalhes das políticas usadas, demonstra a álgebra de encaminhamento e prova que não existem ciclos no envio de pacotes. Na arquitectura DTIA não há ciclos. O resultado final da camada de encaminhamento é uma tabela de encaminhamento, baseada nos números dos SAs. Esta tabela de encaminhamento pode conter mais que um caminho para cada destino e separa-os consoante os seus qualificadores. A possibilidade do uso de multi-caminhos da mesma categoria de ligações sem causar ciclos no envio de pacotes fornece uma boa base para a engenharia de tráfego.

4) *Distribuição de rotas intra-domínio*

O DTIA assume que só existe uma entidade reguladora por SA, o que tem algumas vantagens mas também cria algumas dificuldades. No interior de um SA parte-se do princípio que os encaminhadores de fronteira actuam em concordância e que estão conectados em malha completa. Os encaminhadores de fronteira participam no encaminhamento intra-domínio e anunciam o facto de serem encaminhadores de fronteira aos encaminhadores internos. Quando um encaminhador interno recebe um pacote com o endereço IP externo ao SA, este simplesmente envia-o para um dos encaminhadores de fronteira.

Nível de Envio

Como o envio é baseado nos números dos SAs, para manter o pacote IP inalterado tem de ser construída uma lista com os vínculos entre endereços IP e os números dos SAs. Além disso, todos os SAs que constituem um caminho têm de construir a lista com os vínculos. Uma possível optimização é adicionar um cabeçalho ao pacote com o número do SA de destino, eliminando a necessidade da verificação dos vínculos entre prefixos e números de SAs ao longo do caminho. Este cabeçalho seria acrescentado pelo encaminhador de fronteira do SA de origem e removido pelo encaminhador de fronteira do SA de destino.

O envio na camada de encaminhamento é realizado através da escolha de um caminho da

lista de caminhos disponíveis que tenham o qualificador mais preferido.

Da secção 3.3.1 à secção 3.3.3 são apresentadas as três camadas do modelo DTIA.

3.3.1 Acessibilidade

Na secção 3.3 foi assumido que um mapa da região seria distribuído aos SAs por uma entidade central. Por cada mapa novo gerado, é gerado um número sequencial único. Este mapa *G(V,A)* é estruturado como um mapa directo; onde *V(G)* representa os vértices que descrevem os SAs e *A(G)* representa os arcos que descrevem as ligações entre SAs. Os arcos são etiquetados de acordo com as relações comerciais entre os SAs. A arquitectura DTIA considera quatro tipos de relações inter SAs, que são descritas a seguir.

Tipos de relação inter SA

- Fornecedor-Cliente (*Provider-Customer*) - Um SA (fornecedor) aceita todo o tráfego do outro SA (cliente). São considerados dois arcos: um em cada direcção, fornecedor-cliente (*f2c*) e cliente-fornecedor (*c2f*).
- Fornecedor-Fornecedor (*Peer-to-peer*) - Os SAs fornecem conectividade aos seus clientes directos ou indirectos. Não é permitido tráfego de trânsito de fornecedores. Existe um arco em cada direcção (*f2f*).
- Fornecedor-Fornecedor que permite *backup* (*Peer-to-peer allowing backup*) - O mesmo que o caso anterior, mas permite tráfego de trânsito se não existir outro caminho. Existe um arco em cada direcção (*f2fbkp*).
- Fornecedor-Fornecedor que permite tráfego de trânsito (*Peer-to-peer allowing transit traffic*) - Tráfego de trânsito é permitido em qualquer situação. Existe um arco em cada direcção (*f2fptt*).

De acordo com estas relações, o protocolo define duas tabelas de regras para validar caminhos e construir $P_r(X)$ (conjunto de todos os caminhos válidos que começam num SA X). O DTIA explora todos os caminhos ascendentes (*c2f*), descendentes (*f2c*) e horizontais (*f2f, f2fbkp* e *f2fptt*) num processo salto-a-salto (*hop-by-hop*) no sentido do envio,

começando no SA X. Para evitar um 'vale' nos caminhos, um atributo *Direcção* (D) é acrescentado ao caminho. A direcção do caminho é atribuído de acordo com a relação da primeira ligação:

1. Se *c2f* então D é 1;
2. Se *f2c* então D é 0;
3. Se *f2fbkp* ou *f2fptt* então temos dois caminhos possíveis, com $D = 0$ e $D = 1$;
4. Se *f2f* então D é 0;

O valor do atributo D pode ser alterado na exploração de um caminho. Um caminho descendente (*D=0*) nunca se altera para um caminho ascendente (um 'vale' não é permitido nos caminhos). Um caminho ascendente é alterado para um caminho descendente quando é encontrado no caminho um arco do tipo *f2c*. Arcos do tipo fornecedor-fornecedor colocam problemas extras em termos da garantia da não existência de ciclos nos caminhos. Ciclos são evitados ao não permitir que um SA apareça num caminho mais do que uma vez.

As tabelas: 3.1 e 3.2 mostram respectivamente as regras de validação para os caminhos descendentes e ascendentes; V marca o caminho como válido e o X como inválido. É de notar que um caminho descendente com arcos de partida do tipo *f2f* ou *c2f* nunca é válido.

Tabela 3.1: Validação de caminhos para $D = 0$.

		Arco de Partida				
		f2c	*c2f*	*f2fbkp*	*f2f*	*f2fptt*
Arco de Chegada	*f2c*	V	X	V	X	V
	c2f	-	-	-	-	-
	f2fbkp	V	X	Se(SA já existe)X se não V	X	Se(SA já existe)X se não V
	f2f	X	X	X	X	X
	f2fptt	V	X	Se(SA já existe)X se não V	X	Se(SA já existe)X se não V

Depois de ser obtido o conjunto de caminhos $P_r(X)$, o algoritmo de encaminhamento decide quais os caminhos a serem utilizados para cada destino. Os autores do protocolo provam que os caminhos explorados de $P_r(X)$ não contêm ciclos e enunciam o seguinte

teorema [ABP08].

Teorema 1. *Assumindo que não existem ciclos no relacionamento cliente-fornecedor (i.e. nenhum domínio é fornecedor de um dos seus fornecedores directos ou indirectos assumindo que SAs vizinhos do mesmo tipo também são fornecedores indirectos), um caminho válido numa região entre dois SAs não tem ciclos. (Assuming that there are no cycles in the provider-customer relationships (i.e. no domain is a provider of one of its direct or indirect providers assuming that peers are also indirect providers). A valid path between two AS in the region has no loops.)*

O teorema 1 garante que todos os caminhos são válidos, desde que respeitem as regras indicadas nas tabelas 3.1 e 3.2. Caso contrário, o sistema será instável se alguns SAs tiverem caminhos conflituosos que não respeitam as regras mencionadas.

Tabela 3.2: Validação de caminhos para $D = 1$.

		Arco de Partida				
		f2c	*c2f*	*f2fbkp*	*f2f*	*f2fptt*
Arco de Chegada	*f2c*	-	-	-	-	-
	c2f	V;D=0	V	V	V	V
	f2fbkp	V;D=0	V	Se(SA já existe)X se não V	X	Se(SA já existe)X se não V
	f2f	V;D=0	X	X	X	X
	f2fptt	V;D=0	V	Se(SA já existe)X se não V	X	Se(SA já existe)X se não V

3.3.2 Encaminhamento

Esta secção explica a camada de encaminhamento do trabalho *Amaral et al.'s* [ABP09]. O facto de $P_r(X)$ ser formado por caminhos livres de ciclos não significa que todo o sistema está livre de ciclos. Isto deve-se a duas razões: ao facto de o sistema ser formado por multi-caminhos e de um caminho poder estar em conflito com outro; e mesmo que este aspecto seja resolvido, se existir uma ligação que falhe, ainda podem ocorrer conflitos similares.

Para resolver este problema é definido um mecanismo de qualificação, para classificar os caminhos válidos e um algoritmo de gestão de falhas. O mecanismo de qualificação usa um espaço discreto que atribui um custo aos caminhos. Caminhos que tenham o mesmo valor são tratados do mesmo modo, proporcionando assim uma base para o multi-caminho. O mecanismo de qualificação é explicado a seguir.

Mecanismo de Qualificação

O mecanismo de qualificação é suportado por quatro regras de preferência; as duas primeiras são já conhecidas da Internet actual; as outras duas foram adicionadas devido às novas políticas de ligação entre SAs:

1. Não é enviado tráfego de um fornecedor para outro fornecedor;
2. As rotas de clientes são preferidas em relação às rotas de fornecedores;
3. Caminhos principais são sempre preferidos em relação a caminhos de *backup*;
4. Entre os caminhos principais, caminhos *F2Fptt* e *F2C* são preferidos (com valor igual) aos caminhos *F2F*. Caminhos *C2F* têm a menor preferência.

A aplicação destas regras ao $P_r(X)$ tem dois efeitos: alguns caminhos válidos não são considerados para o encaminhamento; todos os caminhos têm um qualificador que os classifica. É provado em [ABP09] que se cada SA usar os caminhos da maior preferência disponível para o encaminhamento, o algoritmo de encaminhamento converge e os pacotes chegam ao SA de destino sem formarem ciclos no encaminhamento.

O algoritmo de encaminhamento da arquitectura DTIA funciona como um protocolo de Vector de Caminhos, tal como o BGP, já que escolhe as rotas de acordo com os seus atributos e com a preferência estabelecida. Segundo *Amaral* [AGA+09] o DTIA funciona também como um Vector de Caminhos Simulado Localmente (SVCL) (*Local Simulated Path Vector (LSPV)*) [GS05].

Correcção do Encaminhamento

Um protocolo de encaminhamento está correcto se numa rede estável (sem alterações) obtém um conjunto de caminhos sem ciclos, entre todos os pares de nós que estão conec-

tados. A correcção do encaminhamento no DTIA é provado usando o conceito da álgebra de encaminhamento de *Sobrinho* [Sob05]. As propriedades algébricas para assegurar a correcção do encaminhamento são válidas para protocolos de Vector de Caminhos [GS05], e o DTIA pode ser visto como um protocolo de Vector de Caminhos.

Uma álgebra de encaminhamento é definida por um tuplo $A = (\sum, \prec, \bigoplus, L, \phi)$. $\sum$ é um conjunto de *assinaturas* que qualificam os caminhos, $\prec$ é uma relação de preferência sobre as assinaturas (i.e. com $\alpha \prec \beta$, α é preferida), L é um conjunto de etiquetas associadas às ligações, $\bigoplus$ é uma operação binária para obter as assinaturas dos caminhos, ϕ é uma assinatura especial que indica caminhos inválidos.

Para este protocolo, temos o seguinte conjunto de etiquetas $L = \{f2fptt,\ f2c,\ f2f,\ c2f,\ f2fbkp\}$ e as assinaturas $\sum = \{\varepsilon, F2Fptt, F2C, F2F, C2F, F2Fbkp\} \cup \{BKP \times N^+\}$. A assinatura ε é a assinatura inicial do caminho quando só existe um nó no fim do caminho; as restantes assinaturas são similares aos tipos etiqueta/ligação. Cada SA usa um $P_r(X)$ e a tabela 3.3 para efectuar os cálculos das assinaturas dos caminhos usando a operação $\bigoplus$.

Tabela 3.3: Operação binária $\bigoplus$, na coluna mais à esquerda temos a ligação adicionar e na linha mais a cima temos a assinatura do caminho.

		Assinatura						
		ε	$F2Fptt$	$F2C$	$F2F$	$F2Fbkp$	$C2F$	(BKP, y)
Etiqueta	$f2fptt$	$F2Fptt$	$F2Fptt$	$F2C$	ϕ	$(BKP, 1)$	$C2F$	$(BKP, y+1)$
	$f2c$	$F2C$	$F2C$	$F2C$	ϕ	$(BKP, 1)$	ϕ	$(BKP, y+1)$
	$f2f$	$F2F$	$F2F$	$F2F$	ϕ	$(BKP, 1)$	ϕ	ϕ
	$c2f$	$C2F$	$C2F$	$C2F$	$C2F$	$C2F$	$C2F$	$(BKP, y+1)$
	$f2fbkp$	$F2Fbkp$	$(BKP, 1)$	$F2Fbkp$	ϕ	$(BKP, 1)$	$(BKP, 1)$	$(BKP, y+1)$

O procedimento é o seguinte: uma ligação com tipo de etiqueta l é adicionada na direcção do nó de origem ao caminho de assinatura α, resultando numa nova assinatura do caminho $\beta = l \bigoplus \alpha$. Observando a tabela 3.3, ao adicionar uma ligação $c2f$ a um caminho $F2F$, então a assinatura do novo caminho será $C2F$; isto significa que a assinatura resultante é um caminho ascendente para um fornecedor seguido de um caminho com assinatura $F2F$. A tabela 3.4 mostra a ordem de qualificação pelo qual os caminhos são escolhidos; a mais elevada é mais preferida.
Observando cuidadosamente a tabela 3.3, notamos que uma ligação $f2fbkp$ poderia ser

usada como uma ligação regular fornecedor-fornecedor ou como uma ligação de *backup*. Em termos de preferência, uma assinatura $F2Fbkp$ (reflectindo a sua utilização como um regular fornecedor-fornecedor (F2F)) tem o mesmo efeito que uma assinatura $F2F$. Quando uma ligação é usada como *backup*, a assinatura resultante é (BKP, y). y é incrementado cada vez que uma ligação é adicionada. Deste tipo de ligação podemos extrair dois exemplos:

1. Ligações de *backup* usadas como troca normal de tráfego: a assinatura resultante de $f2fbkp \bigoplus F2C$ é $F2Fbkp$. Adicionando uma ligação $f2fbkp$ a um caminho cliente é equivalente a um caminho $F2F$, desde que exista uma ligação normal de troca de tráfego.

2. Ligações de *backup* usadas como *backup*: para um caminho *backup* a assinatura resultante é sempre (BKP, y). O valor de y é incrementado cada vez que uma ligação $f2fbkp$ é usada. Para cada nova ligação num caminho *backup*, o inteiro y é incrementado. É possível usar um caminho com assinatura $f2c \bigoplus (BKP, y)$. Contudo decresce a preferência do caminho, já que estamos adicionar uma ligação fornecedor a um caminho *backup*.

Tabela 3.4: Ordem de Qualificação.

ε
F2C = F2Fptt
F2F = F2Fbkp
C2F
(BKP,1)
...
(BKP,n)

É preciso compreender a definição de ciclo e monotonia antes de delinear a correcção do encaminhamento. Um ciclo é uma sequência de nós distintos, excepto o primeiro e último. Um ciclo é livre se pelo menos um dos nós envia pacotes para o destino fora do ciclo em vez de volta ao ciclo. Ou seja, nesse nó a preferência por outro caminho é maior que a preferência pelo próximo nó pertencente ao ciclo.

Considerando que cada nó i do ciclo tem uma assinatura α_i, mas o nó i tem outros caminhos j para o destino que não seguem o ciclo com assinaturas β_{ij}. Seja $S(x_i, x_{i-1}, x_{i-2})$ a

assinatura do caminho $x_i\ x_{i-1}\ x_{i-2}$; a condição para que um ciclo seja livre é:

Definição 1. *Liberdade dos ciclos: um ciclo x_1, x_2, ..., x_{n-1}, x_n com $x_n = x_1$ é livre, se existir um índice i, $2 \leq i < n$ tal que $\beta_{ij} \prec S(x_{i+1}, x_{i+2}, \ldots, x_n)$.*

Da definição 1, um ciclo L é sempre livre enquanto em cada nó existem caminhos com maior preferência que a de L. Esta definição leva-nos até à propriedade seguinte:

Definição 2. *Liberdade da rede: Uma rede é livre, enquanto todos os ciclos forem livres.*

A monotonia é também uma propriedade importante. Uma álgebra é monótona, se a preferência de um caminho não subir quando uma ligação é adicionada a um caminho.

Definição 3. *Monotonia de uma álgebra: Uma álgebra é monótona se para todos os $\alpha \in \sum$, e para todos os $l \in L$, $\alpha \preceq \alpha \bigoplus l$.*

Define-se também uma propriedade mais forte de monotonia: monotonia estrita. Esta propriedade garante que ao se adicionar uma etiqueta a um caminho, decresce a preferência da assinatura resultante. A partir destas propriedades, o trabalho de *Sobrinho* prova os teoremas seguintes [Sob05]:

Teorema 2. *Numa rede livre, um protocolo de vector de caminhos converge para uma rede em árvore óptima, independentemente da rede. (In a free network, a path-vector protocol converges to local optimal in-trees).*

Teorema 3. *Se um protocolo de vector de caminhos tem uma álgebra monótona, então o protocolo pode convergir para uma rede em árvore, independentemente da rede. (If a path-vector protocol has a monotone algebra, then the protocol can converge to local in-trees, regardless of the network).*

Os teorema 2 e 3 são importantes para provar que um protocolo vector de caminhos converge. A partir destes teoremas, os autores de DTIA provam o seguinte teorema [ABP09]:

Teorema 4. *Supondo que a rede não tem ciclos nas relações cliente-fornecedor, então o protocolo de encaminhamento da arquitectura DTIA converge usando conjuntos de caminhos sem ciclos. (Assuming that the network has no cycles in the provider-customer relationships, then DTIA's routing protocol converges using sets of cycle free paths).*

Teorema 4 é uma propriedade importante porque prova que o DTIA é um protocolo *estável*; todos os SAs convergem com o mesmo conjunto de caminhos livres de ciclos. Contudo, é necessário cuidado com as ligações *f2fptt*, pois são as únicas que conseguem formar ciclos não livres: cada ligação *f2fptt* adicionada não decresce a preferência do caminho. Posto este facto, conclui-se que o protocolo é monótono simples. Para desfazer o empate de caminhos *F2Fptt*, os autores sugerem um solução simples que escolhe o caminho *F2Fptt* com o menor número de ligações, ou com o menor número de ligações com etiquetas *f2fptt*.

Se vários caminhos tiverem o mesmo número de ligações, podem ser todos escolhidos porque o DTIA permite encaminhamento multi-caminho. Por exemplo, dois caminhos *F2C* e o caminho mais curto *F2Fptt* em simultâneo.

Implementação

Dada a monotonia de ambas as álgebras é possível conceber um algoritmo muito simples para calcular as assinaturas de um caminho no sentido *para a frente* ao contrário do sentido *para trás* usado na definição da álgebra. Ao implementar o algoritmo deste modo, é mais simples e tem um grande impacto na complexidade e no tempo de processamento. Este método só é possível porque o cálculo é feito com base em caminhos válidos (usando $P_r(X)$). Caso contrário, existiriam problemas na análise da validade dos caminhos.

Com base na monotonia, pode ser visto que um caminho com uma certa assinatura, estendido por uma ligação na direcção da origem, ou mantém a ordem de preferência ou a diminui.

O algoritmo funciona do seguinte modo: primeiro, é considerada S_i como a assinatura de um caminho, com uma só ligação com etiqueta l_i entre dois SAs, que é definido pela seguinte expressão $S_i = l_i \bigoplus \varepsilon$. Então, seguimos os caminhos na direcção *para a frente* e para cada caminho é calculada a S_i para cada salto até ao destino. A assinatura do caminho completo para um dado SA de destino, SA n, é a menos preferida de todas as S_i.

A figura 3.1 exemplifica o cálculo da assinatura de um caminho do nó X ao nó D. O processamento do cálculo é o seguinte:

1. $S_1 = c2f \bigoplus \varepsilon = C2F$, cálculo efectuado com a etiqueta da primeira ligação (entre o nó X e nó A, $c2f$) e a assinatura inicial (ε);

2. $S_2 = f2fbk \bigoplus C2F = C2F$, cálculo efectuado com a etiqueta da segunda ligação (entre o nó A e nó B, $f2fbkp$) e a assinatura anterior ($C2F$), a assinatura mantêm-se porque o SA B é visto como um fornecedor normal;

3. $S_3 = f2fbk \bigoplus C2F = (BKP, 1)$ cálculo efectuado com a etiqueta da terceira ligação (entre o nó B e nó C, $f2fbkp$) e a assinatura anterior ($C2F$), a assinatura altera-se em relação ao caso anterior porque são consideradas as ligações de *backup*, $A - B$ e $B - C$;

4. $S_4 = c2f \bigoplus (BKP, 1) = (BKP, 2)$, cálculo efectuado com a etiqueta da quarta ligação (entre o nó C e nó D, $c2f$) e a assinatura anterior ($(BKP, 1)$).

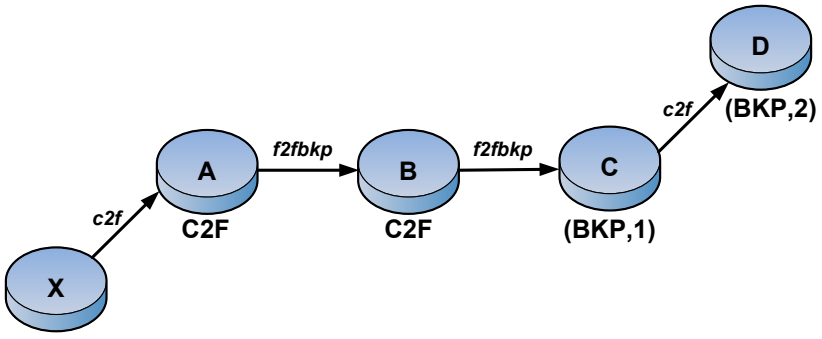

Figura 3.1: Cálculo de assinaturas na direcção *para a frente*.

Como podemos ver, a assinatura do caminho completo $(BKP, 2)$ é a assinatura menos preferida de todas as assinaturas.

Gestão de Falhas

O mapa estático não garante que as ligações estão activas, ou seja, não garante que seja o estado actual da rede. DTIA tem uma parte dinâmica para a acessibilidade e protocolos de encaminhamento, com o intuito de ter consciência das falhas de ligação para se adaptar a elas. Existem dois objectivos: assegurar que não existem ciclos no envio durante as falhas e que nenhum pacote é perdido se existir pelo menos um caminho sem falhas até ao destino.

Assumindo que uma ligação falha, os encaminhadores divulgam um pacote de controlo na camada de acessibilidade ou de encaminhamento. Os pacotes de controlo contêm a identificação da ligação que falhou, a direcção e o número da versão do mapa. Quando um SA recebe um pacote de controlo, é verificado se todos os SAs ainda são alcançáveis sem o uso da ligação que falhou. Se pelo menos um SA ficar inalcançável, a divulgação do pacote de controlo continua; a divulgação é feita segundo as regras das tabelas 3.1 e 3.2. Caso contrário, se todos os SAs continuam alcançáveis a divulgação do pacote de controlo termina. Uma avaliação posterior é feita na camada de encaminhamento.

Falhas em ligações na camada de encaminhamento podem alterar as assinaturas dos caminhos para alguns SAs. No encaminhamento podem ocorrer ciclos se um novo caminho pertencer a uma preferência menor que o caminho utilizado anteriormente. Observemos a figura 3.2.

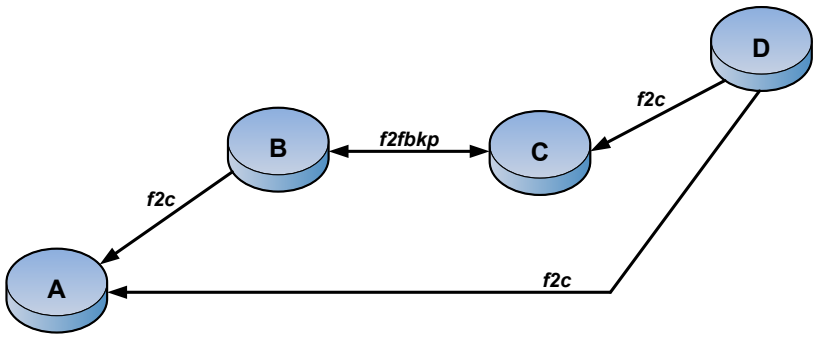

Figura 3.2: Exemplo de uma rede com ciclo no encaminhamento se a ligação $A-B$ falhar.

Se a ligação $A - B$ falhar, de acordo com a camada de acessibilidade A e B podem-se alcançar um ao outro, desta forma não são enviados pacotes de controlo nesta camada. Na camada de encaminhamento, antes da falha a assinatura do SA B ao A é $F2C$. De-

pois da falha a assinatura muda para $(BKP, 1)$ (caminho $B - C - D - A$). Como B está ciente da falha, irá encaminhar pacotes pelo SA C; no entanto, C não tem conhecimento da falha e prefere usar o SA B como próximo salto para encaminhar pacotes; porque a assinatura do caminho $C - B - A$ é $F2Fbkp$ e tem preferência mais elevada que a assinatura do caminho $C-D-A$ ($C2F$). Nesta situação temos um ciclo entre B e C.

Para assegurar a correcção do encaminhamento, o SA C precisa de ser avisado da falha da ligação $A-B$. Como SA B mudou o caminho seleccionado para outro com preferência menor, deve notificar o SA C para assegurar a correcção do encaminhamento. Assim que o SA C for notificado, este irá usar essa informação para encaminhar os pacotes seguintes.

Em termos gerais, as condições de divulgação são as seguintes: se a assinatura de um caminho for alterada então são enviados pacotes de controlo de acordo com as regras. Os pacotes de controlo contêm a identificação de todas as ligações que falharam conhecidas pelo SA que envia o pacote. Após a recepção de um pacote de controlo, um SA só identifica as ligações das quais não conhece a sua falha, se conhecer todas as falhas descarta o pacote. Se todos os caminhos mantiverem o nível de preferência da assinatura então a divulgação termina.

Com o tempo, os mapas que os SAs usam para calcular as rotas podem não ser o mesmo, porque as notificações só são enviadas quando uma falha de ligação afecta um SA. Esta propriedade representa um forte mecanismo de contenção. Os pacotes de controlo não são enviados para SAs onde os seus caminhos válidos não são afectados pela falha da ligação. Se uma ligação for retomada, os critérios de divulgação são os mesmos. Por exemplo, um SA inatingível torna-se acessível por uma falha de ligação recuperada ou existe um novo caminho com maior preferência que os outros que estão a ser usados.

A quantidade de pacotes de controlo enviados está directamente relacionado com o grau de *multihoming* da região. Um elevado grau de *multihoming*, associado a maior redundância, torna o número de pacotes de controlo enviados menor. É menos provável que um SA perca a acessibilidade desde que existam mais caminhos da mesma preferência, interrompendo o envio de pacotes de controlo.

Se um SA falhar, ou seja, todas as suas ligações falharem (é evento raro), então toda a

região é avisada. No entanto, se um SA terminal falhar conectado a um único fornecedor, o fornecedor não irá avisar a região inteira. Pacotes de dados falham no fornecedor. Isto é consistente com a Internet actual. Ainda hoje, pacotes de dados podem chegar a um SA só para verificar que um prefixo não é válido naquele momento.

Correcção do encaminhamento na presença de falhas

Os autores provam [ABP08, p. 5-6] que todos SAs interessados são avisados nos termos de acessibilidade e encaminhamento. Ciclos transientes são contidos e duram apenas enquanto pacotes de controlo são divulgados. Finalmente, os autores provam que, se existir um caminho para o destino, nenhum pacote é perdido e o protocolo converge [ABP08].

Teorema 5: A divulgação do pacote de controlo garante informar todos os SAs que um SA antes acessível tornou-se inacessível após a falha de uma ligação (*The control packet dissemination guarantees to inform all ASes that a previously reachable AS becomes unreachable after a link failure*).

O teorema 5 é provado por contradição. Suponha-se que um SA é suposto receber um pacote de controlo, mas não o recebe. Esta situação só ocorre se o SA perdeu a acessibilidade ou todos os SAs têm caminhos alternativos que não usam a ligação que falhou e continuam a alcançar todos os SAs que antes alcançavam. Ambos os casos não podem ocorrer em simultâneo. Portanto, todos os SAs que têm caminhos válidos que usam a ligação que falhou são avisados.

Se tivermos na presença de múltiplas falhas de ligação, então é possível que alguns pacotes de controlo não cheguem a alguns destinos. Nestas situações, um SA deve guardar pacotes de controlo para os seus vizinhos até estes estarem alcançáveis. Uma vez a ligação seja restabelecida com o vizinho, todos os pacotes de controlo pendentes são enviados para o vizinho.

O teorema 5 também se aplica a ligações restabelecidas. Todos os SAs que tenham um caminho válido usando a ligação restabelecida são notificados. Quando um SA recebe um pacote de ligação restabelecida, este deve cancelar todos os pacotes pendentes para a essa ligação.

Teorema 6: A divulgação do pacote de controlo garante informar todos os SAs que têm de mudar as suas decisões de encaminhamento de modo a manter-se a convergência do encaminhamento (*The control packet dissemination guarantees to inform all AS that have to change routing decisions to maintain routing convergence*).

Garantir a acessibilidade não é suficiente quando ocorre uma alteração na topologia. É imperativo que os caminhos não criem ciclos quando o estado das ligações é alterado. Assumindo que o estado das ligações é alterado num instante t, os autores do DTIA provam que no instante $t = t^+$ todos os SAs que não convergiram para os mesmos caminhos livres de ciclos são avisados com um pacote de controlo [ABP09]. É necessário garantir que as decisões de encaminhamento são uniformes a todos os nós. Na recepção de um pacote de controlo, se a decisão de encaminhamento de um SA não for alterada, i.e. a decisão de encaminhamento é a mesma para os instantes $t = t^+$ e $t = t^-$ (antes da alteração do estado das ligações), então a divulgação do pacote de controlo é interrompida. No entanto, para as assinaturas dos caminhos $F2Fptt$ e (BKP, y) há aspectos subtis. Da secção 3.3.2 aprendemos que estes caminhos só podem ser utilizados um de cada vez. No caso de uma falha de ligação, mesmo que estas assinaturas mantenham a sua preferência para o destino D, o pacote de controlo tem sempre de ser enviado. Os teoremas 5 e 6 têm propriedades fortes, já que ambos combinados restringem o número de avisos na rede.

Teorema 7: Ciclos transitórios causados por inconsistência do pacote de controlo são contidos a um salto e ciclos de pacotes ocorrem no máximo uma vez entre esses dois encaminhadores (*Transient loops caused by control packet inconsistency are contained to one hop and packets loop at most one time between these two routers*).

Os autores do DTIA provam o teorema 7 com base nos teoremas anteriores. Se o estado das ligações se alterar, o teorema 5 garante que todos os SAs que são afectados em termos de acessibilidade são notificados. O teorema 6 também assegura que todos os SAs que alterem as decisões de encaminhamento são notificados. Contudo, ciclos transitórios podem ocorrer se um SA X processar um pacote de controlo mas o seu vizinho X_i não o processar, forçando os pacotes de dados a estarem em ciclo entre X e X_i. Depois do SA X processar o pacote de controlo p, envia-o para o SA X_i. O SA X_i pode ter enviado pacotes de dados por essa ligação, mas agora irá invalidar a ligação $X - X_i$. Se existirem

caminhos alternativos, os pacotes de dados irão utilizá-los; caso contrário os pacotes de dados serão descartados. Esta propriedade garante que o sistema irá permanecer estável, desde que ciclos transitórios sejam contidos num só salto.

Teorema 8: Assumindo que há um caminho alternativo para um destino D durante falhas, nenhum pacote de dados é perdido (*Assuming that there is an alternative path to destination D during failures, no data packets are lost*).

O teorema 8 garante que um pacote p é sempre entregue ao seu destino, já que os teoremas anteriores garantem que o pacote de dados só segue caminhos livres de ciclos durante as falhas. Além disso, os ciclos só existem no máximo a um salto durante falhas transitórios. Segundo os autores do DTIA [ABP08], um pacote p é descartado se não existirem caminhos válidos para o destino D, o que contradiz o pressuposto do teorema 8.

3.3.3 Engenharia de Tráfego

Esta secção abrange a contribuição desta dissertação - Engenharia de Tráfego Leve ao Nível do Inter-Domínio. A Engenharia de Tráfego (ET) é a distribuição do tráfego através de rotas para servir um objectivo específico.

Para aplicar engenharia de tráfego no DTIA ao nível de inter-domínio, é importante conhecer as possibilidades que o DTIA oferece. O uso de um sistema multi-caminho fornece a possibilidade de diferentes fluxos para o mesmo destino usarem diferentes caminhos, e até o mesmo fluxo pode usar mais do que um caminho. O problema da ET coloca-se para:

- Tráfego de saída - como dividir os fluxos pelas múltiplas rotas disponíveis;
- Tráfego de entrada - como influenciar o modo como os outros domínios dividem o tráfego que enviam para o SA em causa.

Vimos na secção 3.3.1 que o objectivo da camada de acessibilidade é calcular o conjunto de todos os caminhos válidos de um SA X para qualquer outro SA numa região r, denominado $P_r(X)$. Na secção 3.3.2 observámos que o protocolo de encaminhamento é realizado com base no $P_r(X)$.

A partir de $P_r(X)$ é possível construir uma tabela com todos os diferentes primeiros saltos para um SA de destino numa região r, denominado $R_r(X)$. $R_r(X)$ pode ser utilizado para implementar engenharia de tráfego, balanceamento da carga e ser utilizado por protocolos das camadas superiores.

É de notar que a tabela $R_r(X)$ pode conter vários primeiros saltos para um destino. Isto deve-se ao facto do DTIA ser um sistema multi-caminho e *multihomed*.

Propriedades da ET

Nos termos da engenharia de tráfego é importante saber o seguinte: dado um caminho válido de um SA X_1 a X_n, o caminho inverso X_n a X_1 também é válido. É de relembrar que todos os tipos de ligações são simétricas excepto $f2c$ e $c2f$.

Vamos considerar o caminho P de X_1 a X_n, $P = \{X_1, X_2, X_3, \ldots, X_{n-1}, X_n\}$. Seja $S(P)$ a assinatura de P, e $L(X_i, X_j)$ a etiqueta da ligação entre X_i e X_j nessa mesma direcção. Por exemplo, se X_j for o fornecedor de X_i então $L(X_i, X_j) = c2f$ e $L(X_j, X_i) = f2c$. Então $S(P) = (((\varepsilon \bigoplus L(X_{n-1}, X_n)) \bigoplus L(X_{n-2}, X_{n-1})) \bigoplus \ldots) \bigoplus L(X_1, X_2)$.

Consideremos agora o caminho inverso utilizando as mesmas ligações na ordem inversa: $rP = \{X_n, X_{n-1}, \ldots, X_3, X_2, X_1\}$. A sua assinatura é $S(rP) = (((\varepsilon \bigoplus L(X_2, X_1)) \bigoplus L(X_3, X_2)) \bigoplus \ldots) \bigoplus L(X_n, X_{n-1})$.

Isto significa que um SA pode calcular todos os caminhos válidos para todos os SAs, e pode do mesmo modo calcular os caminhos de outro SA para ele próprio, ao inverter os caminhos válidos. Resta a questão: existe algum caminho "inverso" válido que não tenha sido obtido ao inverter os caminhos válidos?

A resposta a esta pergunta é dada no capítulo seguinte.

Capítulo 4

Engenharia de Tráfego Leve ao Nível do Inter-Domínio

4.1 Introdução

Este capítulo tem como objectivo apresentar os algoritmos desenvolvidos nesta dissertação para se efectuar engenharia de tráfego usando a arquitectura DTIA. O capítulo 3 descreveu formalmente a arquitectura focando as técnicas e os protocolos usados. Descreveram-se técnicas para a construção de um mapa contendo todas as ligações válidas de um SA X para todos os outros SAs da rede, $P_r(X)$. Este mapa fornece, uma base forte para a aplicação da engenharia de tráfego.

Os algoritmos desenvolvidos nesta dissertação realizam engenharia de tráfego em linha. Os algoritmos usam pedidos de cooperação entre o SA inicial e SAs remotos. Os SAs cooperam para alterarem a distribuição do tráfego, com o intuito de evitar congestão na rede e equilibrar o tráfego na rede.

A lógica dos algoritmos é explicada com a ajuda de fluxogramas.

Os algoritmos são validados e o seu desempenho é quantificado usando o simulador de rede *ns-2* (*The Network Simulator 2*) [nsR10]. Para testar os algoritmos da engenharia de tráfego, foram efectuadas algumas alterações ao núcleo do simulador, explicadas na secção 4.5.

4.2 Princípios Gerais dos Algoritmos

Na secção 3.3.3 do capítulo anterior ficou a seguinte pergunta: existe algum caminho "inverso" válido que não tenha sido obtido ao inverter os caminhos válidos? A seguinte propriedade responde a essa pergunta:

Propriedade 1: Dados dois SAs A e B, não existe um caminho inválido P de A para B tal que o caminho inverso rP, de B para A, seja válido (*Given two ASes A and B there is no invalid path P from A-B such that the reverse path rP from B-A is valid*).

Prova: Em termos de álgebra de encaminhamento, isto significa que para cada caminho P, se $S(P) = \phi$, então $S(rP) = \phi$. A proposição pode ser demonstrada por contradição. Consideremos $P = \{X_1, X_2, X_3, \ldots, X_{n-1}, X_n\}$ e seja $S(X_i, X_n)$ a assinatura do caminho intermédio entre X_i e X_n (entre o salto i e o destino). Temos $S(P) = \phi$ se e só se para algum termo i, $S(X_i, X_n) \bigoplus L(X_{i-1}, X_i) = \phi$ com $i \in \{1, 2, 3, \ldots, n-2\}$ (desde que $S(X_{n-1}, X_n) \neq \phi$). Ou seja, se num caminho uma das assinaturas parciais estendida com a etiqueta seguinte for inválida. Seguindo o mesmo raciocínio, o caminho é válido para o mesmo termo i se e só se $S(X_n, X_i) \bigoplus L(X_i, X_{i-1}) \neq \phi$ com $i \in \{1, 2, 3, \ldots, n\}$. Examinando na tabela 3.3 os casos de assinaturas inválidas, existem seis casos. Vamos analisar cada um e encontrar as suas assinaturas inversas:

$S(1) = F2F \bigoplus f2c = \phi$. A assinatura inversa é $rS(1) = c2f \bigoplus rS(F2F)$ onde $rS(F2F)$ é a assinatura inversa do caminho calculado na direcção *para a frente* com assinatura $F2F$. Como existem enumeras combinações possíveis de ligações que formam um caminho $F2F$ (ver células nas colunas que não a primeira coluna da tabela 3.3), temos de obter todas da tabela 3.3, e invertê-las para encontrar todas as assinaturas possíveis para os caminhos inversos. Obtemos $rS(F2F) \in \{\varepsilon \bigoplus f2fptt \bigoplus f2f;\ \varepsilon \bigoplus f2f \bigoplus f2fptt;\ \varepsilon \bigoplus f2f \bigoplus c2f\}$ substituindo na expressão $rS(1)$ temos ϕ para todas as possibilidades.

Da mesma forma temos os outros cinco casos:
$S(2) = F2F \bigoplus f2f = \phi$, a assinatura inversa é $rS(2) = f2f \bigoplus rS(F2F)$ onde $rS(F2F) \in \{\varepsilon \bigoplus f2fptt \bigoplus f2f;\ \varepsilon \bigoplus f2f \bigoplus f2fptt;\ \varepsilon \bigoplus f2f \bigoplus c2f\}$ substituindo na expressão $rS(2)$ temos ϕ para todas as possibilidades.

$S(3) = F2F \bigoplus f2fbkp = \phi$, a assinatura inversa é $rS(3) = f2fbkp \bigoplus rS(F2F)$ onde

$rS(F2F) \in \{\varepsilon \bigoplus f2fptt \bigoplus f2f; \varepsilon \bigoplus f2f \bigoplus f2fptt; \varepsilon \bigoplus f2f \bigoplus c2f\}$ substituindo na expressão $rS(3)$ temos ϕ para todas as possibilidades.

$S(4) = C2F \bigoplus f2c = \phi$, a assinatura inversa é $rS(4) = c2f \bigoplus rS(C2F)$ onde $rS(C2F) \in \{\varepsilon \bigoplus f2fptt \bigoplus f2c; \varepsilon \bigoplus f2c \bigoplus f2c; \varepsilon \bigoplus f2c \bigoplus f2fptt; \varepsilon \bigoplus f2c \bigoplus c2f; \varepsilon \bigoplus f2c \bigoplus f2f; \varepsilon \bigoplus f2c \bigoplus f2fbkp;\}$ substituindo na expressão $rS(4)$ temos ϕ para todas as possibilidades.

$S(5) = C2F \bigoplus f2p = \phi$, a assinatura inversa é $rS(5) = f2f \bigoplus rS(C2F)$ onde $rS(C2F) \in \{\varepsilon \bigoplus f2fptt \bigoplus f2c; \varepsilon \bigoplus f2c \bigoplus f2c; \varepsilon \bigoplus f2c \bigoplus f2fptt; \varepsilon \bigoplus f2c \bigoplus c2f; \varepsilon \bigoplus f2c \bigoplus f2f; \varepsilon \bigoplus f2c \bigoplus f2fbkp;\}$ substituindo na expressão $rS(5)$ temos ϕ para todas as possibilidades.

$S(6) = (BKP, Y) \bigoplus f2f = \phi$, a assinatura inversa é $rS(6) = f2f \bigoplus rS(BKP, Y)$ onde $rS(BKP, Y) \in \{\varepsilon \bigoplus f2fbkp \bigoplus f2fptt; \varepsilon \bigoplus f2pbkp \bigoplus f2c; \varepsilon \bigoplus \{f2fbkp; c2f; f2f; f2fptt\} \bigoplus f2fbkp \bigoplus c2f; \varepsilon \bigoplus \{f2fbkp; c2f; f2f; f2fptt\} \bigoplus f2fbkp\}$ substituindo na expressão $rS(6)$ temos ϕ para todas as possibilidades.

Portanto, para cada caminho inválido na direcção *para a frente*, o caminho inverso é também inválido, o que conclui a nossa prova. Esta propriedade significa que o SA pode calcular todos os caminhos inversos de qualquer outro SA durante a execução dos protocolos de acessibilidade e encaminhamento.■

4.3 Modelo do Sistema

Nesta secção é apresentado o modelo do sistema que é usado para definir os algoritmos desenvolvidos na dissertação. O modelo define formalmente os saltos para um SA de destino e a carga existente nas ligações com os vizinhos directos.

Na secção 3.3.2 observamos que o protocolo de encaminhamento é realizado com base no $P_r(X)$, conjunto de todos os caminhos válidos de um SA X para qualquer outro SA numa região r. A partir de $P_r(X)$ é possível construir uma tabela com todos os diferentes primeiros saltos para todos os SAs de destino numa região r, denominado $R_r(X)$. É de notar

que a tabela $R_r(X)$ pode conter vários primeiros saltos para um destino. Isto deve-se ao facto do DTIA ser um sistema multi-caminho.

Os algoritmos de engenharia de tráfego são desenvolvidos com base na tabela $R_r(X)$. Determinam-se quais primeiros saltos recebem mais ou menos tráfego, de modo a evitar a congestão.

Para evitar a congestão numa ligação um SA tem de monitorizar as suas ligações. Chama-se à atenção que um SA só pode monitorizar as ligações com os seus vizinhos directos e que a informação a retirar é a carga que cada origem tem nessa ligação.

Com o objectivo de guardar a informação da monitorização de cada uma das ligações, definiu-se uma estrutura de informação, que interliga as origens das cargas com o respectivo destino. Um SA é constituído por n ligações L identificadas pelo número de SA do vizinho. O atributo guardado é o valor da carga, Γ, por origem. A tabela 4.1 ilustra a estrutura dos dados para as cargas existentes, num conjunto de ligações genéricas de um SA.

Tabela 4.1: Carga nas Ligações por SA.

Carga nas Ligações			
Ligação	Origens	Destinos	Cargas - Γ
L_1	$\{O_{11}, O_{12}, \ldots, O_{1m}\}$	$\{D_{11}, D_{12}, \ldots, D_{1m}\}$	$\Gamma = \{\gamma_{11}, \gamma_{12}, \ldots, \gamma_{1m}\}$
L_2	$\{O_{21}, O_{22}, \ldots, O_{2m}\}$	$\{D_{21}, D_{22}, \ldots, D_{2m}\}$	$\Gamma = \{\gamma_{21}, \gamma_{22}, \ldots, \gamma_{2m}\}$
$\ldots$	$\ldots$	$\ldots$	$\ldots$
L_n	$\{O_{n1}, O_{n2}, \ldots, O_{nm}\}$	$\{D_{n1}, D_{n2}, \ldots, D_{nm}\}$	$\Gamma = \{\gamma_{n1}, \gamma_{n2}, \ldots, \gamma_{nm}\}$

Para todos os algoritmos foram definidos os seguintes parâmetros a serem aplicados durante a análise de uma ligação sobrecarregada:

- $\Omega_X(O_i)$ - identifica o conjunto de vizinhos que fornecem dados de uma origem, O_i. $\Omega(O_i) = \{\omega_1, \omega_2, \ldots, \omega_n\}$ num SA X, sendo n o número de vizinhos que fornecem dados da origem;

- ψ - Limite da carga nas ligações para o qual as ligações são consideradas sobrecarregadas, valor definido pelo gestor da rede (em percentagem de largura de banda usada);

- $\Xi_X(i)$ - Conjunto de primeiros saltos do SA X para um SA i, $\Xi_X(i) = \{\xi_1, \xi_2, \ldots, \xi_n\}$, este conjunto é retirado da tabela $R_r(X)$.

Na figura 4.1 é dado um exemplo de uma rede, e o conteúdo da tabela $R_r(X)$ para o SA O_1 é apresentado na tabela 4.2. Considere-se que SA O_1 está a enviar tráfego para o SA D_1. Analisando a tabela 4.2 é possível verificar que o SA X define um ponto crítico no caminho para o SA D_1, que pode ser acedido através de duas ligações: entre V_1 e X; e entre V_2 e X. Caso a ligação entre V_1 e X fique sobrecarregada, o SA O_1 pode alterar o tráfego, pois existe uma alternativa pelo SA V_2. No entanto, se a ligação sobrecarregada for entre o SA X e o SA D_1, o SA O_1 não tem opções para alterar o tráfego.

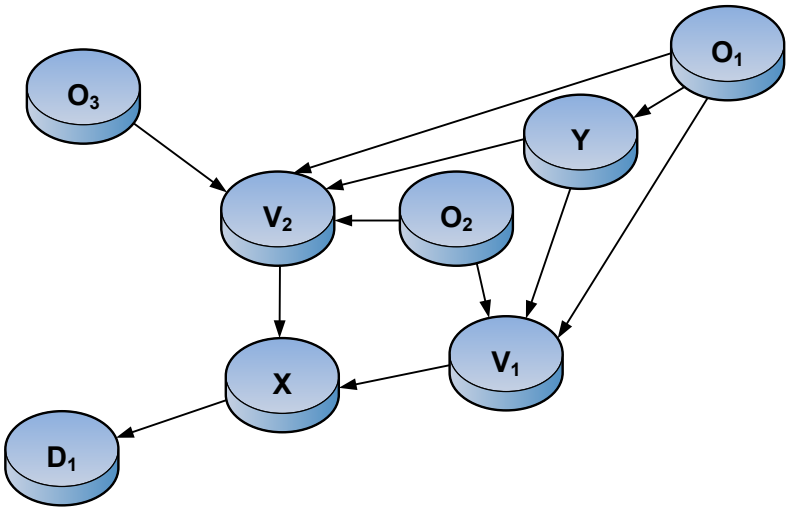

Figura 4.1: Exemplo de uma rede formada por SAs.

Tabela 4.2: Tabela $R_r(O_1)$.

Tabela $R_r(O_1)$	
Destino	Primeiros Saltos
V_1	$\{V_1, Y\}$
V_2	$\{V_2, Y\}$
Y	$\{Y\}$
X	$\{V_1, V_2, Y\}$
D_1	$\{V_1, V_2, Y\}$

O SA X tem dois vizinhos, V_1 e V_2, que lhe fornecem pacotes para um destino, SA D_1.

Num instante representado na tabela 4.3, pela sua ligação ao SA X o vizinho V_1 fornece 20% da capacidade da ligação com pacotes da origem O_1 e 15% com pacotes da origem O_2. Pelo seu lado, o SA V_2, para o mesmo instante t, fornece pela sua ligação ao SA X 30% da capacidade da ligação com pacotes da origem O_1, 20% com pacotes da origem O_2 e 20% com pacotes da origem O_3. O resultado da carga das ligações para o SA X é mostrado na tabela 4.3.

Tabela 4.3: Carga nas Ligações para o SA X.

Carga nas Ligações			
Ligação	Origens	Destinos	Cargas - Γ
V_1	$\{O_1, O_2\}$	$\{D_1, D_1\}$	$\Gamma = \{20, 15\}$
V_2	$\{O_1, O_2, O_3\}$	$\{D_1, D_1, D_1\}$	$\Gamma = \{30, 20, 20\}$

4.4 Algoritmos

Esta secção apresenta os algoritmos desenvolvidos nesta dissertação para a engenharia de tráfego a aplicar na arquitectura DTIA, conhecendo o estado das cargas nas ligações. Os algoritmos desenvolvidos são:

- Cálculo para uma origem - analisa as ligações sobrecarregadas e efectua a engenharia de tráfego com base na origem com maior carga na ligação;
- Cálculo para N origens - analisa as ligações sobrecarregadas e efectua a engenharia de tráfego com base nas origens com mais carga na ligação;
- Distribuição por classes - analisa as ligações sobrecarregadas e efectua a engenharia de tráfego com base na origem com mais carga da ligação. Calcula os caminhos inversos de um SA remoto, e divide os saltos, $\Xi(i)$, por classes.

A construção dos algoritmos foi orientada para resolver a congestão e o desequilíbrio do tráfego nas ligações. Para solucionar estes problemas foram estabelecidos os seguintes objectivos para os algoritmos:

1. Cooperação entre SAs.

2. Redução da quantidade de tráfego na ligação sobrecarregada distribuindo por caminhos alternativos.

3. O SA de destino tem prioridade em relação a outros SAs.

4. Os SAs vizinhos do SA onde teve origem a mensagem de engenharia de tráfego, têm prioridade em relação a outros SAs que não sejam o destino.

O ponto 1 estabelece que tem de existir cooperação entre os SAs, para ser possível realizar engenharia de tráfego inter-domínio.

O ponto 2 define que se um SA tem uma ligação de entrada sobrecarregada, e este tiver mais ligações de entrada para receber o tráfego existente na ligação sobrecarregada, deve pelo envio de mensagens de engenharia de tráfego tentar que outros SAs alterem o envio de tráfego para equilibrar a carga nas ligações de entrada existentes.

O ponto 3 define que se o SA de destino for um dos primeiros saltos na tabela de encaminhamento de um SA, e esta ligação, entre o SA e o destino não for a ligação sobrecarregada, então esta ligação deve receber mais tráfego que as restantes.

O ponto 4 estabelece uma relação semelhante ao demonstrado no ponto anterior, só que neste caso para os vizinhos do SA de origem da mensagem de ET. Se na tabela de encaminhamento um destes vizinhos for primeiro salto para o destino, e a ligação formada entre esse vizinho e o SA que enviou a mensagem não estiver sobrecarregada, então essa ligação deve receber mais tráfego que as restantes (com excepção da ligação para o destino).

Para a distribuição dos pacotes pelos saltos, $\Xi(i)$, são definidas três regras para cada um dos algoritmos. As duas primeiras regras são comuns aos três algoritmos, enquanto a terceira regra difere entre os dois primeiros algoritmos e o terceiro algoritmo. São definidas da seguinte forma:

1. Se o conjunto de saltos $\Xi(i)$ contiver o SA de destino, é-lhe atribuído um máximo de pacotes, ∇, parametrizável;

2. Se o conjunto de saltos $\Xi(i)$ contiver algum dos SAs da lista $\Omega_{X1}(O_i)$, são-lhes atribuídos 30% de ∇;

3. Para os dois primeiros algoritmos é admitida uma distribuição exponencial pelos primeiros saltos, dos pacotes a enviar para o destino, apresentada na secção 4.4.1; Para o terceiro algoritmo é utilizada para o envio dos pacotes uma distribuição por classes dos primeiros saltos, para o envio de pacotes, apresentada na secção 4.4.3.

Para uma leitura mais simples do texto nas secções seguintes, define-se τ como o vizinho na extremidade de uma ligação sobrecarregada; na análise de uma ligação sobrecarregada, τ identifica o vizinho pelo seu número de SA e é único durante o processo; e deve ser interpretado como o SA a evitar no encaminhamento de pacotes.

4.4.1 Carga nas Ligações - Cálculo para uma Origem

Esta secção apresenta uma primeira solução de engenharia de tráfego processando só uma origem dos pacotes existentes. Primeiro é dada uma ideia geral do funcionamento do algoritmo, seguido da introdução às estruturas de dados, terminando com uma explicação detalhada do funcionamento do algoritmo.

Funcionamento Geral

Assim que um SA genérico $X1$ detecta uma ligação sobrecarregada, este identifica a origem, O_i, com mais carga na ligação a partir da tabela 4.1. Com base na mesma tabela, é gerada a lista de vizinhos, $\Omega_{X1}(O_i)$. O SA $X1$ conclui a sua tarefa com o envio de uma mensagem de engenharia de tráfego (ET) para o SA a partir de onde tem origem a ligação sobrecarregada, designado de τ. Por sua vez, o SA τ reenvia a mensagem para outros SAs ($X2$, $X3$, ...) por onde passa o tráfego com origem em O_i.

Um SA V, quando recebe a mensagem, analisa a sua tabela $R_r(V)$ e verifica se tem outras opções para o envio do tráfego de forma a não passar por τ. Se tiver altera o encaminhamento de pacotes para o destino. Termina a sua tarefa com o envio de uma cópia da mensagem de engenharia de tráfego recebida, para o seu conjunto de SAs vizinhos, $\Omega_V(O_i)$. Este processo é executado tanto pelo vizinho da ligação sobrecarregada como pelos outros SAs que recebem uma mensagem de ET.

A divulgação da mensagem de ET termina quando esta chega à origem dos pacotes, O_i. O SA que detecta uma ligação sobrecarregada, só repete o envio de uma mensagem de ET devido a essa ligação quando o trio (destino, τ e O_i) não tenha sido objecto de aviso anterior.

Estruturas de Dados

Uma mensagem de engenharia de tráfego é trocada entre dois SAs para que o SA de destino da mensagem possa tentar efectuar alterações ao tráfego. A mensagem é constituída por vários atributos: destino dos pacotes de dados, Dp; origem dos pacotes de dados, O_i; o vizinho na ligação de $X1$ sobrecarregada a contornar, τ; e os vizinhos de $X1$ que fornecem dados da origem, Ω_{X1}. Tanto os dois SAs que trocam a mensagem como os presentes nos atributos são identificados pelo seu número de SA. A tabela 4.4 ilustra a estrutura dos dados, para uma mensagem genérica.

Tabela 4.4: Mensagem de Engenharia de Tráfego com uma Origem.

Mensagem de Engenharia de Tráfego com uma Origem			
Destino dos Pacotes de Dados	Origem dos Pacotes Dados	Vizinho na Ligação Sobrecarregada	Vizinhos - Ω_{X1}
Dp	O_i	τ	$\Omega = \{\omega_1, \omega_2, \ldots, \omega_n\}$

Um dos critérios para a distribuição dos pacotes pelos primeiros saltos para um SA de destino, Dp, é a distribuição exponencial. Para isso é necessário definir o peso de cada um dos saltos para Dp, que é obtido pela comparação dos conjuntos de primeiros saltos, $\Xi(Dp)$ e $\Xi(\tau)$. Um SA que pertença a ambos os conjuntos vê o seu peso ser incrementado. O destino e os primeiros saltos são identificados com os seus números de SA. Definiu-se uma estrutura de informação que interliga o destino dos pacotes com os saltos para Dp. O atributo guardado é o peso, P, atribuído a cada um dos SAs do conjunto $\Xi(Dp)$. A tabela 4.5 ilustra a estrutura dos dados, para um SA genérico.

Tabela 4.5: Peso por saltos, $\Xi(Dp)$.

Peso por saltos		
Destino	Saltos	Peso
D_1	$\Xi(D_1) = \{\xi_1, \xi_2, \ldots, \xi_n\}$	$\{P_1, P_2, \ldots, P_m\}$
D_2	$\Xi(D_2) = \{\xi_1, \xi_2, \ldots, \xi_n\}$	$\{P_1, P_2, \ldots, P_m\}$
...	...	...
D_n	$\Xi(D_n) = \{\xi_1, \xi_2, \ldots, \xi_n\}$	$\{P_1, P_2, \ldots, P_m\}$

Continuando com o exemplo apresentado na secção 4.3, a estrutura da mensagem enviada pelo SA X seria a dada pela tabela 4.6.

Tabela 4.6: Mensagem de Engenharia de Tráfego com uma Origem gerada pelo SA X.

Mensagem de Engenharia de Tráfego			
Destino dos Pacotes de Dados	Origem dos Pacotes de Dados	Vizinho na Ligação Sobrecarregada	Vizinhos - Ω_{X1}
D_1	O_1	V_2	$\Omega = \{V_1\}$

Consideremos que o SA Y recebe uma mensagem de engenharia de tráfego, e que esta tem os dados contidos na tabela 4.6. O SA Y tem os seguintes conjuntos: $\Xi(D_1) = \{V_1, V_2\}$ e $\Xi(V_2) = \{V_2\}$. O resultado da tabela com os pesos atribuídos aos saltos para o SA D_1 é dado pela tabela 4.7.

Tabela 4.7: Peso por saltos para o SA D_1.

Peso por saltos		
Destino	Saltos	Peso
D_1	$\Xi(D_1) = \{V_1, V_2\}$	$\{0, 1\}$

A figura 4.2 demonstra a divulgação da mensagem de engenharia de tráfego, inicialmente enviada para $\tau = V_2$. Numa segunda fase o SA V_2 envia uma cópia integral da mensagem de ET para os seus vizinhos, $\Omega_{V_2}(O_1)$. A divulgação termina com mais um envio da cópia integral da mensagem de ET, por parte do SA Y com destino ao SA O_1.

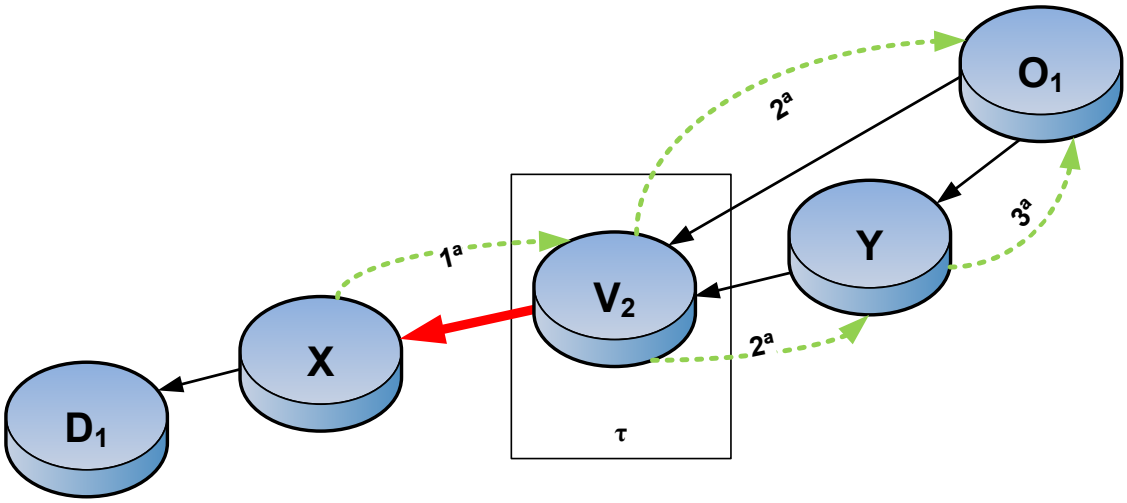

Figura 4.2: Divulgação das Mensagens de ET.

Funcionamento Detalhado

O algoritmo é descrito com a ajuda de fluxogramas. O processo é iniciado num SA $X1$, quando este tem uma ligação sobrecarregada com um dos seus vizinhos que lhe fornece dados, designado de τ. A partir da tabela 4.1 é identificada a origem mais relevante do tráfego, O_i. Se o trio (destino, τ e O_i) já tiver sido detectado na vez anterior, o processo termina. Se não, é obtido o conjunto dos vizinhos, $\Omega(O_i)$, e é enviada uma mensagem de engenharia de tráfego, para o SA, τ. A figura 4.3 apresenta o fluxograma deste processo.

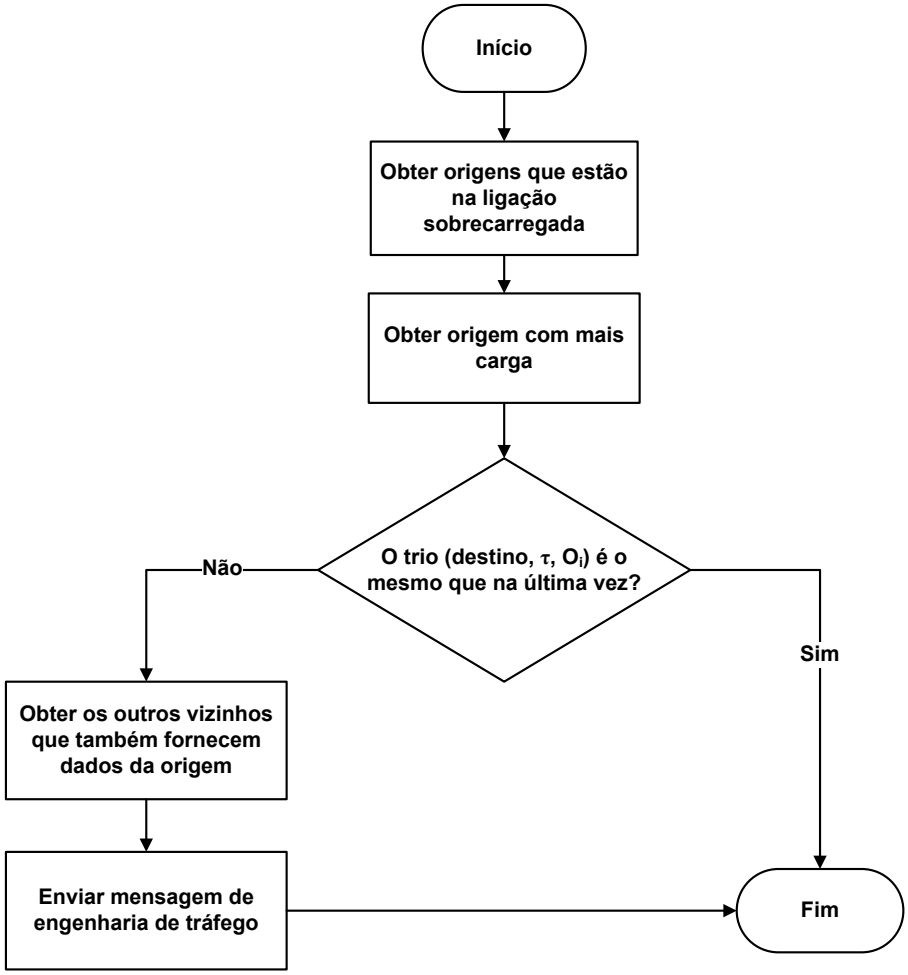

Figura 4.3: Processamento da Detecção de Ligação Sobrecarregada (Uma Origem).

O processo para a recepção e tratamento da mensagem de ET está representado na figura 4.4. Um SA k que seja o destino de uma mensagem de ET verifica se tem saltos para enviar o tráfego para Dp que não passem por τ. Se tiver, altera o envio de tráfego. Analisa a mensagem para verificar se o trio (destino, τ e O_i) se mantém o mesmo que na última mensagem recebida. Se for o mesmo, o processo termina. Se não, são calculados os vizinhos que fornecem dados da origem, $\Omega_k(O_i)$, e é enviada uma cópia da mensagem de engenharia de tráfego para esses vizinhos.

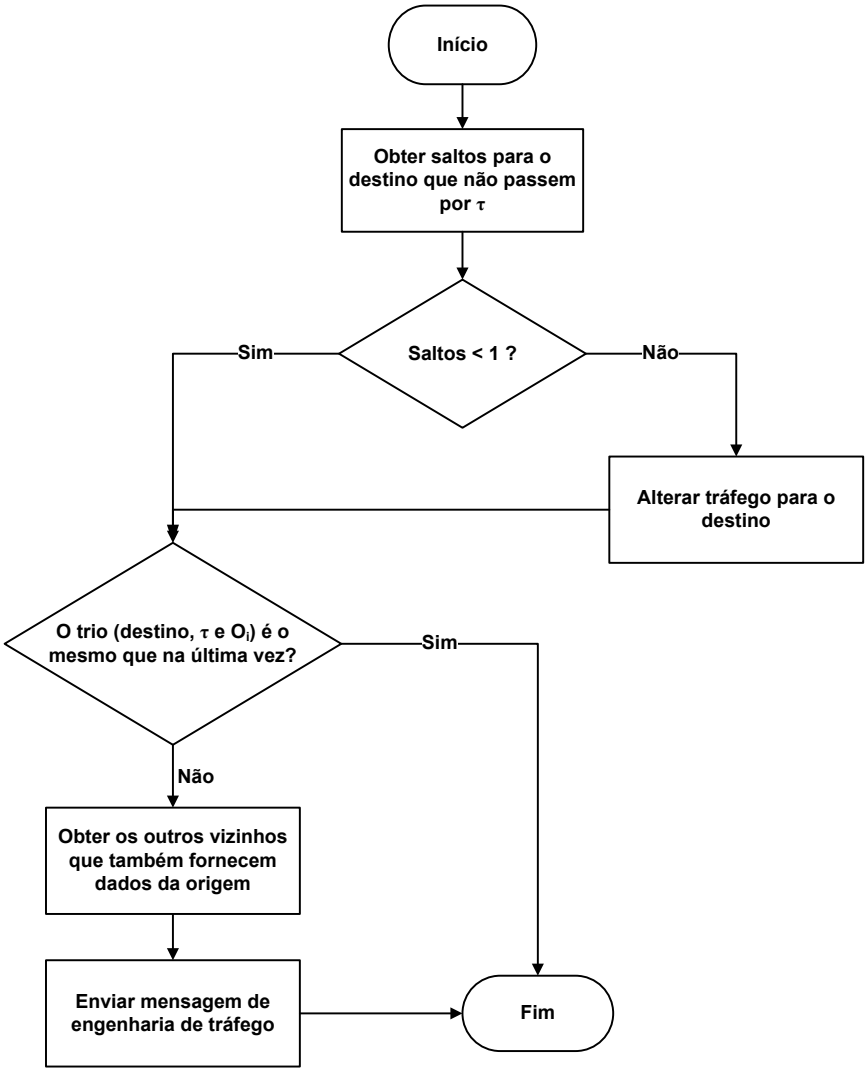

Figura 4.4: Processamento da Recepção da mensagem de ET (Uma Origem).

A alteração do envio de tráfego para o destino é executada de acordo com o fluxograma representado na figura 4.5: é verificado se o destino ou um dos vizinhos directos da origem da mensagem de engenharia de tráfego é um dos saltos para o destino. Se a verificação for positiva, são enviados pacotes segundo o máximo de pacotes ∇ para o salto que é o destino; no caso de ser um dos vizinhos da origem da mensagem, o envio para estes é 30% de ∇. Se o destino e os vizinhos directos não forem primeiros saltos, a distribuição do tráfego para o destino é executado segundo uma distribuição exponencial.

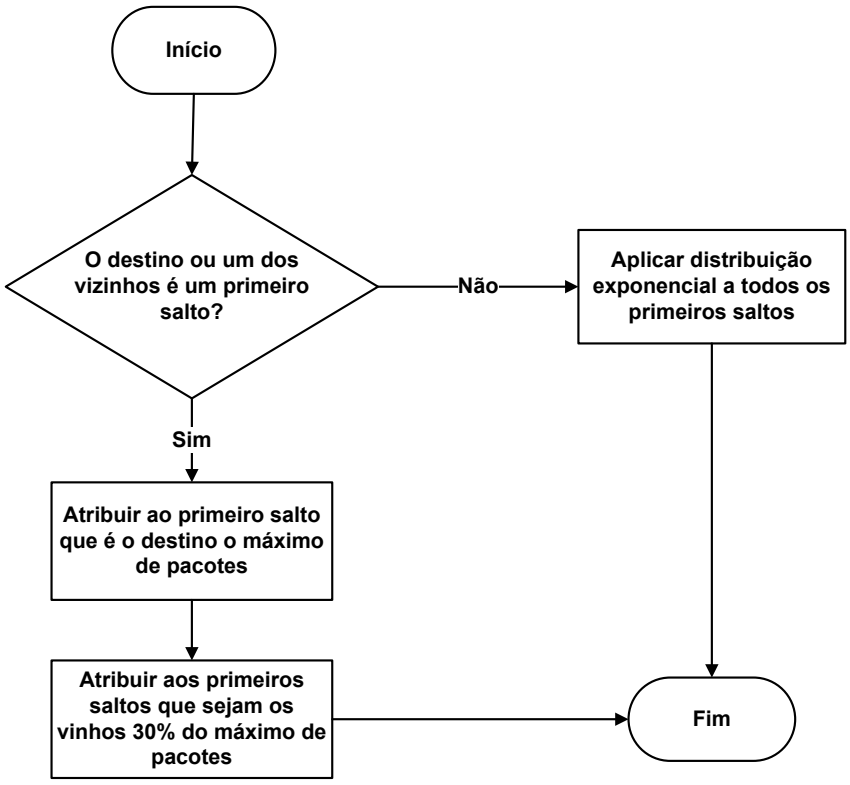

Figura 4.5: Alteração do Envio de Tráfego para um Destino.

A distribuição exponencial dos pacotes é efectuada tendo em conta o peso atribuído a cada um dos elementos de Ξ, apresentado na tabela 4.5, e segundo a seguinte fórmula:

$$Exp(SA_i) = \frac{e^{-P(\xi_i)}}{\sum_{k=1}^{n} e^{-P(\xi_k)}}$$

onde i é o SA a ser analisado e n é o número de saltos para o destino.

A escolha por uma distribuição exponencial foi tomada para que fosse possível agir mais bruscamente sobre os SAs, para que estes desviem o tráfego das ligações indesejadas.

Para terminar a análise do exemplo dado anteriormente na secção 4.4.1, observemos a

tabela 4.5. A distribuição de pacotes é executada para cada um dos primeiros saltos:

$$Exp(V_1) = \frac{e^{-0}}{e^{-0} + e^{-1}} = 0.73$$
$$Exp(V_2) = \frac{e^{-1}}{e^{-0} + e^{-1}} = 0.23$$

O encaminhamento de pacotes não utiliza percentagens mas sim a quantidade de pacotes a enviar por cada salto. Define-se então 1 como o número mínimo de pacotes que se pode enviar para um destino de modo a manter o multi-caminho, atribuindo-o ao salto com menor valor na distribuição exponencial. Assim obtêm-se a seguinte relação na distribuição de pacotes pelos saltos:

$$Pacotes(V_1) = \frac{0.73}{0.23} = 3$$
$$Pacotes(V_2) = \frac{0.23}{0.23} = 1$$

são encaminhados para o SA V_1 3 pacotes de cada vez enquanto que para o SA V_2 só é encaminhado 1 pacote de cada vez.

Para terminar a descrição deste algoritmo temos o envio de uma mensagem de *reset* com o trio (destino, τ e O_i). O envio desta mensagem é efectuado pelo mesmo SA $X1$ que detectou uma ligação sobrecarregada, e quando a carga da ligação (anteriormente sobrecarregada) toma valores inferiores a ψ. A divulgação desta mensagem segue os mesmo parâmetros que a mensagem de engenharia de tráfego. Resumindo, o SA que detectou a ligação sobrecarregada limpa a informação referente ao trio, e envia a mensagem de *reset* para τ, que por sua vez envia para os seus vizinhos $\Omega_\tau(O_i)$ e assim sucessivamente. Os SAs que receberem a mensagem de *reset* efectuam a mesma alteração nas suas bases de dados mas mantêm as alterações efectuadas no envio do tráfego.

4.4.2 Carga nas Ligações - Cálculo para *N* Origens

Esta secção apresenta a segunda solução de engenharia de tráfego processando N origens dos pacotes existentes. Primeiro é dada uma ideia geral do funcionamento do algoritmo,

seguida da introdução às estruturas de dados, terminando com uma explicação detalhada do funcionamento do algoritmo.

Para a selecção das N origens é necessário definir o critério que leva a essa escolha. Define-se ρ, em percentagem, como o limite de ocupação das ligações para cada origem. Ou seja, se uma origem tiver uma carga superior a ρ é uma das N origens a processar.

Funcionamento Geral

O funcionamento deste algoritmo é muito semelhante ao apresentado na secção 4.4.1. Assim que um SA $X1$ detecta uma ligação sobrecarregada, este identifica as origens, $O_{1..N}$, com carga superior a ρ, a partir da tabela 4.1. Com base na mesma tabela, gera a lista de vizinhos, $\Omega_{X1}(O_{1..N})$. O SA $X1$ conclui a sua tarefa com o envio de uma mensagem de engenharia de tráfego (ET) para o SA vizinho a partir de onde tem origem a ligação sobrecarregada, designado de τ. Esta mensagem é depois reenviada sucessivamente, até atingir as várias origens, $O_{1..N}$.

Um SA V quando recebe a mensagem, analisa a tabela $R_r(V)$ e verifica se tem outras opções para o envio do tráfego de forma a não passar por τ. Se tiver altera o encaminhamento de pacotes para o destino.

Para cada uma das N origens é calculado o corresponde conjunto de vizinhos, $\Omega_V(O_i)$. Conhecendo os conjuntos $\Omega_V(O_{1..N})$ é possível identificar as origens comuns para cada vizinho. Então para cada vizinho é criado uma mensagem de engenharia de tráfego semelhante à mensagem recebida, sendo a diferença o conjunto de origens. O processo termina com o envio das mensagens de ET para os vizinhos, cada uma delas com o correspondente conjunto de origens comuns ao vizinho. Este processo é executado tanto pelo vizinho da ligação sobrecarregada como pelos outros SAs que recebem uma mensagem de ET.

A divulgação da mensagem de ET termina quando esta chega às origens dos pacotes. O SA que detecta uma ligação sobrecarregada, só repete o envio de uma mensagem de ET devido a essa ligação quando o trio (destino, τ e o conjunto de origens, $O_{1..N}$) não for o mesmo.

Estruturas de Dados

A mensagem de engenharia de tráfego trocada entre dois SAs no algoritmo com cálculo para N origens é muito parecida com a do algoritmo apresentado na secção 4.4.1. Estas diferem no segundo atributo, que em vez de uma origem tem agora N origens de dados. Os atributos guardados são: destino dos pacotes de dados, Dp; os SA origem dos pacotes de dados, $O_{1..N}$; o SA vizinho na ligação de $X1$ sobrecarregada a contornar, τ; e os SAs vizinhos de $X1$ que fornecem dados da origem, Ω_{X1}. A tabela 4.8 ilustra a estrutura dos dados, para uma mensagem genérica.

Tabela 4.8: Mensagem de Engenharia de Tráfego com N Origens.

Mensagem de Engenharia de Tráfego com N Origens			
Destino dos Pacotes de Dados	Origens dos Pacotes Dados	Vizinho na Ligação Sobrecarregada	Vizinhos - Ω_{X1}
Dp	$O_{1..N} = \{O_1, O_2, \ldots, O_n\}$	τ	$\Omega = \{\omega_1, \omega_2, \ldots, \omega_n\}$

Com base na figura 4.1 e na tabela 4.3 considere-se o seguinte exemplo: para um $\rho = 20\%$, o SA X detecta que a ligação com o SA V_2 está sobrecarregada. A estrutura da mensagem enviada pelo SA X para τ, SA V_2, seria a dada pela tabela 4.9.

Tabela 4.9: Mensagem de Engenharia de Tráfego com N Origens gerada pelo SA X.

Mensagem de Engenharia de Tráfego com N Origens			
Destino dos Pacotes de Dados	Origens dos Pacotes Dados	Vizinho na Ligação Sobrecarregada	Vizinhos - Ω_{X1}
D_1	$O_{1..N} = \{O_1, O_2, O_3\}$	V_2	$\Omega = \{V_1\}$

Por sua vez, o SA V_2 enviaria as mensagens de ET para os seus vizinhos com o conteúdo apresentado na tabela 4.10.

Tabela 4.10: Mensagem de Engenharia de Tráfego com N Origens gerada pelo SA V_2.

	Mensagem de Engenharia de Tráfego com N Origens			
Destino da mensagem de ET	Destino dos Pacotes de Dados	Origens dos Pacotes Dados	Vizinho na Ligação Sobrecarregada	Vizinhos - Ω_{X1}
O_3	D_1	$O_{1..N} = \{O_3\}$	V_2	$\Omega = \{V_1\}$
O_1	D_1	$O_{1..N} = \{O_1\}$	V_2	$\Omega = \{V_1\}$
Y	D_1	$O_{1..N} = \{O_1, O_2\}$	V_2	$\Omega = \{V_1\}$
O_2	D_1	$O_{1..N} = \{O_2\}$	V_2	$\Omega = \{V_1\}$

Funcionamento Detalhado

O algoritmo é descrito com a ajuda de fluxogramas. O processo é iniciado num SA $X1$, quando este tem uma ligação sobrecarregada com um dos seus vizinhos que lhe fornece dados, designado de τ. A partir da tabela 4.1 são identificadas as origens mais relevantes do tráfego, $O_{1..N}$, ou seja as origens com carga superior a ρ. Se o trio (destino, τ e $O_{1..N}$) já tiver sido detectado na vez anterior, o processo termina. Se não, é obtido o conjunto dos vizinhos, $\Omega_X(O_{1..N})$, e é enviada uma mensagem de engenharia de tráfego, para o SA, τ. A figura 4.6 apresenta o fluxograma deste processo.

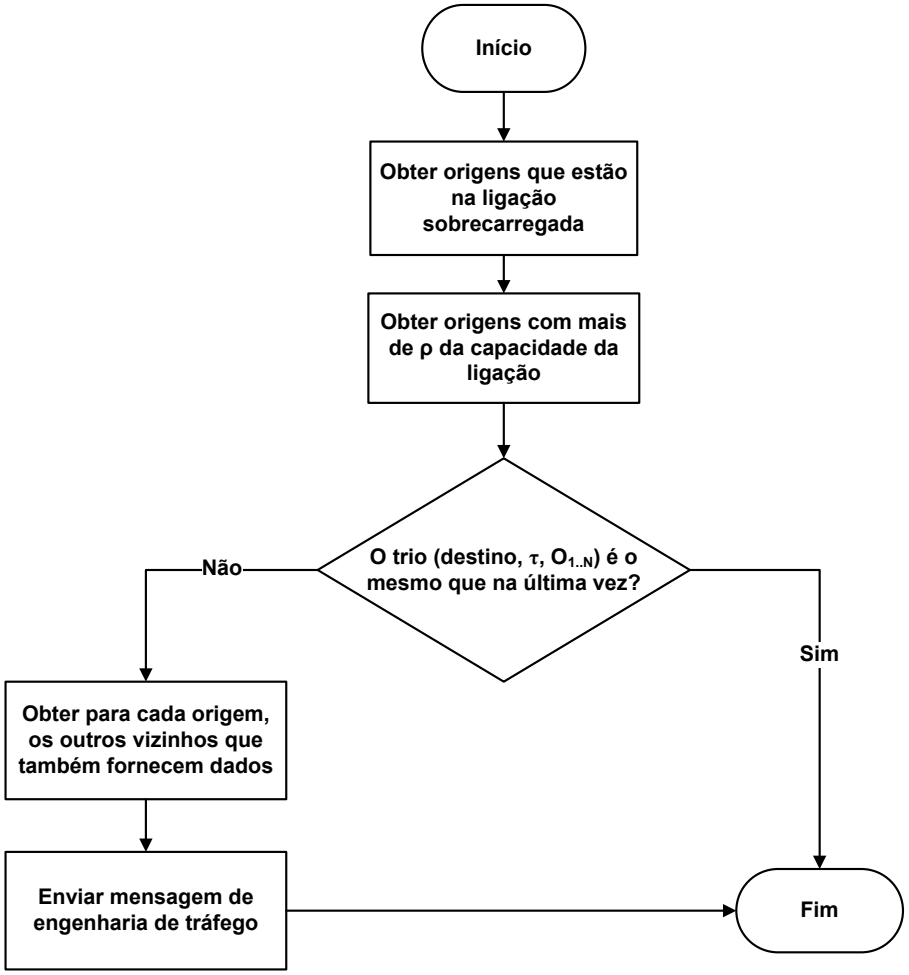

Figura 4.6: Processamento da Detecção de Ligação Sobrecarregada (N Origens).

O processo para a recepção e tratamento da mensagem de ET está representado na figura 4.7. Um SA k que seja o destino de uma mensagem de ET verifica se tem saltos para enviar o tráfego para Dp que não passem por τ. Se tiver, altera o envio de tráfego. Analisa então a mensagem para verificar se o trio (destino, τ e origens $O_{1..N}$) se mantém o mesmo que na última mensagem recebida. Se for o mesmo, o processo termina. Se não, são determinados para cada uma das N origens os conjuntos de vizinhos, $\Omega_k(O_i)$. Conhecendo estes conjuntos é possível identificar as origens comuns para cada vizinho, e enviar para cada um deles uma mensagem de engenharia de tráfego semelhante à mensagem recebida, alterando apenas o conjunto de origens.

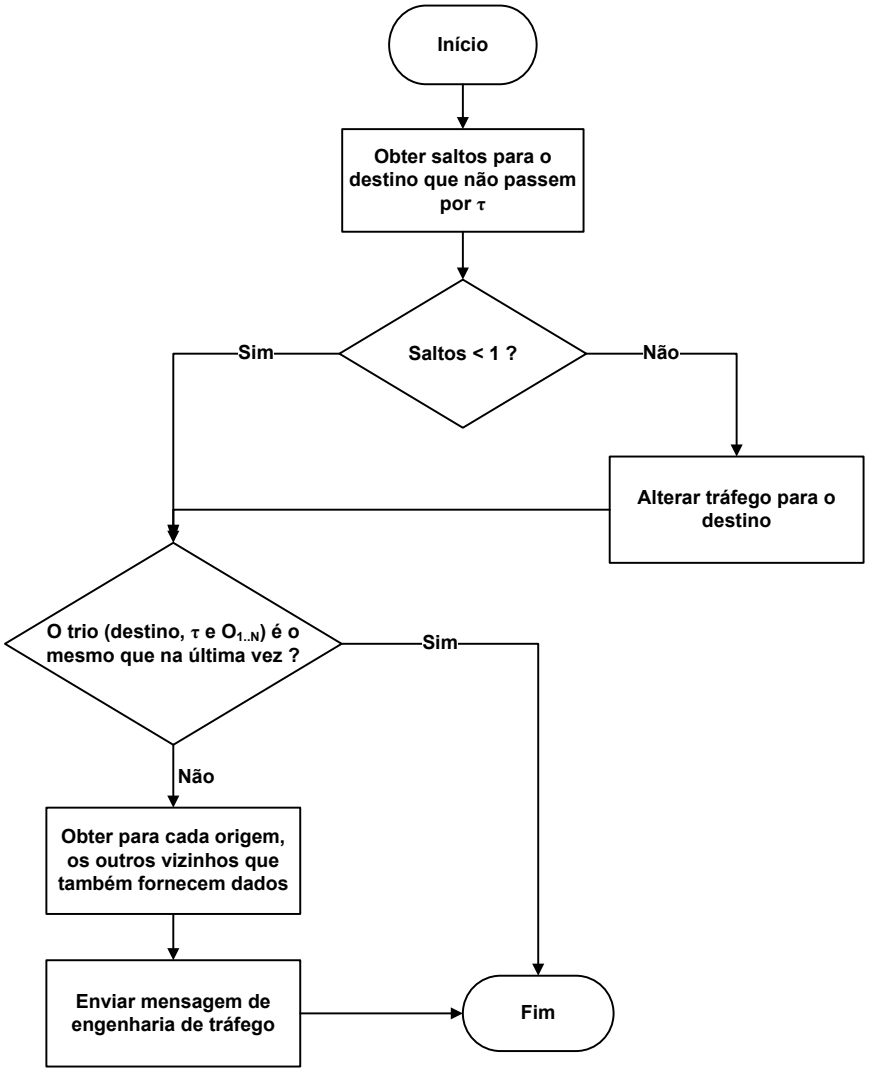

Figura 4.7: Processamento da Recepção da mensagem de ET (N Origens).

A alteração do envio do tráfego para o destino é executado de acordo com o fluxograma representado na figura 4.5 da secção 4.4.1.

Como no algoritmo - cálculo para uma origem, também neste é utilizado uma mensagem de *reset*. O envio da mensagem de *reset* é efectuado pelo mesmo SA $X1$ que detectou uma ligação sobrecarregada e quando a carga da ligação toma valores inferiores a ψ. A divulgação desta mensagem segue os mesmos parâmetros que a mensagem de engenharia de tráfego deste algoritmo. Os SAs que recebem este tipo de mensagem apagam das suas bases de dados a informação correspondente ao trio (destino, τ e origens $O_{1..N}$) contido na mensagem mas mantêm as alterações efectuadas no envio do tráfego.

4.4.3 Carga nas Ligações - Distribuição por Classes

Esta secção apresenta a terceira solução de engenharia de tráfego proposta, que processa só uma origem dos pacotes existentes e distribui os diferentes primeiros saltos para a entrega do tráfego por classes de SAs. É de salientar que neste algoritmo os cálculos são efectuados pelo SA que tem uma ligação sobrecarregada, analisando a rede a partir da origem até ao próprio SA. Os destinos das mensagens de engenharia de tráfego são unicamente os SAs que podem efectuar alterações no seu envio de pacotes.

Para que seja o SA que identifica uma ligação sobrecarregada com um dos seus vizinhos a executar a análise da rede do ponto de vista dos outros SAs, é necessário conhecer os caminhos que os SAs remotos têm para o SA local, ou seja, os caminhos inversos. Na secção 3.3.3 provámos que era possível conhecer esses caminhos.

Este algoritmo utiliza um conjunto de cinco classes de SAs para classificar como bons ou menos bons saltos, os primeiros saltos que um SA remoto tem para um SA $X1$. Um dos passos para a construção dessas classes é a verificação se os primeiros saltos são ou não saltos para os vizinhos, que fornecem dados de uma origem O_i ao SA $X1$. Um salto que seja opção para um vizinho de $X1$ que mantém uma ligação com pouca carga recebe um valor positivo; caso seja opção para um vizinho com uma ligação sobrecarregada recebe um valor negativo. Esta verificação é executada para todos os primeiros saltos que um SA remoto tem para o SA $X1$, classificando-os consoante o estado actual das ligações de $X1$.

As classes de SAs são formadas por quatro classes onde os primeiros saltos são classificados por um conjunto de condições e uma quinta classe extra onde são colocados os SAs considerados capazes de efectuarem grandes alterações no envio do tráfego, como por exemplo, os vizinhos directos do destino ou o próprio destino.

Funcionamento Geral

Assim que um SA genérico $X1$ detecta uma ligação sobrecarregada, identifica a origem, O_i, com mais carga na ligação a partir da medição de tráfego local, mantida numa tabela equivalente à tabela 4.1. Com base na mesma tabela, é gerada a lista de vizinhos, $\Omega_{X1}(O_i)$. O SA $X1$ calcula os caminhos que o SA O_i tem para ele, guardando todos os primeiros saltos, $\Xi_{O_i}(X1)$. Os saltos $\Xi_{O_i}(X1)$ são distribuídos pelas classes de SAs consoante se são primeiros saltos para os vizinhos, $\Omega_{X1}(O_i)$, e tendo em conta a carga existente nas ligações de $X1$ com cada um dos SAs de $\Omega_{X1}(O_i)$.

Se após a distribuição dos primeiros saltos $\Xi_{O_i}(X1)$ pelas classes de SAs, existir mais do que uma classe com SAs significa que o SA O_i pode alterar o tráfego. Neste caso é enviada uma mensagem de engenharia de tráfego (ET) para o SA O_i. Se existir só uma classe com os primeiros saltos, O_i não tem opções para alterar o tráfego. São então verificados os caminhos inversos que os saltos, $\Xi_{O_i}(X1)$ têm para $X1$, e é iniciado o mesmo processo de análise que se teve para o SA O_i, para todos os SAs pertencentes a $\Xi_{O_i}(X1)$. Caso também falhem, passa-se para os segundos saltos. O processo de detecção do(s) SA(s) que podem alterar o envio do tráfego, termina quando é encontrado um ou vários SAs nos caminhos entre O_i e $X1$ que possam alterar o tráfego, ou quando o SA a analisar é o SA $X1$ e não tiver sido detectado nenhum SA capaz de alterar o tráfego nos caminhos entre O_i e $X1$.

Caso o SA $X1$ detecte um SA com capacidade para alterar o tráfego, envia-lhe uma mensagem de ET com a distribuição dos saltos pelas classes de SAs. Quando um SA recebe uma mensagem de ET, altera a distribuição do tráfego pelos primeiros saltos com base nas classes de SAs indicadas na mensagem de ET.

Métodos de Decisão

Este algoritmo utiliza a mesma estrutura referida na secção 4.4.1, para o registo da carga nas ligações, que está representada na tabela 4.1, com o mesmo objectivo: identificar a origem com maior carga na ligação O_i e os vizinhos que fornecem dados dessa origem.

Um SA $X1$ só conhece a taxa de ocupação das ligações que mantém com os SAs vizinhos.

Após ter identificado a origem, O_i, o SA $X1$ gera a lista de vizinhos, $\Omega_{X1}(O_i)$. Para cada um dos SAs de $\Omega_{X1}(O_i)$, $\omega_V(O_i)$, é gerado um índice de carga com base na carga da ligação $\omega_V(O_i) - X1$.

Para classificar a carga nas ligações entre os SAs de $\Omega_{X1}(O_i)$ e o SA $X1$, define-se um índice de carga, que apenas pode ter um conjunto discreto de valores. Cada valor do índice de carga está associado a um intervalo de valores de taxa de ocupação da ligação. O espaço de valores para a taxa de ocupação foi dividido em n intervalos de igual dimensão ($\frac{1}{n}$), que pode ser definido pelo conjunto $\Delta = \{[\delta_0..\delta_1[, [\delta_1..\delta_2[, \ldots, [\delta_{n-1}..\delta_n]\}$, onde $\delta_i = 100 \times \frac{i}{n}$ com $i \in [0..n]$ onde n é um número par.

Cada intervalo do conjunto Δ de níveis de taxas de ocupação é mapeado directamente num índice de carga, λ_j. O índice de carga λ_j está associado ao intervalo $[\delta_{j-1}..\delta_j[$ e tem o valor numérico

$$\lambda_j = \begin{cases} \frac{n}{2} - (j-1) & \text{se } j \leq \frac{n}{2} \\ \frac{n}{2} - j & \text{se } j > \frac{n}{2} \end{cases}$$

com $j \in [1..n]$.

Deste modo é possível associar um índice de carga a cada uma das ligações $\omega_V(O_i) - X1$. Associa-se um índice de carga λ_1 com valor positivo $\frac{n}{2}$ a uma carga no intervalo $[\delta_0..\delta_1[$ indicando uma ligação com pouca carga, e simetricamente, para uma ligação com muita carga, associa-se ao intervalo $[\delta_{n-1}..\delta_n]$ o índice de carga λ_n (um valor negativo $-\frac{n}{2}$).

Conhecido o índice de carga associado às ligações $\omega_V(O_i) - X1$, define-se um nível de importância para os SAs na distribuição de tráfego, que apenas pode ter um conjunto discreto de valores, ζ. O nível de importância é determinado pela soma dos índices de carga das ligações $\Omega_{X1}(O_i) - X1$. Sempre que um SA do conjunto $\Xi_{O_i}(X1)$ é um salto para um dos SAs vizinhos, $\omega_V(O_i)$, é incrementado ao seu nível de importância o índice de carga associado à ligação $\omega_V(O_i) - X1$. Os valores do nível de importância de cada um dos primeiros saltos são guardados numa tabela equivalente à tabela 4.5. Com o nível de importância associado a cada salto do conjunto $\Xi_{O_i}(X1)$ é possível separá-los pelas classes de SAs.

Para distribuir os primeiros saltos $\Xi_{O_i}(X1)$ por nível de importância na distribuição do tráfego, define-se um conjunto de classes de SAs, Υ, composto por quatro classes de SAs, $\Upsilon = \{\upsilon_1, \upsilon_2, \upsilon_3, \upsilon_4\}$. A cada classe υ_k, com $k \in [1..4]$, atribuem-se os SAs cujo os níveis de importância pertencem ao intervalo de valores definido para essa classe. O valor do nível de importância associado a um SA tem um valor máximo quando todas as ligações estão com pouca carga, i.e. $M = (\textit{número de vizinhos} - 1) * \lambda_1$; tem um valor mínimo se estão todas saturadas, i.e. $m = (\textit{número de vizinhos} - 1) * (-\lambda_1)$. Desta forma, a dimensão de cada intervalo é $\mu = \frac{M-m}{4}$. O intervalo de nível de importância de uma classe de SAs é dado por:

$$\begin{cases}]+\infty, (M-\mu)[& \text{se } k = 1 \\ ](M-\mu\times(k-1)), (M-\mu\times k)[& \text{se } k \in [2..3] \\ ](M-\mu\times(k-1)), -\infty[& \text{se } k = 4 \end{cases}$$

Como exemplo, consideremos agora que $n = 8$, originando um conjunto de índices de carga que varia entre $\lambda_1 = 4$ e $\lambda_8 = -4$. Retomemos a figura 4.1 com a carga das ligações para o SA X dada pela tabela 4.3. Pela tabela 4.3 seria determinada a origem dos pacotes, $O_i = O_1$, e verificar-se-ia que o SA X tem 2 vizinhos, o SA V_1 e o SA V_2, respectivamente com os índices de carga, 2 e -2.

Os parâmetros para a definição dos intervalos dos níveis de importância das classes de SAs são os seguintes: $\lambda_1 = 4$; $M = 4$; $m = -4$; e $\mu = 2$; estabelecendo-se assim os intervalos para as classes de SAs: $\upsilon_1 :]+\infty, 2[$, $\upsilon_2 : [2, 0[$, $\upsilon_3 : [0, -2[$ e $\upsilon_4 : [-2, -\infty[$.

A origem O_1 tem três primeiros saltos, $\Xi_{O_1}(X) = \{V_1, V_2, Y\}$, sendo os níveis de importância atribuídos a esses saltos pelo SA X para a distribuição de tráfego dados pela tabela 4.11.

Tabela 4.11: Graus de importância por saltos para o SA O_1.

Peso por saltos		
Destino	Saltos	Peso
D_1	$\Xi(O_1) = \{V_1, V_2, Y\}$	$\{2, -2, 0\}$

Desta forma, os primeiros saltos seriam distribuídos pelas classes de SAs do seguinte modo: $\upsilon_1 = \{\}$; $\upsilon_2 = \{V_1\}$; $\upsilon_3 = \{Y\}$; e $\upsilon_4 = \{V_2\}$. Ao existir mais do que uma classe

com SAs seria gerada uma mensagem de ET para o SA O_1.

É de salientar a existência da quinta classe (classe extra), que pode conter os números de SA do destino ou dos seus vizinhos. Os fundamentos para a definição desta classe são os dois últimos pontos dos objectivos definidos para estes algoritmos, como demonstrado na secção 4.4.

Estruturas de Dados

Conhecidos os métodos de decisão para a construção das quatro classes de SAs por parte de um SA $X1$, apresenta-se em seguida as estruturas de dados da mensagem de engenharia de tráfego (ET) e da estrutura onde são guardados os próximos saltos a analisar após a origem, O_i.

Um SA $X1$ detecta uma ligação sobrecarregada e inicia o processo de análise na origem, O_i, com mais carga na ligação. Se o SA O_i for capaz de alterar o tráfego, os primeiros saltos $\Xi_{O_1}(X1)$ são distribuídos pelas quatro classes de SAs, Υ. Uma mensagem de ET é então enviada do SA $X1$ para o SA O_i. A mensagem é constituída por vários atributos: destino dos pacotes de dados, Dp; origem dos pacotes de dados, O_i; o vizinho na ligação de $X1$ sobrecarregada a contornar, τ; as quatro classes de SAs; e a quinta classe (classe extra). Tanto os dois SAs que trocam a mensagem como os SAs presentes nos atributos são identificados pelo seu número de SA. A tabela 4.12 ilustra a estrutura de dados, para uma mensagem de engenharia de tráfego (ET) genérica.

Tabela 4.12: Mensagem de Engenharia de Tráfego para Distribuição por Classes.

Mensagem de Engenharia de Tráfego para Distribuição por Classes				
Destino dos Pacotes de Dados	Origens dos Pacotes Dados	Vizinho na Ligação Sobrecarregada	Classes	Classe Extra
Dp	O_i	τ	$\Upsilon = \{\upsilon_1, \upsilon_2, \upsilon_3, \upsilon_4\}$	$\{SA_1, SA_2, \ldots, SA_n\}$

Após enviar a mensagem de ET para o SA O_i, o SA $X1$ analisa o conjunto de classes Υ, que enviou na mensagem de ET. Inicia a analisa na classe υ_1 e continua até encontrar uma classe que contenha SAs, e é esse conjunto de SAs que são os próximos SAs a serem

analisados. Para manter o registo dos próximos SAs a serem analisados define-se uma estrutura de dados com os seguintes atributos: destino dos pacotes de dados, Dp; origem dos pacotes de dados, O_i; o vizinho na ligação de $X1$ sobrecarregada a contornar, τ; e os próximos SAs a analisar, Pr_{SAs}. A tabela 4.13 ilustra a estrutura dos dados, de um conjunto genérico de próximos SAs a processar para um SA $X1$.

Tabela 4.13: Próximos SAs a Processar.

Próximos SAs a Processar			
Destino dos Pacotes de Dados	Vizinho na Ligação Sobrecarregada - τ	Origem dos Pacotes de Dados - O_i	Próximos SAs a Processar
D_1	V_1	O_1	$Pr_{SAs} = \{Pr_1, Pr_2, \ldots, Pr_m\}$
D_2	V_1	O_1	$Pr_{SAs} = \{Pr_1, Pr_2, \ldots, Pr_m\}$
D_2	V_2	O_2	$Pr_{SAs} = \{Pr_1, Pr_2, \ldots, Pr_m\}$
$\ldots$	$\ldots$	$\ldots$	$\ldots$
D_n	V_n	O_n	$Pr_{SAs} = \{Pr_1, Pr_2, \ldots, Pr_m\}$

O cálculo dos próximos SAs a serem analisados vem da selecção da primeira classe com SAs porque esses SAs são considerados os mais importantes na distribuição do tráfego, porque são os SAs que têm capacidade para receber mais tráfego. Assim na próxima iteração serão analisados os SAs que podem fazer uma alteração mais significativa no encaminhamento do tráfego.

No exemplo dado na secção 4.4.3 seriam obtidas as classes de SAs $\upsilon_1 = \{\}$, $\upsilon_2 = \{V_1\}$, $\upsilon_3 = \{Y\}$, $\upsilon_4 = \{V_2\}$ e a origem dos pacotes, O_1. Continuando a análise desse exemplo, identificar-se-ia o vizinho na ligação de X sobrecarregada a contornar, $\tau = V_2$ e o destino dos pacotes, $Dp = D_1$. Determinadas as classes de SAs, Υ, o SA X enviaria uma mensagem de ET para o SA O_1, com a estrutura dada pela tabela 4.14. O SA X terminaria o processo com a actualização dos próximos SAs a serem analisados, que é o SA da classe υ_2, por ser a melhor classe calculada por X, como mostra a tabela 4.15.

Tabela 4.14: Mensagem de Engenharia de Tráfego para Distribuição por Classes do SA X para o SA O_1.

Mensagem de Engenharia de Tráfego para Distribuição por Classes				
Destino dos Pacotes de Dados	Origens dos Pacotes Dados	Vizinho na Ligação Sobrecarregada	Classes	Classe Extra
D_1	O_1	V_2	$\Upsilon = \{\{\}, \{V_1\}, \{Y\}, \{V_2\}\}$	$\{\}$

Tabela 4.15: Próximos SAs a Processar pelo SA X.

Próximos SAs a Processar			
Destino dos Pacotes de Dados	Vizinho na Ligação Sobrecarregada - τ	Origem dos Pacotes de Dados - O_i	Próximos SAs a Processar
D_1	V_2	O_1	$Pr_{SAs} = \{V_1\}$

Note-se que não foi tida em conta a quinta classe, a classe extra. Os SAs que a constituem são considerados capazes de efectuarem grandes alterações no envio do tráfego. Por isso não estão sujeitos às mesmas condições de selecção que os SAs das outras classes, e assim não foram incluídos na explicação das estruturas anteriores. A selecção desses SAs é apresentada em pormenor mais à frente.

Conhecido o procedimento por parte de um SA $X1$ após detectar uma ligação sobrecarregada, resta agora analisar como é alterado o envio do tráfego por um SA remoto após receber a mensagem de ET com as classes de SAs.

Métodos de Controlo

Inicialmente os SAs da rede definem para todos os primeiros saltos de qualquer destino, uma classe actual e um número de pacotes a enviar para cada um dos saltos. A classe actual é classificada inicialmente como pertencente à classe υ_4, considerada a pior classe, e é estabelecido para o número de pacotes a enviar o menor número possível, um pacote.

O SA remoto que recebe uma mensagem de engenharia de tráfego, recebe cinco classes com SAs determinadas por um SA $X1$, que indicam uma possível alteração do tráfego. O SA remoto analisa a classe extra, e se esta contiver SAs atribui um máximo de pacotes

Δ ao SA que é o destino, caso este exista no conjunto, e 30% de Δ aos restantes SAs que representam os SAs vizinhos do destino. Esta análise é efectuada de modo a alcançar os dois últimos pontos dos objectivos definidos para os algoritmos, apresentados na secção 4.4. No caso da classe extra estar vazia, o SA remoto analisa as quatro primeiras classes recebidas e altera o envio do tráfego para cada um dos primeiros saltos, com base na classe actual do salto e na classe recebida que contém o salto. Os SAs que pertencem à classe v_1 foram considerados como melhores opções para a distribuição do tráfego, porque são primeiros saltos para as ligações menos congestionadas, enquanto que os SAs da classe v_4 foram considerados as piores opções para o encaminhamento de tráfego por serem saltos para a ligação congestionada e/ou para as ligações com mais carga. Definem-se as regras para a alteração das classes dos primeiros saltos na tabela 4.16.

Tabela 4.16: Alteração das Classes para os Primeiros Saltos.

		Classe Recebida			
		1	2	3	4
Classe Actual	1	1	2	3	4
	2	1	1	3	3
	3	2	2	4	4
	4	1	2	3	4

Se a classe actual de um salto for a classe 1 ou 4, é alterada para classe recebida, para que seja possível uma resposta rápida a picos de tráfego ou a congestionamento de ligações. Se a classe actual for a classe 2 ou 3, só é alterada para classes adjacentes, para que suavemente se possa chegar a uma classe *ideal* para os saltos.

Feita a alteração das classes de cada primeiro salto, a alteração do número de pacotes a enviar para cada um dos saltos é efectuada do mesmo modo que a alteração das classes, com base na classe actual e na classe recebida que continha o salto. Se um salto transitar de uma classe pior para uma classe melhor (por exemplo da classe 4 para a classe 2) passa a receber mais tráfego, por ser considerado uma melhor opção para o encaminhamento do tráfego. Se ocorrer o oposto (por exemplo um salto passar da classe 1 para a 2) passa a receber menos tráfego. Definem-se as regras para a alteração do número de pacotes, a enviar para cada um dos primeiros saltos, na tabela 4.17.

Tabela 4.17: Alteração do Envio de Pacotes para os Primeiros Saltos.

		Classe Recebida			
		1	2	3	4
Classe Actual	1	+2	-2	-4	-6
	2	+2	+2	-2	-2
	3	+2	+2	-2	-2
	4	+6	+4	+2	-2

O número mínimo de pacotes possíveis a enviar para um salto é um pacote e número máximo de pacotes a enviar é ∇, tal como foi definido na secção 4.4.

Como exemplo considere-se a figura 4.1 onde o SA O_1 recebe uma mensagem de ET do SA X com a informação dada pela tabela 4.14. A alteração das classes e do número de pacotes a enviar para os primeiros saltos que o SA O_1 tem para o destino D_1 seria a seguinte:

- Para o SA V_1: a classe actual é a 4 e a classe recebida é a 2, então a classe seria alterada de $4 \rightarrow 2$ e o número de pacotes de $1 \rightarrow 5$;
- Para o SA Y: a classe actual é a 4 e a classe recebida é a 3, então a classe seria alterada de $4 \rightarrow 3$ e o número de pacotes de $1 \rightarrow 3$;
- Para o SA V_2: a classe actual é a 4 e a classe recebida é a 4, então a classe não seria alterada nem o número de pacotes porque o mínimo é um pacote.

Funcionamento Detalhado

O algoritmo é descrito com a ajuda de fluxogramas. O processo é iniciado num SA $X1$, quando este tem uma ligação sobrecarregada com um dos seus vizinhos que lhe fornece dados, designado de τ. A partir da tabela 4.1 é identificada a origem mais relevante do tráfego, O_i. Se o trio (destino, τ e O_i) já tiver sido detectado anteriormente, é retirado da tabela 4.13 com os próximos SAs a serem analisados. Se o trio ainda não tiver sido detectado, o próximo SA a analisar é $Pr_{SA} = O_i$. Para cada um dos SAs a analisar, Pr_{SA}, são distribuídos os seus primeiros saltos, $\Xi_{Pr_{SA}}(X1)$, para $X1$ pelas classes de SAs, $\Upsilon_{Pr_{SA}}$. Se Pr_{SA} tiver opções para alterar o tráfego é-lhe enviado uma mensagem de engenharia de tráfego. O processo termina com a actualização dos próximos SAs a serem analisados.

A figura 4.8 apresenta o fluxograma deste processo.

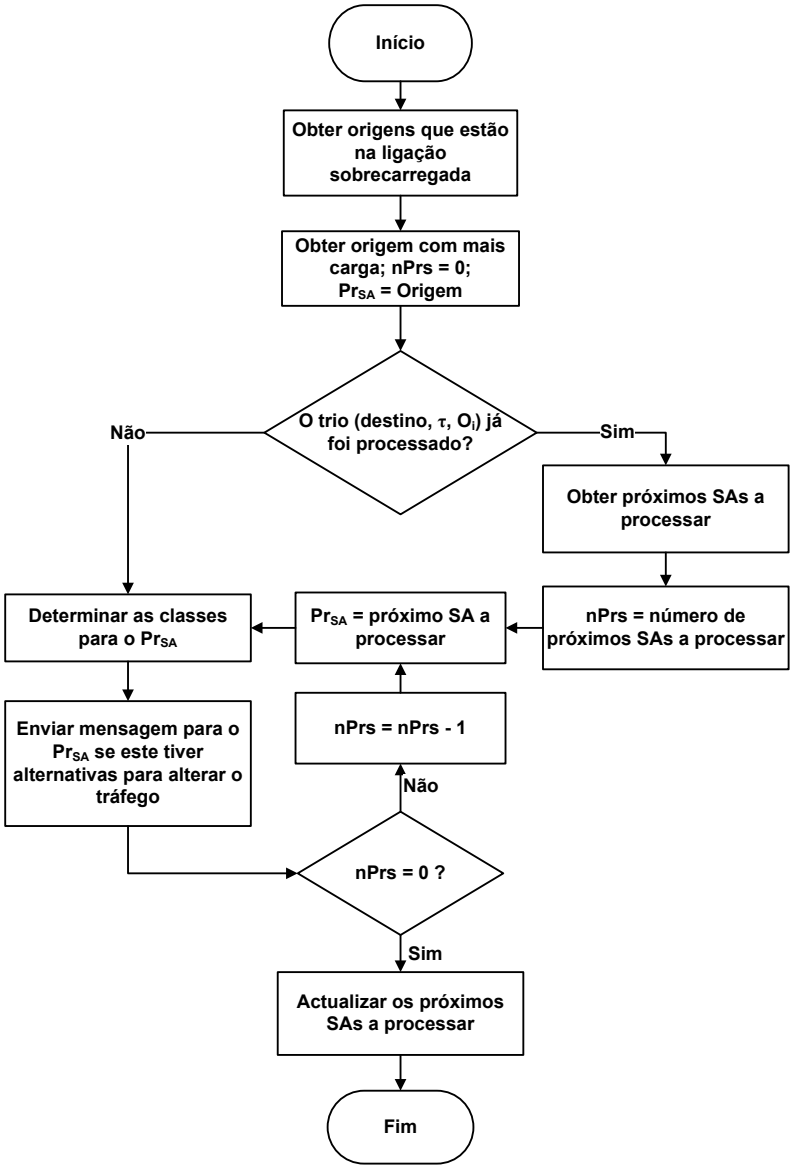

Figura 4.8: Processamento de Detecção de Ligação Sobrecarregada (Distribuição por Classes).

Os próximos SAs a serem analisados são os SAs que formam a melhor classe atribuída a cada um dos Pr_{SA} analisados, pois são os SAs contidos na melhor classe, que irão receber mais tráfego em relação aos outros saltos. Sendo assim, são estes SAs que no futuro podem fazer uma alteração mais significativa na distribuição do tráfego.

A figura 4.9 apresenta o fluxograma da distribuição dos primeiros saltos pelas cinco classes definidas, tendo em conta um primeiro salto genérico, Pr_{SA}. Inicialmente é criada uma lista dos vizinhos do SA $X1$, $\Omega_{X1}(O_i)$, que fornecem dados da origem, O_i, pela análise da tabela 4.1. Analisando os caminhos inversos de $Pr_{SA} - X1$, obtém-se a lista de primeiros saltos, $\Xi_{Pr_{SA}}(X1)$, para o SA $X1$. É então iniciada a distribuição dos SAs de $\Xi_{Pr_{SA}}(X1)$ da seguinte forma e pela seguinte ordem:

1. Se um dos primeiros saltos for o destino, esse salto é remetido para a classe extra;
2. Distribuição dos primeiros saltos, por estado das ligações dos vizinhos, $\Omega_{X1}(O_i)$;
3. Afastar o mais possível as classes com SAs (o funcionamento deste processo é apresentado em pormenor mais à frente);
4. Se o SA local for o destino: Os seus vizinhos que sejam primeiros saltos são remetidos para a classe extra.

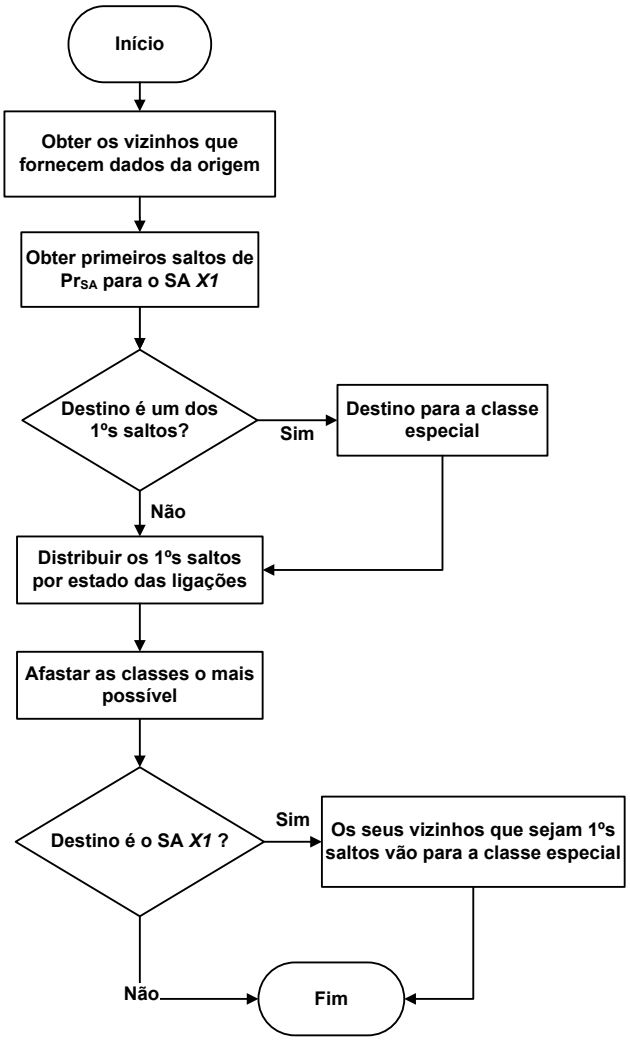

Figura 4.9: Distribuição dos Primeiros Saltos pelas Classes.

A figura 4.10 apresenta o fluxograma da distribuição dos primeiros saltos, $\Xi_{Pr_{SA}}(X1)$, consoante o estado das ligações dos vizinhos, $\Omega_{X1}(O_i)$. Para cada um dos saltos $\xi_S(X1)$ é verificado se também é primeiro salto para algum dos vizinhos, $\omega_V(O_i)$; sempre que for é incrementado ao seu nível de importância, ζ_{ξ_S}, o índice de carga associado à ligação $\omega_V(O_i) - X1$. O processo termina com a distribuição dos primeiros saltos, $\Xi_{O_i}(X1)$, pelas quatro classes.

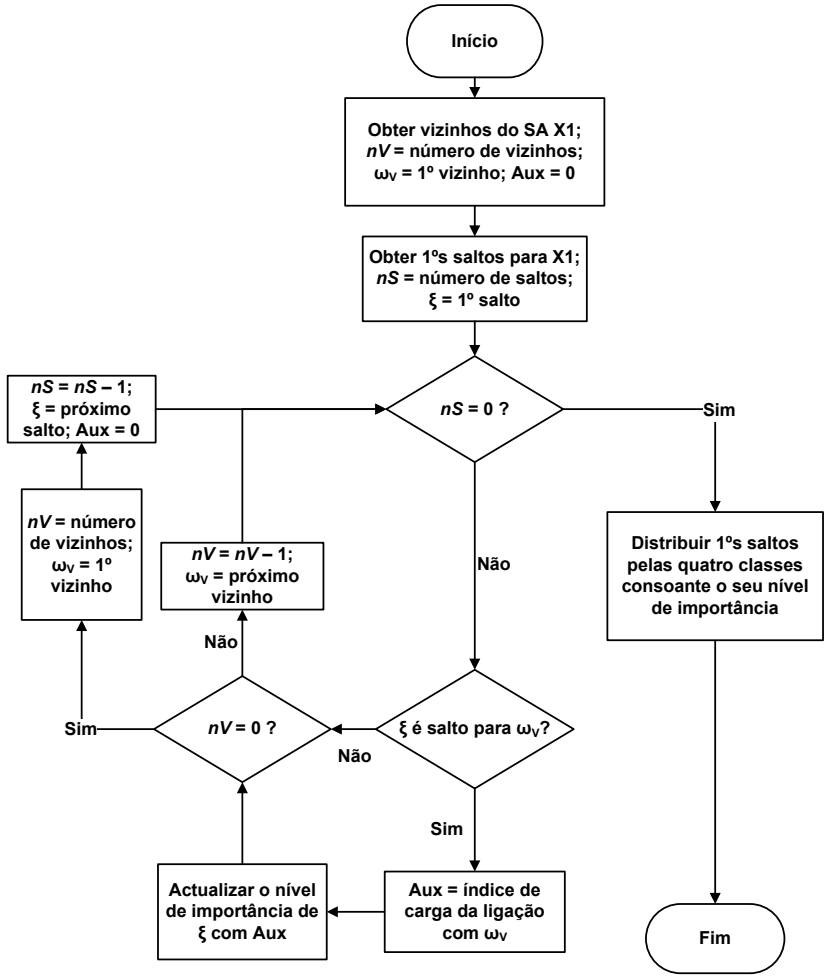

Figura 4.10: Distribuição dos Primeiros Saltos Consoante o Estado das Ligações.

A figura 4.11 apresenta o fluxograma do processo afastar o mais possível as classes com SAs. O processo inicialmente contabiliza o número de classes com SAs. Se só existir uma classe com SAs significa que não há opções para a alteração do tráfego e o processo termina. Se existir mais do que uma classe com SAs, estas são afastadas o mais possível umas das outras. As classes 1 e 4 se estiverem vazias são preenchidas respectivamente com os SAs da melhor e pior classe calculada até esse momento. As classes 2 e 3 são limpas se o seu conteúdo for remetido para outra classe.

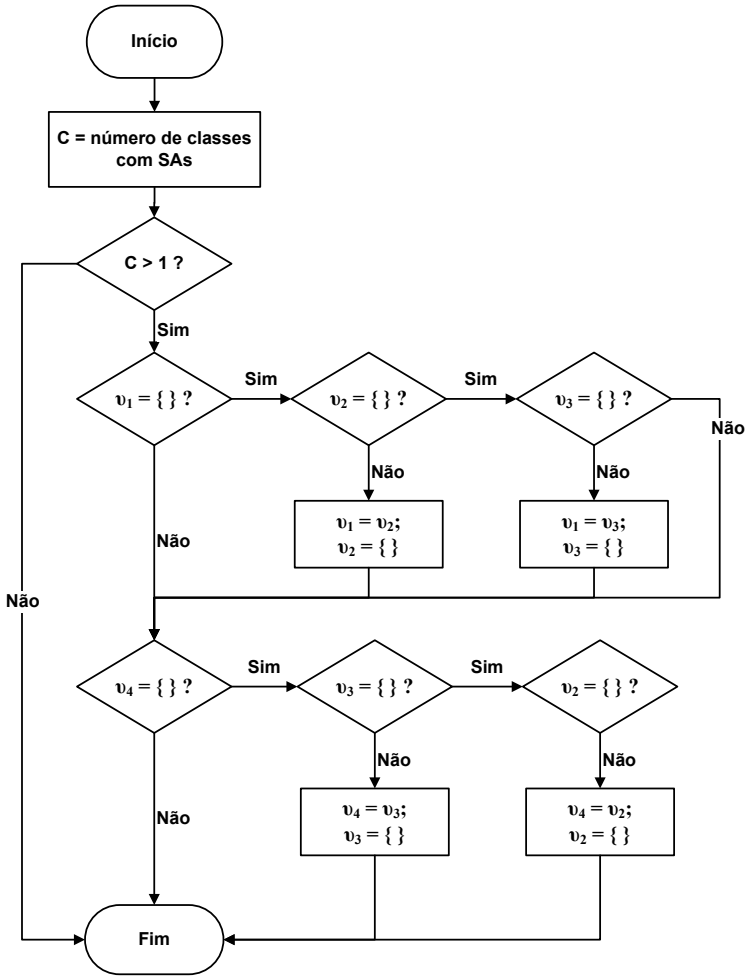

Figura 4.11: Afastar o mais Possível as Classes com SAs.

Um SA recebe uma mensagem de engenharia de tráfego quando pode alterar o envio do tráfego. Este SA verifica se a classe extra da mensagem contém SAs. Se tiver, passa a enviar o máximo de pacotes, ∇, para esses primeiros saltos. Se não, altera o envio de pacotes para os SAs contidos no atributo, classes, da mensagem com base nas tabelas 4.16 e 4.17.

Se o SA que recebe a mensagem ET tiver primeiros saltos para o destino que não estão nos atributos, classes e classe extra, significa que esses saltos são opções para o destino mas não o são para o SA que detectou a ligação sobrecarregada. Assim, esses SAs são considerados como SAs pertencentes à melhor classe, υ_1, e o envio do tráfego para eles é feito segundo as regras das tabelas 4.16 e 4.17.

4.4.4 Comparação dos Algoritmos

Esta secção tem como objectivo fornecer uma comparação dos algoritmos apresentados. Esta comparação foca: a complexidade na execução dos algoritmos, que SAs é que executam os cálculos, como é efectuada a divulgação das mensagens da engenharia de tráfego, e termina com as vantagens e desvantagens que cada algoritmo apresenta. A tabela 4.18 apresenta a comparação dos algoritmos.

Tabela 4.18: Comparação dos Algoritmos.

	Algoritmos		
	Cálculo para uma Origem	Cálculo para N Origens	Distribuição por Classes
Complexidade	Baixa	Baixa/Média	Alta
Quem executa os cálculos	O SA que recebe uma mensagem de ET.	O SA que recebe uma mensagem de ET.	O SA que tem uma ligação sobrecarregada.
Divulgação da mensagem de ET	Inundação Selectiva - Só para os SAs vizinhos que fornecem dados da origem em causa.	Inundação Selectiva - Só para os SAs vizinhos que fornecem dados das origens em causa.	Selectiva - Só para os SAs que podem efectuar alterações no envio do tráfego.
Vantagens	• Processamento reduzido por SA que recebe um mensagem de ET.	• Processamento reduzido por SA que recebe um mensagem de ET.	• Número reduzido de SAs a fazerem cálculos. • Número reduzido de mensagens de ET. • Análise com base na carga de todas as ligações que fornecem dados de uma origem.
Desvantagens	• Número elevado de SAs a fazerem cálculos. • Número elevado de mensagens de ET. • Análise só com base na carga da ligação sobrecarregada.	• Número elevado de SAs a fazerem cálculos. • Número elevado de mensagens de ET. • Análise só com base na carga da ligação sobrecarregada.	• Processamento elevado por SA que detecta uma ligação sobrecarregada.

Os algoritmos podem ser divididos em dois grupos: o primeiro grupo formado pelo algoritmo com cálculo para uma origem dos dados e pelo algoritmo com cálculo para N origens. Estes dois algoritmos são muito semelhantes, apresentando para todos os pontos de comparação, à excepção da complexidade, os mesmos resultados. Diferem na complexidade porque no algoritmo com cálculo para N origens a construção das mensagens de ET é mais elaborada; uma mensagem para um SA vizinho é construída consoante as origens dos dados que esse SA vizinho fornece, e não com todas as N origens.

O segundo grupo é formado pelo algoritmo com distribuição por classes. Os dois grupos apresentam grandes diferenças entre eles. O algoritmo com distribuição por classes é mais complexo, porque o SA que detecta a ligação sobrecarregada analisa os caminhos que a origem dos dados tem para ele, e distribui os saltos dos SAs remotos por classes consoante a carga das ligações que tem com os seus vizinhos. Esta complexidade do algoritmo faz com que este possa realizar uma melhor análise do estado da rede, e assim, aplicar uma engenharia de tráfego mais robusta e reduzir bastante o número de mensagens de ET. O primeiro grupo apresenta a vantagem do processamento em cada SA ser reduzido mas com um grande custo em termos de sinalização.

4.5 O Simulador de Rede 2 (*The Network Simulator 2 (ns-2)*)

Esta secção apresenta o simulador de rede 2 [nsR10], e as alterações introduzidas para implementar os algoritmos de engenharia de tráfego propostos nesta tese. O *ns-2* é uma ferramenta valiosa para os investigadores testarem protocolos de redes, com ou sem fios. Este simulador dá uma base adequada para modificar ou criar mecanismos em cada camada do modelo OSI [osi10]. O programa é de código aberto, o que permite aos utilizadores uma grande flexibilidade para o modificar e corrigir alguns das suas falhas. No entanto, a utilização deste software implica um grande conhecimento dos seus mecanismos. Nesta secção assume-se que o leitor tem conhecimentos básicos dos mecanismos[3] do *ns-2*.

[3]Para uma leitura completa do manual do software ver [nsM10]. Um tutorial rápido está também disponível na página de Marc Greis [nsT10].

Os mecanismos do *ns-2* são suportados através do *Tcl* [tcl10], uma linguagem de *scripting*, e da linguagem de programação $C++$. O Tcl é utilizado para a definição dos ficheiros de configuração, mas também como uma interface para executar comandos sobre os objectos de $C++$ que são mapeados no *Tcl*. Quanto à linguagem $C++$, é usada para definir os mecanismos dos protocolos. Como exemplo, um ficheiro em *Tcl* pode executar um comando que dispara um evento para avisar que uma ligação está sobrecarregada. Esse evento pode invocar uma rotina em $C++$ para advertir a camada da engenharia de tráfego que uma ligação está sobrecarregada. Como consequência, o algoritmo da engenharia de tráfego irá proceder aos cálculos necessários para inverter a situação da ligação.

Para testar um cenário de uma rede, o *ns-2* lê um ficheiro de simulação em *Tcl* e realiza uma simulação em tempo discreto. Para cada protocolo, independente da sua camada, um objecto *Tcl* é criado para cada nó. Este objecto é geralmente designado como *agente* e que serve como parâmetro para os pacotes. A interacção entre as camadas é suportada por um módulo especial, chamado de *classificador*. Um *classificador* também é utilizado para outros fins, tais como o encaminhamento de pacotes para outros nós.

Esta tese utiliza o trabalho realizado por Francisco Ganhão na sua tese de mestrado [Gan09], na qual implementou o protocolo de encaminhamento multi-região - DTIA em *ns-2*.

Os comandos Tcl são usados em cada uma das ligações entre dois SAs, para definir qual o agente de encaminhamento está associado a essa ligação. As próximas linhas apresentam um exemplo de um ficheiro *Tcl* a definir um agente de encaminhamento para uma ligação.

1. $ns duplex-link $n(i) $n(j) largura-banda atraso tipo-lista
2. set link [$ns link $n(i) $n(j)]
3. set rtobj [$n(j) rtObject?]
4. set rtproto [$rtobj rtProto ? DTIA]
5. $link add-link-agent rtProto $rtproto(j)

A primeira linha de código cria uma ligação nos dois sentidos entre os nós i e j. A segunda linha de código obtém o objecto da ligação do nó i para o nó j. A terceira linha de

código é usada para obter o objecto de encaminhamento de um nó, que contém todos os agentes do protocolo de encaminhamento que o nó está a usar. A quarta linha de código é usada para obter o protocolo de encaminhamento - DTIA. Por fim, a quinta linha de código interliga o objecto da ligação com o agente de encaminhamento do nó final, j, dessa ligação.

Para se simular a detecção do aviso de ligação sobrecarregada, foi usado um comando Tcl na lista de espera da ligação para se monitorizar o seu estado. O comando interliga a lista de pacotes com o protocolo de encaminhamento, para que este possa remeter o estado da ligação para a camada da engenharia de tráfego. É definido da seguinte forma:

1. $rtproto(j) cmd set-link-status id-vizinho estado-lista

O comando invoca no objecto do protocolo de encaminhamento a função $set-link-status$ com a identificação do vizinho com o qual é estabelecida a ligação e a percentagem da ocupação da ligação.

Outra mudança incluída no *ns-2*, foi a alteração no número de pacotes a enviar para cada um dos primeiros saltos de um destino. A alteração é efectuada no *classificador*. A camada da engenharia de tráfego decide qual o novo número de pacotes a atribuir a um primeiro salto e passa essa informação para a camada de encaminhamento, pois é esta camada que conhece o objecto do nó. As próximas linhas apresentam um exemplo de um ficheiro Tcl a definir um número de pacotes para um próximo salto.

1. $Obj-Nó setPathWeight $Id-Destino $Id-Prox-Salto $N-Pacotes
2. $Classificador($Id-Destino) cmd setPathWeight $Id-Prox-Salto $N-Pacotes

A primeira linha, executada na camada de encaminhamento, invoca no objecto do nó um procedimento com a seguinte informação: identificação do nó de destino; identificação do próximo salto para esse destino; e o novo número de pacotes que devem ser enviados para o próximo salto. Este passo é necessário pois é na classe *Nó* que está a interface do objecto do *classificador* para cada destino. O comando $setPathWeight$ ordena o *classificador* do destino, $\$Classificador(\$Id-Destino)$, a executar a alteração do envio de

pacotes para o próximo salto, $\$Id - Prox - Salto$, estabelecendo um novo número de pacotes, $\$N - Pacotes$, para esse salto.

Capítulo 5

Análise de Resultados

5.1 Introdução

Este capítulo apresenta uma análise do desempenho dos algoritmos desenvolvidos. Esta análise baseia-se na comparação do desempenho da arquitectura DTIA, sem engenharia de tráfego e com engenharia de tráfego. Salienta-se também a comparação entre algoritmos relativamente ao número de mensagens de engenharia de tráfego utilizado.

A secção 5.2 apresenta a topologia testada nas simulações. Esta topologia é baseada em dados da CAIDA [cai]. Depois, a secção 5.3 avalia o desempenho relativamente às seguintes métricas:

- Número de pacotes entregues versus os não entregues;
- Percentagem de utilização das ligações consideradas relevantes;
- Número de mensagens de engenharia de tráfego enviadas.

5.2 Características da Topologia

Apresentamos agora a topologia que serviu de base às experiências efectuadas. A topologia foi definida a partir de dados do Projecto de Investigação do Relacionamento entre SAs do CAIDA [cai]. Foram seleccionados 54 SAs e 517 ligações para a nossa investigação. A topologia contém um sub-conjunto formado por SAs terminais de Portugal, e um conjunto de SAs de trânsito que eles usam até aos *tier-1*. Inclui também, 10 SAs do tipo

tier-1 (sem fornecedores) no topo, e um conjunto de *tiers* de trânsito no segundo nível da hierarquia conectados com ligações *f2c*. Teria sido possível definir uma topologia maior a partir dos dados do CAIDA. Porém, há limitações de escala significativas devido a uma utilização intensiva de recursos computacionais do *ns-2*. A figura 5.1 ilustra a topologia vista no ns/nam; ns/nam é o pacote de animação de redes utilizado pelo *ns-2*.

As instituições de SAs de cada nó da figura 5.1 estão listadas na tabela 5.1.

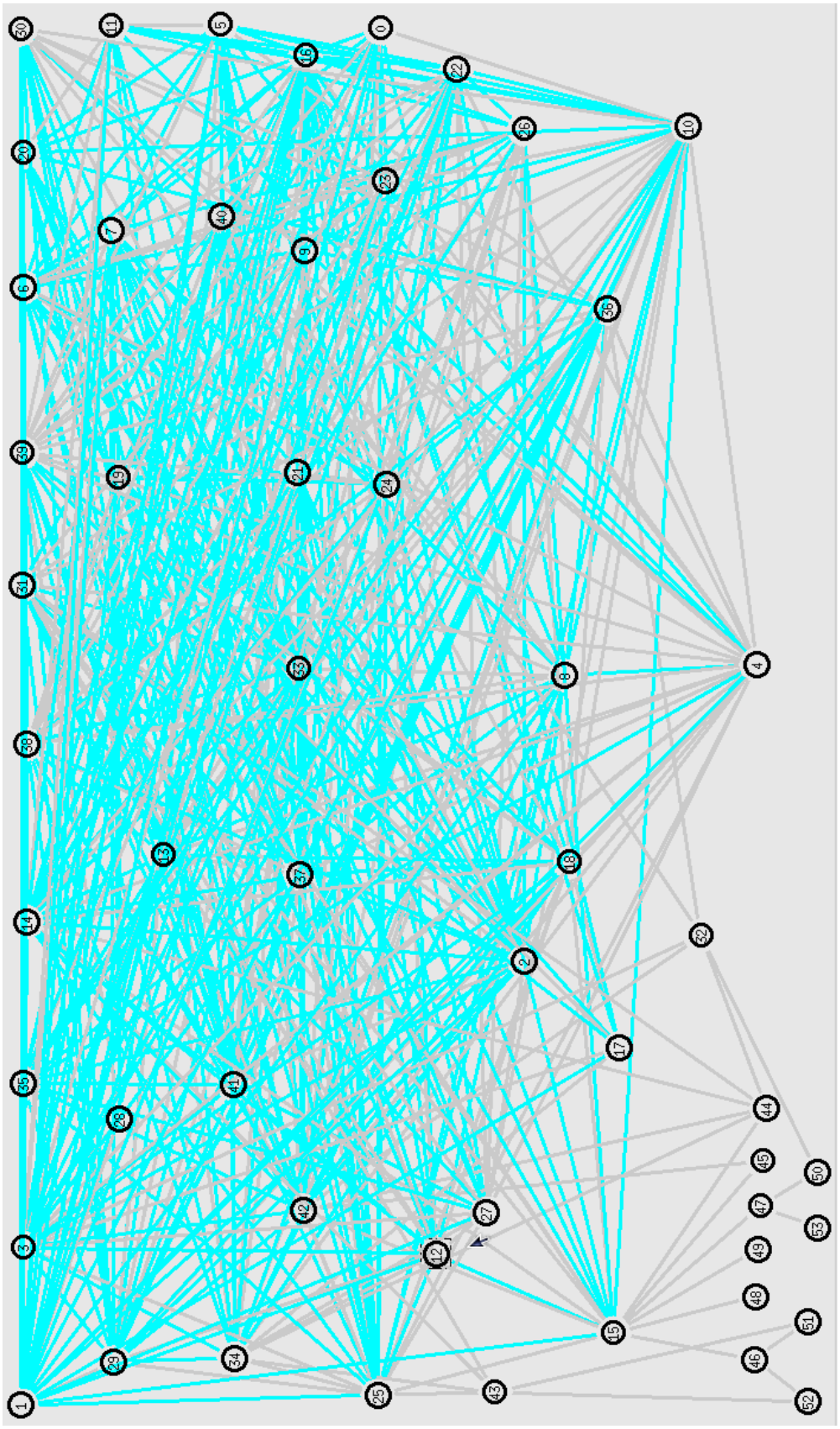

Figura 5.1: Topologia ns/nam. As ligações a azul são ligações fornecedor-fornecedor; As ligações a cinzento representam relações fornecedor para cliente de cima para baixo.

Tabela 5.1: Identificação dos nós da figura 5.1.

Nó	SA	Nome da Instituição		Nó	SA	Nome da Instituição
0	8708	RDSNET		27	3491	Beyond The Network America
1	6939	Hurricane Electric		28	20932	IP-MAN.Net Engineering
2	2497	Internet Initiative Japan Inc.		29	6461	MFN - Metromedia Fiber Network
3	3549	Global Crossing		30	7018	AT&T WorldNet Services
4	12956	Telefonica		31	701	MCI Communications Services
5	6830	UPC Broadband		32	2860	Novis
6	4323	TW Telecom Holdings		33	3561	Savvis
7	9002	RETN Limited		34	702	MCI Communications Services
8	5400	BT Global Services		35	209	Qwest Communications Company
9	4766	Korea Telecom		36	5511	France Telecom - Orange
10	6762	Telecom Italia Sparkle		37	3257	Tinet SpA
11	22773	Cox Communications		38	1239	Sprint
12	5413	GX Networks		39	3356	Level 3 Communications
13	1299	TeliaSonera AB Networks		40	3320	Deutsche Telekom AG
14	174	Cogent Communications		41	2914	NTT America, Inc.
15	8657	Portugal Telecom		42	6453	TELEGLOBE IP ENGINEERING
16	3303	SWISSCOM		43	9186	ONI TELECOM
17	3216	Golden Telecom		44	13156	CABOVISAO
18	1273	Cable and Wireless IP GSOC Europe		45	12542	TVCABO
19	19151	WV FIBER LLC		46	3243	TELEPAC
20	2828	XO Communications		47	15525	PT PRIME
21	13237	LambdaNet Communications		48	15457	Cabo Tv Madeirense
22	2516	KDDI Corp.		49	42863	TMN
23	3786	LG DACOM Corporation		50	35038	INESC
24	8928	Interoute Communications		51	34873	IGIF-Ministério da Saúde
25	286	KPN Internet Solutions		52	25253	Caixa Geral de Depósitos
26	6539	Bell Canada		53	43643	Tap Air Portugal

Familiarizados com a topologia da rede, podemos avançar para a próxima secção que descreve as experiências no *ns-2*.

5.3 Experiências

Esta secção descreve as experiências feitas no simulador *ns-2*. As análises efectuadas focam-se nas seguintes métricas:

1. Número de pacotes entregues versus os não entregues na secção 5.3.2;

2. Percentagem de utilização das ligações consideradas relevantes na secção 5.3.3;

3. Número de mensagens de engenharia de tráfego enviadas na secção 5.3.4.

São comparados os seguintes algoritmos de ET:

- Algoritmo com base numa só origem;
- Algoritmo com base em N origens;
- Algoritmo com distribuição por classes;

Na primeira experiência, é verificado o desempenho dos algoritmos no que diz respeito à entrega de pacotes. É comparada a percentagem de pacotes entregues sem engenharia de tráfego e com engenharia de tráfego para cada um dos algoritmos.

Na segunda experiência é verificado o equilíbrio das ligações, ou seja, se um SA está a receber o tráfego de uma forma equilibrada pelos seus vizinhos. Para esta análise é comparado o valor do coeficiente de *Jain* [JCH84].

A terceira e última análise, contabiliza os pacotes de sinalização utilizados por cada algoritmo para aplicar a engenharia de tráfego. Para as experiências anteriores a comparação é feita tendo em conta os mesmos cenários de envio de tráfego, sem engenharia de tráfego e com engenharia de tráfego.

A topologia da figura 5.1 foi utilizada para realizar as experiências. Foram escolhidos 20 SAs dos níveis de topo para serem as origens do tráfego e 9 SAs dos níveis inferiores para serem os destinos.

Foram criadas aleatoriamente 110 simulações diferentes, cada uma com um conjunto de 4 origens e 1 destino. Nas análises efectuadas são representados os intervalos de confiança, que representam um conjunto de valores que contém o valor medido com uma probabilidade de 95%. A definição dos intervalos de confiança é feita assumindo-se que os dados

têm uma distribuição normal. Para $n = 110$, é dado por:

$$\overline{x} - a\frac{\sigma}{\sqrt{n}} \leq \mu \leq \overline{x} + a\frac{\sigma}{\sqrt{n}} \text{ com } a : P(Z > a) = \frac{\alpha}{2}$$

$$\alpha = 0.05 \iff P(Z > a) = 0.025 \iff$$
$$\iff P(Z < a) = 0.975 \iff a = \Phi^{-1}(0.975) = 1.96$$

5.3.1 Cenário e Parâmetros das Simulações

Para avaliar o desempenho relativamente às métricas definidas na secção 5.1, foram definidos para as 517 ligações da topologia os parâmetros dados pela tabela 5.2.

Tabela 5.2: Parâmetros das 517 ligações.

Largura de banda	Atraso	Fila de espera
1 Megabit	10ms	DropTail

A cada uma das quatro origens foi anexado um gerador de tráfego com uma taxa de bits constante. O gerador de tráfego envia um pacote com 500 bytes em cada 0.005 segundos (i.e. 200 pacotes por segundo). Cada uma das origens injecta tráfego na rede durante 5 segundos, e o início do envio do tráfego está espaçado em 0.1 segundos entre elas (i.e. a segunda origem inicia o envio do tráfego 0.1 segundos depois da primeira o ter feito e assim sucessivamente até estarem as quatro origens a enviar tráfego).

Foram estabelecidos os seguintes parâmetros dos algoritmos para as simulações:

- O limite de ocupação das ligações para o qual as ligações são consideradas sobrecarregadas, $\psi = 75\%$;
- Número máximo de pacotes a enviar por ligação, $\nabla = 100$;
- O limite de ocupação das ligações por origem, $\rho = 20\%$;
- O número de classes, $n = 4$;

5.3.2 Análise: Número de pacotes entregues versus os não entregues

Nesta secção é efectuada uma análise do desempenho dos algoritmos em termos da redução do número de pacotes não entregues. Foi registado o número de pacotes não entregues sem aplicação e com a aplicação dos algoritmos de engenharia de tráfego para as diferentes simulações, procedendo de seguida ao cálculo da percentagem de recuperação de pacotes não entregues para cada simulação. Os parâmetros definidos para as simulações estão descritos na secção 5.3.1.

O gráfico da figura 5.2 representa a função cumulativa da percentagem de pacotes que passaram a ser entregues com a aplicação de ET nas simulações realizadas; compara os três algoritmos registando a percentagem de recuperação dos pacotes não entregues.

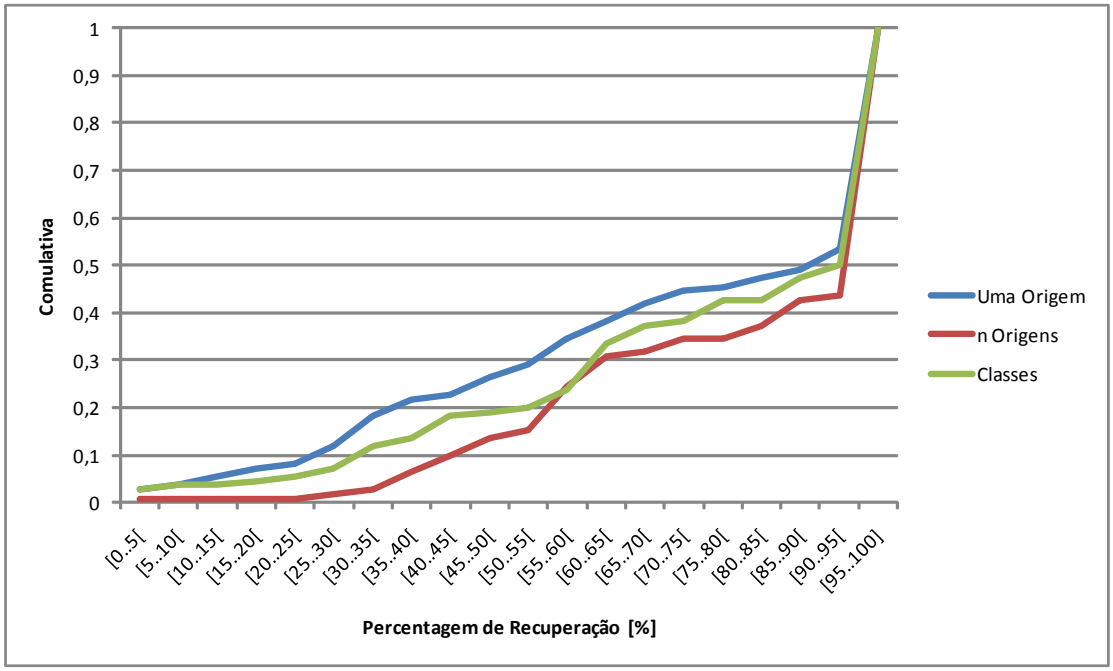

Figura 5.2: [%] de Recuperação na Entrega de Pacotes.

Analisando a figura 5.2 verifica-se que a aplicação dos algoritmos nunca aumenta o número de pacotes não entregues, pois não existem simulações abaixo dos 0% (i.e. nunca se perderam mais pacotes com ET do que sem ET). Os algoritmos ou mantêm o número de

pacotes não entregues ou o reduzem. Salienta-se que todos algoritmos têm mais de 50% das simulações com recuperação na entrega de pacotes superior a 90%. Salienta-se ainda que o algoritmo *N origens*, apresenta o melhor resultado com quase 60% das simulações com recuperação entre os 95% e 100%.

As percentagens de recuperação da entrega de pacotes de todas as simulações foram posteriormente processadas para obter a média, $\bar{x}$, de recuperação para cada um dos algoritmos e respectivos desvios padrão, σ, e o intervalo de confiança a 95%, $I.C._{95\%}(\mu)$; a tabela 5.3 mostra esses resultados.

Tabela 5.3: Média, Desvio Padrão e Intervalo de Confiança dos três Algoritmos na Recuperação da Entrega de Pacotes.

	Média - $\bar{x}[\%]$	Desvio Padrão - σ	$I.C._{95\%}(\mu)$
Uma Origem	$72,31$	31.06	$[66.51; 78.11]$
N Origens	$81,40$	23.77	$[76.96; 85.84]$
Classes	$76,92$	28.41	$[71.61; 82.23]$

5.3.3 Análise: Equilíbrio na distribuição da carga em SAs congestionados

Nesta secção é efectuada uma análise do equilíbrio na entrega de tráfego aos SAs congestionados. Estes SAs são escolhidos por terem, pelo menos, uma ligação sobrecarregada. A partir destes SAs identificam-se as ligações pelas quais cada um recebe tráfego, definindo assim um conjunto de ligações para cada SA. O equilíbrio na distribuição de carga no conjunto de ligações é medido com o coeficiente de *Jain*. O coeficiente de *Jain* é dado pela seguinte fórmula:

$$Jain = \frac{(\sum x_i)^2}{(n \times \sum x_i^2)}$$

onde x_i é a carga na ligação i por onde o SA recebe tráfego e n o número de ligações. Quanto mais o valor de *Jain* se aproximar de 1, mais equilibrada é a distribuição do tráfego pelas ligações.

Foram realizadas 110 simulações sem ET e com os três algoritmos de ET, com os parâmetros na secção 5.3.1. No conjunto das simulações foram detectadas 195 ligações sobrecarregadas, que compõem o conjunto de amostra considerado. Foi calculado o coeficiente de *Jain* aos 195 conjuntos de ligações, representando-se na figura 5.3 a função cumulativa do factor de *Jain*.

Sem a aplicação da engenharia de tráfego registam-se aproximadamente 70% dos conjuntos das ligações com um coeficiente de *Jain* inferior a 0.5. Com a aplicação de engenharia de tráfego através do algoritmo com cálculos para uma origem, só 40% dos conjuntos de ligações mostram um coeficiente de *Jain* inferior a 0.5.

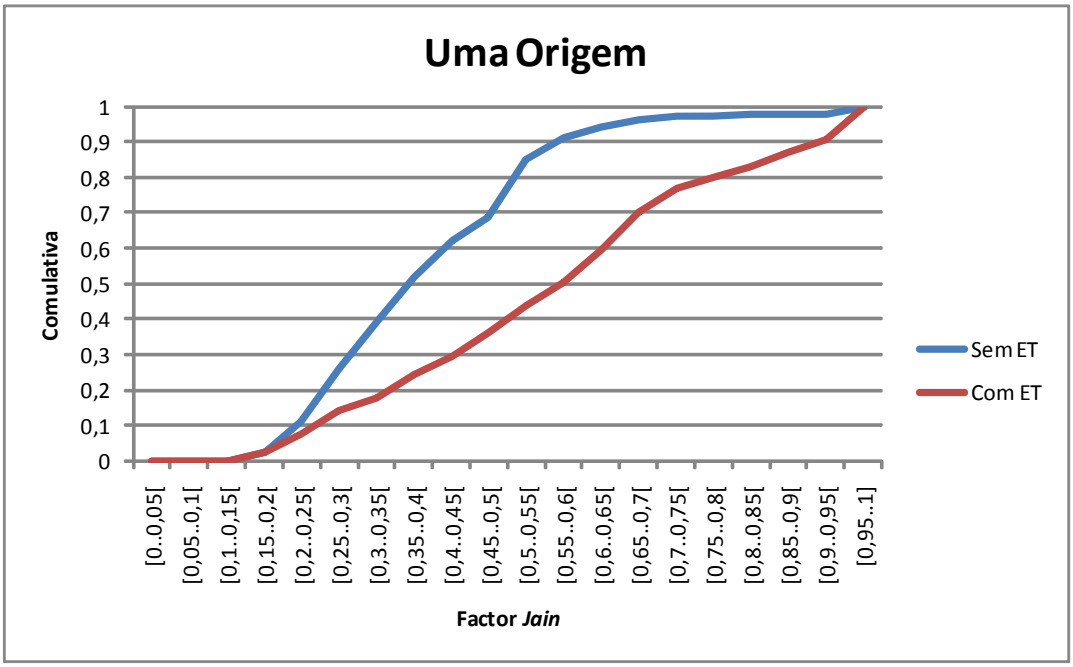

Figura 5.3: Equilíbrio das Ligações para o Algoritmo: Cálculos para uma Origem.

Para as 110 simulações do algoritmo de ET com cálculos para N origens foram detectadas 200 ligações sobrecarregadas, que compõem o conjunto de amostra considerado. Foi calculado o coeficiente de *Jain* aos 200 conjuntos de ligações, representando-se na figura 5.4 a função cumulativa do factor de *Jain*.

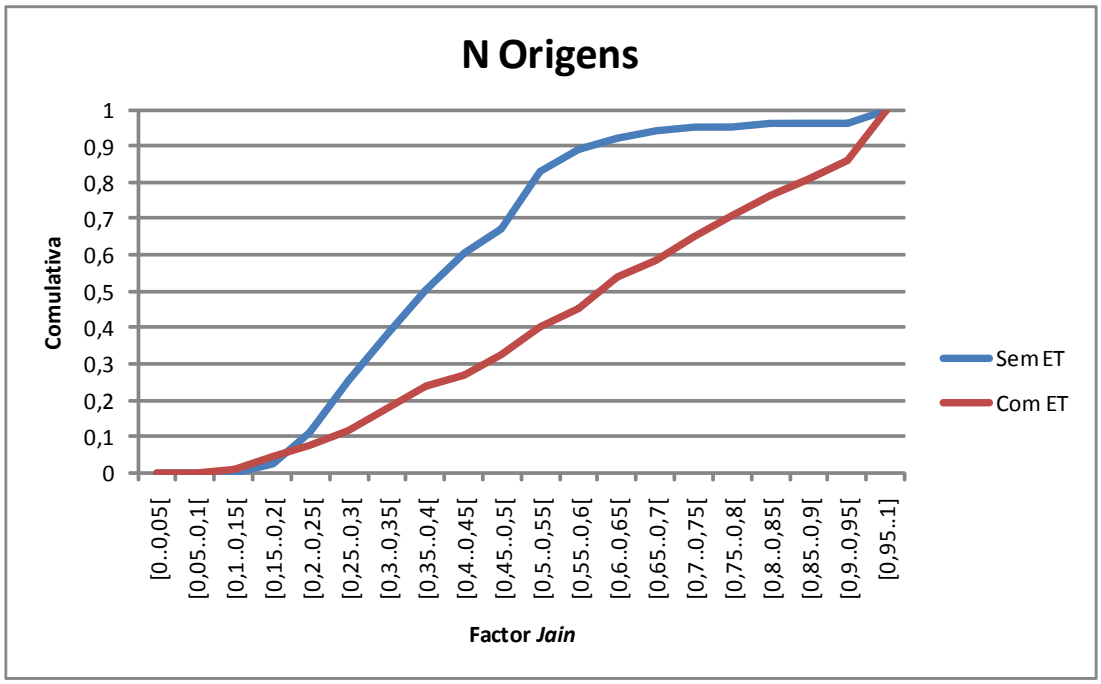

Figura 5.4: Equilíbrio das Ligações para o Algoritmo: Cálculos para N Origens.

Sem a aplicação da engenharia de tráfego regista-se aproximadamente 75% dos conjuntos das ligações com um coeficiente de *Jain* inferior a 0.5. Enquanto com a aplicação de engenharia de tráfego através do algoritmo com cálculos para N origens, só 35% dos conjuntos de ligações mostram um coeficiente de *Jain* inferior a 0.5.

Para as 110 simulações do algoritmo de ET com distribuição por classes foram detectadas 196 ligações sobrecarregadas, que compõem o conjunto de amostra considerado. Foi calculado o coeficiente de *Jain* aos 196 conjuntos de ligações, representando-se na figura 5.4 a função cumulativa do factor de *Jain*.

Sem a aplicação da engenharia de tráfego regista-se aproximadamente 80% dos conjuntos das ligações com um factor de *Jain* inferior a 0.5. Enquanto com a aplicação de engenharia de tráfego através do algoritmo com distribuição por classes, só 40% dos conjuntos de ligações mostram um factor de *Jain* inferior a 0.5.

Figura 5.5: Equilíbrio das Ligações para o Algoritmo: Distribuição por Classes.

Os coeficientes de *Jain* de todas as simulações com e sem engenharia de tráfego, foram posteriormente processados para obter a média, $\bar{x}$, dos coeficientes de *Jain* e respectivos desvios padrão, σ, e os intervalos de confiança a 95%, $I.C._{95\%}(\mu)$; a tabela 5.4 mostra esses resultados. O desempenho dos três algoritmos de ET é muito semelhante, pois todos melhoram o equilíbrio do tráfego nas ligações em relação aos resultados obtidos sem ET. O algoritmo com cálculos para N origens mostra ligeiramente melhores resultados que os outros dois.

Tabela 5.4: Média, Desvio Padrão e Intervalo de Confiança dos coeficientes de *Jain* para os três Algoritmos.

	Sem ET			Com ET		
	Média, $\bar{x}$	Desvio Padrão, σ	$I.C._{95\%}(\mu)$	Média, $\bar{x}$	Desvio Padrão, σ	$I.C._{95\%}(\mu)$
Uma Origem	0.418	0.156	$[0.396; 0.44]$	0.59	0.23	$[0.558; 0.623]$
N Origens	0.431	0.176	$[0.407; 0.456]$	0.626	0.248	$[0.591; 0.66]$
Classes	0.414	0.15	$[0.393; 0.435]$	0.587	0.228	$[0.555; 0.619]$

5.3.4 Análise: Número de mensagens de engenharia de tráfego

Nesta secção é feita a análise do número de mensagens de sinalização utilizadas por cada algoritmo para efectuar engenharia de tráfego. Para o mesmo conjunto de simulações das secções anteriores foi contabilizado o número de mensagens de ET utilizadas por cada um dos algoritmos. O gráfico da figura 5.6 representa a função cumulativa das simulações realizadas; compara os três algoritmos registando o número de mensagens de engenharia de tráfego enviadas.

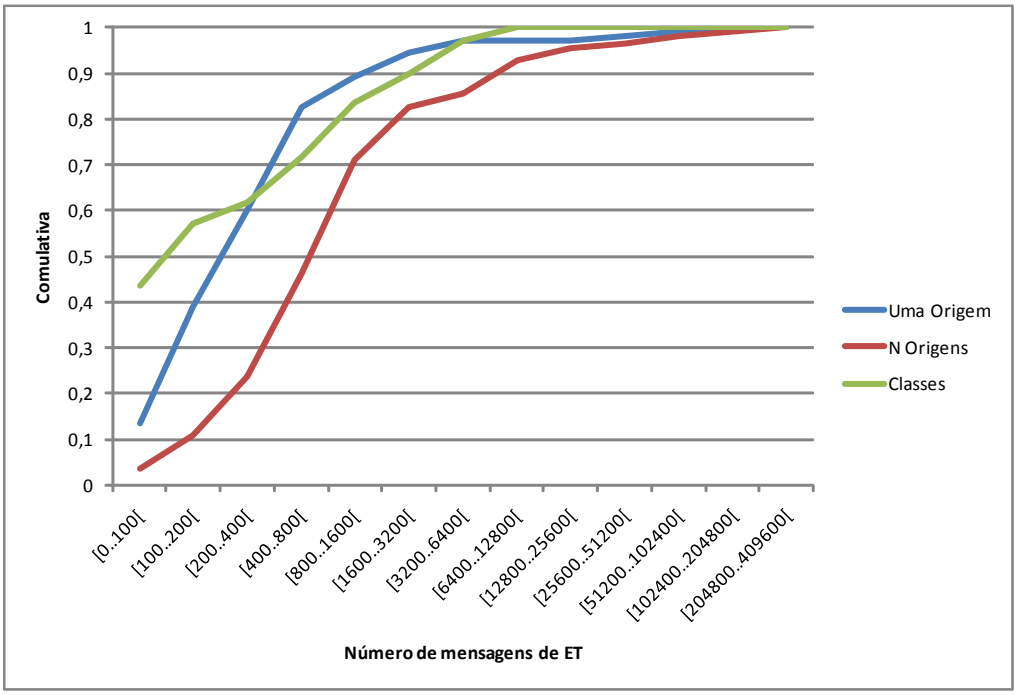

Figura 5.6: Número de Mensagens de Engenharia de Tráfego.

A figura 5.6 revela que o algoritmo com distribuição por classes tem aproximadamente 45% das simulações com um número de mensagens de ET inferior a 100, e que o máximo de mensagens enviadas numa simulação não ultrapassa as 12800. Os outros dois algoritmos apresentam uma quantidade reduzida de simulações onde o número de mensagens enviadas é inferior a 100, em comparação com o algoritmo com distribuição por classes. Pode-se ver também que os algoritmos com uma origem e N origens apresentam um número máximo de mensagens enviadas muito mais elevado que no algoritmo com distribuição por classes devido à utilização de um método de inundação.

Recolheram-se o número de mensagens de engenharia de tráfego enviadas de todas as simulações, e estes valores foram posteriormente processados para obter o valor médio, $\bar{x}$, o máximo e o mínimo das mensagens enviadas para cada um dos algoritmos; a tabela 5.5 mostra esses resultados.

Tabela 5.5: Média, Máximo e Mínimo das Mensagens Enviadas.

	Mínimo	Média - $\bar{x}$	Máximo
Uma Origem	10	2551	114059
N Origens	12	7521	241662
Classes	0	1012	12364

A tabela 5.6 mostra os valores do desvio padrão, σ, e os intervalos de confiança a 95% obtidos para cada um dos algoritmos no envio de mensagens de ET. Os resultados da análise ao número de mensagens de sinalização utilizadas por cada algoritmo para efectuar ET mostram uma grande diferença entre algoritmos. O algoritmo com distribuição por classes apresenta uma redução de 60% nas mensagens de sinalização utilizadas em relação ao algoritmo com cálculo para uma origem e um redução de 86% em relação ao algoritmo com cálculos para N origens.

Tabela 5.6: Desvio Padrão e Intervalo de Confiança das Mensagens Enviadas.

	Desvio Padrão - σ	$I.C._{95\%}(\mu)$
Uma Origem	13015	$[119; 4984]$
N Origens	31046	$[1720; 13323]$
Classes	2035	$[632; 1393]$

5.3.5 Resumo

A tabela 5.7 mostra um resumo dos resultados obtidos nas análises efectuadas para os três algoritmos apresentados. Os algoritmos apresentam resultados muito semelhantes nas duas primeiras análises, recuperação do número de pacotes não entregues e no equilíbrio das ligações, sendo o algoritmo com cálculos para N origens o que apresenta melhores resultados e o algoritmo com cálculos para uma origem o que apresenta os piores resultados

dos três algoritmos.

A terceira análise efectuada teve como objectivo contabilizar a sinalização utilizada por cada um dos algoritmos, pois abrange um dos principais temas da engenharia de tráfego: efectuar ET com a menor sinalização possível. Nesta análise registou-se uma grande diferença entre algoritmos, como referido na secção 4.4.4. O algoritmo com distribuição por classes registou um menor número de mensagem de ET em comparação com os outros dois algoritmos, evidenciando-se assim em relação a eles.

Tabela 5.7: Resumo dos Resultados Obtidos.

	Algoritmos		
	Cálculo para uma Origem	Cálculo para N Origens	Distribuição por Classes
Redução do número de pacotes não entregues	Em média tem uma recuperação de 72.31%	Em média tem uma recuperação de 81.40%	Em média tem uma recuperação de 76.92%
Equilíbrio das ligações	Em média o coeficiente de *Jain* é 0.59	Em média o coeficiente de *Jain* é 0.626	Em média o coeficiente de *Jain* é 0.587
Número de mensagens de ET	*Mínimo* $= 10$, $\bar{x} = 2551$; *Máximo* $= 114059$	*Mínimo* $= 12$, $\bar{x} = 7521$; *Máximo* $= 241662$	*Mínimo* $= 0$, $\bar{x} = 1012$; *Máximo* $= 12364$

Se fosse necessário escolher um algoritmo, a escolha recairia sobre o algoritmo com distribuição por classes, porque apresenta uma recuperação do número de pacotes não entregues e um equilíbrio das ligações muito semelhante aos outros algoritmos, mas com um número de mensagens de ET muito menor. Em contrapartida, exige a utilização de maior capacidade computacional nos SAs.

Capítulo 6

Conclusões

Este capítulo resume as conclusões desta dissertação, baseado nos capítulos anteriores. A secção 6.1 contém uma pequena síntese e a secção 6.2 refere as considerações finais da dissertação. A secção 6.3, enumera alguns tópicos para trabalho futuro.

6.1 Síntese

Esta secção descreve brevemente o conteúdo de cada capítulo. Na secção 1.1 foi formulada a hipótese principal desta dissertação. Os capítulos seguintes validam a hipótese formulada.

No capítulo 2 discutiu-se uma taxonomia para a engenharia de tráfego. Foi feita uma breve discussão sobre o estado da arte, comparando os vários modelos em que se pode efectuar engenharia de tráfego. Analisa-se os prós e contras dos métodos discutidos, com base nos estudos realizados recentemente. No capítulo 3 é feita uma descrição da arquitectura DTIA, uma solução para encaminhamento inter-região, que serve como base para a principal contribuição desta dissertação. São apresentadas as possibilidades que o DTIA oferece para aplicar engenharia de tráfego no nível de inter-domínio.

O capítulo 4 apresenta uma visão global da implementação desta dissertação, com a ajuda de fluxogramas. São propostas três soluções para efectuar engenharia de tráfego na arquitectura DTIA. As alterações feitas no simulador ns-2 foram também apresentadas para a posterior validação dos algoritmos apresentados. O capítulo 5 descreve a topologia da rede

experimentada no simulador ns-2, e compara os resultados obtidos com os três algoritmos.

6.2 Conclusões

O capítulo 2 apresenta diferentes abordagens para efectuar engenharia de tráfego, geralmente com uma complexidade considerável para a arquitectura da Internet actual. São assim bases difíceis para os problemas de optimização. O protocolo utilizado pela Internet para o encaminhamento é o BGP, sendo a engenharia de tráfego realizada através da manipulação de atributos na selecção de rotas.

A engenharia de tráfego, ao nível do inter-domínio ainda é uma área desafiadora. As técnicas actuais do BGP para realizar engenharia de tráfego são primitivas, e além disso, o seu efeito é muitas vezes difícil de prever. Na engenharia de tráfego inter-domínio é essencial uma cooperação entre domínios. Pesquisas recentes demonstram que quando domínios adjacentes executam ET inter-domínio de forma egoísta, não só o desempenho global da rede não é optimizado, como as estratégias de cada domínio para a ET inter-domínio pode afectar negativamente outros domínios.

Nesta dissertação propusemos novas soluções que estendem o trabalho DTIA, adicionando-lhe a capacidade de efectuar engenharia de tráfego. Estas propostas melhoram o controlo e optimização do encaminhamento e o equilíbrio do tráfego nas ligações, mantendo a escalabilidade da rede. Estes objectivos são atingidos com a cooperação entre SAs, através de um único pacote de sinalização. São usadas apenas informações das ligações e as características locais da arquitectura DTIA para evitar o congestionamento e conseguir uma melhor distribuição do tráfego no topo da arquitectura DTIA. É um custo menor, especialmente porque o DTIA não precisa de pacotes (nem mesmo para os anúncios de rotas do BGP). Os resultados do capítulo 5 comprovam isso mesmo.

6.3 Trabalho Futuro

Durante o desenvolvimento das soluções propostas, algumas hipóteses foram feitas em termos de que parâmetros deviam ser analisados. As soluções propõem que a engenharia

de tráfego seja realizada com base nas taxas de ocupação das ligações. Poderia ter sido implementado um modelo que interligasse a largura de banda e taxas de ocupação com o atraso das ligações.

As soluções propostas detectam uma ligação sobrecarregada e efectuam engenharia de tráfego. Após a conclusão do envio do tráfego as alterações efectuadas no encaminhamento de pacotes mantêm-se. Outro ponto para trabalho futuro é a realização de um algoritmo distribuído que identifique o que serve melhor os objectivos da engenharia de tráfego; manter as alterações ou revertê-las ao seu estado inicial. Se o algoritmo indicasse que o melhor é reverter as alterações, isto poderia ser alcançado usando a sinalização "*reset*" que já está prevista no protocolo.

Bibliografia

[ABG+01] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, and G. Swallow. Rsvp-te: Extensions to rsvp for lsp tunnels. RFC 3209, dec 2001.

[ABP08] Pedro Amaral, Luís Bernardo, and Paulo Pinto. Dtia: an inter-domain reachability architecture technical report. Technical report, September 2008.

[ABP09] Pedro Amaral, Luís Bernardo, and Paulo Pinto. Dtia - routing at the inter-domain level technical report. 2009.

[AGA+09] P. Amaral, F. Ganhao, C. Assuncao, L. Bernardo, and P. Pinto. Scalable multi-region routing at inter-domain level. In *Proc. IEEE Global Telecommunications Conf. GLOBECOM 2009*, pages 1–8, 2009.

[AnCK03] Sharad Agarwal, Chen nee Chuah, and Randy H. Katz. Opca: Robust inter-domain policy routing and traffic control, January 18 2003.

[Awd99] D.O. Awduche. Mpls and traffic engineering in ip networks. *Communications Magazine, IEEE*, 37(12):42 –47, December 1999.

[Bon07] Olivier Bonaventure. Reconsidering the internet routing architecture. Internet draft, draft-bonaventure-irtf-rira-00.txt, March 2007.

[cai] Caida data. see http://www.caida.org/data/, august 2009.

[CL05] R.K.C. Chang and M. Lo. Inbound traffic engineering for multihomed ass using as path prepending. *Network, IEEE*, 19(2):18 – 25, 2005.

[CR05] M. Caesar and J. Rexford. Bgp routing policies in isp networks. *Network, IEEE*, 19(6):5 – 11, 2005.

[DB08] Benoit Donnet and Olivier Bonaventure. On BGP communities. *Computer Communication Review*, 38(2):55–59, 2008.

[DKF+07] Xenofontas Dimitropoulos, Dmitri Krioukov, Marina Fomenkov, Bradley Huffaker, Young Hyun, kc claffy, and George Riley. As relationships: inference and validation. *SIGCOMM Comput. Commun. Rev.*, 37(1):29–40, 2007.

[FVA06] A Farrel, J.-P. Vasseur, and J. Ash. A path computation element (pce)-based architecture. IETF RFC 4655, August 2006.

[Gan09] Francisco Ganhão. Multi-region routing. Master's thesis, UNIVERSIDADE NOVA DE LISBOA, Faculdade de Ciências e Tecnologia, Departamento de Engenharia Electrotécnica e de Computadores, 2009.

[Gao00] Lixin Gao. On inferring autonomous system relationships in the internet. In *Global Telecommunications Conference, 2000. GLOBECOM '00. IEEE*, volume 1, pages 387 –396 vol.1. 2000.

[GR01] Lixin Gao and J. Rexford. Stable internet routing without global coordination. *Networking, IEEE/ACM Transactions on*, 9(6):681 –692, December 2001.

[GS05] Timothy G. Griffin and João L. Sobrinho. Metarouting. In Roch Guérin, Ramesh Govindan, and Greg Minshall, editors, *SIGCOMM*, pages 1–12. ACM, 2005.

[JCH84] R. Jain, D. Chiu, and W. Hawe. A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. Technical Report TR-301, DEC Research, September 1984.

[KLS03] M. Kodialam, T.V. Lakshman, and S. Sengupta. Online multicast routing with bandwidth guarantees: a new approach using multicast network flow. *Networking, IEEE/ACM Transactions on*, 11(4):676 – 686, 2003.

[LR07] Yong Liu and A.L. Narasimha Reddy. Multihoming route control among a group of multihomed stub networks. *Comput. Commun.*, 30(17):3335–3345, 2007.

[MC05] R. Musunuri and J.A. Cobb. An overview of solutions to avoid persistent bgp divergence. *Network, IEEE*, 19(6):28 – 34, 2005.

[MWA04] Ratul Mahajan, David Wetherall, and Thomas Anderson. Towards coordinated interdomain traffic engineering. In *In Proc. SIGCOMM Workshop on Hot Topics in Networking*, 2004.

[MWA05] Ratul Mahajan, David Wetherall, and Thomas Anderson. Negotiation-based routing between neighboring isps. In *in Proc. USENIX Symposium on Networked Systems Design and Implementation*, 2005.

[MZF07] D. Meyer, L. Zhang, and K. Fall. Report from the iab workshop on routing and addressing. RFC 4984, September 2007.

[nsM10] Manual do ns-2. ver http://www.isi.edu/nsnam/ns/ns-documentation.html., Setembro 2010.

[nsR10] ns-2 - network simulator 2. ver http://isi.edu/nsnam/ns/, Setembro 2010.

[nsT10] Simulador de rede - tutorial do marc greis. ver http://www.isi.edu/nsnam/ns/tutorial/, Setembro 2010.

[OPW+10] R. Oliveira, Dan Pei, W. Willinger, Beichuan Zhang, and Lixia Zhang. The (in)completeness of the observed internet as-level structure. *Networking, IEEE/ACM Transactions on*, 18(1):109 –122, 2010.

[osi10] Osi: Open system interconnection. ver http://standards.iso.org/ittf/licence.html, Setembro 2010.

[QB05] B. Quoitin and O. Bonaventure. A cooperative approach to interdomain traffic engineering. In *Proc. Next Generation Internet Networks*, pages 450–457, 2005.

[QTUB04] Bruno Quoitin, Sébastien Tandel, Steve Uhlig, and Olivier Bonaventure. Interdomain traffic engineering with redistribution communities. *Computer Communications*, 27(4):355–363, 2004.

[QUP+03] Bruno Quoitin, Steve Uhlig, Cristel Pelsser, C. Pelsser, Louis Swinnen, and Olivier Bonaventure. Interdomain traffic engineering with bgp. *IEEE Communications Magazine*, 41, 2003.

[RIP] Ripe database, http://www.ripe.net/db/index.html.

[RL06] Y. Rekhter and T. Li. A border gateway protocol 4 (bgp-4). RFC 4271, 2006.

[SARK02] L. Subramanian, S. Agarwal, J. Rexford, and R. H. Katz. Characterizing the internet hierarchy from multiple vantage points. In *Proc. IEEE Twenty-First*

Annual Joint Conf. of the IEEE Computer and Communications Societies INFOCOM 2002, volume 2, pages 618–627, 2002.

[SCE+05] Lakshminarayanan Subramanian, Matthew Caesar, Cheng Tien Ee, Mark Handley, Morley Mao, Scott Shenker, and Ion Stoica. Hlp: a next generation inter-domain routing protocol. *SIGCOMM Comput. Commun. Rev.*, 35(4):13–24, 2005.

[SE01] P. Srisuresh and K. Egevang. Traditional ip network address translator (traditional nat). RFC 3022, jan 2001.

[Sob05] J.L. Sobrinho. An algebraic theory of dynamic network routing. *Networking, IEEE/ACM Transactions on*, 13(5):1160 – 1173, 2005.

[TAP+01] P. Trimintzios, I. Andrikopoulos, G. Pavlou, P. Flegkas, D. Griffin, P. Georgatsos, D. Goderis, Y. T'Joens, L. Georgiadis, C. Jacquenet, and R. Egan. A management and control architecture for providing ip differentiated services in mpls-based networks. *Communications Magazine, IEEE*, 39(5):80 –88, May 2001.

[tcl10] Sitio de desenvolvimento da linguagem tcl. ver http://www.tcl.tk/, Setembro 2010.

[WHPH08] Ning Wang, Kin Ho, G. Pavlou, and M. Howarth. An overview of routing optimization for internet traffic engineering. *IEEE Communications Surveys & Tutorials*, 10(1):36–56, 2008.

[YCB07] X.. Yang, D.. Clark, and A.W. Berger. Nira: A new inter-domain routing architecture. *Networking, IEEE/ACM Transactions on*, 15(4):775 –788, 2007.