



Universidade de Aveiro
2009

Departamento de Electrónica
Telecomunicações e Informática

Ana Francisca
Carvalho de
Almeida Sampaio e
Melo

Engenharia de Tráfego de Redes Ethernet
baseadas em Árvores de Suporte



Universidade de Aveiro
2009

Departamento de Electrónica
Telecomunicações e Informática

**Ana Francisca
Carvalho de
Almeida Sampaio e
Melo**

**Engenharia de Tráfego de Redes Ethernet
baseadas em Árvores de Suporte**

dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre Integrado em Engenharia Electrónica e Telecomunicações, realizada sob a orientação científica de Amaro de Sousa, Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

o júri

presidente

Professor Doutor Rui Luís Andrade Aguiar
Universidade de Aveiro

vogal – Arguente principal

Professor Doutor Carlos Manuel da Silva Rabadão
Dep. De Eng^a Informática da Esc. Sup. De Tecnologia e Gestão do Inst. Politécnico
de Leiria

vogal – Orientador

Professor Doutor Amaro Fernandes de Sousa
Universidade de Aveiro

palavras-chave

Multiple Spanning Tree, Ethernet, Engenharia de Tráfego.

resumo

Dado o seu custo, a tecnologia Ethernet tem vindo progressivamente a ser utilizada nas redes dos operadores públicos, nos seus segmentos de rede de acesso e rede metropolitana. Tradicionalmente, o encaminhamento em redes Ethernet é baseado numa única árvore de suporte. Esta forma de funcionamento, apropriada para redes de área local (LANs), não responde aos requisitos de uma rede de operador. Num passado recente, a Ethernet tornou-se mais atractiva como tecnologia para um operador com a introdução dos protocolos IEEE 802.1w *Rapid Spanning Tree Protocol* (RSTP), que permite tempos de reconfiguração da árvore de suporte de menos de um segundo quando há falhas de rede, e IEEE 802.1s *Multiple Spanning Tree Protocol* (MSTP), que permite balancear o tráfego pelas diferentes ligações de rede com base em múltiplas árvores de suporte, conseguindo-se, assim, uma melhor utilização dos recursos da rede.

Este trabalho aborda primeiro a evolução dos protocolos de encaminhamento para redes de camada 2 baseadas na Ethernet até à proposta do MSTP. Depois, faz um estudo da literatura científica do modo como o MSTP tem sido explorado para melhorar os diferentes aspectos do funcionamento de uma rede de telecomunicações tais como o balanceamento de carga, optimização da utilização da rede, qualidade de serviço, etc.

keywords

Multiple Spanning Tree, Ethernet, Traffic Engineering.

abstract

Due to its cost, Ethernet technology is being progressively used on the access and metropolitan segments of public operator networks. Traditionally, routing on Ethernet networks was based on a single spanning tree. This paradigm, which is appropriate for LAN environments, does not answer to the requirements of a public operator network. Recently, Ethernet became more attractive to operators with the introduction of two new protocols: IEEE 802.1w Rapid Spanning Tree Protocol (RSTP), enabling spanning tree reconfigurations below 1 second, and IEEE 802.1s Multiple Spanning Tree Protocol (MSTP), enabling a better load balance among all network links based on multiple spanning trees and, thus, improving the network resource utilization.

This work addresses, firstly, the evolution of layer 2 routing protocols up to MSTP used in Ethernet networks. Then, it makes a comprehensive scientific state-of-art on the different ways that MSTP has been proposed in order to improve different aspects of telecommunication networks like load balancing, optimization of network utilization, quality of service, etc.

Índice

| | |
|---|----|
| CAPÍTULO 1 - INTRODUÇÃO | 7 |
| 1.1 Enquadramento | 7 |
| 1.2 Objectivos..... | 8 |
| 1.3 Organização da dissertação | 8 |
| CAPÍTULO 2 – A origem do Multiple Spanning Tree e o seu funcionamento | 9 |
| 2.1 Spanning Tree Protocol | 9 |
| 2.1.1 Construção da árvore de suporte | 10 |
| 2.1.2 Caso de falha na rede..... | 13 |
| 2.1.3 Estado das portas..... | 14 |
| 2.2 Rapid Spanning Tree Protocol | 15 |
| 2.2.1 Estado e Função das portas | 15 |
| 2.2.2 Bridge Protocol Data Unit (BPDU)..... | 18 |
| 2.2.3 Fast Aging | 19 |
| 2.2.4 UplinkFast, BackboneFast e PortFast | 19 |
| 2.2.5 Novos mecanismos para detecção de alteração da rede | 25 |
| 2.3 Melhorias e compatibilidade do RSTP em relação ao STP | 25 |
| 2.4 Virtual LANs..... | 26 |
| 2.5 Per-VLAN Spanning Tree | 27 |
| 2.6 Multiple Spanning Tree Protocol | 27 |
| 2.6.1 Multiple Spanning Tree com Regiões..... | 28 |
| 2.6.2 Instâncias MST | 29 |
| 2.6.3 Operações dentro de uma região | 31 |
| 2.6.4 Operações entre regiões MST | 32 |
| 2.6.5 Mecanismo de reconfiguração..... | 32 |
| CAPÍTULO 3 – Utilização do MSTP em diferentes contextos | 33 |
| 3.1 Engenharia de Tráfego em Redes Metro Ethernet | 33 |
| 3.2 Cenários Dinâmicos..... | 45 |
| 3.3 Cenários Estáticos | 56 |
| 3.4 Divisão da rede em regiões..... | 70 |
| 3.5 Outros aspectos de Engenharia de tráfego..... | 74 |
| CAPÍTULO 4 - CONCLUSÕES | 90 |
| REFERÊNCIAS..... | 92 |

CAPÍTULO 1 - INTRODUÇÃO

1.1 Enquadramento

Actualmente, os *switches* Ethernet são compatíveis com os protocolos IEEE 802.1d e o IEEE 802.1q. O 802.1d *Spanning Tree Protocol* (STP) atribui percursos de encaminhamento entre qualquer par de *switches* baseados num conjunto lógico de ligações, que abrangem todos os *switches*, sem ciclos, ou seja, uma *Spanning Tree* (ST). Este protocolo inclui detecção de mudanças da topologia de rede e reconstrução da ST para se atingir conectividade global. Numa rede com n *switches* interligados por ligações ponto-a-ponto, há $n-1$ ligações activas e as restantes ligações actuam como ligações de *backup*. Por outro lado, o protocolo 802.1q VLAN permite a definição de diferentes VLANs, criando desta forma diferentes domínios *broadcast* dentro do mesmo conjunto de *switches* (uma VLAN é um conjunto de portas, que pertencem aos mesmos ou a diferentes *switches*, com conectividade total entre elas). O tráfego de diferentes clientes é suportado por diferentes VLANs prevenindo assim que pacotes de um cliente cheguem aos portos de outro, otimizando desta forma os recursos da rede e prevenindo ataques de segurança na camada 2.

O STP foi proposto há anos atrás quando, após uma falha de rede, a recuperação da conectividade em alguns segundos era considerado um desempenho adequado. Recentemente, dois novos protocolos foram propostos pelo IEEE para melhorar a sobrevivência e a capacidade de engenharia de tráfego das redes de *switches*. Um destes protocolos é o IEEE 802.1w *Rapid Spanning Tree Protocol* (RSTP), uma evolução do STP onde os estados das portas e as suas funções são redefinidas e um mecanismo de negociação é usado para acelerar a convergência da ST sempre que a topologia da rede é alterada. O outro protocolo é o IEEE 802.1s *Multiple Spanning Tree Protocol* (MSTP) que usa os protocolos RSTP e VLAN. Com o MSTP, o operador de rede pode definir regiões diferentes onde uma *Common Spanning Tree* (CST) liga todos os *switches* de todas as regiões de tal forma que uma mudança de topologia dentro de uma região não afecta a CST fora da região. O MSTP oferece também a possibilidade de ter STs adicionais a serem configuradas dentro de cada região.

O MSTP não define como é que as redes se devem dividir em regiões, que STs devem ser criadas em cada região e como devem ser atribuídas as VLANs a cada ST. Existem, portanto, inúmeras possibilidades de melhorar este protocolo, em termos de engenharia de tráfego, proporcionando balanceamento de carga, tolerância a falhas, qualidade de serviço (QoS) e proporcionando também a possibilidade da sua aplicação em redes móveis, com a vantagem de ser uma tecnologia de baixo custo e bastante flexível.

1.2 Objectivos

Os objectivos do trabalho desenvolvido são:

- (1) estudar a evolução dos protocolos de encaminhamento para redes de camada 2 baseadas na Ethernet até à proposta do MSTP;
- (2) estudar o protocolo MSTP;
- (3) fazer um estudo da literatura científica do modo como o MSTP tem sido explorado para melhorar os diferentes aspectos do funcionamento de uma rede de telecomunicações.

1.3 Organização da dissertação

Esta dissertação está organizada em 4 capítulos.

O capítulo 1 (o presente capítulo) apresenta o enquadramento e objectivos do trabalho desenvolvido.

O capítulo 2 tem como finalidade o estudo do MSTP, começando por rever os protocolos que lhe deram origem. Compreender o funcionamento do MSTP e tudo que lhe é inerente é fundamental para que se possa analisar, com espírito crítico, o que até hoje foi feito com base neste protocolo.

No capítulo 3 é feito um estado da arte alargado, analisando diversos artigos que focam diferentes aspectos do protocolo, onde se propõem algoritmos e métodos que proporcionam melhorias em termos de engenharia de tráfego.

Finalmente, o capítulo 4 apresenta as principais conclusões do trabalho realizado.

CAPÍTULO 2 – A origem do Multiple Spanning Tree e o seu funcionamento

O IEEE 802.1s *Multiple Spanning Tree Protocol* (MSTP), proposto em 2002, é uma evolução do IEEE 802.1d *Spanning Tree Protocol* (STP). Com o crescente número de problemas associados ao aparecimento constante de esquemas mais complexos para as redes assentes na camada 2 do modelo de OSI (*Open Systems Interconnection*), especialmente referentes à redundância e ao balanceamento de carga, foi desenvolvido o MSTP, tendo sempre em vista obter o menor impacto possível em termos de desempenho. Este novo protocolo veio trazer numerosas vantagens fazendo uso de vários aspectos de outros protocolos como o *Rapid Spanning Tree Protocol* (RSTP) e *Per-Vlan Spanning Tree* (PVST), tornando-o bastante atraente.

Para melhor se poder compreender o protocolo *Multiple Spanning Tree* e a forma como este funciona, este capítulo começa por descrever cada um dos protocolos atrás mencionados e evolui, de seguida, para a sua descrição.

2.1 Spanning Tree Protocol

O *Spanning Tree Protocol* (STP) foi definido pela norma IEEE 802.1d em 1990, tendo sido actualizado em 1998. Este é um protocolo para equipamento de rede que assenta na camada 2. Este protocolo estabelece o caminho entre qualquer par de *switches* baseado numa atribuição lógica de ligações activas abrangendo todos os *switches* sem ciclos, i.e., formando uma árvore de suporte. O seu algoritmo determina as ligações a activar e considera as ligações adicionais como ligações de *backup*. Em caso de falha de uma ligação activa, o protocolo determina uma nova árvore de suporte de ligações activas (fazendo uso das ligações de *backup*) sem o perigo da existência de ciclos nem da necessidade de configuração manual das mesmas ligações. Os ciclos numa rede de *switches* devem ser evitados pois a sua existência resulta no colapso das comunicações uma vez que, quando um switch não sabe para onde deve enviar os seus pacotes, ele envia-os para todas as portas, exceptuando a porta de onde recebeu o pacote. Este fenómeno é designado por *Flooding*.

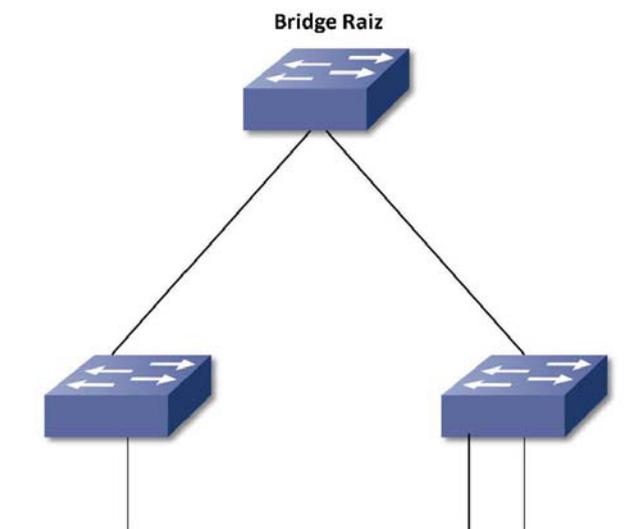


Figura 1 – Esquema de uma árvore de suporte

A Figura 1 ilustra uma árvore de suporte bastante simples onde as linhas a negrito identificam as ligações activas da mesma.

No encaminhamento baseado em *Spanning Trees* é então escolhido um *switch* como sendo a bridge raiz (*Root Bridge*). As outras bridges usam o algoritmo de *Bellman-Ford* assíncrono e distribuído para calcular o vizinho no percurso de custo mínimo para a bridge raiz. Desta forma, as ligações compostas pelos percursos de custo mínimo (de todas as outras bridges para a raiz) definem uma árvore de suporte (*Spanning Tree*).

2.1.1 Construção da árvore de suporte

A construção da árvore de suporte, no STP, é baseada em dois tipos de parâmetros inteiros positivos: o *BridgeID* atribuído a cada *switch* e o *PortCost* atribuído a cada porta de cada *switch*. Estes parâmetros podem ser configurados pelo gestor de rede com diferentes valores. O *BridgeID* de cada *switch* é identificado por um endereço que contém: 2 bytes de prioridade, configurável pelo gestor de rede, e 6 bytes fixos (um dos endereços MAC das portas da bridge ou qualquer outro endereço único desse mesmo tamanho). De notar que os 2 bytes de prioridade têm precedência sobre os 6 bytes fixos.

No início, o *switch* que tiver menor *BridgeID* passa a ser a bridge raiz (*Root Bridge*), isto é, o *switch* que fica na raiz da árvore de suporte. Seguidamente, as ligações activas são aquelas que pertencem ao percurso de menor custo de cada *switch* para a bridge raiz. Cada *switch* tem associado um custo do percurso para a raiz (*Root Path Cost*) que é a soma dos custos das portas que recebem os pacotes enviados pela raiz (porta raiz) no percurso de menor custo para o *switch*.

Se um *switch*, numa rede local é responsável pelo envio de pacotes dessa mesma rede local para a bridge raiz e vice-versa, é denominada de bridge designada (*Designated Bridge*), sendo que a bridge raiz é a

bridge designada de qualquer rede local a que esteja ligada. À porta que liga uma rede local à sua bridge designada é designada por porta designada. Assim, cada bridge designada possui uma porta designada, a qual é responsável por, numa rede local, enviar pacotes dessa mesma rede local para a bridge raiz e vice-versa. Por outro lado, numa bridge, a porta que fica encarregue de receber/transmitir pacotes de/para a bridge raiz é designada de porta raiz (*Root Port*).

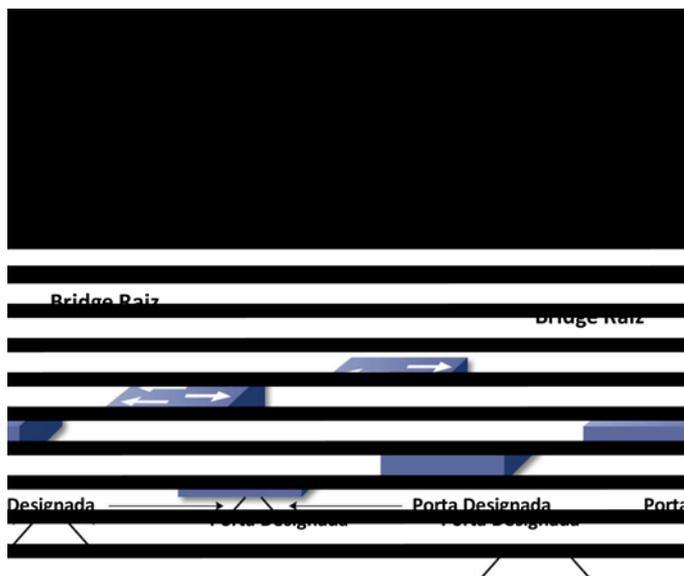


Figura 2 – Esquema de árvore de suporte com indicação das portas raiz e designadas

A construção e manutenção desta árvore de suporte são baseadas na troca de mensagens protocolares, denominadas BPDUs (*Bridge Protocol Data Units*), entre *switches*. Existem dois tipos de BPDUs: *Configuration* e *Topology Change Notification* (TCN). O *Configuration* BPDU é enviado periodicamente, de acordo com o *Hello time*, cujo valor por defeito é de 2 segundos. O TCN BPDU é emitido sempre que é detectada uma mudança na topologia por um *switch*.



Figura 3 – Formato do pacote trocado entre bridges para a construção e manutenção da *Spanning Tree*

Como se pode ver na Figura 3, no pacote trocado entre bridges existem quatro campos que antecedem a mensagem BPDU:

- **Destination:** endereço *multicast* atribuído ao protocolo;
- **Source:** endereço MAC da porta que envia o BPDU;
- **DSAP = SSAP = 01000010** (42 hexadecimal): SAP atribuído ao protocolo.

A mensagem BPDU é composta por diversos parâmetros:

- **Protocol ID:** 2 bytes;
- **Version:** 1 byte;
- **Type:** 1 byte (*Configuration BPDU* ou *Topology Change Notification*);
- **Flags:** 1 byte (1º bit é a flag *Topology Change* e o 8º bit é a flag *Topology Change Acknowledgment*);
- **Root ID:** 8 bytes;
- **Root Path Cost:** 4 bytes;
- **Bridge ID:** 8 bytes;
- **Port ID:** 2 bytes;
- **Message Age:** 2 bytes;
- **Max Age:** 2 bytes;
- **Hello Time:** 2 bytes;
- **Forward Delay:** 2 bytes;
- **Frame Padding:** 8 bytes (necessária para cumprir com o tamanho mínimo do campo de dados de uma trama Ethernet).

Desta listagem de parâmetros salientam-se os seguintes: *Root ID*, que é a estimativa actual do ID da bridge raiz, *Root Path Cost*, que é a estimativa actual do custo para a bridge raiz, *Bridge ID*, que identifica a bridge que envia a mensagem de configuração e *Port ID*, que identifica a porta que envia a mensagem de configuração.

Inicialmente, cada bridge assume que é a bridge raiz e envia mensagens de configuração para todas as suas portas. Estas mensagens têm o campo *Root Path Cost* igual a zero (por convenção, o custo do percurso da bridge raiz para si própria é zero). Todas as bridges vão armazenando as melhores mensagens de configuração recebidas em cada uma das suas portas, sendo que o ID da bridge raiz é o que está associado ao menor valor no campo *Root ID*. A bridge assume que o custo para a raiz (a sua estimativa de *Root Path Cost*) é igual à soma do custo da sua porta raiz com o *Root Path Cost* da melhor mensagem de configuração recebida até esse instante. A bridge assume que a sua porta raiz é aquela em que tenha sido recebida uma mensagem de configuração com um valor de *Root ID* coincidente com a estimativa actual do endereço da bridge raiz e que fornece menor custo para a bridge raiz, ou seja, a porta para a qual a soma do *Root Path Cost* (da melhor mensagem de configuração recebida na porta) e do *Path Cost* da porta for a menor. Se duas ou mais portas oferecerem igual custo para a raiz, recorre-se a um desempate. Em primeiro lugar desempata-se verificando o Bridge ID das mensagens recebidas nas portas. Se este valor também coincidir

o desempate é feito através do *Port ID*. Em qualquer um dos casos, a porta que prevalece é a que tiver associada menor valor de *Bridge ID* ou *Port ID*.

Depois de o algoritmo convergir, apenas uma bridge em cada rede local transmite mensagens de configuração, geradas pela bridge raiz.

2.1.2 Caso de falha na rede

Um switch recebe regularmente BPDUs de configuração enviados pela bridge raiz na sua *Root Port* (um switch nunca envia BPDUs de configuração em direcção à raiz). Para tal foi introduzido um BPDUs especial denominado *Topology Change Notification (TCN) BPDUs*. Assim, quando um switch precisa de notificar uma alteração de rede, começa a enviar TCNs pela sua *Root Port*. A *Designated Bridge* recebe o TCN, toma conhecimento da alteração, e gera outro TCN e envia pela sua *Root Port*. O processo continua assim sucessivamente até chegar à *Root Bridge*. O TCN é um BPDUs muito simples que contem apenas os três primeiros campos do BPDUs de configuração (*Protocol ID, Version, Message Type*). A *Designated Bridge* toma conhecimento do TCN enviando p próximo BPDUs de configuração com o bit do *Topology Change Acknowledgement (TCA)* a 1. A bridge que detectou a alteração da rede não pára de enviar TCNs até que a sua *Designated Bridge* tome conhecimento dessa alteração. Logo, a *Designated Bridge* responde ao TCN enviando o próximo BPDUs de configuração da sua *Root Bridge* com o bit do *Topology Change Acknowledgement (TCA)* a 1.

Quando a raiz se apercebe de que ocorreu uma alteração na topologia na rede, começa a enviar BPDUs de configuração com o bit *Topology Change (TC)* a 1. Esses BPDUs são reenviados para a rede por cada *switch*. Desta forma, todos os switches tomam conhecimento da alteração na topologia e o *Aging Time* das tabelas de encaminhamento é reduzido para o *Forward Delay*.

Ao ocorrer uma alteração na topologia da rede pode existir uma perda temporária de conectividade se uma porta que estava inactiva na topologia antiga ainda não passou ao estado de activa na topologia nova. Podem também existir ciclos temporários se uma porta que estava activa na topologia anterior ainda não passou ao estado de inactiva na topologia actual. Esta última possibilidade é mais grave. Para minimizar a probabilidade de ocorrência de ciclos temporários, os *switches* são obrigadas a esperar algum tempo antes de permitirem que uma das suas portas passe do estado inactivo para o estado activo. Este tempo de espera é calculado em função do parâmetro *Forward Delay*.

Quando o STP foi desenvolvido, este protocolo era capaz de fazer face às necessidades das redes da época. O STP foi capaz de oferecer redes de camada 2 redundantes ao mesmo tempo que evitou ciclos, sendo esta a sua grande vantagem. Durante bastante tempo, este protocolo foi capaz de servir os seus utilizadores com uma certa qualidade mas, com o crescimento tecnológico e crescimento do número de utilizadores, as empresas começaram a enfrentar diversos problemas com a sua utilização. Este protocolo tornou-se insuficiente devido à lenta convergência para uma topologia activa. Mover uma porta do estado *Blocking* para o estado *Forwarding* demora, regra geral, 30 segundos. Isto significa que um computador

recentemente ligado à rede só poderá receber/enviar dados decorrido esse tempo, sendo este um exemplo da lenta convergência. Outro exemplo é a falha de uma ligação que liga dois *switches*. Esta falha acciona a convergência do *Spanning Tree Protocol*, fazendo com que parte da rede fique isolada até que o processo de convergência seja concluído, o que demora, regra geral, 50 segundos.

2.1.3 Estado das portas

Cada porta de um *switch* deverá passar por vários estados de portas diferentes à medida que o protocolo STP evolui para formar a árvore de suporte. Neste protocolo os estados das portas são:

1. **Blocking:** estado para o qual a porta transita quando a porta não está no percurso de custo mínimo para a *bridge raiz* (quando se liga um equipamento, quer seja um outro switch ou um terminal, a uma porta de um switch activo, essa mesma porta fica inicialmente no estado *Blocking*). Os processos de aprendizagem e de expedição de pacotes estão inibidos, a porta apenas recebe e processa mensagens de configuração.
2. **Listening:** do estado *Blocking*, a porta passa para o estado *Listening* e aqui permanecerá durante um período de tempo igual a *Forward Delay*. Os processos de aprendizagem e de expedição continuam inibidos, recebe e processa mensagens BPDU de configuração.
3. **Learning:** após um tempo no estado *Listening*, a porta passa para o estado *Learning* e aqui permanece durante um período de tempo igual a *Forward Delay*. O processo de aprendizagem está activo mas o de expedições de pacotes está inibido, continuando a receber e processar mensagens BPDU de configuração.
4. **Forwarding:** é o estado activo, tanto o processo de aprendizagem como o de expedição de pacotes estão activos, recebendo e processando mensagens de BPDU configuração.
5. **Disabled:** os processos de aprendizagem e de expedição de pacotes estão inibidos, as portas neste estado não participam no algoritmo de *Spanning Tree*.

A Figura 4 apresenta as transições possíveis entre os diferentes estados e quais os eventos que as determinam.

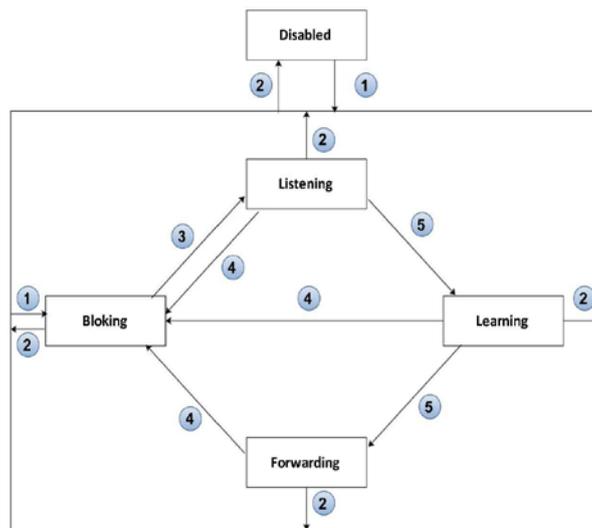


Figura 4 – Diagrama de estados das portas no STP

Legenda:

1. Porta activada por gestão ou por inicialização;
2. Porta desactivada por gestão ou falha;
3. Algoritmo selecciona como sendo porta raiz ou porta designada;
4. Algoritmo selecciona como não sendo porta raiz ou porta designada;
5. *Forwarding Timer* expira.

2.2 Rapid Spanning Tree Protocol

Em 2001 foi desenvolvido o *Rapid Spanning Tree Protocol* (RSTP), norma 802.1w [3], com a finalidade de acelerar o processo de alteração da árvore de suporte quando há alteração da topologia da rede por forma a que este processo possa acontecer num tempo na ordem das dezenas de milissegundos.

O seu funcionamento é bastante semelhante ao STP, sendo compatível com o mesmo (conforme será explicado mais à frente), existindo algumas diferenças.

2.2.1 Estado e Função das portas

Uma das alterações introduzidas neste protocolo está nos estados das portas, como é apresentado na Tabela seguinte:

| Estado Operacional | Estado STP | Estado RSTP |
|--------------------|-------------------|-------------------|
| <i>Disabled</i> | <i>Disabled</i> | <i>Discarding</i> |
| <i>Enabled</i> | <i>Blocking</i> | <i>Discarding</i> |
| <i>Enabled</i> | <i>Listening</i> | <i>Discarding</i> |
| <i>Enabled</i> | <i>Learning</i> | <i>Learning</i> |
| <i>Enabled</i> | <i>Forwarding</i> | <i>Forwarding</i> |

Tabela 1 – Relação entre o estado operacional das portas e os estados STP e RSTP [3]

Pela observação da Tabela 2 sabe-se que as portas no RSTP têm apenas três estados (*Discarding*, *Learning* e *Forwarding*), sendo que os estados *Disabled*, *Blocking* e *Listening* foram agregados no estado *Discarding*. Já os estados *Learning* e *Forwarding* são os mesmos em ambos os protocolos.

Qualquer porta pode encontrar-se no estado *Discarding* quando existe uma topologia activa ou na fase de sincronização de uma topologia. Este estado impede o envio de tramas para evitar ciclos. Como no estado anterior, o estado *Learning* pode encontrar-se numa topologia activa ou na fase de sincronização da mesma, mas as portas neste estado analisam as tramas de dados para construir as suas tabelas de encaminhamento. O estado *Forwarding* existe apenas quando a respectiva porta pertence à topologia activa (o conjunto das portas em *forwarding* determina a topologia activa em cada momento). Neste caso, após uma alteração de topologia ou durante a sincronização, o encaminhamento de tramas apenas acontece depois de haver acordo de topologia.

Para além das alterações em termos do estado das portas, há também diferenças a nível funcional das portas. Neste protocolo existem duas novas funções das portas em relação às do STP. Estas funções das portas são designadas de *Alternate Port* e *Backup Port*, como se pode ver na Tabela 2:

| Função de portas – STP | Função de portas - RSTP |
|------------------------|-------------------------|
| <i>Root Port</i> | <i>Root Port</i> |
| <i>Designated Port</i> | <i>Designated Port</i> |
| <i>Blocking</i> | <i>Alternate Port</i> |
| <i>Blocking</i> | <i>Backup Port</i> |

Tabela 2 – Relação entre a função das portas do STP e RSTP [3]

Pode-se também associar funções de portas a estados específicos de portas RSTP. No estado *Forwarding* temos as *Root Port* e as *Designated Port*, e no estado *Discarding* temos as *Alternate Port* e as *Backup Port*.

Considere-se o exemplo apresentado na Figura 5.

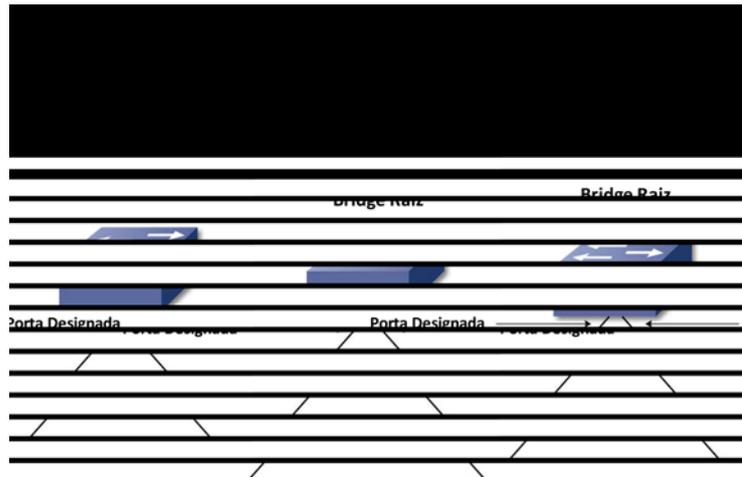


Figura 5 – Esquema de árvore de suporte com indicação de porta *Alternate* e porta de *Backup*

Uma *Alternate Port* é uma porta que oferece um caminho alternativo para uma bridge, não sendo esta de uma *Designated Bridge*. Esta assume o estado *Discarding* numa topologia activa estável. Uma *Alternate Port* só pode existir num *switch* que não seja bridge designada para uma determinada rede local. Em caso de falha da *Designated Bridge* dessa mesma rede, a *Alternate Port* altera o seu estado para *Designated Port*. De notar que no STP esta porta estaria no estado *Blocking*, enquanto que na nova forma, havendo uma falha, a recuperação da rede é efectuada de forma mais rápida.

Quando uma *Designated Bridge*, de uma determinada rede local, tem para além de uma *Designated Port*, tem uma porta adicional para essa mesma rede, correspondendo a uma ligação redundante, diz-se que possui uma *Backup Port*. Esta porta tem um *Port ID* superior ao da designada assumindo o estado *Discarding* numa topologia activa. O protocolo em questão oferece uma maior importância a estas portas, que normalmente estariam no estado *Blocking* no STP.

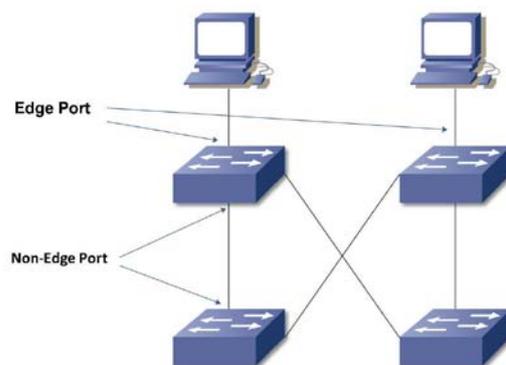


Figura 6 – Esquema de uma rede com indicação de *Edge Port* e *Non-Edge Port*

O RSTP faz também a distinção entre *Edge Port* e *Non-Edge Port*. Pela análise da Figura 6 pode-se dizer que quando uma porta não faz ligação a um outro *switch* é denominada por *Edge Port*. Caso contrário, se uma porta faz ligação a um outro *switch* é uma *Non-Edge Port*. Se uma *Edge Port* receber um BPDU passa a *Non-Edge Port*, o que quer dizer que em vez de fazer ligação a um equipamento terminal (por exemplo um PC) passa a fazer ligação a um *switch*.

2.2.2 Bridge Protocol Data Unit (BPDU)

O formato do BPDU permanece o mesmo em relação ao STP, garantindo a interoperabilidade com o 802.1d *Spanning Tree Protocol*.

As diferenças entre 802.1d e 802.1w em relação às mensagens trocadas na rede, consiste numa alteração nos campos *Protocol Version Identifier*, *BPDU type* e *Flags*, como se pode observar na Figura 7.

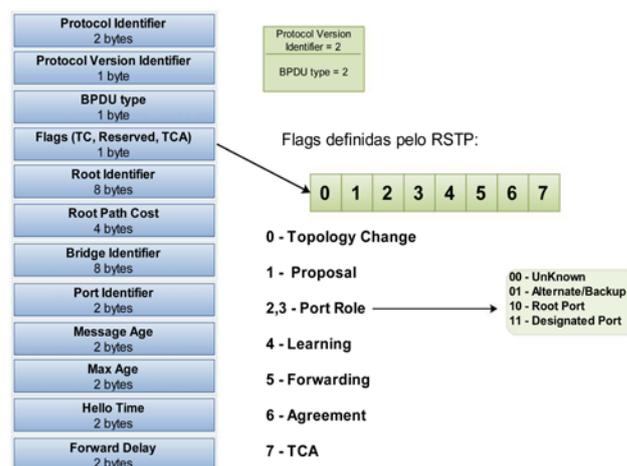


Figura 7 – Esquema das alterações nas *Flags* das mensagens BPDu [5]

No STP são apenas utilizadas duas *Flags*, *Topology Change* e *Topology Change Acknowledgement*, sendo que os restantes bits não têm qualquer significado. Já no RSTP, os seis bits restantes do campo *Flags* são usados para conferir uma maior funcionalidade a este protocolo.

O funcionamento das *Flags Proposal/Agreement* desempenha um papel fundamental para acelerar a mudança de estados das portas, num *switch* que participe no RSTP (este será explicado mais à frente com detalhe). Para identificar o *Port Role* são usados dois bits conforme a função da porta em questão. As *flags Learning* e *Forwarding* servem para identificar o estado da porta que estiver a transmitir o BPDu. Estes dois campos auxiliam bastante na tarefa de prevenção dos ciclos.

Os benefícios trazidos pela norma IEEE 802.1w RSTP vão para além dos novos estados de portas, as suas funções e o formato do BPDU. A forma como este protocolo opera é bastante diferente em comparação com o protocolo de norma IEEE 802.1d STP. Neste último somente a *Root Bridge* poderá gerar os BPDUs de configuração, enviando-os para a rede pelas suas *Designated Ports*, os restantes *switches* recebem estes BPDUs de configuração pela sua *Root Port*, fazem a alteração nos campos necessários e reenviam, de seguida, os BPDUs alterados pelas suas *Designated Ports*. Este procedimento foi alterado no RSTP, onde todos os *switches* são capazes de gerar os seus próprios BPDUs e enviá-los em intervalos regulares (definidos pelo *Hello Time*). Desta forma, a cada 2 segundos (tempo predefinidos) os *switches* criam os seus próprios BPDUs que são enviados pelas suas *Designated Ports* permitindo desta forma o processo de *Fast Aging*.

2.2.3 Fast Aging

No RSTP os BPDUs são utilizados como um mecanismo de *Keep Alive* entre os *switches*. O *Keep Alive* é um método bastante conhecido para determinar a disponibilidade de equipamentos de rede, etc. Como todos os *switches* geram os seus BPDUs e enviam-nos em intervalos regulares (*Hello time*, 2 segundos por defeito) pelas suas portas designadas, os *switches* que os recebem, sabem que aquele equipamento está activo, mantendo-o na sua tabela de endereços MAC. Caso contrário, se uma determinada porta não receber os BPDUs consecutivamente da sua *Designated Bridge*, o *switch* elimina as informações sobre a topologia da rede apreendidas via ligação/porta para o qual o *switch* não recebeu os referidos BPDUs.

O grande benefício trazido pelo *Fast Aging* é o facto de permitir a detecção de falhas muito mais rapidamente do que os métodos suportados pelo STP (que faz uso do campo *Max Age*). No RSTP, os *switches* utilizam os BPDUs para assegurar a conectividade da rede e detectar mais rapidamente *switches* vizinhos que se desligaram, em vez de depender do *Max Age*, resolvendo assim a questão dos 20 segundos de atraso associados a esse campo. Uma outra vantagem em relação a este mecanismo reside no facto de qualquer *switch* que não receba três *Keep Alive* (BPDUs) consecutivos, sabe que o *switch* que está na outra ponta da ligação está com problemas. Já no STP, após a expiração do *Max Age*, os *switches* não possuem conhecimento do local exacto onde ocorreu a falha na rede.

2.2.4 UplinkFast, BackboneFast e PortFast

O IEEE, ao desenvolver a norma 802.1w RSTP, procurou tirar partido do melhor da norma 802.1d STP, utilizando também conceitos criados pela Cisco, como Backbonefast, UplinkFast e PortFast, adicionando recursos extra.

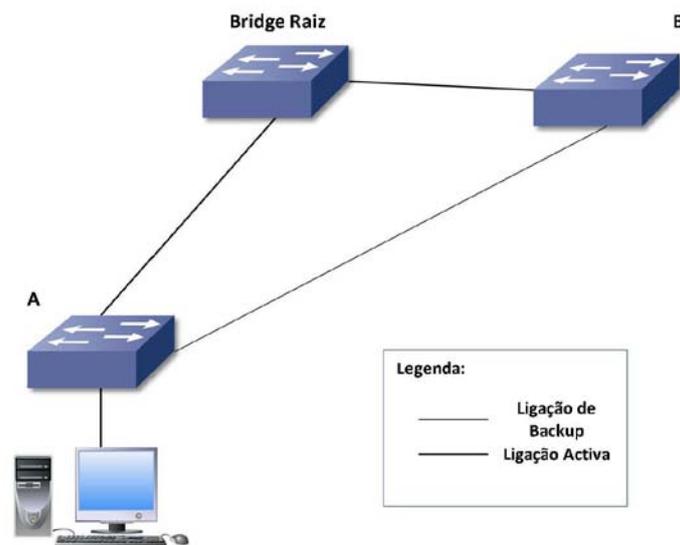


Figura 8 – Esquema de rede local

Conforme ilustra a Figura 8, pode-se verificar que o recurso *UplinkFast* permite que o *switch A* tenha uma porta prontamente disponível para ser usada, caso a ligação activa no momento seja sujeita a uma falha. A porta de *Backup* é mantida no estado *Blocking* sendo instantaneamente activada caso seja detectada uma falha na porta principal, passando a enviar/receber informação para/de a raiz através do *switch B*. Este método faz alterações relativas ao custo da porta e do *switch* para que não seja usado o Backup link para chegar à raiz da *Spanning Tree*.

Já o método *BackboneFast* permite mudar o estado das portas rapidamente mas de forma diferente. Quando um *switch* recebe informação inferior proveniente da sua bridge designada ou da raiz, este aceita-a imediatamente e substitui pela informação guardada anteriormente. Considere-se o exemplo da Figura 8. Se existir uma falha entre o *switch B* e a raiz, o *switch B* começa a enviar BPDUs com a informação de que este é a nova raiz. No entanto, o *switch A* sabe que a raiz ainda se encontra activa, logo envia imediatamente um BDU ao *switch B* contendo informação da raiz e que pode ser acedida através dele. Como resultado, o *switch B* pára de enviar os seus BPDUs e aceita a porta do *switch A* como uma nova porta raiz.

Na versão proprietária da Cisco, é utilizado o termo *Edge Port* para permitir a configuração do *PortFast*. Como referido anteriormente, uma *Edge Port* é simplesmente uma porta onde não há *switches* ligados (apenas terminais). Por ser possível receber BPDUs numa *Edge Port*, são nestas portas onde a configuração do *PortFast* é recomendada. Com este método, eliminam-se os 30 segundos necessários para mover uma porta do estado *Blocking* para o estado *Forwarding*. O problema começa quando uma bridge é ligada inadvertidamente numa *Edge Port* da bridge onde deveria estar ligado um terminal. Ao receber BPDUs, o *PortFast* é desactivado e a porta passa pelos procedimentos *normais*. Mesmo assim é possível que ocorram ciclos temporários.

Uma melhoria do *PortFast* efectuada pelo RSTP possibilita também a activação do *PortFast* em ligações ponto-a-ponto, mesmo havendo dois *switches* ligados nos seus extremos. Para que possa funcionar, a ligação deverá estar em modo de operação Full-Duplex, sem excepção. Uma vez existindo ligações Full-Duplex, é quase certo estar-se na presença de uma ligação ponto-a-ponto, não oferecendo assim condições para a formação de ciclos. De notar que a porta que faz a ligação ponto-a-ponto não poderá levar à raiz, pois dessa forma já existiria possibilidade de ciclos.

De seguida, considera-se um exemplo que permite comparar o STP e o RSTP em termos de convergência da árvore de suporte.

Spanning Tree Protocol (IEEE 802.1d):

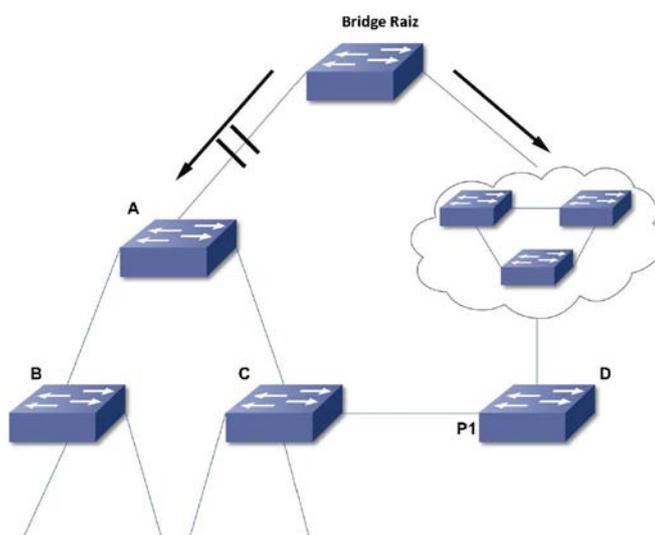


Figura 9 – Exemplo de uma rede com STP configurado [5]

A Figura 9 representa a adição de uma ligação entre o *switch* A e a raiz, sendo o acesso anterior do *switch* A para a raiz efectuada através do *switch* C (ligação C-D). Ao ser estabelecida esta nova ligação, as portas entre o *switch* A e a raiz são colocadas no estado *Listening* e não deixam fluir o tráfego pela ligação. Como o *switch* A consegue receber BPDUs directamente da raiz pela nova ligação, este propaga-os pelas suas portas designadas.

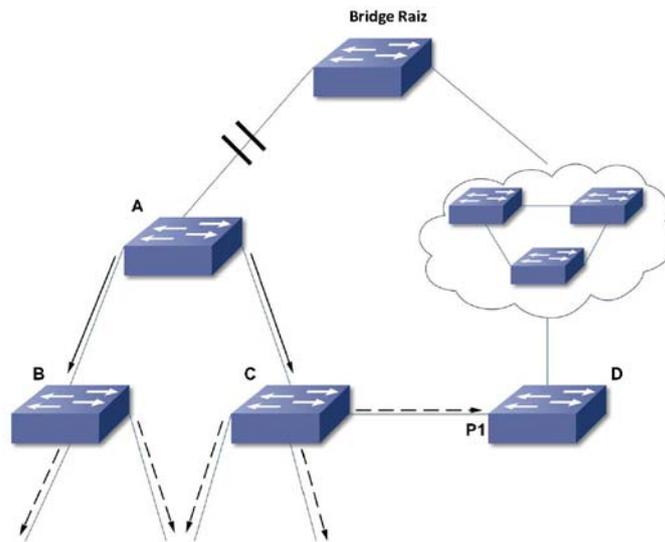


Figura 10 - Exemplo de uma rede com STP configurado [5]

Quando os *switches* B e C recebem este novo BPDU indicando que o custo para a raiz é menor pelo *switch* A, estas reencaminham imediatamente essa informação pelas suas portas.

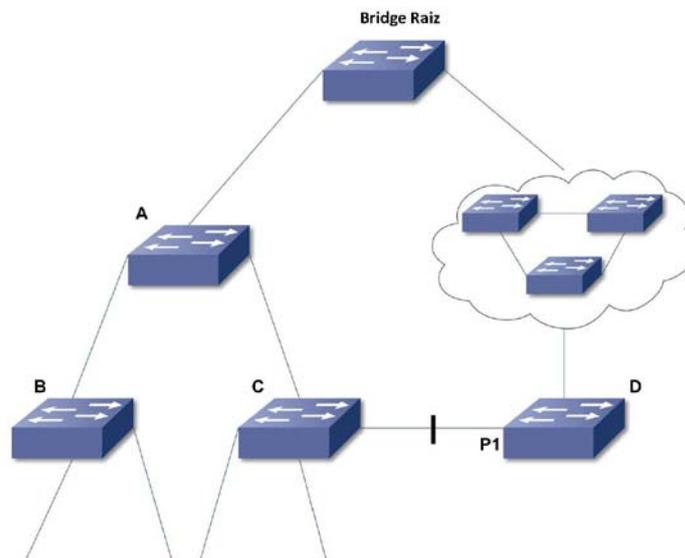


Figura 11 - Exemplo de uma rede com STP configurado [5]

Em alguns segundos o *switch* D recebe o BPDU da raiz e bloqueia instantaneamente a sua porta P1.

A construção da árvore foi eficazmente calculada com esta nova alteração na rede. No entanto, o único problema é o tempo que este processo demora a estar terminado, que é igual ao dobro do *Forward Delay* (30 segundos por defeito). Durante este tempo há uma quebra em todo o tráfego da rede. Analisemos o caso do RSTP na mesma situação.

Rapid Spanning Tree Protocol (IEEE 802.1w):

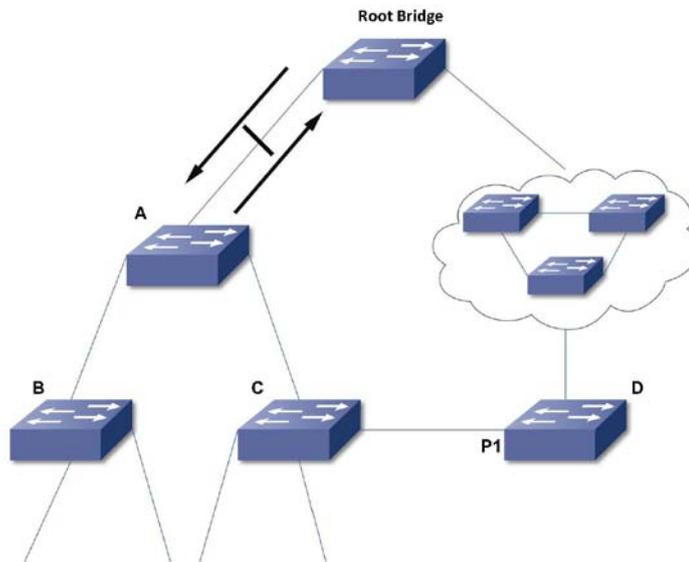


Figura 12 - Exemplo de uma rede com RSTP configurado [5]

Quando esta nova ligação é estabelecida (*switch A – raiz*) ambas as portas ficam no estado *Blocking* e inicia-se então uma negociação entre os *switches*.

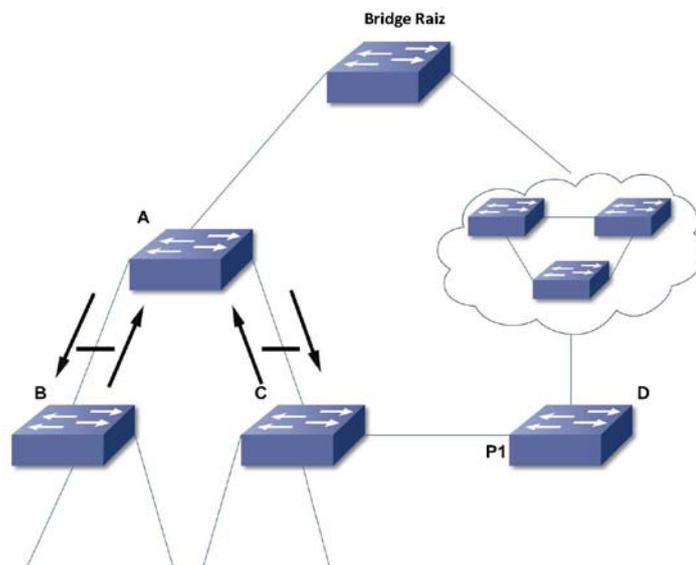


Figura 13 - Exemplo de uma rede com RSTP configurado [5]

O *switch A* recebe o BPDU da raiz e bloqueia instantaneamente as suas *Designated Ports* que não sejam *Edge Ports*, esta operação é denominada *sync*. Quando o *switch A* bloqueia essas portas coloca de imediato a sua porta para a raiz no estado *Forwarding*. Esta é a grande diferença para o STP, uma vez que

neste protocolo as portas vão sendo bloqueadas à medida que se vão originando novos BPDUs. Neste ponto irá também haver uma negociação entre o *switch* A e os seus vizinhos (*switches* B e C). O *switch* B como, para além da sua ligação com o *switch* A, só possui *Edge Ports*, vai colocar as suas portas no estado *Forwarding*.

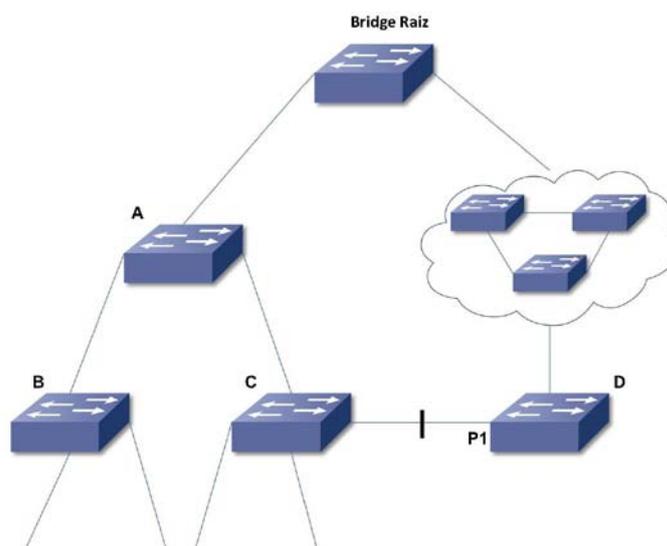


Figura 14 - Exemplo de uma rede com RSTP configurado [5]

Realiza-se o mesmo processo entre o *switch* C e o *switch* D e chega-se à mesma situação que no STP, ficando bloqueada a porta P1.

A árvore de suporte foi gerada sem a utilização de um *timer*, o único mecanismo novo introduzido pelo RSTP é a passagem imediata para o estado *Forwarding* que um *switch* pode assumir na sua nova porta raiz, não necessitando assim o tempo de espera de transição para o estado *Forwarding* (não utiliza os passos *Listening* e *Learning*). A negociação que cada *switch* é sujeita chama-se de *Proposal/Agreement handshake* em que utiliza os bits referidos anteriormente no campo *Flags* do BPDUs, isto é assinala o bit *Proposal* quando um *switch* inicia uma negociação e se houver acordo entre estas, o *switch* destino assinala o bit *Agreement* e reenvia o BPDUs para a origem para indicar o sucesso da negociação.

Em suma, o mecanismo *Proposal/Agreement* é accionado quando um *switch* recebe uma mensagem *Proposal* numa das suas portas e essa porta é seleccionada como porta raiz. Então, todas as outras portas serão sincronizadas com a nova informação. Só depois de obter a confirmação de que todas as portas estão sincronizadas com a nova informação, o *switch* envia uma mensagem *Agreement* para a *Designated Bridge* correspondente à sua *Root Port*. As portas, que não são *Root Port* nem *Edge Port*, actuam de forma idêntica para decidir se ficam no estado *Forwarding* ou *Blocking* relativamente aos outros *switches*.

2.2.5 Novos mecanismos para detecção de alteração da rede

No RSTP, somente as portas que não são *Edge Ports* e que estão a transitar para o estado *Forwarding* podem causar um *Topology Change* (TC), ou seja, uma porta que está no estado *Blocking* não pode gerar um TC como acontecia no STP. Assim, quando um *switch* detecta uma alteração na rede acontece o seguinte:

- Inicia um *TC While Timer* com um valor igual ao dobro do *Hello Time* para todas as *Designated Ports* e *Non Edge Ports* e, se necessário, a sua *Root Port*;
- Inicia as suas portas com os endereços MAC associados a essas portas;
- Enquanto o *Timer* estiver activo numa porta, são enviados BPDUs com o TC bit a um, sendo enviado também pela sua *Root Port*.

Quando um *switch* recebe um BPDUs com o TC bit activo de um vizinho desencadeia-se os seguintes procedimentos:

- O *switch* limpa os seus endereços MAC apreendidos em todas as suas portas com a excepção da porta onde recebeu este BPDUs;
- Inicia o seu *TC While Timer* e envia BPDUs com TC bit activo em todas as suas *Designated Ports* e na *Root Port*.

O *switch* detecta uma alteração na rede e envia BPDUs com essa informação para todos os *switches* vizinhos, ao contrário do STP em que somente a raiz poderia efectuar esta operação. Assim a propagação do TC é bastante mais rápida pois já não existe a necessidade de esperar que a raiz seja notificada e somente depois colocar a rede em estado de alteração durante um tempo especificado pela soma de *Max Age* e *Forward Delay*.

2.3 Melhorias e compatibilidade do RSTP em relação ao STP

O formato do RSTP BPDUs é usado somente quando existem *switches* com o protocolo activo. Os *switches* com o STP não reconhecem os RSTP BPDUs e descartam estas mensagens. Contudo, os *switches* com RSTP activo, reconhecem os BPDUs com o formato do STP. Quando um *switch* tem o STP configurado numa ligação particular, este adapta os seus BPDUs e TCN para o STP nas suas portas associadas, garantindo desta forma uma interoperabilidade entre estes dois protocolos.

O RSTP age de forma proactiva não sendo necessários os *Timers Delay* existentes no STP. O processo de reiniciar uma nova topologia pode ser despoletado por falta de mensagens *Hello* e também por detecção de anomalia de uma ligação. O RSTP é portanto uma evolução do STP, otimizando uma série de procedimentos que lhe garante um menor tempo de convergência.

2.4 Virtual LANs

Uma rede local virtual, chamada de *VLAN*, é uma rede logicamente independente, sendo que podem coexistir várias *VLANs* num mesmo comutador (*switch/bridge*). O protocolo, predominantemente utilizado nestas redes virtuais, é o IEEE 802.1q [2].

A implementação destas redes virtuais permite a partição de uma rede local em diferentes segmentos lógicos (criação de diferentes domínios *Broadcast*), permitindo que utilizadores fisicamente ligados a diferentes switches estejam virtualmente ligados a um único switch. Uma vez que as *VLANs* proporcionam uma alta flexibilidade numa rede local, estas redes tornam-se ideais em ambientes corporativos. As *VLANs* possuem inúmeras vantagens, entre os quais:

- **Controle do tráfego *Broadcast*:** em redes onde o tráfego *Broadcast* é responsável por grande parte do tráfego total, as *VLANs* reduzem o número de pacotes enviados para destinos desnecessários, aumentando a capacidade total da rede;
- **Segmentação lógica da rede:** as redes virtuais podem ser criadas com base na organização sectorial de uma empresa; cada *VLAN* pode estar associada aos utilizadores de um departamento ou de um grupo de trabalho, mesmo que os seus membros estejam fisicamente distantes, como referido anteriormente;
- **Redução de custos e facilidade de gestão:** grande parte do custo de uma rede deve-se ao facto da movimentação dos utilizadores, sendo necessária uma nova ligação através do uso de cabos, novo endereçamento e nova configuração dos equipamentos; com o uso de *VLANs*, a inclusão ou movimentação de utilizadores pode ser efectuada de forma remota pelo gestor de rede (da sua própria estação), sem a necessidade de movimentação física, conferindo uma maior flexibilidade;
- **Independência da topologia física:** as *VLANs* proporcionam uma independência da topologia física da rede, permitindo que grupos de trabalhos, fisicamente diversos, possam ser ligados logicamente por um único domínio *Broadcast*;
- **Maior segurança:** o tráfego de uma *VLAN* não pode ser interceptado por terminais de outra rede virtual, já que estas não se comunicam sem que haja um dispositivo de rede desempenhando a função de um Router entre elas; desta forma, o acesso a servidores que não estejam na mesma *VLAN* é restrito, proporcionando uma maior segurança.

Outra característica das *VLANs* é a possibilidade de criação de uma *Common Spanning Tree (CST)*, sendo definida como uma única árvore ST para toda a rede, independentemente da quantidade de *VLANs* existentes nos *switches*. Consequentemente, ao ser posta em operação uma rede de *switches*, todas as *VLANs* que participem nessa rede vão ser mantidas por uma única instância STP. De facto, implementar redes com o modelo CST é muito simples e exige pouco do administrador de rede.

Por este conjunto de características das *VLANs*, pode-se afirmar que apresentam um desempenho superior às tradicionais redes locais.

2.5 Per-VLAN Spanning Tree

Baseado na norma IEEE 802.1d STP, o *Per-VLAN Spanning Tree* (PVST) proposto pela CISCO é um protocolo que permite o balanceamento de carga em redes de camada 2. O PVST mantém uma instância STP para cada VLAN agindo como uma rede independente. O PVST consiste, essencialmente, na configuração dos *switches* para que os mesmos possam distribuir as *VLANs* pelas diferentes ligações, permitindo desta forma um melhor *Balanceamento de carga* da rede. Embora este modelo tenha esta vantagem, o PVST traz também algumas desvantagens, uma vez que para cada VLAN suportada pela rede vai existir uma instância STP. Por este motivo, em redes de maior escala, cada *switch* poderá ter dezenas ou mesmo centenas de instâncias STP em execução. Em termos de desempenho, este modelo pode levar a uma utilização de recursos (CPU e memória) muito mais intensa do que o modelo CST.

2.6 Multiple Spanning Tree Protocol

O PSVT proposto pela Cisco foi vantajoso ao permitir o chamado balanceamento de carga em esquemas altamente redundantes. Para além de ser uma norma proprietária, o grande problema do PVST é o excesso de instâncias STP que pode existir. O *Multiple Spanning Tree Protocol* (MSTP), definido pela norma IEEE 802.1s [4], foi proposto em 2002 e é inspirado na implementação do *Multiple Instances Spanning Tree Protocol* (MISTP) da Cisco.

Considere-se o exemplo apresentado na Figura 15.

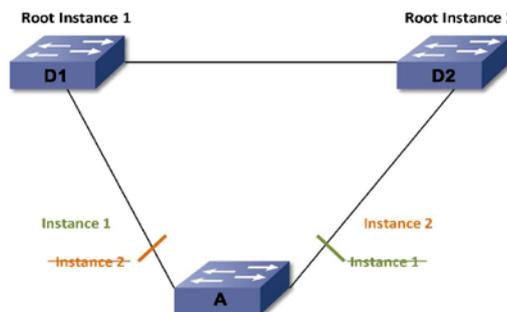


Figura 15 – Rede com o MSTP configurado [6]

Esta figura mostra uma rede comum de acesso do *switch A*, estando 1000 *VLANs* ligadas em redundância, a dois *switches* de distribuição, D1 e D2. Pretende-se que os utilizadores ligados ao *switch A* dividam o tráfego pelos dois *switch* D1 e D2.

Usando apenas as *VLANs*, a norma original IEEE 802.1q define uma CST que apenas considera uma instância ST para toda a rede de *switches*, independentemente do número de *VLANs*. Se for aplicado este modelo a este exemplo, como resultado tem-se o seguinte:

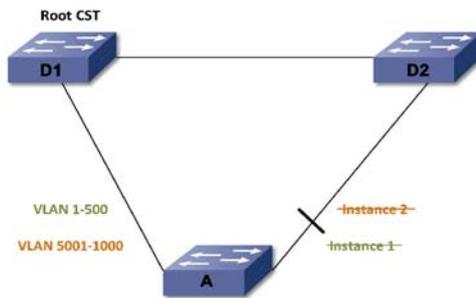


Figura 16 – Rede considerada anteriormente configurada com a norma 802.1q [6]

Numa rede configurada como ilustra a Figura 16, não é possível ter-se *balanceamento de carga*, onde uma ligação activa tem que ser bloqueada para todas as VLANs. Por outro lado o processamento do CPU é mínimo uma vez que se usa apenas uma instância ST.

Num ambiente *Per-VLAN Spanning Tree (PVST)*, os parâmetros ST são atribuídos de forma a que as VLANs sejam igualmente distribuídas pelas duas ligações activas. Desta forma, vamos ter na instância 1: VLAN 1 até à VLAN 500 e na instância 2: VLAN 501 até à VLAN 1000. Neste caso específico consegue-se obter resultados óptimos em relação ao balanceamento de carga sendo mantida uma instância ST para cada VLAN, isto é, 1000 instâncias para duas topologias lógicas diferentes. Este panorama tem a desvantagem referida anteriormente, exige demasiado processamento do CPU para todos os switches da rede.

Usando o protocolo IEEE 802.1s MSTP, este protocolo combina o melhor do PVST e da norma 802.1q *Virtual Bridged LAN*. O objectivo é suportar várias VLANs no menor número de instâncias ST possível, uma vez que a maior parte das redes necessitam de um número reduzido de topologias lógicas. Na topologia analisada na Figura 15 existem apenas duas topologias lógicas logo, só são necessárias duas instâncias ST. Se se atribuir metade das VLANs a cada instância podemos atingir o *balanceamento de carga* pretendido e o processamento do CPU é reduzido (uma vez que há apenas duas instâncias).

Assim, do ponto de vista técnico, o MST é a melhor solução. Na perspectiva do administrador de rede, o maior inconveniente associado ao uso do MSTP é o facto de este ser mais complexo do que o STP e requerer um maior conhecimento do funcionamento do protocolo.

2.6.1 Multiple Spanning Tree com Regiões

Como anteriormente mencionado, a principal melhoria introduzida pelo MSTP é o facto de várias VLANs poderem ser atribuídas a uma única instância ST. Isto levanta o problema de como determinar quais as VLANs que são associadas a cada instância, mais precisamente, como rotular BPDUs para que os equipamentos que recebem as mensagens possam identificar as instâncias e as VLANs suportadas pela rede.

Este aspecto é irrelevante no caso da norma 802.1q *Virtual Bridged LAN*, uma vez que todas as VLANs são atribuídas a uma única instância. No caso do PVST, diferentes VLANs transportam os BPDUs pelas suas VLANs respectivas (um BPDU por VLAN).

O MISTP da Cisco envia um BPDU para cada instância, incluindo uma lista de VLANs que o BPDU é responsável. No caso de dois *switches* serem mal configurados e terem gamas diferentes de VLANs associadas a uma mesma instância, é difícil para o protocolo recuperar desta situação.

O IEEE 802.1s adoptou uma abordagem mais fácil e simples que introduz o conceito de regiões MST. Pode-se pensar numa região como sendo um grupo de *switches* debaixo de uma administração comum.

Cada *switch* que corre o MSTP na rede tem uma configuração MST única que consiste em três atributos:

- Um nome alfanumérico configurado (32 bytes);
- Um número de revisão configurado (2 bytes);
- Uma tabela com 4096 elementos que associa cada uma das potenciais 4096 VLANs a uma determinada instância.

Para fazer parte de uma região MST comum, um grupo de *switches* deve partilhar os mesmos atributos configurados. Cabe ao administrador de rede configurar correctamente todos os switches de uma mesma região. Actualmente, esta etapa só é possível através da interface da linha de comando ou através do *Simple Network Management Protocol* (SNMP).

Para assegurar uma atribuição das VLANs às instâncias de forma consistente, é necessário que o protocolo consiga identificar as fronteiras das regiões. Para que tal aconteça, as características da região são incluídas nos BPDUs. A atribuição das VLANs às instâncias não se propaga nos BPDUs, uma vez que os *switches* só precisam de saber se estão na mesma região que o vizinho. Apenas um resumo da tabela de atribuições é enviado, juntamente com o número e o nome de configuração. Quando um *switch* recebe um BPDU, esse *switch* extrai o resumo e compara com o seu próprio resumo. Se os dois resumos diferirem, a porta na qual o *switch* recebeu o BPDU está na fronteira de uma região.

2.6.2 Instâncias MST

De acordo com as especificações do IEEE 802.1s, um *switch* MST deve ser capaz de suportar (i) uma *Internal Spanning Tree* (IST) e (ii) uma ou mais *Multiple Spanning Tree Instance(s)* (MSTIs). De seguida descreve-se separadamente cada uma.

Instância IST

Para se perceber o papel de uma instância IST, convém lembrar que o MSTP foi proposto pelo IEEE. Assim, o MST foi concebido de forma a interagir com a norma IEEE 802.1q *Virtual Bridged LAN*, uma vez que

esta última também foi proposta pela mesma entidade. Para o 802.1q, uma rede de *switches* implementa apenas uma ST (CST). A instância IST é simplesmente uma instância RSTP que estende a CST para dentro das regiões MST. Uma instância IST recebe e envia BPDUs para a CST, uma vez que uma IST representa inteiramente uma região MST.

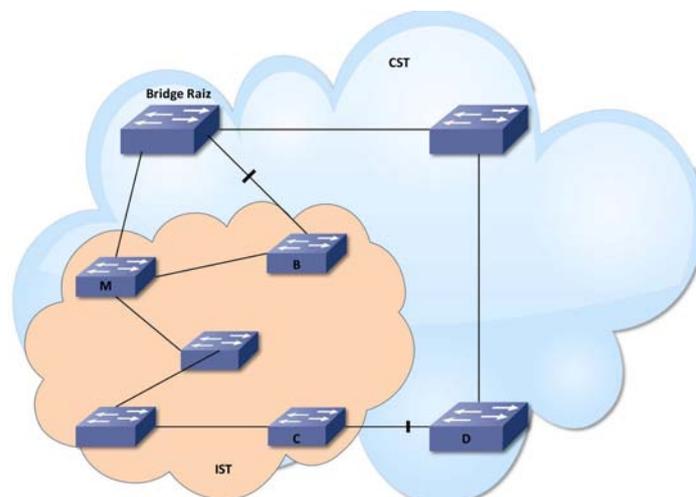


Figura 17 – Rede de *switches* MST com IST [6]

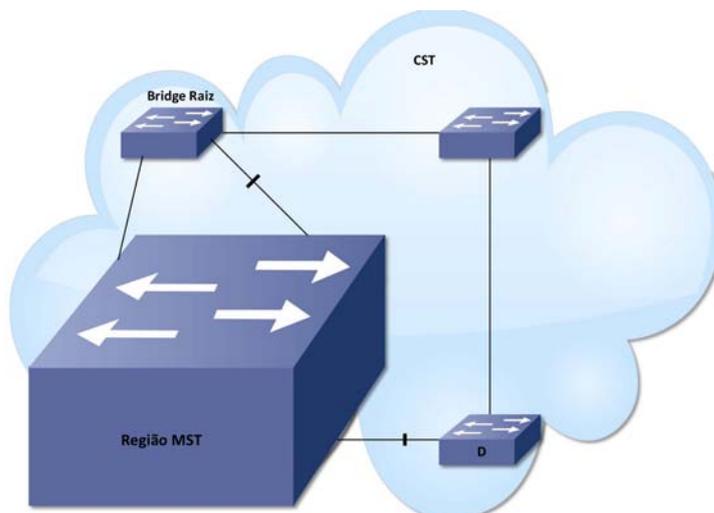


Figura 18 – Rede de *switches* MST onde o IST é visto como um *switch* virtual [6]

Considere-se o exemplo apresentado na Figura 17 em que os *PortCosts* são todos iguais. A Figura 17 e a Figura 18 são diagramas equivalentes em termos funcionais. De notar a localização das portas no estado *Blocking*. Numa rede de *switches* sem o MSTP, seria de esperar que uma porta entre os *switches* M e B estivesse bloqueada. Além disso, em vez de se ter a ligação entre os *switches* C e D bloqueada, seria de esperar que o segundo ciclo fosse quebrado numa porta no meio da região MST. No entanto, devido à instância IST, toda a região aparece como um *switch* virtual que corre uma única *Spanning Tree* (CST). Isto faz com que seja possível compreender que o *switch* virtual bloqueia uma *Alternate Port* na porta B. Por

outro lado, o *switch* virtual contem o *switch* C, que faz a ligação com o *switch* D, o que leva a que o *switch* D tenha a sua porta bloqueada.

Instâncias MST

As MSTIs são instâncias RSTP simples que apenas existem dentro de uma região. Estas instâncias correm o RSTP automaticamente por defeito, sem qualquer trabalho extra de configuração. Ao contrário da IST, as MSTIs nunca interagem com o que está fora da sua própria região. É importante referir que o MST usa apenas uma ST fora da região, exceptuando a instância IST, em geral as instâncias dentro de cada região não têm qualquer participação com o exterior. De notar que as MSTIs não enviam BPDUs para fora da sua região, apenas as IST o fazem.

As MSTIs não enviam BPDUs individualmente. Dentro de uma região MST, os *switches* trocam MST BPDUs que podem se vistos como RSTP BPDUs normais para a IST, mesmo contendo informação adicional para cada MSTI. A Figura 19 ilustra a troca de BPDUs entre os *switches* A e B dentro de uma região MST. Cada *switch* apenas envia um BDU, mas cada um deles inclui um *MRecord* (Informação do protocolo para o MST que está presente na porta) por cada MSTI presente nas suas portas.

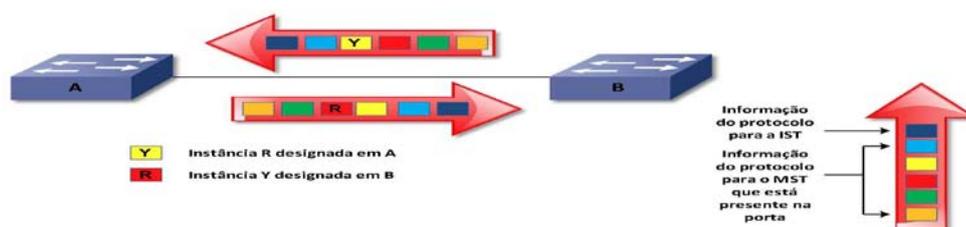


Figura 19 – Troca de BPDUs entre o *switch* A e o *switch* B dentro de uma região MST [6]

De notar que o primeiro campo de informação a ser enviado pelo MST BDU contem informação acerca da IST. Isto implica que a IST (instância 0) está sempre presente em todo o lado dentro de uma região MST (note-se, no entanto, que o administrador de rede não tem que atribuir VLANs à instância 0).

Ao contrário do que acontece normalmente numa topologia ST, ambas as extremidades de uma ligação podem enviar e receber BPDUs simultaneamente. Isto deve-se ao facto de, como ilustra a Figura 19, cada *switch* pode ser a bridge designada para uma ou mais instâncias e precisa transmitir BPDUs. Assim que alguma instância MST for designada numa porta, um BDU que contenha informação para todas as instâncias (IST e MSTIs) deve ser enviado.

2.6.3 Operações dentro de uma região

A IST interliga os *switches* que são internos a uma dada região. Quando existe a convergência da IST, a raiz da IST torna-se a raiz regional que é o *switch* que possui o menor custo para a raiz da CST e menor

Bridge ID em caso de empate (a bridge raiz da CST é o *switch* que apresenta o menor Bridge ID no conjunto de todos os *switches*). Note-se que, no caso de a rede estar configurada como uma única região, a raiz da CST é a raiz regional (da única região existente). Note-se também que os custos das ligações dentro de uma região na IST são vistos como nulos pelos *switches* fora dessa região, ou seja, todas os *switches* de uma região são vistos como uma única bridge virtual.

2.6.4 Operações entre regiões MST

Quando existem várias regiões, o protocolo assume que entre regiões corre o RSTP. O MSTP estabelece e mantém a CST, que inclui todas as regiões MST e todos os *switches* com o RSTP activo. A IST interliga todos os *switches* configurados com o MSTP dentro da região e representa uma sub-árvore na CST, tendo a sub-árvore uma raiz regional, como foi descrito anteriormente. A região MST aparece como uma bridge virtual quer para as outras regiões MST quer para as *switches* fora das regiões. Apenas a CST envia e recebe BPDUs, e as instâncias MST adicionam a informação das suas STs a estes BPDUs para poderem interagir com os seus *switches* vizinhos e construir a topologia final da ST. Assim, os parâmetros ST relativos à transmissão de BPDUs, tais como o *Hello Time*, *Forward Delay*, *Max-Age*, etc., são configurados apenas na instância CST mas afectam todas as instâncias. Parâmetros relativos à topologia ST (como por exemplo, prioridade da bridge, custo de cada porta, etc.) podem ser configurados tanto na CST como em cada instância MST. De notar que o uso de regiões não é visível às restantes regiões vizinhas, ou seja, no caso de falha de uma ligação interna numa região, as restantes árvores presentes nas regiões vizinhas não são afectadas, não dando mesmo por esta alteração na topologia.

2.6.5 Mecanismo de reconfiguração

As instâncias IST e MSTIs não usam a informação *Message-Age* nem a *Maximum-Age* no BPDUs de configuração para determinar a topologia alcançada. Estas instâncias usam o custo de percurso para a raiz e um mecanismo de contagem designado por *Hop-Count*. O resultado alcançado desta forma é o mesmo, ou seja, permite determinar quando é necessário proceder a uma reconfiguração. A bridge raiz da instância que corresponde envia sempre um BPDUs com um custo de 0 e o valor do *Hop-Count* com o valor máximo. Quando um *switch* recebe este BPDUs, decrementa uma unidade no valor do *Hop-Count* e propaga este valor como o valor restante do *Hop-Count* nos BPDUs que gera. Quando este valor chega a 0, o *switch* descarta o BPDUs e a informação nele presente. No entanto, a informação do *Message-Age* e *Maximum-Age* da porção RSTP do BPDUs continua a ser a mesma por toda a região, e os mesmos valores são propagados pelas portas designadas da região na fronteira.

CAPÍTULO 3 – Utilização do MSTP em diferentes contextos

O uso do MSTP como um meio de melhorar as capacidades da engenharia de tráfego em proveito dos serviços Ethernet tem sido recentemente abordado por diferentes autores [7][8][9][10]. Em todos estes trabalhos são abordados cenários dinâmicos onde as VLANs são solicitadas dinamicamente e as múltiplas *Spanning Trees* devem ser dinamicamente activadas e desactivadas. Por outro lado, os artigos [11][12][13] focam-se na melhoria do balanceamento de carga e minimização do impacto de falhas, num cenário estático. No artigo [14], os autores abordam o problema da divisão da rede em regiões. Outros trabalhos propõem usar o MSTP para melhorar aspectos da rede de forma a suportar qualidade de serviço [15][16][17][18] e mobilidade[19].

Neste capítulo é apresentado um estado da arte alargado, onde se detalha a utilização do MSTP enquanto ferramenta de Engenharia de Tráfego. O objectivo é aprender em que situações foi utilizado o MSTP, que alterações lhe foram introduzidas, e qual a finalidade das soluções propostas. Desta forma, pretende-se adquirir uma visão abrangente das inúmeras possibilidades e potencialidades deste protocolo e das suas aplicações práticas.

O primeiro artigo a ser analisado [7] foca vários aspectos inerentes ao MSTP e à sua utilização, tais como: esquema de agrupamento, balanceamento de carga, *Multiple Spanning Trees*, interacção entre o agrupamento e a largura de banda fornecida e finalmente discute-se a sobrevivência das redes Ethernet de nova geração.

O contínuo domínio da Ethernet nas áreas locais, em campus e em redes empresariais, tem conduzido num passado recente a um esforço no sentido de estender esta tecnologia para as redes metropolitanas do domínio dos operadores públicos e existem já exemplos de operadores de rede a implementar serviços metro com baixo custo e escalabilidade.

A tecnologia Ethernet, que foi inicialmente desenvolvida para a aplicação em LANs, actualmente suporta diferentes tipos de tráfego, tais como dados, vídeo e voz, em redes empresariais. O maior desafio da implementação desta tecnologia em redes de acesso e redes metro é a garantia de qualidade de serviço (QoS) usando esquemas de gestão de engenharia de tráfego inteligentes. Outros desafios incluem a escalabilidade, fiabilidade e operação, administração e manutenção (OAM).

3.1 Engenharia de Tráfego em Redes Metro Ethernet

Os clientes exigem a existência de serviços Ethernet extremo-a-extremo, nas quais, LANs que estejam geograficamente distantes sejam interligadas como se fossem uma só LAN. Esta LAN virtual do cliente (C-VLAN) oferece ligação entre LANs e permite que uma estação numa LAN comunique em *unicast*, *multicast* e *broadcast* com qualquer estação que pertença à C-VLAN. Este tipo de comunicação é referido como multiponto. Os clientes estão ligados à rede metro Ethernet através dos pontos de acesso (APs). Em

[7], os APs podem ser referidos como *Provider Edge* (PE) routers ou bridges. O operador metro implementa, normalmente, a ligação dos seus APs usando uma interface do tipo ST. O interface lógico do cliente para a metro Ethernet é denominado *User-to-Network Interface* (UNI), enquanto a ligação lógica entre UNIs, que constituem todos os sites de um dado cliente, é denominada por *Ethernet Virtual Connection* (EVC). No fundo, este trabalho reflecte sobre problemas básicos de Engenharia de Tráfego (TE) em redes metro Ethernet. O artigo analisa os cenários de rede seguintes.

Cenários de Rede

Existem várias arquitecturas disponíveis para transportar tramas Ethernet ao longo da rede metro. No entanto, há duas abordagens que têm captado maior atenção da indústria:

- Usar o *Multiprotocol Label Switching* (MPLS) como tecnologia de transporte na rede metro;
- Estender um protocolo nativo da Ethernet.

MPLS em redes metro

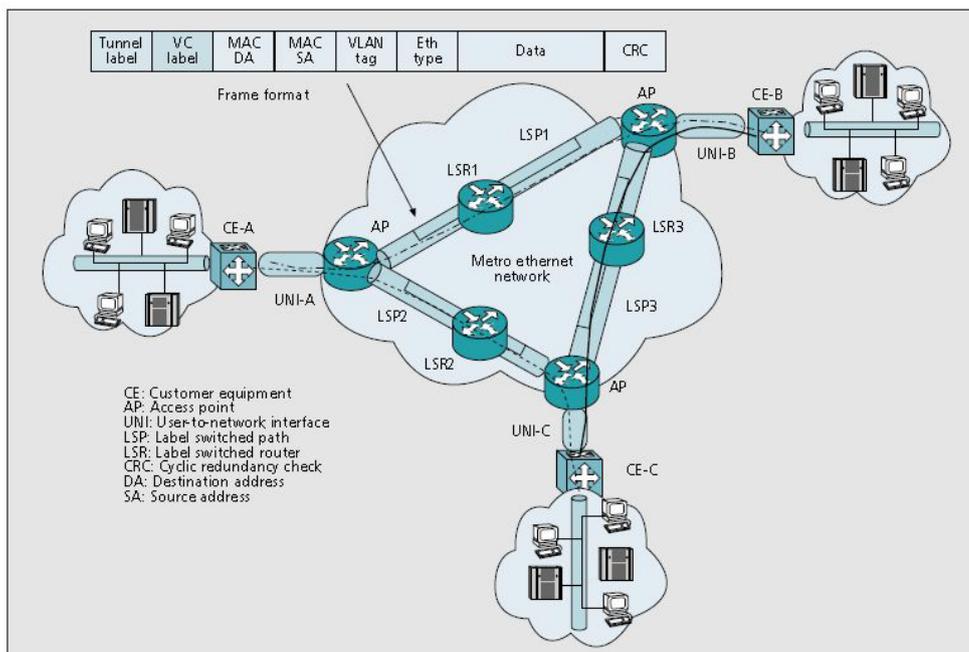


Figura 20 - Cenário de rede 1: uma rede metro MPLS [7]

Como está ilustrado na Figura 20, a rede metro compreende os APs dados em forma de PE routers, *Label Switch Routers* (LSRs) e *Label Switched Paths* (LSPs) entre os dois PE routers. Os PEs são ligados aos equipamentos do cliente através da UNI. Neste cenário, o encapsulamento da camada 2 MPLS facilita o transporte das tramas através de um domínio MPLS do operador de rede. São inseridas duas etiquetas MPLS nas tramas Ethernet do cliente, baseadas na informação do endereço/porta MAC de destino, nos nós de entrada. A primeira etiqueta, no topo da *stack*, é a etiqueta do túnel que é usado para transportar as

tramas. Os LSRs do núcleo olham apenas para a etiqueta do topo da *stack* para transportar essa trama ao longo do domínio MPLS. Esta etiqueta, que fica no topo da *stack*, é normalmente removida pelo penúltimo salto do túnel, ou seja, o salto anterior ao *Label Edge Router* (LER) de saída. Na segunda posição da *stack* está a etiqueta denominada *Virtual Circuit* (VC), que é usada pelo LER de saída para determinar como proceder com a trama e onde a entregar na rede de destino. O LER de saída percebe, a partir da etiqueta VC, como processar a trama e depois envia-a pela porta adequada. Esta etiqueta não é visível até à altura em que a trama chega ao LER de saída por causa da hierarquia do túnel MPLS. Posto isto, para o encapsulamento MPLS, são necessárias duas etiquetas (túnel/VC).

Provider Bridged Networks

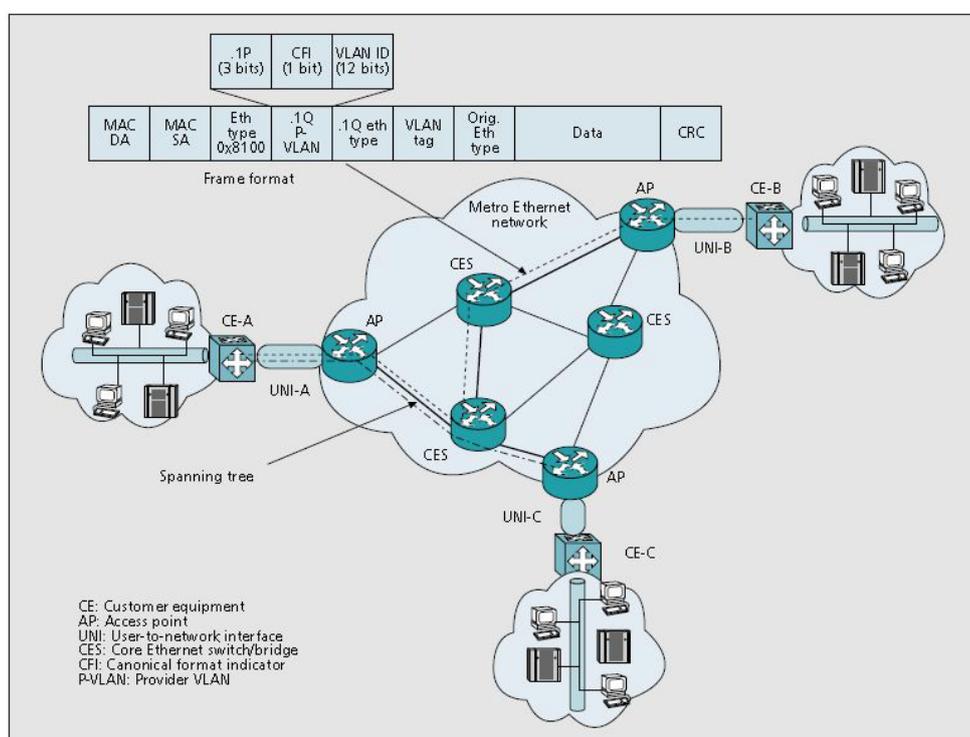


Figura 21 - Cenário 2: *provider bridged network* [7]

Este cenário é uma extensão do protocolo nativo Ethernet nas redes metro. Como se pode verificar na Figura 21, a rede metro abrange *switches/bridges* Ethernet. Um protocolo ST é usado para estabelecer uma ou mais árvores de suporte que abrangem todos os APs. Cada árvore fornece um percurso entre todos os sites dos clientes na VLAN desse mesmo cliente.

São principalmente usados dois esquemas de encapsulamento para fazer frente aos problemas de escalabilidade da implementação Ethernet num domínio metro. O primeiro problema de escalabilidade é devido ao número limitado de VLANs que podem ser suportadas. Como o VLAN *tag* (também referido como *Q-tag*) é limitado a 12 bits, uma *Q-tag* pode ser adicionada, pelo operador de rede, às tramas Ethernet dos clientes no *switch* de entrada do domínio metro. Usando este esquema, o operador de rede pode

transportar mais VLANs dos clientes do que as que seriam possíveis. O esquema em questão é denominado *Q-in-Q* ou *VLAN stacking*.

O outro problema de escalabilidade é a sobrecarga da tabela de endereços MAC. Esta questão surgiu por causa da necessidade por parte dos *switches* do núcleo aprenderem todos os endereços MAC dos clientes. Para minimizar este problema, um esquema de encapsulamento do tipo *Mac-in-Mac* (MiM) é usado para transportar tráfego do cliente através do domínio metro. Em MiM, cada nó de entrada insere dois campos adicionais com endereços MAC que correspondem aos respectivos APs e têm importância local dentro do domínio metro. Desta forma, os endereços MAC dos utilizadores terminais são associados com o endereço MAC do nó de saída (AP) correspondente. Cada um dos *switches* do núcleo tem apenas que aprender os endereços MAC dos *switches* terminais. Como resultado, muitas entradas de endereços MAC são evitadas nos *switches* do núcleo.

Serviços metro Ethernet

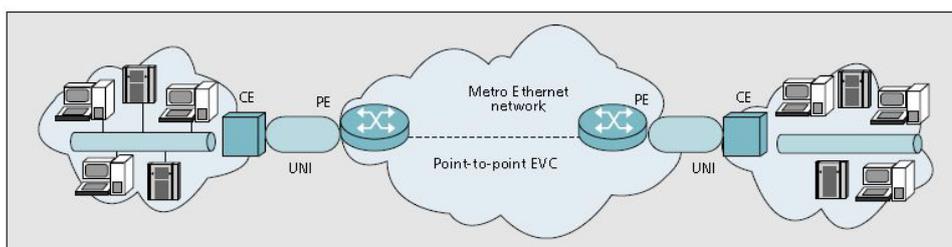


Figura 22 - Serviço do tipo E-Line usando EVC ponto-a-ponto [7]

Com base na arquitectura da rede da Figura 22 foram definidas um conjunto de novos serviços fornecidos pela metro Ethernet. O equipamento do cliente é ligado com a rede metro Ethernet através das portas Ethernet. Esta interface com a rede (UNI) actua como um ponto fronteira físico entre a responsabilidade do fornecedor de serviços e a responsabilidade do assinante do serviço. Dentro das redes metro, a ligação entre UNIs é feita por uma EVC (*Ethernet Virtual Circuit*) (uma camada de transporte, tal como o *Synchronous Digital Hierarchy* (SDH), o Ethernet MPLS ou o *Synchronous Transfer Mode* (ATM), proporciona uma ligação virtual). Do ponto de vista do utilizador, a rede metro assemelha-se a um cabo Ethernet.

Dois tipos de serviços são definidos com base na conectividade de rede: Ethernet *E-Line* (*E-Line*) e *E-LAN*.

O serviço E-Line, que fornece uma EVC de ponto-a-ponto entre duas UNIs, pode ser usado para se obter um grande alcance dos serviços ponto-a-ponto, tais como o serviço *Ethernet Private Line* (EPL) e o serviço *Ethernet Virtual Private Line* (EVPL). Serviços multiplexados de mais do que um EVC podem ocorrer numa, em ambas, ou nenhuma das UNIs.

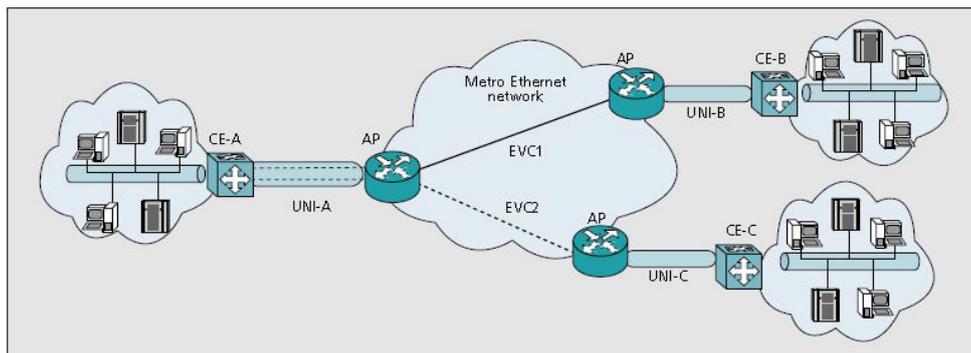


Figura 23 - Serviço EVPL usando um serviço do tipo E-Line [7]

A Figura 23 apresenta um exemplo de um serviço EVPL em que uma porta física do cliente A (UNI A) suporta a ligação de dois serviços Ethernet. Algumas tramas de serviço EVP podem ser enviados para o EVC1 (cliente B), enquanto outras tramas de serviços podem ser enviadas para o EVC2 (cliente C).

Para ligar três ou mais sites, o assinante pode usar o serviço E-LAN para o qual a rede de transporte executa funções de *switching/bridging*.

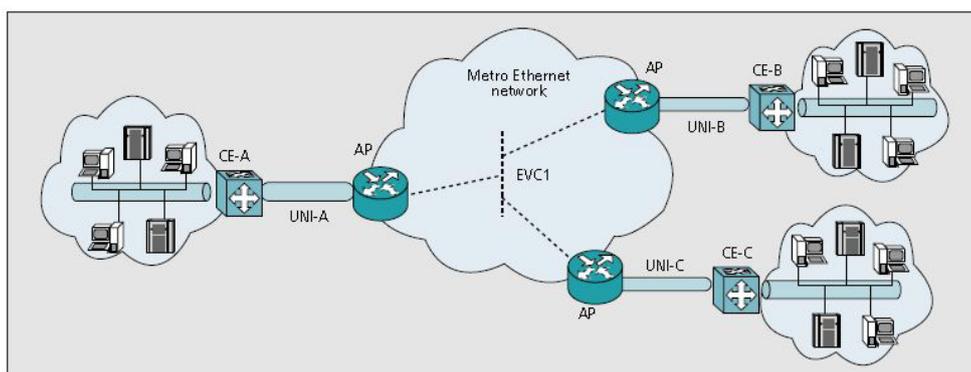


Figura 24 - Extensão LAN usando um serviço E-LAN [7]

A Figura 24 apresenta uma extensão da LAN usando um serviço E-LAN. Os nós terminais actuam com funções de *switching/bridging*, enquanto o STP é usado para prevenir ciclos da rede do operador.

De seguida os autores expõem um conjunto de questões e, após uma reflexão, propõem uma solução.

Questão 1 – Escalabilidade do número de VLANs activas

Como referido anteriormente, o VLAN ID (VID) tem apenas 12 bits disponíveis, o que permite que, no máximo, estejam 4096 VLANs activas ao mesmo tempo, no mesmo domínio. Para ultrapassar esta limitação, é benéfico agrupar várias C-VLANs, atribuir uma VID para esse grupo e associar esse grupo a uma ST. Note-se que a VID atribuída tem significado local dentro do domínio do operador e o nó de saída pode isolar o tráfego C-VLAN respectivo. Contudo, durante a fase de aprendizagem (dos endereços MAC), o

tráfego *broadcast* de uma C-VLAN pode terminar em alguns APs onde a C-VLAN em causa não se encontra presente. A este fenómeno dá-se o nome de *leakage (fuga)*. Os APs devem descartar estas tramas, prevenindo potenciais problemas a nível de segurança. Apesar de o agrupamento resultar numa melhor utilização do tamanho do VID, este método pode levar a um desperdício de recursos de rede. Além disso, o agrupamento pode resultar em rigorosos pedidos de largura de banda agregada e/ou podem tornar alguns das ST inadequada. Uma dada ST torna-se inadequada para transportar tráfego da VLAN X se a VLAN X está agrupada com a VLAN Y que tem essa ST como inadequada de início. Por este motivo, deve-se prestar a atenção necessária quando se está a desenvolver o algoritmo de agrupamento de associação.

Questão 2 – Escassez de largura de banda

Em várias aplicações é benéfico aplicar rotas alternativas. Estas rotas alternativas são importantes para a protecção, o encaminhamento com QoS e para a engenharia de tráfego. O sistema de gestão de rede é feito com MSTs para serem usadas na atribuição das VLANs dos clientes. Devido aos problemas de escalabilidade, a maior parte dos operadores limitam o número total de STs activas a 256. Na prática, em média são usadas 64 STs. Estas STs podem ser construídas quer com o STP ou ter a distribuição de tráfego das C-VLANs em conta, com o objectivo de construir STs adaptadas ao tráfego do cliente. Como exemplo da segunda abordagem, pode ser construída uma ST para um grupo de C-VLANs, de tal forma que o consumo da largura de banda total dos clientes desse grupo nas ligações físicas da ST seja minimizado. Para a abordagem com recurso ao STP, podem ser construídas STs genéricas, cada uma com a raiz num AP diferente. Em qualquer um dos casos, essas STs oferecem um mecanismo de rotas alternativas que pode proporcionar uma melhor utilização e balanceamento dos recursos da rede. Este artigo aborda apenas a construção das STs usando a abordagem com STs. Esses problemas são mais tarde endereçados.

Questão 3 – Sobrevivência baseada na QoS

A terceira questão colocada é o problema de fornecimento de níveis diferenciados de protecção em ambientes metro Ethernet. O artigo foca o problema da atribuição do tráfego EVC com pedidos de protecção dados em MSTs, utilizando novos esquemas baseados em engenharia de tráfego. Esta questão é endereçada mais à frente.

Agrupamento de C-VLANs

Como referido anteriormente, é benéfico agrupar C-VLANs múltiplas e partilhar o mesmo VID. No entanto, o agrupamento de C-VLANs múltiplas na rede do operador pode introduzir *fugas* nos APs do operador para as mensagens *broadcast*. As fugas afectam, se não todos os APs, uma parte das C-VLANs do grupo, logo, a largura de banda total usada na rede do operador não é óptima. Mais importante, a fugas pode tornar-se um grave problema de segurança e precisa de ser devidamente controlada. Note-se que o tráfego *unicast* não é afectado pelo agrupamento.

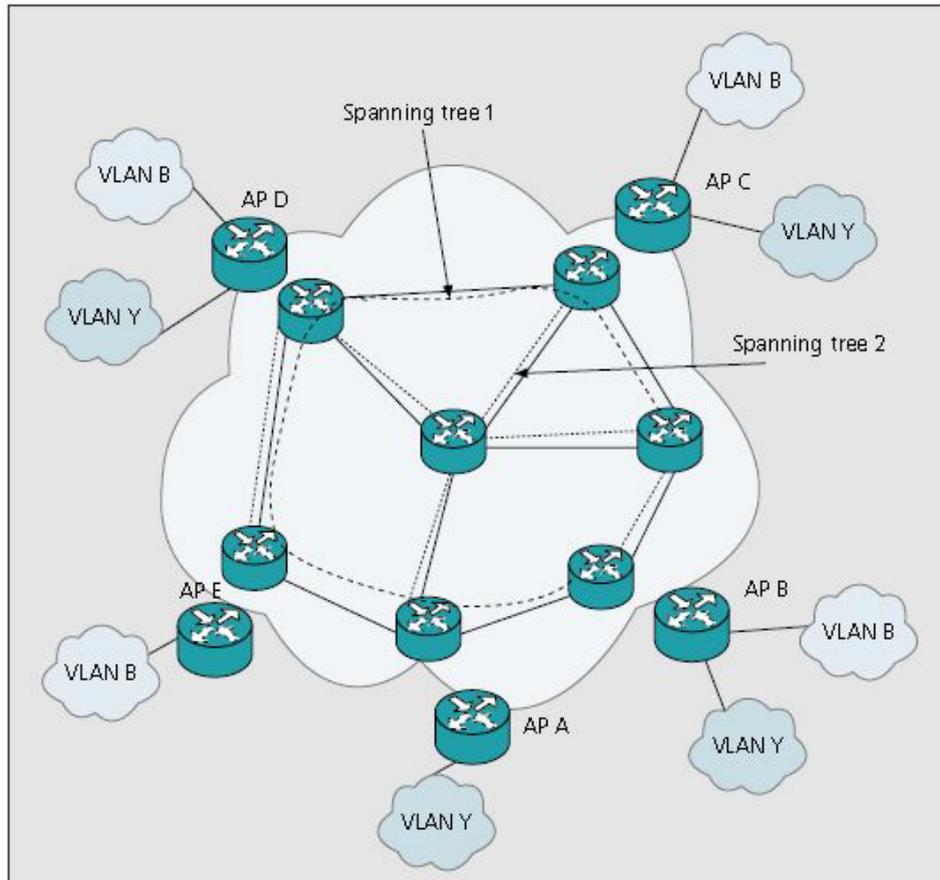


Figura 25 - Ilustração do conceito de *leakage* [7]

A Figura 25 ilustra o potencial problema de *fuga*. Duas C-VLANs, VLAN Y e VLAN B, são agrupadas. A VLAN B é ligada ao domínio metro através dos nós terminais B, C, D e E. A VLAN Y é ligada ao domínio metro através dos nós terminais A, B, C e D. Contudo, as duas C-VLANs partilham os nós terminais B, C e D. Se um dos terminais da VLAN B envia uma mensagem *broadcast*, esta vai chegar a todos os nós terminais (A, B, C, D e E). Entre elas, a VLAN B não implica o nó terminal A. Como resultado, o tráfego *broadcast* levado ao nó terminal A é descartado. O fenómeno de *fuga* baseado em APs dá-se quando a fuga do tráfego é apenas medido nos nós terminais sem ser considerado o seu percurso. Por outro lado, o desperdício de largura de banda nas ligações que não fazem parte do caminho de uma C-VLAN no que respeita a uma ST é referido como fuga de rede.

Para quantificar a fuga de rede, o percurso actual do tráfego, ou uma ST específica, precisa de ser incluído no processamento da fuga. Quando existem MSTs na rede, a fuga de rede varia com a selecção da ST para um dado grupo. Por exemplo, quando o segmento da VLAN Y ligado ao nó terminal A envia uma trama *broadcast*, a mensagem introduz fuga de rede na ST 2 no nó terminal E. Por outro lado, não há fuga de rede na ST 1 uma vez que o nó terminal E é um salto necessário para que a trama chegue aos outros terminais.

Uma vez que a fuga de rede precisa de uma atribuição actual do grupo para a ST, a interacção entre a matriz de tráfego *unicast* e *broadcast* de diferentes VLANs é essencial. Análises computacionais intensivas são exigidas para uma solução óptima na procura da mínima fuga de rede.

Cálculo de fuga

Note-se que uma C-VLAN é ligada ao operador de rede através de múltiplos APs distantes geograficamente. Duas VLANs que tenham um conjunto de APs idêntico, são consideradas um par perfeito para o agrupamento. No entanto, quando um ou mais APs não são usados por todas as VLANs agrupadas, ocorre uma fuga. Um esquema de agrupamento deve ter como objectivo a minimização de fugas através do agrupamento inteligente de VLANs similares. Há diversas formas de quantificar o critério de combinação.

Pedidos de largura de banda, balanceamento de carga e atribuição de STs

Nesta secção são discutidos vários aspectos do balanceamento de carga e a atribuição apropriada de C-VLANs às STs. Dado um conjunto de grupos obtidos a partir de um esquema de agrupamento, o objectivo é a obtenção do melhor encaminhamento em termos de balanceamento de carga desses grupos. Certamente, com o aumento do número de STs, estão disponíveis mais caminhos alternativos, proporcionando maior flexibilidade no balanceamento do tráfego que é possível obter. O balanceamento de carga é aqui abordado segundo três passos. Inicialmente é obtida uma solução que minimiza a largura de banda total utilizada pelos grupos da rede, sendo de seguida encontrada a ligação com a maior quantidade de largura de banda usada (MAX). Seguidamente, a ligação física de maior capacidade deixa de ter o valor infinito e passa a guardar o MAX para todas as ligações. Finalmente, é outra vez obtida uma solução para o encaminhamento dos grupos, mas desta vez tendo em conta o balanceamento do tráfego. Admitindo que U e L correspondem ao máximo e mínimo de largura de banda reservada nas ligações da rede, respectivamente, o balanceamento tem em vista a minimização U-L. A solução assim encontrada é uma solução de encaminhamento onde as ligações físicas são altamente balanceadas.

Agrupamentos e pedidos de largura de banda

A fuga devido ao agrupamento de C-VLANs múltiplas pode afectar a selecção da ST óptima para C-VLANs levando a um aumento de utilização de largura de banda na rede metro Ethernet. No caso de estudo seguinte, 100 VLANs são agrupadas com um método inteligente, tendo um máximo de fugas baseadas em APs, para cada VLAN, de 0,3. É verificado que o menor aumento de largura de banda pode ser atingido usando métodos de optimização mais agressivos, tal como o método inteligente usado. Houve, no entanto, um pequeno aumento de cerca de 2% na largura de banda comparando com o não agrupamento. De qualquer forma, este ligeiro aumento de largura de banda é tolerável se se pensar que são poupados cerca de 34% no espaço de etiquetas.

Como discutido anteriormente, a fuga para um dado grupo depende da ST seleccionada. É relevante o estudo do impacto nos pedidos de largura de banda de abastecimento devido à restrição

imposta pelas fugas para um grupo. Se uma ST tiver duras restrições de fuga para um dado grupo, algumas das STs podem não ser atribuídas a esse grupo. De outra forma, as restrições de fuga são violadas. Como o número de STs disponíveis para encaminhamento vai decrescendo devido a esta restrição, era esperado que a quantidade de largura de banda necessária para fornecimento aumentasse por causa da eliminação de alguns caminhos mais curtos das STs. A diferença de percentagem de *threshold* é definida da seguinte maneira: para cada grupo, o conjunto de 21 STs é avaliada para calcular a fuga da rede (entre zero e um). A diferença de valor para cada grupo é a máxima fuga para uma árvore menos a fuga menor. A diferença de percentagem de *threshold* de X permite que todas as STs que não excedam (X *valor da diferença da fuga mínima) para serem incluídas no encaminhamento daquele grupo. Assim, uma diferença de percentagem de *threshold* de zero permite que apenas STs com a mínima fuga sejam incluídas, enquanto uma percentagem de 100 de *threshold* permite a inclusão de todas as STs. A Figura 26 mostra a largura de banda total como uma função da diferença de percentagem de *threshold* para o agrupamento, resultando em 66 grupos, e usando a topologia da rede Italiana.

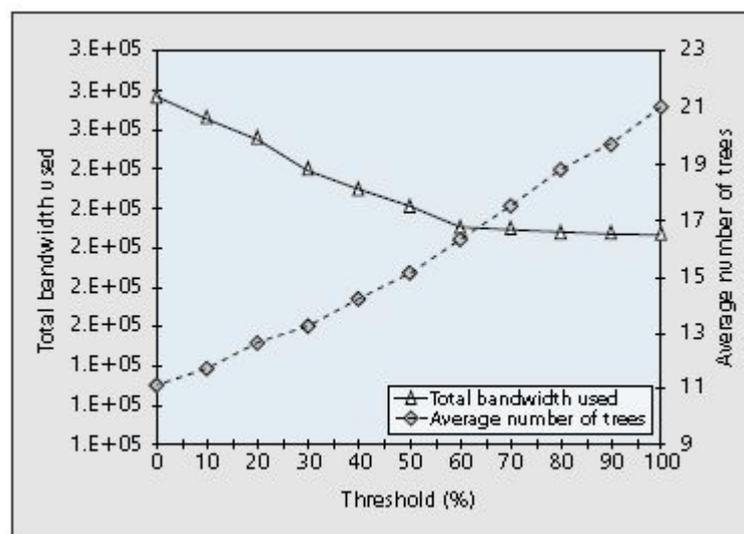


Figura 26 - Largura de banda total e média do número de árvores em função da diferença de percentagem de *threshold* para o agrupamento obtido para a rede Italiana [7]

Como demonstra a Figura 26, requisitos restritos na fuga da rede têm um impacto importante no total da largura de banda necessária. Assim, um grupo que necessite de menos fuga de rede pode ter de pagar cada vez mais pelo uso de largura de banda. A figura mostra que de uma percentagem de 60 para cima, o total de largura de banda necessária não é afectado. Isto sugere que requisitos moderados para a fuga de rede podem ser acomodados sem grande influência no total de largura de banda usada. A figura mostra também o número médio de árvores por grupo como função da diferença de percentagem de *threshold*.

Um encaminhamento bem sucedido de C-VLANs pode ser medido pela distribuição de tráfego na rede. Um esquema de encaminhamento que equilibra o tráfego das C-VLANs através da rede é melhor que

outro esquema qualquer de encaminhamento que tenda a usar as ligações físicas de um modo desequilibrado. Claramente, conforme se tem mais opções de encaminhamento (isto é, mais STs) para as C-VLANs, mais oportunidades existem para o equilíbrio do tráfego.

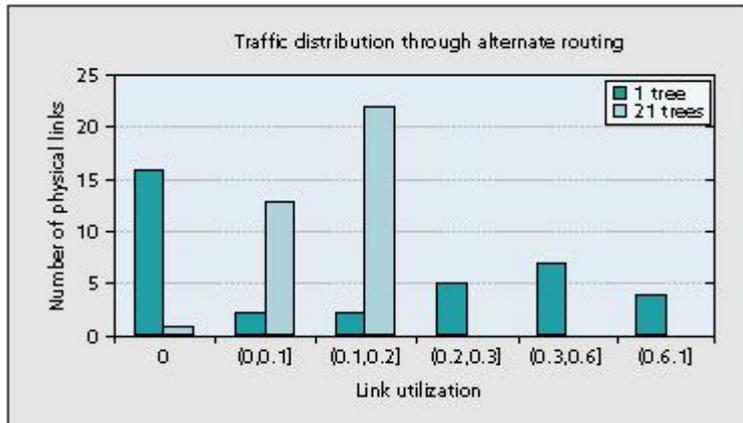


Figura 27 - Impacto de encaminhamento alternativo no balanceamento de carga para a rede Italiana [7]

A Figura 27 mostra que a utilização das ligações da rede Italiana como uma função do número de caminhos alternativos (isto é, árvores) disponíveis. O eixo horizontal define os limites do valor da utilização da ligação, definida como a quantidade de tráfego encaminhado numa ligação dividida pela capacidade da ligação. O eixo vertical mostra o número de ligações físicas cuja utilização está dentro do respectivo intervalo. Note-se como o tráfego fica desequilibrado quando apenas uma ST é usada. Algumas das ligações estão a ser bastante utilizadas enquanto outras não. Quando são usadas 21 STs, o esquema demonstrado é capaz de espalhar o tráfego através da rede de modo a que todas as ligações sejam igualmente utilizadas. Como mostra a figura, cada ligação individual não é utilizada mais de 20%. Este encaminhamento é altamente desejável já que proporciona um balanceamento de carga nos *switches* e equipamentos de transmissão, e é preferível para esquemas de sobrevivência.

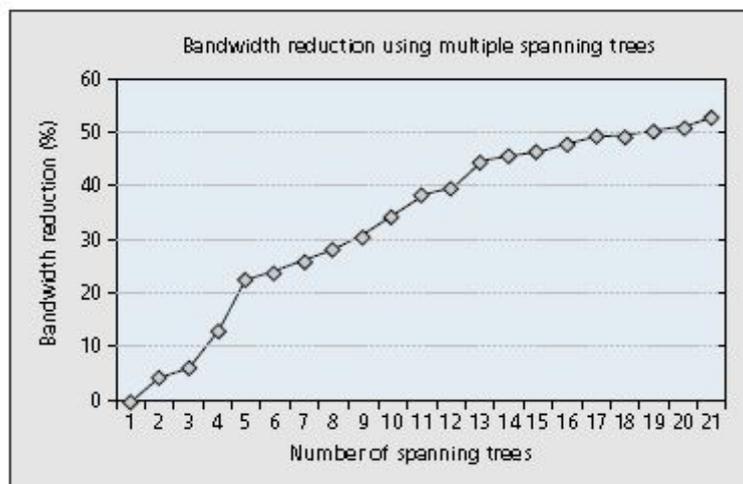


Figura 28 - Redução da largura de banda usada utilizando MSTs para a rede Italiana [7]

A Figura 28 ilustra o benefício de múltiplas STs na redução do uso de largura de banda na rede Italiana. No eixo horizontal varia-se o número de STs atribuídas de 1 a 21. Como foi mencionado antes, estas STs estão calculadas de acordo com o MSTP (IEEE 802.1s) para uma rede com 21 APs. O eixo vertical quantifica a redução da percentagem de largura de banda calculada relativamente ao uso de apenas uma ST. Pela figura pode-se verificar os benefícios do uso de MSTs. Usando todas as 21 STs permitidas o uso da largura de banda é reduzido até 50%. De notar que a taxa de melhoria decresce quando o número de STs aumenta acima de 11. Assim, apenas metade das STs geram 80% da redução da largura de banda.

Questões de sobrevivência no ambiente *Ethernet* têm vindo a tornar-se mais importantes à medida que as ligações de rede carregam grandes quantidades de tráfego. Numa rede *Ethernet*, os *switches* participam no protocolo de distribuição para construírem uma ou mais STs. Uma ST fornece uma ligação sem ciclos entre as fontes de *Ethernet* e define o que é chamada uma *topologia activa*. No caso de uma ligação falhar, o protocolo STP ou a sua versão melhorada, RSTP, impõe que a ligação seja restabelecida na sua totalidade activando alternadamente portas/ligações. No entanto, este processo leva tempo, e melhorias ao protocolo estão a ser investigadas. A norma IEEE 802.3 fornece um mecanismo para juntar um grupo de portas numa entidade lógica. No caso de uma dessas portas falhar, um protocolo com o nome *Link-Aggregate Control Protocol (LACP)* é activado para mudar o tráfego afectado para uma capacidade redundante na ligação lógica conjunta. A reconvergência do protocolo ST não é activada neste caso, e a topologia lógica da topologia activa é mantida.

Cientes EVCs diferentes podem ter diferentes requisitos para a ruptura de tráfego quando ocorrem falhas nas ligações. Um EVC, por exemplo, transportando tráfego multimédia em tempo real vai, provavelmente, ter mais requisitos rigorosos na perda/recuperação de dados que outro EVC que transporta tráfego de baixa prioridade. Para permitir tal diferenciação, são propostas MSTs diferenciadas que variam em robustez quando ocorrem falhas nas ligações. A robustez é definida como uma fracção de ligações de agregados de ligação e o número de ligações que constituem uma ST. Num extremo, uma ST completamente robusta usa $(n-1)$ ligações que são baseadas em agregados de ligações. No caso de uma ligação falhar de um dos componentes de um agregado de ligações lógicas, todo o tráfego dos EVCs no componente que falhou pode ser mudado para os componentes subjacentes do agregado de ligações lógicas. Assim, os EVCs experienciam uma degradação da QoS (em termos de largura de banda) e atraso adicional na activação da ST reconvergente. No outro extremo, uma ST que não seja robusta, não tem nenhuma $(n-1)$ das ligações de agregados de ligação. No caso de uma das ligações falhar, a convergência das ST é activada (por exemplo o protocolo RSTP), e a ligação lógica da *Ethernet* é alterada para EVCs transportadas. Usando a continuidade de STs entre estes dois casos extremos, as STs podem ser construídas de uma maneira em que estas possuam vários graus de robustez. Isto permite que o fornecedor chegue à atribuição diferenciada de EVCs nas STs dependendo do ranking da robustez e das baixas probabilidades de

interrupção de tráfego no conjunto dado de STs. Clientes que exigem menor interrupção de tráfego podem pagar mais para ter os seus EVCs atribuídos nas STs mais robustas.

De seguida, são ilustrados os benefícios da atribuição de VLANs inteligente, para acomodar os requisitos necessários à sobrevivência baseada em QoS de uma VLAN. O problema de gestão implica a atribuição das VLANs às STs de tal forma que o tráfego de VLANs diferenciadas projectado esteja de acordo com o *Service Level Agreements* (SLAs). Estes SLAs são definidos em termos de requisitos de largura de banda e nível de protecção.

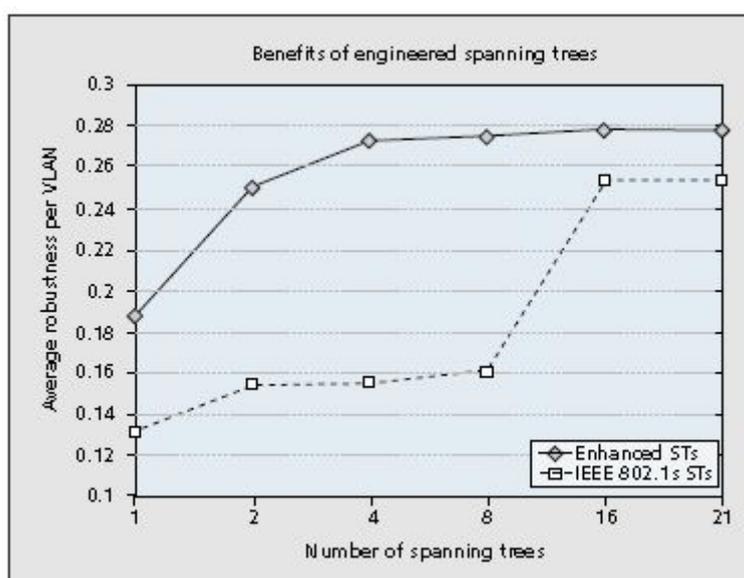


Figura 29 - Melhoria da sobrevivência VLAN para a rede Italiana pela construção de STs e atribuição das VLANs às MSTs [7]

A Figura 29 ilustra esta ideia. Esta mostra o desempenho em termos de robustez média por VLAN, numa simulação. Assume-se que 10% das ligações da rede são agregados de ligações. No eixo horizontal observa-se o número de STs disponíveis para o encaminhamento e no eixo vertical observa-se o valor da robustez por VLAN. A robustez é definida pela percentagem do total da largura de banda da VLAN que se encontra protegida. Os resultados são de dois conjuntos diferentes de STs. O primeiro conjunto (com o nome IEEE 802.1s STs) corresponde a um algoritmo de construção agnóstico LACP de uma ST. Estas árvores são as que se esperavam ao correr o algoritmo *standard* MSTP. O segundo conjunto de árvores (com nome *enhanced* STs) é construído tirando vantagem das ligações LACP discutidas. O algoritmo empregue para a atribuição de VLANs otimiza a robustez como uma função objectiva para cada VLAN. A figura mostra a melhoria na sobrevivência da VLAN pela construção de STs inteligentes e atribuição de VLANs às MSTs de forma esclarecedora.

A capacidade disponível usada na ligação de equipamento agregado pode ser usada para transporte de *bronze-type traffic* (isto é, tráfego que não necessite de mais nada para além do RSTP para atenuar a

falha das ligações). No caso de ocorrer falha, este tráfego pode ser retirado para ser reencaminhado usando RSTP permitindo que o *gold traffic* (isto é, tráfego protegido) seja movido para a capacidade disponível.

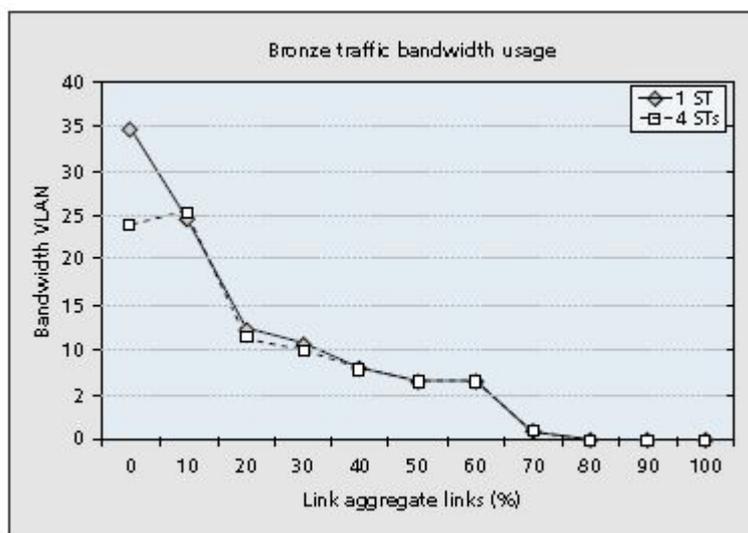


Figura 30 - Requisitos de largura de banda na rede Italiana por VLAN para o *bronze traffic* como função da configuração de rede [7]

Seria oportuno encontrar a poupança em termos médios de utilização de largura de banda para uma *bronze* VLAN em ambientes que usam agregados de ligações. Assim, na Figura 30 observa-se a largura de banda média usada por VLAN do tipo *bronze* através de uma função da percentagem dos agregados de ligações existentes na rede. A figura mostra resultados para dois casos, usando uma ST e quatro STs, para encaminhamento nas VLANs. Como se esperava, a capacidade disponível do agregado de ligações reduz drasticamente a largura de banda usada pelo *bronze-traffic*. Este atenuar pode ser alcançado por cerca de 40% dos agregados de ligações distribuídos na rede. Note-se que as MSTs não ajudam muito na redução de *bronze-traffic* em redes com agregados de ligações já que a atenuação é, na sua maioria, causada pela utilização da capacidade disponível dos agregados de ligações. Além disso, o investimento inicial para o *upgrade* do agregado de ligações pode ser recuperado fornecendo serviços de sobrevivência aos clientes.

3.2 Cenários Dinâmicos

O artigo [8] tem como base uma arquitectura *Global Open Ethernet* (GOE), sendo esta uma solução escalável e rentável de próxima geração de *Virtual Private Networks* (VPNs) assente em infraestruturas de redes ópticas. Esta arquitectura satisfaz vários requisitos ao mesmo tempo: flexibilidade na topologia de rede, funcionalidades de rede acessíveis, baixo custo de equipamento e baixo custo operacional e é baseada nas extensões do STP, tais como o RSTP e o MSTP. Em geral, estas extensões usam uma métrica de custo de ligação que é inversamente proporcional à capacidade da ligação. Com o objectivo de obter um melhor balanceamento de carga na rede, os autores apresentam uma métrica de custo de ligação que é

uma função da carga de ligação corrente. São então propostos dois algoritmos de engenharia de tráfego diferentes e comentados acerca dos seus desempenhos para várias topologias de rede e para diferentes cenários de tráfego.

Arquitectura *Global Open Ethernet* (GOE)

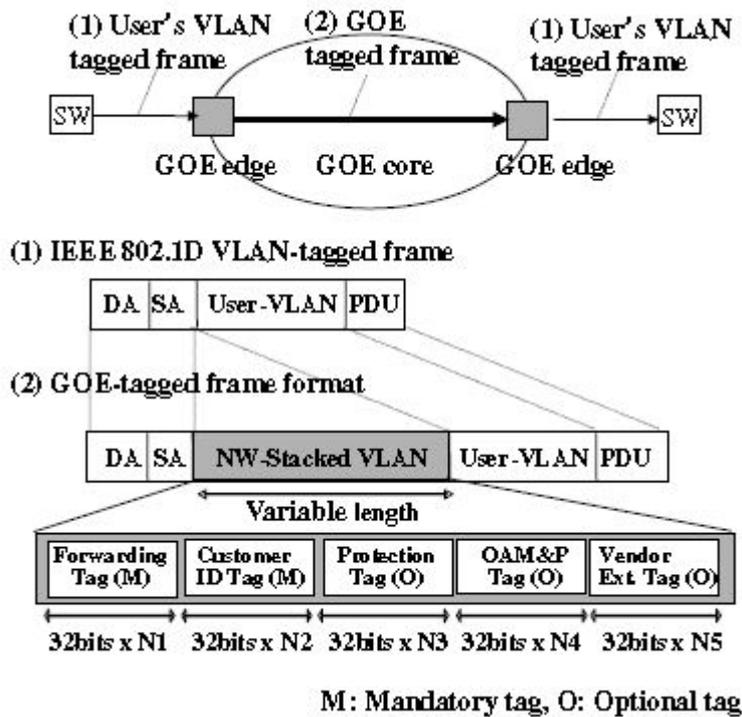


Figura 31 - Trama GOE *tagged* e trama VLAN *tagged* [8]

A Figura 31 representa uma rede de transporte baseada na arquitectura GOE, que consiste no núcleo GOE (*GOE core*) e nos nós das extremidades. A Figura 31 mostra também o formato dos pacotes de entrada dos utilizadores e dos pacotes de encapsulamento GOE transportados para dentro da rede de transporte. O nó GOE da extremidade, ao receber tramas Ethernet do utilizador, com ou sem IEEE 802.1d *VLAN-tag*, adiciona um cabeçalho GOE imediatamente a seguir ao campo do endereço MAC de origem. Este cabeçalho é flexível e extensível com uma estrutura de tamanho variável, que consiste em dois *tags* obrigatórios (*Forwarding tag* e *Customer ID tag*) e outros *tags* opcionais (*protection tag*, *OAM&P tag*, *security tag*, *QoS tag*, e *vendor extension tag*).

O *Forwarding tag* é um elemento chave do GOE para o encaminhamento de pacotes na rede de transporte, ao passo que o *Customer ID tag* é usado apenas para identificar os utilizadores e para solicitar o processamento e tráfego específico do utilizador. Desacoplar o *Forwarding* e o *Customer information* em campos *tag* diferentes pode separar a gestão da rede e a gestão do utilizador, que leva a um funcionamento de rede simples e escalável. Em particular, para uma maior escalabilidade nas redes VPN,

Hierarchical Multiple Rapid Spanning Trees são efectuadas pela combinação do IEEE 802.1s MSTP e o 802.1w RSTP em cada nível.

O Forwarding GOE é baseado no endereço do nó de destino com o propósito de ser usado de forma semelhante ao que acontece nas redes IP. Basicamente uma *VLAN-ID* é atribuída a um *switch* Ethernet em vez de uma porta Ethernet, de modo que um *VLAN-tag* é usado como endereço de encaminhamento. Por outras palavras, o *VLAN-ID* torna-se um tipo de endereço que pode ser processado por *switches* Ethernet da mesma forma que os routers IP fazem com os endereços IP.

Problema de Engenharia de Tráfego

Assume-se que há N nós GOE num domínio plano. É atribuído um *tag VLAN-ID* único a cada nó na rede. Em cada um deles é criada uma tabela de encaminhamento de pacotes apenas para o topo da *stack* de Forwarding *tags*. Por outras palavras, cada nó do núcleo GOE verifica o topo da *stack* dos Forwarding *tags* apenas para determinar o percurso de encaminhamento. Note-se que nos *switches* Ethernet usuais verificam o VLAN tag e o endereço MAC de destino para o mesmo efeito. Os pacotes de informação etiquetados com o formato GOE têm que ser transportados pelas STs geradas para cada ID de nó (VLAN ID). O GOE utiliza o MSTP mas de uma forma diferente da habitual. É criada uma ST para cada nó de destino com uma VLAN-ID única. A árvore parece-se com uma ST ao contrário para cada destino, onde o nó raiz desta árvore é sempre o nó de destino.

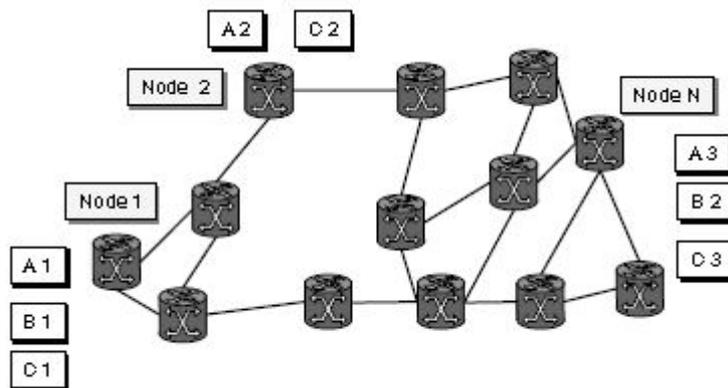


Figura 32 - Pedido de serviços de tráfego na arquitectura GOE [8]

É assumido pelos autores que a matriz de requisitos de tráfego é conhecida, e pode ser baseada num conjunto de subscrições de utilizadores para um serviço de linhas alugadas virtuais. Considere-se o exemplo da Figura 32. Há N nós e cada um dos utilizadores A, B e C pretendem estabelecer um serviço VLAN, do tipo multiponto-multiponto, entre as cidades seleccionadas. A localização do utilizador A e C está nos nós 1, 2 e N , e a localização do utilizador B está nos nós 1 e N . Neste tipo de serviços, cada pacote enviado pelo utilizador A localizado no nó 1, deve ser enviado para o nó 2 e N . Na mesma linha de raciocínio, cada pacote enviado pelo utilizador A localizado no nó N deve ser enviado para o nó 1 e 2, e

assim sucessivamente. Note-se que um pedido de serviço, que é feito em termos de largura de banda, entre qualquer par de localizações do utilizador A é o mesmo e igual a uma constante, i.e., assume-se que os requisitos de tráfego são sempre simétricos. O mesmo pressuposto é assumido aos requisitos de tráfego entre as localizações dos utilizadores B e C.

Posto isto, é formulado o problema: dados pedidos de tráfego entre pares de nós e capacidade disponível em todas as ligações de rede, como criar uma ST ao contrário para cada nó de destino activo de forma a minimizar o número de pedidos de tráfego rejeitados. Esta árvore especial, chamada de *per-destination MRSTP* (PD-MRSTP), encontra o caminho mais curto de cada nó para um nó raiz (o nó de destino) usando o inverso da largura de banda das ligações físicas com sendo o peso da ligação.

Para resolver o problema apresentado é usada uma abordagem popular, uma heurística baseada no algoritmo de *Dijkstra*, devido à sua baixa complexidade. Contudo, estas heurísticas podem variar, dependendo da definição da função de custo de ligação e do método utilizado para escolher o pedido de tráfego. São então apresentados dois algoritmos, o primeiro é chamado o algoritmo *Constant Residual Bandwidth* (CRB) e o segundo é o algoritmo *Reciprocal Residual Bandwidth* (RRB).

Estes dois algoritmos são comparados com o algoritmo standard do STP que usa uma métrica fixa de custo de ligação.

- CRB – usa uma métrica de custo de ligação dinâmica que é uma função constante da largura de banda residual de ligação;
- RRB – usa uma métrica de custo de ligação dinâmica que é o recíproco da largura de banda da ligação residual;
- RLC – usa uma métrica de custo de ligação estática que é o recíproco da capacidade de ligação.

São usadas diversas métricas de desempenho para comparar estes três algoritmos. A primeira é a probabilidade de bloqueio da largura de banda, dado por B_{bw} , que expressa a razão entre a quantidade total de largura de banda rejeitada e a quantidade total de largura de banda requisitada. Na segunda métrica de desempenho é calculada a probabilidade de bloqueio de serviço, dada por B_{sr} , que expressa a razão entre o número total dos pedidos de serviços de tráfego rejeitados e o número total de pedidos de serviços de tráfego. Note-se que quando um pedido de serviço para $r_{i,j}$ unidades de largura de banda entre a origem i e o destino j é rejeitado, o número de pedidos rejeitados aumenta em uma unidade, e o total de largura de banda rejeitada aumenta em $r_{i,j}$ unidades de largura de banda. O objectivo é manter o B_{bw} e o B_{sr} o mais baixo possível. A terceira métrica de desempenho, dada por η , é a utilização média das ligações, ou seja, a utilização da rede. Entre dois algoritmos com probabilidades de bloqueio semelhantes, o que tiver menor média de ocupação de ligação é preferível uma vez que há menos recursos de rede ocupados.

É então usado o método de *WaxMan* para gerar topologias *mesh* aleatórias. Nas simulações, e para cada topologia de rede, são usados dois tipos de tráfego: tráfego baseado em distribuições uniformes

e e tráfego baseado em distribuições de *Zipf*. São feitas 5 simulações em cada caso para, a partir dos resultados médios, serem tiradas conclusões. Os resultados são sumariados na Figura 33 e Figura 34.

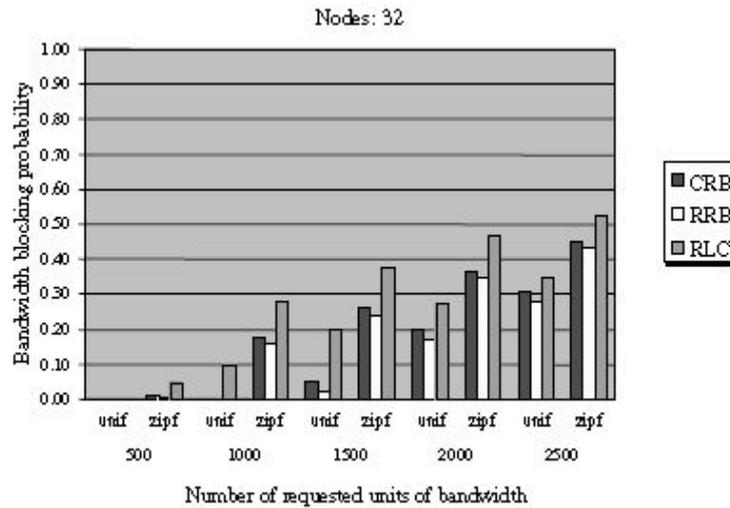


Figura 33 - Probabilidade de bloqueio de largura de banda (B_{bw}) [8]

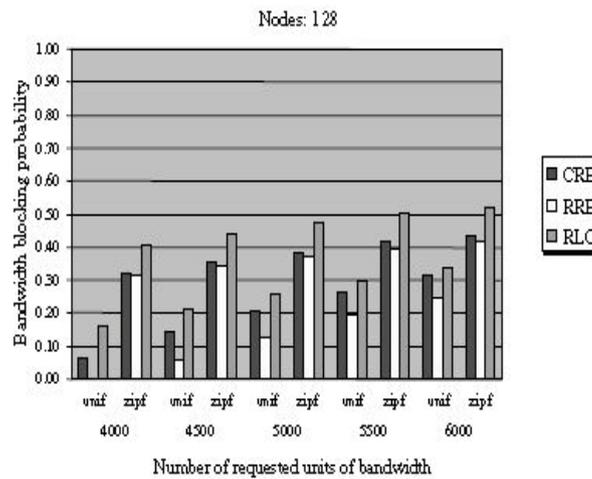


Figura 34 - Probabilidade de bloqueio de largura de banda (B_{bw}) [8]

A partir da observação destas figuras, pode-se tirar as seguintes conclusões:

- Para todas as topologias de rede e de tráfego, e em todos os três algoritmos, a probabilidade de bloqueio aumenta quando aumenta a carga de tráfego;
- Para todas as topologias de rede e todos os algoritmos, a probabilidade de bloqueio para a distribuição *Zipf* é sempre maior do que a distribuição uniforme. Esta constatação é coerente com o esperado, uma

vez que no caso da distribuição *Zipf* há menos nós destino activos, mas há bastante mais pedidos de tráfego do que no caso de distribuição uniforme.

- Em termos de probabilidade de bloqueio, o algoritmo RRB tem melhor desempenho. O algoritmo CRB tem um desempenho ligeiramente pior do que o RRB considerando que o algoritmo RLC é significativamente pior em relação aos dois anteriores. Isto aplica-se a todas as topologias de rede e de tráfego. Em alguns casos, a probabilidade de bloqueio segundo o algoritmo RLC é 0.15 superior do que segundo o algoritmo RRB. A razão para tal acontecer é o facto de o algoritmo RRB faz um melhor balanceamento de carga do tráfego de tal forma que o número de ligações sobrecarregadas e subcarregadas é reduzido. O tráfego, no caso do algoritmo RLC, é transportado através de um menor número de ligações. Neste algoritmo, o custo das ligações é fixo e não corresponde ao estado actual da rede de tal forma que o algoritmo de *Dijkstra* não distribui a carga por todas as ligações da rede.

Os resultados para B_{sr} no caso da topologia de rede ter 32 e 128 nós vêm confirmar todas as conclusões retiradas da análise dos resultados para o B_{bw} . O resultado para o η no caso da topologia de rede ter 32 e 128 nós é apresentada na Figura 35 e na Figura 36, respectivamente.

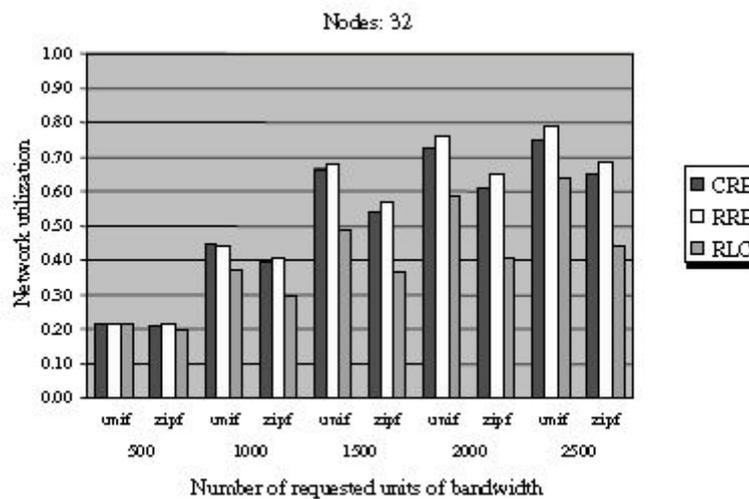


Figura 35 - Utilização da rede (η) [8]

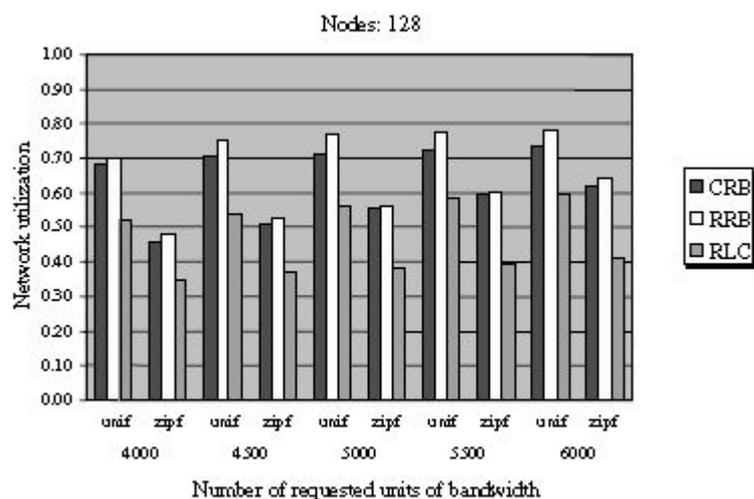


Figura 36 - Utilização da rede (η) [8]

As duas figuras mostram que a utilização das ligações da rede é maior no caso do RRB, ligeiramente menor no caso do CRB e a mais baixa é atingida quando se usa o RLC. No caso da distribuição *Zipf*, a utilização da rede é sempre menor do que no caso da distribuição uniforme. Em alguns casos, a utilização de rede usando o RLC é menor em 0.2 do que se for usado o RRB. A fraca utilização de rede e a alta probabilidade de bloqueio segundo o algoritmo RLC demonstra que o método de optimização é bastante pobre com a métrica de ligação estática em termos de uso dos recursos da rede.

Como conclusão, é evidente pelos resultados apresentados que o algoritmo RLC (usado no STP) é bastante pior do que o RRB e o CRB, em termos de probabilidade de bloqueio e de eficiência no uso dos recursos da rede. Dentro dos dois algoritmos que usam uma métrica de custo de ligação dinâmica, o algoritmo RRB faz um melhor balanceamento de carga do que o algoritmo CRB, logo atinge um melhor desempenho.

O artigo [9] propõe uma arquitectura MST Ethernet, denominada por Viking, que melhora o *throughput* agregado e a tolerância a falhas abordando a tecnologia VLAN de uma forma diferente.

A ideia nuclear de um sistema *Viking* é a de usar MSTs em conjunto com a tecnologia VLAN com o objectivo de melhorar o desempenho do *throughput* em toda a rede usando ligações múltiplas redundantes. Além disso, o *Viking* proporciona características de tolerância de falha proporcionando um mecanismo para desviar as comunicações afectadas para percursos alternativos após a detecção de falha. De facto, o Viking esforça-se por proporcionar uma solução de engenharia de tráfego tolerante a falhas para redes metro Ethernet e redes cluster.

O sistema *Viking* assenta na tecnologia VLAN para a selecção do percurso apropriado. As VLANs são convencionalmente usadas para simplificar a administração de redes, a redução de custos e melhorar a segurança. O sistema Viking afasta-se do paradigma convencional e usa *tags* baseados nas VLANs para

seleccionar o percurso apropriado entre pares de terminais. Todos os caminhos que possivelmente podem ser usados para os percursos são suportados em STs diferentes. Uma vez que cada instância ST corresponde a uma VLAN particular, uma selecção explícita de uma VLAN resulta numa selecção implícita de um percurso associado, com uma ST correspondente. Em caso de falha, os terminais precisam apenas de mudar o VLAN ID nas tramas subsequentes para seleccionar um percurso alternativo.

Com o objectivo de utilizar os recursos de rede disponíveis de forma eficaz, os seguintes factores devem ser tomados em conta na construção das STs.

Para a tolerância a falhas, deve haver pelo menos dois percursos para qualquer par de nós em duas STs diferentes que não partilhem nós nem ligações. Um desses percursos pode ser o percurso primário, e o outro o percurso de *backup*. Para o eficiente balanceamento de carga, a selecção do percurso deve maximizar a utilização das ligações marginalmente carregadas e minimizar o uso de ligações muito carregadas. Para maximizar o número de ligações activas globalmente, as STs devem-se sobrepor minimamente. O número máximo de VLANs, e por isso o número de STs possíveis na rede, é ditado pelo equipamento de rede usado. O tamanho do VLAN ID é limitado a 4096 entradas, mas o número máximo de STs suportadas pelos *switches* é bastante inferior a esse valor. Tipicamente, os *switches* suportam um máximo de 1024 instâncias ST, havendo mesmo *switches* que apenas suportam um máximo de 64 STs.

Assim, torna-se imperativo minimizar o número de STs necessárias enquanto se maximiza o número de ligações activas, de forma balanceada, com espaço para percursos de *backup* em caso de falha. O sistema Viking endereça estas questões seguindo uma abordagem de engenharia de tráfego. Tipicamente, o tráfego que corresponde aos percursos de *backup* para um par de nós pode ter menor prioridade do que o tráfego correspondente aos percursos primários de outros pares de nós que partilhem o mesmo conjunto de ligações. Isto pode ser feito marcando os bits de prioridade nas tramas de transmissão em concordância com as especificações da norma IEEE 802.1p.

A possibilidade de implementar MSTs na camada 2 de rede faz com que seja possível explorar as redundâncias da rede física e aumentar o *throughput* agregado, restabelecer percursos após uma falha e diminuir o tempo de recuperação da mesma. Devido a uma melhoria no desempenho e das características de recuperação de falha, o sistema Viking forma uma construção de um bloco melhor para MANs, para redes de armazenamento e para redes cluster. É desenvolvida uma construção de MSTs centralizada que tem como objectivo o balanceamento de carga de todas as ligações da rede física, assumindo uma topologia de rede estática e um tráfego fixo. Através de um conjunto de simulações exaustivo do algoritmo proposto, é demonstrado que a arquitectura Ethernet com MSTs e a construção de STs associada pode efectivamente aumentar o *throughput* agregado e acelera o processo de recuperação de falhas. Para provar a viabilidade da arquitectura proposta, são também apresentados detalhes de implementação de um protótipo Viking. Medições empíricas neste protótipo estão em linha com os resultados das simulações. Em particular, a tolerância a falhas por parte do sistema Viking proporciona uma recuperação transparente no caso de ocorrer uma falha de ligação, com o período de recuperação reduzido de dezenas de segundos, na arquitectura de uma única ST, para 400 a 600 milissegundos. Apesar de o sistema Viking poder ser

construído em *switches* Ethernet usando SNMP, um design mais estreitamente integrado pode levar a um ainda melhor tempo de recuperação.

O artigo [10] debruça-se sobre os problemas de engenharia de tráfego relacionados com a construção das STs que é influenciada pela métrica de custo de ligação considerada. Para atingir um bom balanceamento de carga e diminuir o atraso médio da rede, os autores propõe o uso de uma métrica com o custo de ligação configurável que depende do atraso e da carga de ligação corrente.

Em vez de ser usada uma abordagem estática no custo da ligação, que é inversamente proporcional à capacidade da ligação, pode ser usada uma atribuição de custos de forma dinâmica em função da carga da ligação para atingir um melhor balanceamento de carga. Os custos das ligações são atribuídos tendo em conta a largura de banda disponível. É demonstrado que o uso de uma função de custo de ligação dinâmica leva a um aumento notável na utilização da rede relativamente ao caso onde o custo de ligação é constante.

São propostas funções de custo de ligação configuráveis para algoritmos de pré-planeamento de construção de árvores, tentando atingir o melhor compromisso entre o balanceamento de carga e o atraso médio possível. Nesta primeira abordagem, onde são construídas MSTs baseadas na origem para avaliar as diferentes métricas de custo de ligação, não foi atingido o compromisso óptimo entre o balanceamento de carga e o atraso médio, uma vez que não é levado em conta o atraso de ligação nem a largura de banda disponível na métrica do custo da ligação. Assim, é proposta uma alteração ao algoritmo anterior que colmata essa lacuna. No entanto, é muito simples perceber que os resultados da engenharia de tráfego são altamente dependentes da relação entre a variação do atraso da ligação e a variação da largura de banda da ligação usada. É verificado que com este esquema não se consegue atingir uma boa distribuição de tráfego quando há uma grande variação do atraso de ligação. Nos algoritmos de construção de árvores pré-planeadas, são preferidas métricas de custo de ligação mais complicadas e flexíveis. Como resultado, é considerada uma nova função de custo de ligação que depende de α , sendo α o parâmetro de custo de ligação. Este parâmetro toma o valor 1 quando o custo de ligação é determinado tendo em conta apenas o atraso de ligação; pelo contrário, toma o valor 0 se é determinado apenas pela carga de ligação. Com α controla-se portanto o peso entre o atraso e a carga de ligação. O algoritmo de construção de árvores que usa esta função, denominada $(\alpha, 1-\alpha)$, atinge um forte balanceamento de carga.

Em comparação com este, são usados dois algoritmos sugeridos por outros autores para avaliar o desempenho em redes Metro Ethernet no que toca a engenharia de tráfego. Considere-se os diferentes algoritmos considerados:

- Algoritmo $(\alpha, 1-\alpha)$ – usa a métrica de custo de ligação exposta anteriormente;
- Algoritmo $1/C$ – usa a métrica de custo de ligação proposta em [8];
- Algoritmo D/C – usa a métrica de custo de ligação proposta em [14].

A simulação é feita com o uso dos três algoritmos referidos anteriormente aplicados à topologia de rede dos E.U.A.. Em todas as simulações, os pedidos de tráfego são gerados aleatoriamente. A capacidade das ligações é de 1000 Mb/s. A saída da simulação é o formato das STs. De acordo com as árvores formadas e os requisitos do tráfego, é calculado o atraso médio e a variação da largura de banda usada em todas as ligações. A largura de banda utilizada é uma métrica importante para se poder avaliar o desempenho do balanceamento de carga.



Figura 37 - Rede dos E.U.A [10]

A rede dos E.U.A. é apresentada na Figura 37 que contém 12 nós e 17 ligações. Os valores observados nas ligações são o atraso de ligação correspondente à mesma, que é o comprimento em milhas da ligação. São gerados 50 fluxos de tráfego gerados aleatoriamente e, seguidamente, são formadas as MSTs usando os três algoritmos considerados.

| Algorithm | Average Used Bandwidth | Load Variance | Average Delay |
|------------|------------------------|---------------|---------------|
| (0, 1) | 122.71 | 21.48 | 1823.46 |
| (1, 0) | 114.12 | 37.42 | 1587.89 |
| (0.5, 0.5) | 116.30 | 27.51 | 1669.00 |
| 1/C | 114.12 | 29.22 | 1707.00 |
| D/C | 114.12 | 36.24 | 1589.53 |

Tabela 3 – Resultados da simulação na rede dos E.U.A. [10]

A Tabela 3 mostra a largura de banda média usada na rede, a variação da largura de banda usada em todas as ligações e o atraso médio da rede, tendo sido, para o efeito, efectuadas em média 20 simulações.

Das simulações tiram-se as seguintes ilações:

- A variação da largura de banda de ligação usada é mais pequena quando o algoritmo $(\alpha, 1-\alpha)$, com $\alpha = 0$, é usado, apesar de o atraso médio ser significativamente maior do que nos outros casos. Isto deve-se ao facto de o algoritmo tentar o seu melhor para atingir o balanceamento de cargas sem considerar o atraso do caminho;
- O algoritmo $(\alpha, 1-\alpha)$, com α a 1, atinge o menor atraso médio, mas a variação da largura de banda de ligação usada é a maior. Por α ser igual a 1, o objectivo do algoritmo é a optimização do atraso médio da rede sem considerar o balanceamento de carga;
- O algoritmo $(\alpha, 1-\alpha)$, com o α a 0.5 atinge um balanceamento de carga considerado bom e o atraso médio é pequeno quando comparado com outros casos. Neste caso, tem-se um compromisso, atrás referido, entre o balanceamento de carga e o atraso médio. O algoritmo usa o caminho de carga marginal cujo atraso não é significativamente grande quando comparado com outros caminhos, logo, o atraso médio não se vai deteriorar de forma significativa;
- O algoritmo 1/C atinge um bom balanceamento de carga. Contudo, o atraso médio é imprevisível, uma vez que o algoritmo é proposto para otimizar o balanceamento de carga sem considerar o atraso médio. De qualquer forma, o balanceamento de carga resultante é pior do que no algoritmo $(\alpha, 1-\alpha)$;
- O algoritmo D/C atinge um desempenho em termos de balanceamento de carga e de atraso médio comparável ao algoritmo $(\alpha, 1-\alpha)$ com α igual a 1. Apesar do algoritmo considerar os dois casos (atraso e largura de banda de ligação), os resultados da engenharia de tráfego dependem altamente da relação entre a variação do atraso de ligação e a variação da largura de banda de ligação. Na simulação efectuada, a variação do atraso de ligação é superior à variação de largura de banda da ligação, logo, o algoritmo D/C não consegue atingir uma boa distribuição de tráfego.

Considere-se a figura seguinte que é também referente à rede dos E.U.A. ilustrada anteriormente:

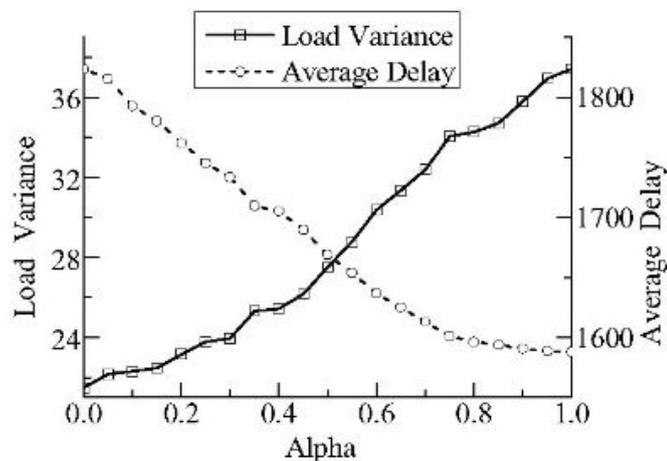


Figura 38 - Resultados da engenharia de tráfego com diferentes parâmetros [10]

A Figura 38 mostra de que forma o parâmetro α afecta o desempenho em termos de balanceamento de carga e de atraso médio. Verificamos que, à medida que α aumenta, a variação de carga aumenta e o atraso médio diminui.

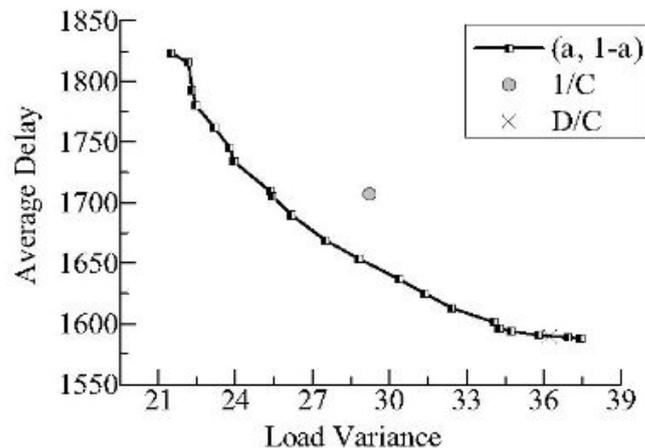


Figura 39 - Variação de carga vs atraso médio [10]

A forma como é escolhido o parâmetro α vai depender dos requisitos da engenharia de tráfego. Com o resultado das simulações é obtida a Figura 39, com a variação da largura de banda de ligação usada no eixo horizontal e o atraso médio no eixo vertical. Pode-se concluir que, se o objectivo é a obtenção de uma melhor balanceamento de carga, o atraso médio tem que subir. Também através da Figura 39 pode-se dizer que dados os mesmos requisitos de balanceamento de carga, o algoritmo $(\alpha, 1-\alpha)$ consegue obter o atraso médio mais baixo do que o algoritmo 1/C. Como já foi referido, no ambiente de simulação deste artigo, o algoritmo D/C é equivalente ao algoritmo $(\alpha, 1-\alpha)$ quando α se aproxima de 1. De qualquer forma, este último consegue abranger vários requisitos de engenharia de tráfego, atribuindo o valor apropriado a α , enquanto que os outros dois algoritmos são rígidos e altamente dependentes do contexto da rede. Pode-se concluir, portanto, que o algoritmo $(\alpha, 1-\alpha)$ consegue atingir uma melhor engenharia de tráfego e é mais flexível.

3.3 Cenários Estáticos

Os três artigos [11][12][13] abordam a melhoria do balanceamento de carga, usando para isso um cenário estático. No caso de [11], o autor propõe também uma melhoria da resistência da rede Ethernet a falhas, tudo isto assente no caso de existir apenas uma região. É apresentada uma solução que determina os parâmetros de configuração MSTP, que optimize o balanceamento de carga e minimize o impacto das falhas de rede. O artigo [12] faz a passagem para o caso de existirem múltiplas regiões e compara este caso,

com o estudado no artigo anterior. Em [13] é também explorada a relação entre o balanceamento de carga e o *service disruption* assente numa falha de ligação quando a rede tem apenas uma região ou com múltiplas regiões, e com diferente número de STs adicionais.

Em [11] é dada uma rede Ethernet composta por um conjunto de *switches* ligados através de ligações ponto-a-ponto e por um conjunto de VLANs, cada uma definida por um conjunto de pedidos de tráfego que devem ser suportados pela rede. Posto isto, são determinados os parâmetros MSTP apropriados para a implementação das STs que tem que estar operacionais e a atribuição de cada VLAN de tráfego a uma das STs. A solução aqui apresentada separa a determinação das ligações activas que definem a ST, da determinação dos parâmetros MSTP apropriados. Para qualquer ST desejada, é sempre possível determinar um conjunto de parâmetros MSTP que façam com que as ligações pretendidas fiquem activas.

Cada fluxo de tráfego é encaminhado pela rede pelo percurso definido pela ST a que a sua VLAN foi atribuída. A carga de um arco pode ser determinada somando os pedidos de fluxo de tráfego que utilizaram esse mesmo arco. É definido um *array* de cargas de uma solução particular, o *array* que é formado por todos os valores das cargas resultantes de todos os arcos da rede, numa ordem decrescente: o primeiro elemento contém o valor mais alto de carga, o segundo contém o segundo valor mais alto de carga e assim sucessivamente. Este *array* é utilizado para comparar o balanceamento de carga entre diferentes soluções. Para duas soluções dadas, 1 e 2, as quais possuem um array de carga L1 e L2, respectivamente, considera-se que a solução 1 é melhor do que a 2 se L1 tiver menor valor do que L2 na primeira posição do *array* em que o valor de ambos é diferente. O objectivo é seleccionar a solução com menor valor de pior carga.

O algoritmo elaborado foi testado em dois casos de estudo diferentes. O primeiro considera um pequeno exemplo que permite ilustrar os resultados da configuração dos parâmetros MSTP atribuídos. O segundo caso de estudo lida com um cenário de rede mais realista. O primeiro caso considera o exemplo apresentado na Figura 40.

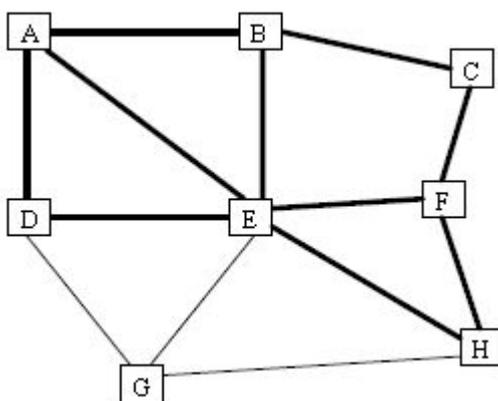


Figura 40 - Rede do primeiro caso de estudo [11]

Este caso é baseado numa rede composta por 8 *switches* e 13 ligações ponto-a-ponto. Esta rede tem ligações com capacidade de 1 Gbps (linhas grossas) e ligações com capacidade de 100 Mbps (linhas

finas). Os parâmetros iniciais da ST são: o *switch* A tem o menor Bridge ID, os valores de *Port Cost* eram iguais a 1 para todas as portas de 1 Gbps e iguais a 10 para todas as portas de 100 Mbps.

No que toca aos pedidos de tráfego, são consideradas 12 VLANs e 32 fluxos de tráfego (2 VLANs com 6 fluxos de tráfego cada e 10 VLANs com 2 fluxos de tráfego cada). Os nós de origem e destino dos fluxos de tráfego são seleccionados aleatoriamente entre todos os *switches*, e as suas capacidades foram geradas aleatoriamente entre 10 Mbps e 100 Mbps. O conjunto de fluxos de tráfego gerados tem a particularidade de a largura de banda total dos fluxos de tráfego originados no *switch* G ser igual 113 Mbps, que é um valor acima da capacidade de qualquer das suas ligações de saída.

Esta técnica foi aplicada para valores de n entre 1 e 4, sendo que n representa o número de instâncias ST.

| Index of Array L | n | | | |
|--------------------|--------|-------|-------|-------|
| | 1 | 2 | 3 | 4 |
| 1 | 113,00 | 60,00 | 60,00 | 60,00 |
| 2 | 65,00 | 56,00 | 56,00 | 56,00 |
| 3 | 37,90 | 50,00 | 50,00 | 50,00 |
| 4 | 37,60 | 42,00 | 42,00 | 42,00 |
| 5 | 27,00 | 40,00 | 40,00 | 40,00 |
| 6 | 26,00 | 28,90 | 20,00 | 20,00 |
| 7 | 21,80 | 28,00 | 18,70 | 18,40 |
| 8 | 20,10 | 26,00 | 16,80 | 16,80 |
| 9 | 18,50 | 25,00 | 16,10 | 16,40 |
| 10 | 14,70 | 23,90 | 16,00 | 16,00 |
| 11 | 12,80 | 20,00 | 14,70 | 15,00 |
| 12 | 10,70 | 19,00 | 14,10 | 14,10 |
| 13 | 7,20 | 17,20 | 13,70 | 13,70 |
| 14 | 6,00 | 16,90 | 13,00 | 13,00 |
| 15 | 0,00 | 14,80 | 13,00 | 13,00 |
| 16 | 0,00 | 14,00 | 13,00 | 11,00 |
| 17 | 0,00 | 12,00 | 11,00 | 10,10 |
| 18 | 0,00 | 12,00 | 10,10 | 10,00 |
| 19 | 0,00 | 11,20 | 9,50 | 9,90 |
| 20 | 0,00 | 11,20 | 9,20 | 9,20 |
| 21 | 0,00 | 7,40 | 8,40 | 8,10 |
| 22 | 0,00 | 7,00 | 8,00 | 8,00 |
| 23 | 0,00 | 3,00 | 7,70 | 7,70 |
| 24 | 0,00 | 2,00 | 7,00 | 7,00 |
| 25 | 0,00 | 0,00 | 6,90 | 6,50 |
| 26 | 0,00 | 0,00 | 2,20 | 5,20 |

Tabela 4 - Array L da melhor solução encontrada para cada valor de n [11]

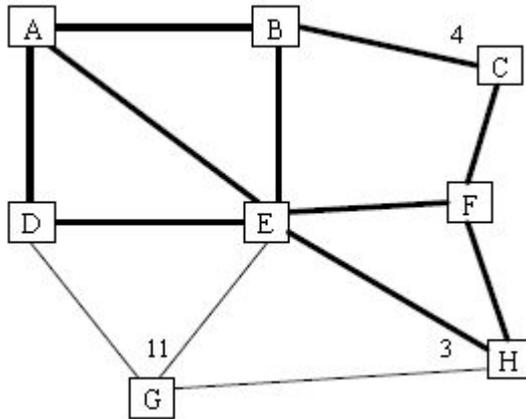


Figura 41 - Árvore com valores de *Port Cost* [11]

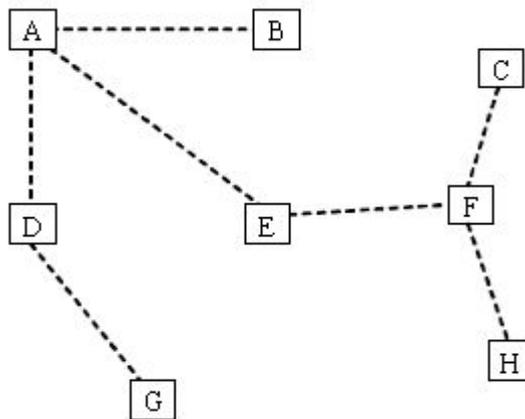


Figura 42 - Ligações activas da ST resultante [11]

O caso $n = 1$ representa o caso STP tradicional: a Figura 41 mostra os valores de *Port Cost* alterados pela técnica usada e a Figura 42 mostra as ligações activas resultantes. Neste caso, o resultado presente na Tabela 4 (o primeiro valor de L é superior a 100% da carga de ligação) mostra que não é possível proporcionar uma configuração STP sem haver congestionamento e perda de alguns fluxos.

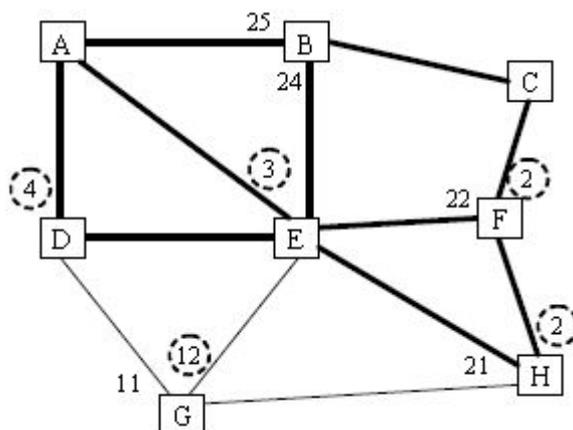


Figura 43 - Árvore com Port Cost [11]

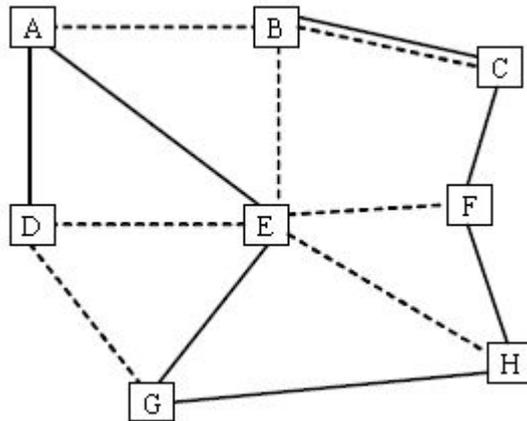


Figura 44 - Ligações activas das duas STs [11]

Para $n = 2$, é possível determinar uma configuração onde o valor da pior carga é de 60%, que fica bastante abaixo do nível de congestionamento de ligação. A Figura 43 mostra os valores do *Port Cost* determinados para as duas *Spanning Trees* e a Figura 44 mostra as duas STs com as respectivas ligações activas, uma representada pelas linhas contínuas e a outra por linhas com tracejado. Este caso demonstra que com o MSTP, a rede pode suportar mais tráfego do que com o tradicional STP. Note-se que o número de ligações comuns entre as duas STs é mínimo, o que também minimiza o impacto de falhas de ligação em fluxos de tráfego.

Para o caso de existirem 3 ou 4 instâncias ST, a Tabela 4 apenas apresenta ganhos marginais e não são obtidas melhoras nos 5 piores casos de carga. Daqui resulta a seguinte conclusão: um balanceamento de carga aproximadamente óptimo pode ser obtido com um número pequeno de STs, não penalizando significativamente com isto o processamento dos *switches*.

Considere-se o segundo caso de estudo.

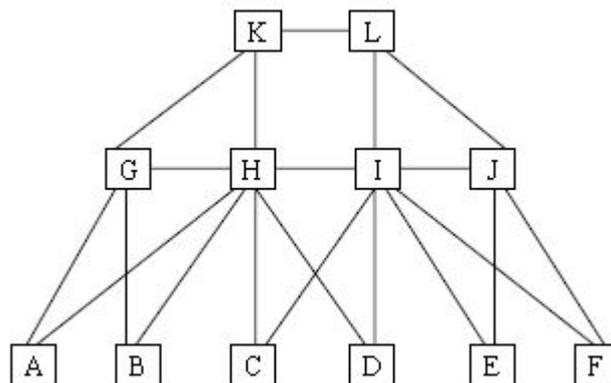


Figura 45 - Esquema da rede A [11]

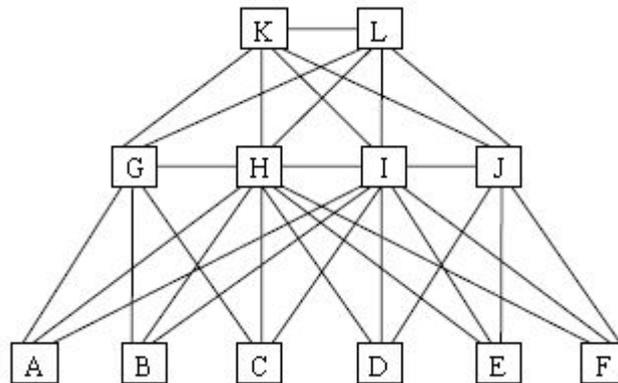


Figura 46 - Esquema da rede B [11]

Este segundo caso de estudo considera cenários de rede mais realistas. A primeira rede, a rede A, possui 12 *switches* e 20 ligações ponto-a-ponto (Figura 45). Nesta rede, os *switches* de A a F são *switches* de acesso, onde o equipamento do utilizador está ligado, e os *switches* de G a L são *switches* de transporte que proporcionam ligações redundantes entre *switches* de acesso. Cada *switch* de acesso está ligado a dois *switches* de transporte diferentes para poder recuperar de uma falha de ligação. A segunda rede, a rede B, é baseada na rede A, à qual se adicionaram 10 ligações ponto-a-ponto. Nesta rede, cada *switch* de acesso está ligado a 3 *switches* de transporte diferentes, aumentando número de caminhos alternativos entre *switches*, e há também mais ligações entre *switches* de transporte. Em ambas as redes, todas as ligações têm a capacidade de 1 Gbps.

No que toca a pedidos de tráfego, são considerados 30 VLANs e 60 fluxos de tráfego (cada VLAN com 2 fluxos de tráfego). Os pedidos de capacidade para os fluxos de tráfego foi considerado com a seguinte distribuição: 10 fluxos com 100 Mbps, 15 fluxos com 50 Mbps, 15 fluxos com 10 Mbps e 20 fluxos com 2 Mbps. Os nós origem e destino de cada fluxo de tráfego são seleccionados aleatoriamente de entre o conjunto de *switches* de A a F. O mesmo conjunto de fluxos de tráfego foi considerado para ambas as redes. Mais uma vez foi usada a técnica anteriormente referida para os casos em que n varia entre 1 e 4. A Tabela 5 apresenta os 14 primeiros elementos do array de cargas L da melhor solução encontrada, para a rede A.

| Index of Array L | n | | | |
|--------------------|-------|-------|-------|-------|
| | 1 | 2 | 3 | 4 |
| 1 | 75,40 | 37,80 | 37,80 | 37,80 |
| 2 | 75,40 | 37,80 | 37,80 | 37,80 |
| 3 | 75,40 | 37,80 | 37,80 | 37,80 |
| 4 | 75,40 | 37,80 | 37,80 | 37,80 |
| 5 | 75,40 | 37,60 | 37,60 | 37,60 |
| 6 | 75,40 | 37,60 | 37,60 | 37,60 |
| 7 | 64,20 | 37,60 | 37,60 | 37,60 |
| 8 | 64,20 | 37,60 | 37,60 | 37,60 |
| 9 | 58,40 | 35,00 | 32,20 | 32,20 |
| 10 | 58,40 | 35,00 | 32,20 | 32,20 |
| 11 | 58,00 | 34,40 | 32,00 | 32,00 |
| 12 | 58,00 | 34,40 | 32,00 | 32,00 |
| 13 | 56,60 | 34,40 | 31,00 | 30,60 |
| 14 | 56,60 | 34,40 | 31,00 | 30,60 |

Tabela 5 - Resultados do caso de estudo da rede A [11]

Neste caso, a melhor configuração STP necessita de um pior caso de 75.4% de carga de ligação em 6 ligações diferentes, e o menor valor de carga, entre todas as ligações activas, é de 56.6% (a Tabela 5 mostra apenas os 14 piores casos). Com uma configuração MSTP, baseada em duas instâncias ST, o resultado é muito melhor, o pior caso de carga de ligação diminuiu para 37.8%, que é aproximadamente metade do valor no caso do STP, e este valor de pior caso é ainda melhor do que todos os 14 piores valores de carga de ligação no caso do STP. Em relação ao caso de termos 3 ou mais STs, os resultados da Tabela 5 mostram ganhos marginais e não se fizeram sentir melhoras nos valores dos 8 piores casos de carga.

A Tabela 6 apresenta os primeiros 14 elementos do *array* de cargas L para as melhores soluções encontradas para o caso de estudo B.

| Index of Array L | $\%$ | | | |
|--------------------|-------|-------|-------|-------|
| | 1 | 2 | 3 | 4 |
| 1 | 75,40 | 33,20 | 25,20 | 25,20 |
| 2 | 75,40 | 33,20 | 25,20 | 25,20 |
| 3 | 75,40 | 33,20 | 25,20 | 25,20 |
| 4 | 75,40 | 33,20 | 25,20 | 25,20 |
| 5 | 64,20 | 33,20 | 25,20 | 25,20 |
| 6 | 64,20 | 33,20 | 25,20 | 25,20 |
| 7 | 58,40 | 33,00 | 25,20 | 25,20 |
| 8 | 58,40 | 33,00 | 25,20 | 25,20 |
| 9 | 58,00 | 32,80 | 25,00 | 25,00 |
| 10 | 58,00 | 32,80 | 25,00 | 25,00 |
| 11 | 56,60 | 32,80 | 25,00 | 25,00 |
| 12 | 56,60 | 32,80 | 25,00 | 25,00 |
| 13 | 0,00 | 32,60 | 21,40 | 21,40 |
| 14 | 0,00 | 32,60 | 21,40 | 21,40 |

Tabela 6 - Resultados do caso de estudo da rede B [11]

Neste caso, como há mais ligações, a topologia possibilita a existência de mais STs com menos ligações em comum e um melhor balanceamento de carga pode ser obtido. Na rede B, as duas instâncias ST levam a que o valor do pior caso de carga diminua de 75.4% para 33.2% e as três instâncias ST oferecem a possibilidade de diminuir ainda mais esse valor, de 33.2% para 25.2%. Em relação a ter 4 instâncias ST, não há melhorias significativas.

Estes resultados mostram a relação entre o número óptimo de instâncias ST e a topologia da rede: o uso de n instâncias ST no MSTP torna-se benéfico se a topologia possibilitar a existência de n STs disjuntas (ou quase disjuntas), ou seja, que não partilham ligações. No caso da rede A, a topologia possibilita a existência de duas STs praticamente disjuntas pelo que foram verificadas melhorias significativas de balanceamento de carga com duas instâncias ST, enquanto que com mais instâncias ST as melhorias pouco se fizeram sentir. No caso da rede B, a sua topologia permite a obtenção de três STs disjuntas pelo que a melhoria no balanceamento de carga é fortemente sentido quando se usam três instâncias ST ao passo que com quatro instâncias ST as melhorias são marginais.

Comparando os resultados da rede A com a rede B, pode-se concluir:

- Uma ST não consegue utilizar as ligações adicionais para obter um balanceamento de carga significativo;
- Duas instâncias ST conseguem melhorar o balanceamento de carga com a adição de ligações, mas com pequenas melhorias;
- Três e quatro instâncias ST conseguem melhorar significativamente o balanceamento de carga com a adição de ligações.

Estes resultados mostram que, com a configuração adequada, o MSTP permite ao gestor de rede adicionar novas ligações e desta forma balancear a carga de tráfego dentro das ligações disponíveis.

No que toca ao desempenho da técnica proposta, é usado como critério de paragem um número fixo de 100000 iterações. Na prática, uma solução que minimiza um número pequeno de piores casos da carga é suficiente, uma vez que os restantes valores se tornam pequenos. Com este critério, são necessárias bastante menos iterações. A técnica proposta é um processo estocástico, o que quer dizer que simulando duas vezes, pode-se encontrar duas soluções diferentes. Para cada caso foram efectuadas 10 simulações. A melhor solução para os 8 valores do pior caso de cargas foi sempre encontrada abaixo das 1000 iterações para $n = 1$, e abaixo das 100 iterações para $n > 1$. Isto quer dizer que esta técnica consegue encontrar uma boa solução com um tempo de processamento sempre abaixo dos 5 segundos numa plataforma computacional standard. Para o primeiro caso de estudo, os tempos de processamento foram muito menores. Estes resultados mostram que a técnica proposta é uma solução válida e simples para obter automaticamente os parâmetros MSTP de configuração para melhorar o balanceamento de carga e a resistência das redes Ethernet a falhas.

O artigo [12] debruça-se sobre a mesma problemática do [11], explorando a alternativa de redes Ethernet usando o MSTP com múltiplas regiões, uma vez que esta proporciona uma melhor resistência da rede, ou seja, a rede é menos susceptível à ocorrência de falhas.

Aos parâmetros de rede Ethernet dados pelo trabalho anterior, junta-se também outro dado inicial: um conjunto de regiões definidas na rede. Com isto determinam-se, para além da atribuição de VLANs às STs, os parâmetros MSTP apropriados à implementação da CST e das STs adicionais.

Algumas alterações têm que ser efectuadas no algoritmo. De referir que é adicionado código referente à determinação do conjunto de ligações da CST que optimizam o resultado do *array* de carga L . Em cada iteração é gerada uma ST assente no grafo G , posteriormente esta ST é melhorada em relação ao *array* de cargas de ligações, L_G e, se o resultado de L_G for melhor do que o L_G já existente, é guardado o último L_G obtido como a melhor solução encontrada até ao momento. Seguidamente, é tratada cada região separadamente. É determinada um conjunto de ligações da CST dentro da região e de todas as STs adicionais optimizando o resultado do seu *array* de carga. Para cada região é gerada um conjunto de STs assentes no grafo G_i , o que determina uma atribuição de um *array* que indica a ST atribuída a cada VLAN, juntamente com o resultado do *array* de carga L_{G_i} . Mais uma vez compara-se este último *array* com o que estava previamente guardado em L_{G_i} para se guardar o melhor caso entre os dois. De seguida são determinados os parâmetros MSTP apropriados para a CST e todas as STs adicionais assentes no grafo.

Os casos de estudo deste artigo consideram a rede apresentada na Figura 47.

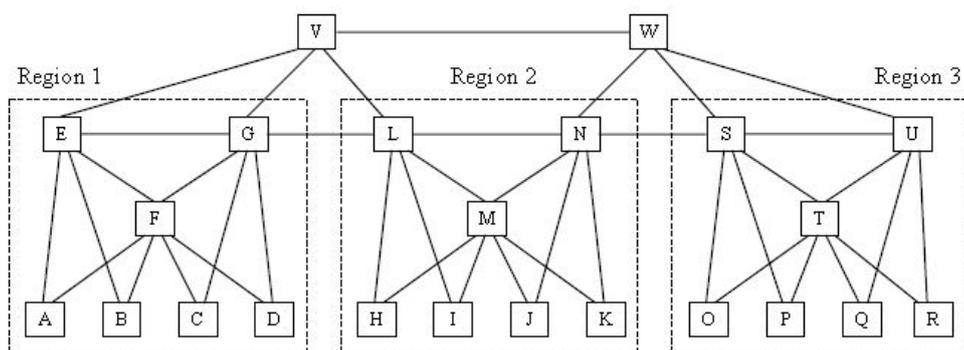


Figura 47 - Rede Ethernet com 23 nós, 42 ligações e 3 regiões [12]

Na Figura 47 está representada uma rede Ethernet com 23 nós, 42 ligações e 3 regiões, em que todas as ligações têm a capacidade de 1 Gbps. Os switches de A a D, de H a K e de O a R são switches de acesso, onde os equipamentos dos clientes são ligados, e os switches V e W são switches gateway que fazem ligação entre a rede Ethernet e outras redes core.

No que toca aos pedidos de tráfego, são consideradas 51 VLANs (cada VLAN com 2 fluxos de tráfego) seleccionadas aleatoriamente entre todos os pares de switches de acesso e todos os pares de gateway de acesso. Os valores dos pedidos foram seleccionados aleatoriamente entre 4 valores diferentes (10, 20, 50 e 100 Mbps) considerando três casos de estudo: a percentagem total de pedidos internos a uma região é de 20% para o caso de estudo A, 50% para o caso de estudo B e de 80% para o caso de estudo C.

Os três casos foram resolvidos considerando um número n de STs adicionais, dentro de cada região, igual a 1 e 2. Em todos os casos, o caso $n = 2$ não encontrou melhor solução do que a melhor solução do caso $n = 1$. A Tabela 7 apresenta os resultados obtidos.

| Case Study | n | Index of first 10 positions of Load Array (%) | | | | | | | | | |
|------------|-----|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| A | 0 | 76 | 76 | 70* | 70* | 68 | 68 | 58 | 58 | 46* | 46* |
| | 1 | 76 | 76 | 68 | 68 | 59* | 59* | 58 | 58 | 40* | 40* |
| B | 0 | 55* | 55* | 51 | 51 | 49 | 49 | 46* | 46* | 43 | 43 |
| | 1 | 51 | 51 | 49 | 49 | 43 | 43 | 36* | 36* | 23* | 23* |
| C | 0 | 41* | 41* | 39* | 39* | 34* | 34* | 32* | 32* | 32* | 32* |
| | 1 | 29 | 29 | 22* | 22* | 22* | 22* | 21* | 21* | 20* | 20* |

Tabela 7 - Os 10 valores mais altos do array de carga para cada caso de estudo [12]

No caso de estudo A, a maior parte dos valores das piores cargas estão em ligações fora da região e, uma vez que as MSTs apenas conseguem melhorar o balanceamento de carga dentro das regiões, há uma melhoria mínima no uso de uma ST adicional. Por outro lado, a ST adicional proporciona uma melhoria significativa no caso de estudo C, uma vez que todos os valores das piores cargas na solução melhor de uma única CST correspondem a ligações dentro de regiões. O caso de estudo B está no meio dos dois resultados anteriores: a ST adicional possibilita uma pequena diminuição da pior carga de ligação de 55% a 51%.

Para todos os casos o número de iterações foi igual a 10000. A primeira parte do algoritmo demorou 160 segundos, no máximo, com as melhores soluções a serem encontradas nas primeiras 10 iterações para todos os casos. A segunda parte do algoritmo demorou, no máximo, 2 segundos, com as melhores soluções a serem encontradas nas primeiras 1000 iterações para todos os casos. Estes resultados mostram que o procedimento proposto é bastante eficiente e capaz de obter soluções com um pequeno tempo de processamento.

Para comparar as duas abordagens, é usado o algoritmo de [11] para obter a melhor solução para estes três casos, assumindo a rede global como uma só região. Os resultados são apresentados na Tabela 8.

| Case Study | n | Index of first 10 positions of Load Array (%) | | | | | | | | | |
|------------|---|---|----|----|----|----|----|----|----|----|----|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| A | 1 | 70 | 70 | 68 | 68 | 59 | 59 | 58 | 58 | 52 | 52 |
| | 2 | 31 | 31 | 31 | 31 | 30 | 30 | 30 | 30 | 30 | 30 |
| | 3 | 31 | 31 | 31 | 31 | 30 | 30 | 26 | 26 | 25 | 25 |
| B | 1 | 55 | 55 | 51 | 51 | 46 | 46 | 43 | 43 | 40 | 40 |
| | 2 | 23 | 23 | 23 | 23 | 21 | 21 | 21 | 21 | 21 | 21 |
| | 3 | 23 | 23 | 23 | 23 | 20 | 20 | 20 | 20 | 20 | 20 |
| C | 1 | 41 | 41 | 39 | 39 | 34 | 34 | 32 | 32 | 32 | 32 |
| | 2 | 21 | 21 | 20 | 20 | 20 | 20 | 19 | 19 | 18 | 18 |
| | 3 | 21 | 21 | 20 | 20 | 20 | 20 | 19 | 19 | 17 | 17 |

Tabela 8 - Os 10 valores mais elevados do array de carga, considerando a rede como uma só região [12]

Considere-se inicialmente o caso de estudo C. Neste caso, a maior parte do tráfego é interno às regiões e a solução para a abordagem das 3 regiões apresentada na Tabela 7 (com o valor de pior carga igual a 29%) é apenas ligeiramente pior do que a melhor solução com a abordagem de uma só região apresentada na Tabela 8 (com o valor de pior carga igual a 21%). Neste caso, a abordagem das 3 regiões é uma boa solução dado que pode atingir um balanceamento de carga quase tão bom como a solução óptima de uma única região e isso minimiza o impacto de falha de ligação na rede. No caso A e B, a solução com apenas uma região obtém soluções bastante melhores de balanceamento de carga. De notar que, no caso de estudo A, a abordagem com 3 regiões é ainda pior do que a solução do STP (a pior carga é 70%) e isto acontece porque o STP não necessita de um conjunto de ligações activas. Nestes casos, há um compromisso entre o balanceamento de carga e o impacto de falha de ligação.

Para finalizar, os resultados apresentados na Tabela 8 mostram que soluções aproximadamente óptimas foram obtidas com duas STs adicionais. Não esquecer que, no trabalho anterior [11], uma ST adicional era suficiente para obter o melhor balanceamento de carga. Estas observações mostram que é possível otimizar o balanceamento de carga com um número pequeno de STs, sem que isso penalize significativamente o elevado (*overhead*) processamento dos switches.

O artigo [13] estuda o relacionamento entre o balanceamento de carga e o *service disruption* assente na falha de uma ligação, no caso de quer uma região ou diferentes regiões serem adoptadas, e para diferentes números de STs adicionais. De notar que todo o tráfego cujo percurso de encaminhamento é

alterado sofre temporariamente *service disruption*. O trabalho [12] já abordava o tema da otimização do balanceamento de carga, mas o algoritmo de otimização proposto é menos eficiente do que o apresentado neste e o impacto de falhas de ligações no *service disruption* não foi abordado.

Aqui são melhorados, quer o algoritmo de atribuição dos valores de *Port Cost*, quer o algoritmo de otimização propriamente dito.

Foram solucionados diferentes problemas de instâncias gerados aleatoriamente com o algoritmo proposto. Os casos de estudo aqui apresentados consideram a rede apresentada na Figura 47, a mesma utilizada em [12], com as mesmas características de rede e de tráfego. Os casos de estudo A, B e C são também idênticos.

Todos os casos de estudo foram resolvidos pelo algoritmo de otimização proposto. Seguidamente, as STs obtidas foram usadas para atribuir parâmetros MSTP apropriados, usando o algoritmo de atribuição de *Port Cost*.

Os 3 casos de estudo foram resolvidos assumindo o número máximo de n STs dentro de cada região entre 0 e 2. A Tabela 9 mostra os piores valores de carga de ligação de cada solução obtida.

| Case Study | n | Index of first 8 positions of Load Array (%) | | | | | | | |
|------------|---|--|------|------|------|------|------|------|------|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| A | 0 | 76.0 | 76.0 | 70.0 | 70.0 | 68.0 | 68.0 | 58.0 | 58.0 |
| | 1 | 76.0 | 76.0 | 68.0 | 68.0 | 59.0 | 59.0 | 58.0 | 58.0 |
| | 2 | 76.0 | 76.0 | 68.0 | 68.0 | 59.0 | 59.0 | 58.0 | 58.0 |
| B | 0 | 55.0 | 55.0 | 51.0 | 51.0 | 49.0 | 49.0 | 46.0 | 46.0 |
| | 1 | 51.0 | 51.0 | 49.0 | 49.0 | 43.0 | 43.0 | 36.0 | 36.0 |
| | 2 | 51.0 | 51.0 | 49.0 | 49.0 | 43.0 | 43.0 | 36.0 | 36.0 |
| C | 0 | 41.0 | 41.0 | 39.0 | 39.0 | 34.0 | 34.0 | 32.0 | 32.0 |
| | 1 | 29.0 | 29.0 | 22.0 | 22.0 | 21.0 | 21.0 | 21.0 | 21.0 |
| | 2 | 29.0 | 29.0 | 21.0 | 21.0 | 21.0 | 21.0 | 20.0 | 20.0 |

Tabela 9 - Resultados do caso de estudo baseado em 3 regiões [13]

Estes resultados mostram que uma ST adicional obtém:

- Uma melhoria significativa no balanceamento de carga para o caso de estudo C onde o tráfego é na maioria interno às regiões (pior carga de ligação diminui de 41% para 29%);
- Uma pequena melhoria no balanceamento de carga para o caso de estudo B (pior carga de ligação diminui de 55% para 51%);
- Não há melhorias no caso de estudo A.

Estes resultados mostram que quando se adoptam regiões múltiplas, a quantidade de balanceamento de carga que se pode obter com STs adicionais é limitada.

Foram também considerados os três casos de estudo mas assumindo toda a rede como uma só região. Os 8 valores de pior carga de ligação de cada solução obtida são apresentados na Tabela 10.

| Case Study | n | Index of first 8 positions of Load Array (%) | | | | | | | |
|------------|---|--|------|------|------|------|------|------|------|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| A | 0 | 70.0 | 70.0 | 68.0 | 68.0 | 59.0 | 59.0 | 58.0 | 58.0 |
| | 1 | 31.0 | 31.0 | 31.0 | 31.0 | 30.0 | 30.0 | 26.0 | 26.0 |
| | 2 | 31.0 | 31.0 | 31.0 | 31.0 | 30.0 | 30.0 | 26.0 | 26.0 |
| B | 0 | 55.0 | 55.0 | 51.0 | 51.0 | 46.0 | 46.0 | 43.0 | 43.0 |
| | 1 | 23.0 | 23.0 | 23.0 | 23.0 | 21.0 | 21.0 | 20.0 | 20.0 |
| | 2 | 23.0 | 23.0 | 23.0 | 23.0 | 20.0 | 20.0 | 20.0 | 20.0 |
| C | 0 | 41.0 | 41.0 | 39.0 | 39.0 | 34.0 | 34.0 | 32.0 | 32.0 |
| | 1 | 21.0 | 21.0 | 20.0 | 20.0 | 20.0 | 20.0 | 19.0 | 19.0 |
| | 2 | 21.0 | 21.0 | 20.0 | 20.0 | 20.0 | 20.0 | 19.0 | 19.0 |

Tabela 10 - Resultados do caso de estudo baseado numa região [13]

Estes resultados mostram que uma ST adicional obtém uma melhoria significativa no balanceamento de carga para todos os casos de estudo (pior carga de ligação diminui de 70% para 31%, no caso A, de 50% para 23% no caso B e de 41% para 21% no caso C). Comparando as duas tabelas anteriores, observa-se que em todos os casos, uma única região é melhor. Esta diferença é bastante acentuada no caso de estudo A e B e é apenas ligeira no caso C uma vez que este é o caso onde o tráfego é maioritariamente interno às regiões e, como observado anteriormente, a ST adicional foi eficiente na melhoria do balanceamento de carga. Além disso, duas STs adicionais não proporcionam uma melhoria no desempenho significativa em relação a uma ST adicional, o que significa que o balanceamento de carga aproximadamente óptimo é obtido com uma única ST adicional, e sem penalizar significativamente o elevando processamento dos *switches*.

Foi também analisado neste artigo o *service disruption* de todas as melhores soluções com os parâmetros MSTP atribuídos pelo algoritmo de atribuição de *Port Cost*. Para cada solução, e para cada ligação usada pelo menos numa ST, foi testado:

- A nova ST quando a ligação em questão falha;
- Os fluxos de tráfego cujos percursos são modificados pelas novas STs;
- A soma dos pedidos de fluxos;
- A percentagem de todos os pedidos que sofrem *service disruption*.

Com estes dados foi determinado:

- A média de todas as percentagens de tráfego para todas as ligações que são usadas pelo menos por uma ST;
- A percentagem de todos os pedidos de tráfego que sofre *service disruption* no pior caso de falha de ligação, como se pode verificar na Tabela 11.

| Case Study | n | Average service disruption | | Worst case service disruption | |
|------------|---|----------------------------|----------|-------------------------------|----------|
| | | 3 regions | 1 region | 3 regions | 1 region |
| A | 0 | 16.64 | 16.69 | 37.81 | 34.83 |
| | 1 | 9.59 | 9.29 | 37.81 | 15.42 |
| | 2 | 9.59 | 8.34 | 37.81 | 15.42 |
| B | 0 | 16.19 | 16.51 | 27.36 | 27.36 |
| | 1 | 8.90 | 7.72 | 25.37 | 11.44 |
| | 2 | 8.42 | 7.52 | 25.4 | 11.44 |
| C | 0 | 13.25 | 13.25 | 20.40 | 20.40 |
| | 1 | 7.89 | 6.72 | 14.43 | 10.45 |
| | 2 | 7.60 | 6.81 | 14.43 | 10.45 |

Tabela 11 - Resultados do *service disruption* [13]

Quer na análise da média, quer na análise do pior caso, os resultados mostram um ponto importante. A abordagem de uma única região é sempre melhor quando são consideradas STs adicionais (mesmo no caso de estudo C, onde o tráfego é na maioria interno às regiões). Isto pode parecer inesperado uma vez que os cenários com múltiplas regiões garantem que as falhas de ligações dentro das regiões não alteram as STs fora da região. De qualquer forma, quanto melhor for o balanceamento de carga da rede e o algoritmo de atribuição de parâmetro MSTP proposto, que minimiza o número de ligações que mudam de estado quando se dá uma falha, melhores são os resultados de *service disruption* na abordagem de uma única região. Além disso, duas STs adicionais não proporcionam melhorias significativas no *service disruption* em relação a uma ST adicional, o que significa que, como em análises anteriores, soluções aproximadamente óptimas são obtidas com uma única ST adicional, não penalizando significativamente o processamento elevado dos *switches*.

O algoritmo de atribuição de parâmetros MSTP proposto tem uma complexidade polinomial relativa ao número de *switches* e ao número de ligações e, desta forma, o algoritmo corre com tempos computacionais bastante baixos. O algoritmo de optimização proposto foi corrido em todos os casos com um tempo máximo de 30 minutos. Na prática, uma solução que minimiza um número pequeno de valores de pior carga (assumam-se os 6 valores de pior carga) é suficiente uma vez que os restantes valores se tornam pequenos. Com este critério, é necessário muito menos tempo de processamento. Mais uma vez, o algoritmo proposto é um processo estocástico, o que significa que corrido em diferentes alturas pode obter resultados diferentes. Foram corridos 5 vezes todos os casos aqui presentes e, de todas as vezes, a melhor solução dos 6 valores de pior carga foi sempre encontrada decorridos menos de 2 minutos de tempo de processamento para casos com 1 ou 2 STs adicionais. Estes tempos deixam os autores confiantes de que os resultados obtidos são de facto óptimos para os 6 valores de pior carga.

3.4 Divisão da rede em regiões

O artigo [14] propõe um algoritmo para a definição de regiões MST num domínio de rede empresarial que tem como objectivo proporcionar o melhor tempo de convergência, a reutilização dos VLAN *tags*, a protecção contra falhas e um tamanho de domínio *broadcast* óptimo.

Nesta abordagem, membros de diferentes VLANs podem ser associadas a um *switch*, ou seja, o *switch* pode fazer parte de múltiplas regiões. Quando um *switch* faz parte de várias regiões, as ligações associadas a esse *switch* podem agir como membros de mais do que uma região. Esta técnica é bastante flexível e escalável. Os *switches* podem ser adicionados ou removidos de uma região com base no facto de pertencerem a uma VLAN da região. Por exemplo, adicionar um novo membro a uma VLAN *v* implica a adição desse membro a um *switch*. Se o *switch* não fizer parte da região onde *v* pertence, o *switch* é adicionado a essa mesma região. No entanto, este método usado para construir regiões deve ir de encontro aos seguintes requerimentos:

- Proporcionar protecção;
- Garantir no mínimo duas ligações físicas entre regiões;
- Oferecer uma rápida recuperação de falhas;
- Assumir um compromisso entre o tamanho da região e o desempenho;
- Garantir um tamanho de domínio *broadcast* óptimo;
- Permitir a reutilização dos VLAN *tags*.

A protecção contra falhas de ligação pode ser conseguida garantindo, pelo menos, duas ligações físicas entre regiões pelo que o primeiro e o segundo ponto estão relacionados. O tamanho do domínio *broadcast* depende do tamanho do domínio e o seu diâmetro. A rápida recuperação de falhas é atingida mais facilmente se a região onde se dá a falha for pequena pelo que, os terceiro, quarto e quinto objectivos estão também relacionados. No entanto, ter demasiadas regiões pequenas na rede resulta na circulação de mais mensagens de controlo, o que faz com que o uso de regiões deixe de ter sentido. Assim sendo, surge a necessidade de encontrar um tamanho óptimo para uma região, que deve ter em consideração o número de nós e o diâmetro da mesma.

O algoritmo desenvolvido está dividido em duas partes. A primeira parte consiste na formação de regiões, ligadas entre si por duas ligações físicas (*2-connected*). A segunda parte é a expansão da região. O algoritmo começa por gerar aleatoriamente uma VLAN e assume-a como sendo uma região. O passo seguinte é fazer com que esta região seja do tipo *2-connected* adicionando novos nós. Se a região puder albergar mais nós, é adicionada outra VLAN e o processo continua. São constantemente geradas regiões até não existirem mais VLANs.

Quando são consideradas regiões mais pequenas, algumas VLANs podem não ser adequadas para nenhuma região disponível. O problema pode passar pelo tamanho da VLAN ou pelo facto de os nós da VLAN estarem muito dispersos. A estas VLANs dá-se o nome de *blocked* VLAN. Com base na disponibilidade

dos VLAN *tags*, ou se faz com que algumas regiões representem uma única VLAN ou se forma uma região com todas as *blocked* VLANs.

Antes de mais, é oportuno introduzir alguns termos/procedimentos utilizados neste artigo, para uma melhor compreensão do mesmo.

VLAN-Region share list (VR list)

Uma *VR list* expressa a percentagem de nós (*switches*) partilhados entre uma VLAN e uma região específica. Esta lista vai ser inicializada a zeros e vai sendo actualizada à medida que a rede se vai dividindo. A lista em questão é, portanto, uma ajuda para escolher as VLANs candidatas para dada uma região.

Observe-se um exemplo de uma *VR list* na Tabela 12.

| | |
|--------|------|
| VLAN 1 | 100% |
| VLAN 2 | 60% |
| VLAN 3 | 70% |
| ... | ... |

Tabela 12 - Exemplo de uma *VLAN-Region share list (VR list)* [14]

Tagging

Este procedimento é usado para identificar a distância a que um nó está da sua região. Quanto menor for o *tag*, menor é a distância entre o nó e a região. Inicialmente, todos os nós da rede que pertencem à região são atribuídos com o *tag* 1. Seguidamente, cada nó com o *tag* 1, encontra todos os nós adjacentes que não têm nenhum *tag*, aos quais é atribuído o *tag* 2. Este processo continua até todos os nós da rede possuírem um *tag*.

Para evitar inconsistências, nunca se atribui um *tag* a um nó que já o possua. Durante o processamento do algoritmo, vão ser adicionados à região nós novos que passam a ter o *tag* 1. Isto resulta no *retagging* de outros nós na rede.

Voting

O procedimento de *voting* é usado para identificar o quanto um nó está ligado a uma região. Todos os nós da rede têm uma hipótese de votar nos seus vizinhos. No entanto, o voto só é atribuído a nós que não façam parte da região actual. Um nó com *tag* k pode dar um voto de $1/k$ a todos os seus vizinhos. Considere-se a seguinte Figura 48.

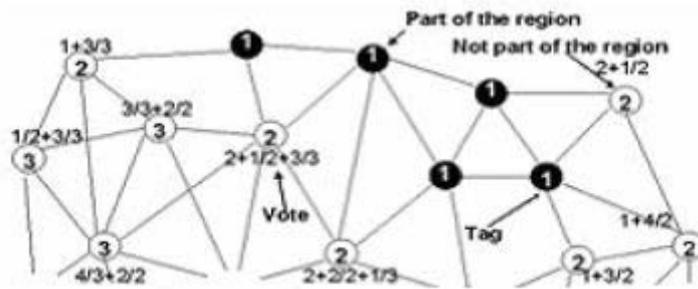


Figura 48 - Exemplo de *tagging* e *voting* [14]

A Figura 48 mostra um grafo com *tags* e *votes*. Esta votação usando '*1/tag*' explica com que actividade um nó está ligado à região e conforme o valor do *tag* aumenta, os nós mais distantes da região vão ter menos votos de baixo valor, o que significa que estes últimos nós vão ser evitados na expansão da região. O processo de *voting* com *tagging* selecciona nós que são imediatamente vizinhos e estão fortemente ligados à região. Isto faz com que se mantenha o diâmetro da região num nível aceitável. Como explicado na secção *tagging*, quando um novo nó passa a fazer parte da região, é necessário fazer um *revoting*.

Coarsening

O procedimento de *coarsening* é usado para reduzir o tamanho de um grafo sem perder as suas propriedades. Este método é utilizado no decorrer do algoritmo num grafo simplificado, reduzindo a complexidade do algoritmo. Observe-se a Figura 49 que representa este método, onde o ponto negro na ilustração da direita, representa todos os pontos e ligações negras da ilustração da esquerda.

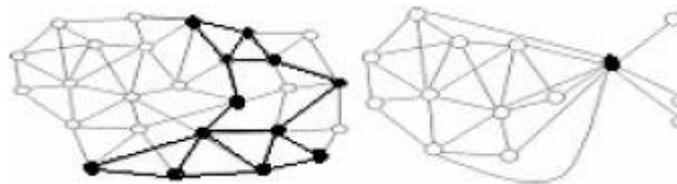


Figura 49 - Exemplo de *coarsening* [14]

Assegurar 2-connectivity

Depois dos processos de *tagging* e de *voting*, o algoritmo extrai um subgrafo *2-connected* da rede com todos os nós da região. Este processo implica a adição de alguns nós à região para garantir que haja sempre duas ligações físicas entre todas as regiões dadas. Estes nós são chamados de *Bridge-nodes* (agem como bridges entre os nós da região).

Apesar de os autores terem como objectivo a minimização do número de *bridge-nodes* na região, os nós extra não vão afectar o desempenho da região ou qualquer dos objectivos propostos. Os *bridge-nodes* adicionais na região proporcionam mais recursos. Uma vez que os nós são partilhados entre regiões,

não há desperdício de recursos e estes *bridge-nodes* vão também participar na selecção de outras VLANs que são bastante próximas da região que está a ser processada quando se dá a expansão do tamanho da região.

As simulações são efectuadas em redes geradas aleatoriamente para estudar o desempenho da abordagem de construção de regiões proposta pelos autores. Estas simulações foram testadas com várias redes com diferente número de *switches* e VLANs. O gerador de regiões da simulação foi testado primeiro para a conectividade dupla (*2-connectivity*). Os resultados provam que pode ser fornecida protecção contra, pelo menos, um nó ou uma falha de ligação.

O tamanho do domínio *broadcast* da região foi testada usando o diâmetro, que é o maior caminho mais curto na rede. Considere-se a Figura 50 que mostra os diferentes diâmetros para regiões com diferente número de nós e com diferente número de VLANs.

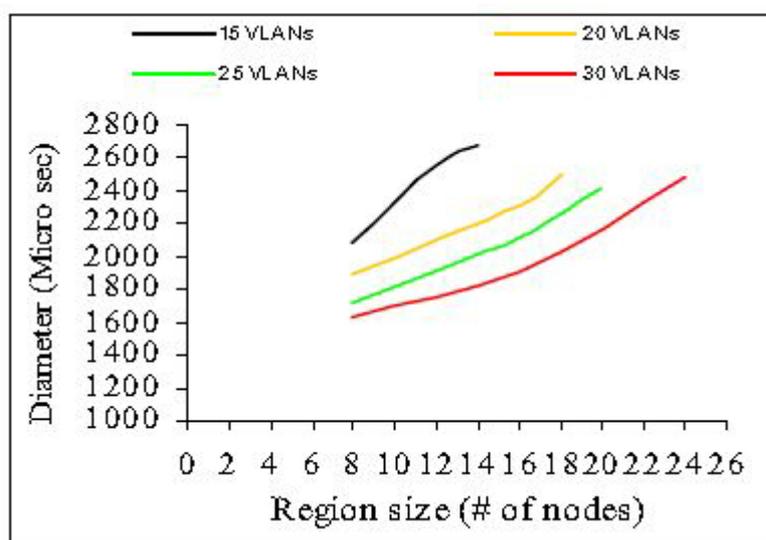


Figura 50 – Diâmetro [14]

Ao observar a Figura 50 percebe-se que quando o tamanho da rede aumenta, o diâmetro também aumenta. Contudo, o aumento do diâmetro é menos acentuado quando há um grande número de VLANs na rede. Note-se que, com o aumento do número de VLANs, a região fica mais pequena, logo, o diâmetro obtido é melhor. Isto indica claramente que, quando se tem mais VLANs para escolher, as regiões estão mais perto do óptimo. Consequentemente, fica provado que a abordagem dos autores é apropriada para redes de grande dimensão com elevado número de VLANs e nós.

Observe-se a Figura 51.

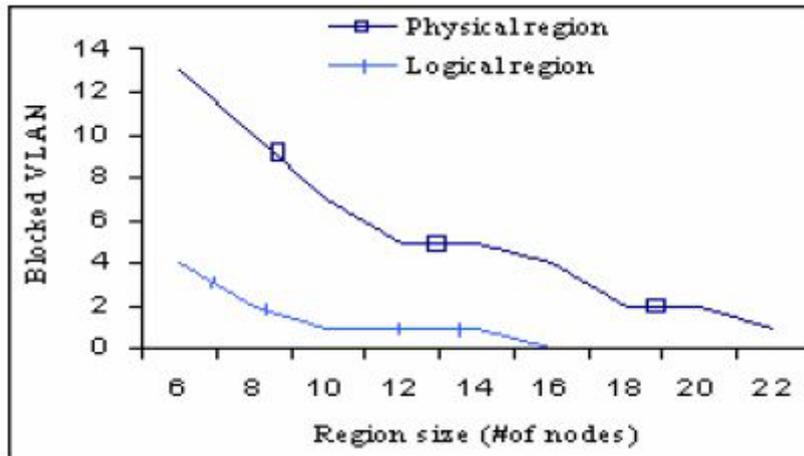


Figura 51 - VLANs bloqueadas [14]

Na Figura 51 é apresentada uma comparação entre a abordagem do autor (regiões lógicas) e as regiões físicas baseadas no número de *blocked VLANs*. Na região física, um *switch* tem permissão para fazer parte de apenas uma região.

Pode-se concluir que os métodos propostos neste artigo suportam a natureza dinâmica das VLANs, onde os nós VLAN podem ser adicionados, mudados de posição ou removidos sem muito esforço. Este método fornece recursos suficientes a uma região para que esta consiga recuperar pelo menos de falhas isoladas. Com as simulações, conclui-se também que esta abordagem pode ser usada para melhorar o desempenho do MSTP dividindo a rede em regiões com base nas VLANs. Este algoritmo pode ser facilmente alargado de forma a construir regiões baseadas em prioridades de tráfego, departamentos, acessos privilegiados, e requisitos de segurança.

3.5 Outros aspectos de Engenharia de tráfego

3.5.1 QoS

As normas do IEEE adoptam uma estrutura *Spanning Tree*. Esta abordagem foi plausível durante algumas décadas, no entanto, recentemente as aplicações multimédia têm chamado a atenção da comunidade de redes, e vários tipos de informação com diferentes requisitos de serviço emergiram. Neste contexto, as abordagens *standard* que assentam numa única ST acarretam os problemas. Em primeiro lugar, uma vez que todo o tráfego de rede passa pela mesma árvore, é frequente dar-se o congestionamento de rede, mesmo em situações onde as ligações bloqueadas possuem bastante largura de banda disponível. Isto causa a degradação da qualidade de serviço para fluxos *QoS-sensitive*. Desta forma, toda a rede é subutilizada e as questões do balanceamento de carga e da engenharia de tráfego tornam-se difíceis de solucionar nestas condições. Em segundo lugar, os standards IEEE referidos possuem uma disciplina de agendamento que dispõe de filas de prioridade simples. No que toca a este assunto, uma vez que o tráfego multimédia é prioritário em relação ao tráfego *best effort*. Estes fluxos de menor prioridade

podem sofrer de *starvation*, (ou seja, podem não ser servidos) se os fluxos multimédia, de maior prioridade, constituírem uma fracção significativa do tráfego. Uma solução para esta situação implica que o gestor de rede decida manualmente as configurações da ST de diferentes VLANs para atingir um balanceamento de carga apropriado. É claramente poupado muito tempo e esforço, se o MSTP puder convergir para uma boa topologia, que tenha em conta o tráfego, sem a intervenção humana. Estes assuntos foram abordados por diversos autores.

O artigo [15] propõe um novo, simples e melhorado MSTP com o objectivo de atingir elevado grau de QoS, tendo em consideração as diferentes características dos vários tipos de tráfego. É proposto um mecanismo MST baseado em QoS para redes WAN que constrói STs baseadas no tipo de tráfego e melhora a qualidade de serviço e a eficácia do balanceamento de carga. Para assegurar a compatibilidade com os protocolos em uso, o mecanismo não requer nenhuma modificação mas apenas uma extensão.

O IEEE 802.1s não especifica os detalhes de atribuição de um grupo de VLANs a uma ou mais MSTIs. No mecanismo baseado na QoS, é proposta uma política de atribuição baseada nas características do tráfego. Para tráfego *não multimédia*, uma ST para cada tipo de tráfego é partilhada entre todas as VLANs para evitar a explosão de mensagens de controlo. No caso do tráfego multimédia (por exemplo, tráfego de voz e vídeo), uma ST é construída para cada par de VLAN/tipo de tráfego, uma vez que tanto o *throughput* como o atraso *end-to-end* são factores igualmente importantes na qualidade de serviço. Neste novo mecanismo de QoS, quando uma VLAN é criada, o seu tipo de tráfego é também assinalado com a sua VLAN ID.

Neste trabalho, é utilizada a correspondência seguinte entre tipos de tráfego e valores de prioridade:

- 0 – Tráfego *best effort*;
- 1 – Tráfego *background*;
- 2 – *Spare*;
- 3 – Tráfego *excellent effort*;
- 4 – Tráfego *controlled load*;
- 5 – Tráfego *video*;
- 6 – Tráfego *voice*;
- 7 – Tráfego *network control*.

Para diferentes tipos de tráfego, as suas métricas de custo mais relevantes podem ser diferentes. Por exemplo, o tráfego 7 de mais alta prioridade precisa que os pacotes não sejam perdidos. Por outro lado, o tráfego 5 e 6 são tráfegos de vídeo e voz e necessitam de um atraso e de um jitter limitado. O tráfego do tipo 0 a 4 (tráfego *não multimédia*) tenta ao máximo evitar não ser servido. O protocolo *standard* usa apenas a velocidade do interface como métrica de custo do percurso para construir a ST. Esta métrica de custo é estática, não reflectindo correctamente o estado da rede, e a ST resultante é sempre a mesma a

menos que haja falha de ligações. Para ultrapassar esta limitação, é adoptado o uso de métricas adicionais baseadas nos tipos de tráfego bem como na velocidade das interfaces para adaptar o custo do percurso.

| Link Speed | Recommended Value |
|-----------------------|-------------------|
| $\leq 100\text{Kb/s}$ | 200000000 |
| 1Mb/s | 20000000 |
| 10Mb/s | 2000000 |
| 100Mb/s | 200000 |
| 1Gb/s | 20000 |
| 10Gb/s | 2000 |
| 100Gb/s | 200 |
| 1Tb/s | 20 |
| 10Tb/s | 2 |

Tabela 13 - Valores do parâmetro custo do percurso [15]

No mecanismo de construção de MSTs baseadas no QoS, inicialmente é considerado um custo para cada porta baseado na sua velocidade de transmissão de acordo com o que é feito pelo protocolo standard (Tabela 13). Seguidamente, o custo do percurso é adaptado pela métrica adicional de acordo com o tipo de tráfego a ser servido.

No início, a ST para o tráfego não multimédia é construída de acordo com a velocidade dos interfaces. Como referido anteriormente, o IEEE 802.1s MSTP considera uma IST e uma ou mais MSTIs. A ST, para o tráfego *best effort* no novo mecanismo baseado em QoS, desempenha um papel de IST no protocolo *standard*. Por outras palavras, a ST em questão faz a ligação com uma ST fora do campus, e as outras STs para o tráfego de maior prioridade estão relacionadas com as MSTIs.

No caso do tráfego multimédia, o atraso extremo-a-extremo é um dos factores de maior importância. De qualquer forma, a informação da fila é dinâmica e imprevisível, especialmente no que toca a filas de prioridade simples. Assim, é introduzida uma nova métrica, o número de VLANs activas, que indica o número actual de fluxos VLAN que estão a passar pelo *switch*. Uma vez que a ST abrange todos as *switches* da rede, e o tráfego actual pode fluir apenas por uma pequena parte da ST, são usadas entradas *up-to-date* na base de dados de filtragem para se saber quantas VLANs estão activas através de uma porta em particular. Usando esta informação, quando um *switch* recebe uma trama de controlo para a construção da ST para o par VLAN/tipo de tráfego, o *switch* calcula a largura de banda viável, $F_{bw}(tc)$, e a largura de banda efectiva, $E_{bw}(tc)$, baseada nas especificações de classe do tráfego *tc* no BPDU.

Para avaliar o desempenho deste novo mecanismo foi adoptado o simulador de rede *Qualnet 3.5* que suporta as normas do IEEE, logo, a construção de uma única ST e a criação e manutenção de VLANs. No entanto, este simulador não suporta MSTs pelo que foi implementado o IEEE 802.1s MSTP para MST e, de

seguida, os módulos MST baseados no QoS propostos. Foi comparado o desempenho entre estes mecanismos de construção de STs em termos de *throughput* e de atraso extremo-a-extremo.

A topologia simulada consiste numa rede com 16 *switches* e ligações de 10 Mbps. Para tráfegos padrão, foram gerados quatro tipos de tráfego: *best effort*, *video*, *voice* e *controlo de rede*. Uma vez que não existe nenhuma mistura ideal dos vários tipos de tráfego, variou-se o a distribuição destes quatro cenários de tipo de tráfego. São usados dois mecanismos assentes na QoS que constroem uma ST baseada no estado da rede. O primeiro mecanismo, QoS-MST, anula as STs que não modificam os seus caminhos de acordo com o estado actual da rede, reconstruindo-as. O segundo mecanismo, QoS-DYN-MST, no entanto, guarda uma tabela com o número de VLANs que atravessam as portas mesmo depois de as novas árvores serem construídas.

| 10 VLANs | | | | |
|--------------|-------------|-------------|---------|-------------|
| | Best Effort | | Video | |
| Dist. | QoS-MST | QoS-DYN-MST | QoS-MST | QoS-DYN-MST |
| 20 / 70 / 10 | 0.019 | 0.003 | 5100 | 335 |
| 30 / 60 / 10 | 1570 | 1570 | 941 | 941 |
| 45 / 45 / 10 | 0.01 | 0.01 | 12 | 12 |
| 20 VLANs | | | | |
| | Best Effort | | Video | |
| Dist. | QoS-MST | QoS-DYN-MST | QoS-MST | QoS-DYN-MST |
| 20 / 70 / 10 | -4400 | -64 | 5260 | 483 |
| 30 / 60 / 10 | -3260 | 322 | 1570 | 611 |
| 45 / 45 / 10 | 693 | 693 | 0.157 | 0.157 |
| 30 VLANs | | | | |
| | Best Effort | | Video | |
| Dist. | QoS-MST | QoS-DYN-MST | QoS-MST | QoS-DYN-MST |
| 20 / 70 / 10 | -2700 | -404 | 6900 | 554 |
| 30 / 60 / 10 | -3800 | -1800 | 1000 | 757 |
| 45 / 45 / 10 | 180 | 909 | 40 | -1000 |

Tabela 14 - Benefícios do *throughput* para cada tipo de tráfego (Kbps) [15]

A Tabela 14 mostra a melhoria apresentada pelas diferentes distribuições de tráfego para cada classe de tráfego quando comparada com o caso de implementação de uma única ST. É obvio que quer o MSTP, QoS-MSTP e QoS-DYN-MSTP têm um desempenho muito melhor do que o STP.

Os resultados apresentados são a diferença média do atraso e do *throughput* observados pelas várias classes de tráfego no que respeita ao cenário base MSTP.

| 10 VLANs | | | | | | |
|--------------|-------------|-------------|---------|-------------|---------|-------------|
| Dist. | Best Effort | | Video | | Voice | |
| | QoS-MST | QoS-DYN-MST | QoS-MST | QoS-DYN-MST | QoS-MST | QoS-DYN-MST |
| 20 / 70 / 10 | -0.02 | -0.07 | 22 | 11 | 28 | 23 |
| 30 / 60 / 10 | -0.03 | -0.03 | 17 | 17 | 0.61 | 0.61 |
| 45 / 45 / 10 | 0.38 | 0.38 | 0.33 | 0.33 | 0.39 | 0.39 |
| 20 VLANs | | | | | | |
| Dist. | Best Effort | | Video | | Voice | |
| | QoS-MST | QoS-DYN-MST | QoS-MST | QoS-DYN-MST | QoS-MST | QoS-DYN-MST |
| 20 / 70 / 10 | -60 | -5 | 23 | 11 | 75 | -1.7 |
| 30 / 60 / 10 | -172 | -27 | 11 | 0.2 | 29 | 12 |
| 45 / 45 / 10 | 36 | 36 | 0.01 | 0.01 | 9.6 | 9.6 |
| 30 VLANs | | | | | | |
| Dist. | Best Effort | | Video | | Voice | |
| | QoS-MST | QoS-DYN-MST | QoS-MST | QoS-DYN-MST | QoS-MST | QoS-DYN-MST |
| 20 / 70 / 10 | 135 | 142 | 58 | 15 | 126 | 72 |
| 30 / 60 / 10 | -430 | -107 | 4.4 | 1.5 | 15 | -34 |
| 45 / 45 / 10 | 52 | 77 | 0.26 | -2.0 | 91 | 53 |

Tabela 15 - Benefícios do atraso extremo-a-extremo para cada tipo de tráfego (ms) [15]

São usadas 10, 20 e 30 VLANs com o objectivo de saturar a rede completamente. A Tabela 14 e a Tabela 15 mostram o *throughput* e o atraso *end-to-end* para cada classe de tráfego no que respeita à distribuição de tráfego. Como o tráfego de controlo de rede é o tráfego de maior prioridade, nunca é sacrificado em relação a outros tráfegos e raramente é descartado, a menos que a ligação esteja congestionada por outros tráfegos de controlo de rede. Desta forma, o artigo foca-se no desempenho de rede do tráfego *best effort*, *video* e *voice*. Uma vez que o *throughput* do tráfego *voice* (VoIP) é pequeno e não é um factor decisivo, o *throughput* foi comparado entre o tráfego *best effort* e o tráfego *video*. A distribuição de tráfego indica a percentagem atribuída a cada tipo de tráfego. Logo, 20/70/10 implica 20% de todo o tráfego da rede é *best effort*, 70% é tráfego multimédia (*voice* e *video*) e 10% é tráfego de controlo de rede. Se a porção de tráfego multimédia for muito pequeno comparado com o tráfego *best effort*, este último não sofre *starvation* (i.e., não deixa de ser servido). Este é o motivo pelo qual é mantida uma quantidade de tráfego multimédia significativa para clarificar a diferença de desempenho.

As tabelas mostram claramente que o QoS-MSTP e o QoS-DYN-MSTP apresentam melhorias notáveis quando comparadas com o MSTP simples. Alguns resultados são bastante interessantes e validam o que os autores tinham em mente. Por exemplo, para 20 VLANs, com distribuição de tráfego de 20/70/10 e de 30/60/10, verifica-se que o tráfego *best effort* (BE) sofre muito mais no caso do QoS-MSTP do que no caso QoS-DYN-MSTP, enquanto o tráfego multimédia ganha muito mais no caso QoS-MSTP do que no caso QoS-DYN-MSTP. Isto deve-se ao facto de o QoS-MSTP fixar a ST pelo que o tráfego de classe 5 e 6 permanece com uma boa árvore de suporte, fazendo uso dela, deixando o tráfego BE, que vem de seguida, sem ser servido. Por outro lado, o QoS-DYN-MSTP monitoriza continuamente e altera a ST se houver necessidade de servir tráfego de classe 5 ou 6. Isto implica uma melhor partilha de recursos entre as várias classes. De facto, na distribuição 30/60/10 para 20 VLANs, pode-se verificar que quer o tráfego BE quer o

tráfego multimédia ganham comparativamente ao MSTP, enquanto para o QoS-MSTP, o tráfego multimédia ganha muito, mas o tráfego BE é sacrificado.

O artigo [16] aborda também a engenharia de tráfego em termos de QoS de uma forma dinâmica. O tradicional STP e MSTP são do tipo *topology-driven*, ou seja, as árvores são construídas apenas com base nas informações da rede. Em contraste com a abordagem *topology-driven*, o método de engenharia de tráfego proposto neste artigo tem a finalidade de construir as árvores e atribuir-lhe VLANs de uma forma *traffic driven* onde o objectivo é a optimização da atribuição de VLANs e a garantia por parte das árvores de requisitos de QoS e de utilização de rede. Depois de uma optimização, são atribuídos pesos (custos) às ligações da rede Ethernet.

Aqui é considerado o MSTP com apenas uma região em toda a rede de acesso. A atribuição de árvores e VLANs é baseada no destino dos pacotes: uma VLAN pertence a uma árvore se o destino dos pedidos, que formam essa VLAN, é o nó raiz da árvore. Se todas as VLANs tiverem os seus destinos a pertencer a uma mesma árvore, a atribuição das VLANs às árvores é do tipo *straightforward*; caso contrário, é necessário distribuir as VLANs pelas árvores. Quando se faz essa atribuição, quer a engenharia de tráfego, quer os requisitos de QoS são considerados.

Posto isto, são formulados 3 casos de estudo:

- *Optimized Spanning Tree Protocol (STP_{opt})*: é considerada apenas uma árvore na rede. No entanto, em vez de uma abordagem do tipo *topology driven*, é assumida uma abordagem do tipo *traffic driven*, na qual se assume que todos os pedidos de todas as classes de tráfego têm que ser transportados por essa árvore única.
- *Optimized MSTP com 1 árvore por raiz (MSTP_{opt1})*: é assumido que há múltiplas árvores e existe uma árvore para cada nó *edge*. A cada árvore é atribuída uma S-VLAN (conjunto de clientes com o mesmo tipo de tráfego/serviços) baseada no destino e na classe do tráfego dessa mesma S-VLAN. Note-se que pode haver múltiplos pedidos encaminhados através de diferentes árvores para diferentes raízes, possibilitando que o tráfego seja distribuído entre as ligações sem correr o risco de ocorrência de ciclos em qualquer das árvores.
- *Optimized MSTP com 2 árvores por raiz (MSTP_{opt2})*: é assumido que há duas (ou até mais) árvores para cada nó *edge*. A cada árvore são atribuídas S-VLANs para cada classe, mas as C-VLANs são atribuídas às S-VLANs de uma forma óptima. Isto permite que uma única ligação possa transportar ambas as árvores que mais tarde podem usar ligações diferentes de novo. Esta abordagem suporta melhor a engenharia de tráfego. Aqui não só se define o encaminhamento nas árvores, como simultaneamente se atribuem VLANs a essas árvores.

Para testar estes três métodos, são usadas as normas STP e MSTP para comparar resultados. A única restrição é o facto de a raiz das árvores terem que ser um dos nós *edge*.

A maior parte dos critérios seleccionados focam-se na qualidade das soluções dadas. Entre eles, o *throughput* disponível é o critério mais importante, uma vez que quanto mais pedidos poderem ser transmitidos ao mesmo tempo, mais pedidos podem ser servidos no âmbito global.

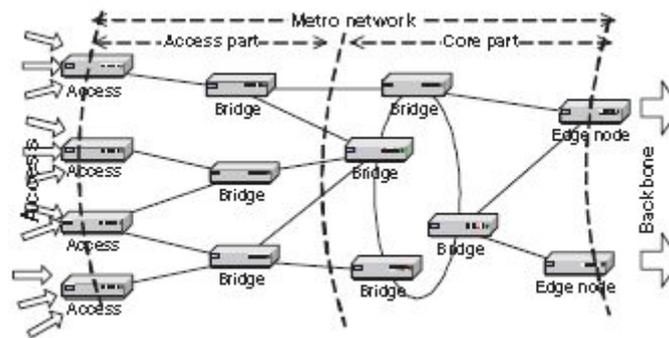


Figura 52 - Topologia de uma rede de acesso típica [16]

São usadas três redes diferentes: uma rede pequena (12 nós) e uma de tamanho médio (18 nós), ambas *dual-homed*, e uma rede em árvore como a representada na Figura 52. A capacidade das ligações na parte do núcleo é de 1 Gbps enquanto na parte de acesso é de 100 Mbps.

Para simplificar, são aqui usados apenas 4 classes como mostra a Tabela 16. É atribuído o maior peso ($w^1 = 4$ neste caso) à classe de tráfego de maior prioridade tendo, no entanto, o menor volume de tráfego, ou seja, taxa de recursos de ligações que lhe é atribuída é a menor ($\beta^1 = 10\%$ neste exemplo).

| Number | Name | Resource Ratio | Priority Weight |
|--------|-------------|------------------|-----------------|
| 1 | Platinum | $\beta^1 = 10\%$ | $w^1 = 4$ |
| 2 | Gold | $\beta^2 = 20\%$ | $w^2 = 3$ |
| 3 | Silver | $\beta^3 = 30\%$ | $w^3 = 2$ |
| 4 | Best Effort | $\beta^4 = 40\%$ | $w^4 = 1$ |

Tabela 16 - Classes de tráfego [16]

É atribuído o maior peso às classes de mais alta prioridade de tráfego embora estas tenham menor volume pelo que é-lhes também atribuído a mais baixa parcela de recursos de ligação.

| Methods | Total used capacity [Mbps] | Link utilis [%] | | Ave. path length per QoS class | | | |
|----------------------------|----------------------------|-----------------|------|--------------------------------|------|--------|------|
| | | Ave. | Max. | Platinum | Gold | Silver | BE |
| <i>STP</i> | 898.898 | 32 | 100 | 4.50 | 4.50 | 4.50 | 4.50 |
| <i>STP_{opt}</i> | 1997.572 | 54 | 100 | 5.00 | 5.00 | 5.00 | 5.00 |
| <i>MSTP</i> | 1198.455 | 44 | 100 | 3.00 | 3.00 | 3.00 | 3.00 |
| <i>MSTP_{opt1}</i> | 2397.862 | 73 | 100 | 3.00 | 3.00 | 3.00 | 3.00 |

Tabela 17 - Parâmetros de capacidade de uma topologia de tamanho médio [16]

A Tabela 17 mostra que se forem usadas mais STs, o comprimento do percurso diminui, uma vez que não há necessidade de desviar o tráfego através do nó raiz de uma única árvore. Além disso, o decréscimo da média do comprimento do percurso explica a razão dos pequenos aumentos de atribuição da capacidade resultarem num ganho em termos de *throughput*. Apesar de ser transmitido mais tráfego, este é encaminhado pelos caminhos mais curtos, logo, é atingida uma maior utilização das ligações (*average link utilization*).

As simulações feitas (12 por caso) indicaram que o *throughput* depende maioritariamente da topologia e do método de optimização de árvores utilizado. A vantagem do MSTP em relação ao STP é óbvia pois as duas árvores atribuídas aos dois nós *edge* podem usar percursos diferentes para transmitir o tráfego nas partes de acesso, o que aumenta o *throughput*.

A maior limitação da *topology-driven* STP e MSTP vêm ao de cima. Todos os nós de acesso numa parte de acesso, são ligados a uma única bridge interna que tem o menor valor de Bridge ID. Ao poder escolher entre duas ou mais VLANs (MSTP_{opt2}) um consequente ganho pode ser alcançado. De qualquer forma, os métodos *traffic driven* têm aproximadamente o mesmo desempenho que os métodos *topology driven* quando as partes de acesso têm topologias em árvore.

O artigo [17] volta a incidir na construção e manutenção de MSTs para uma distribuição de tráfego próxima do óptimo. Aqui, para que uma VLAN seja atribuída a cada ST e as tramas enviadas para uma árvore sejam identificadas pelo VLAN ID, as STs são construídas de tal forma que o tráfego é bem distribuído e não há ligações sobrecarregadas. As ligações têm que ser escolhidas de tal forma que possam lidar com os pedidos de tráfego gerado pelos nós terminais.

Uma solução simples é a utilização do algoritmo *Dijkstra*. Este algoritmo constrói uma ST para cada C-VLAN e depois actualiza o peso das ligações para construir o novo tráfego. Uma vez que um cliente não usa todos os APs, pode não ser obtido o melhor percurso entre VLANs e APs para o tráfego dado, mesmo que a ST tenha custo (peso) mínimo. Também a ordem pela qual as ligações são seleccionadas, onde o nó raiz é colocado e a ordem da construção da ST vão afectar a solução final. Posto isto, os autores propõe uma solução considerando todos estes requisitos que constrói STs. Apesar disso, esta abordagem pode não lidar com mudanças bruscas de carga na rede, mas a constante monitorização e uma reconfiguração simples melhoram o desempenho.

Aqui o peso é uma função da capacidade da ligação dada e do atraso: o peso de uma ligação é directamente proporcional ao atraso e inversamente proporcional à sua capacidade.

São então apresentados os resultados das simulações efectuadas ao algoritmo de construção de MSTs optimizado. O algoritmo é simulado para determinar o desempenho baseado nos percursos da ST, nos pedidos de tráfego e distribuição do mesmo. As entradas da simulação são: a topologia de rede em formato de matriz e os pedidos de tráfego para cada C-VLAN. Ambas são geradas aleatoriamente baseadas em funções de probabilidades. Em todas as topologias, a largura de bandas das ligações usada foi de 100 Mbps. O atraso de propagação é seleccionado aleatoriamente entre 0,2 e 1,0 milissegundos. A saída da

simulação é dada em formato de STs. Seguidamente é analisado o desempenho destas árvores comparando com árvores construídas usando o algoritmo de Dijkstra.

| C-VLANs | % of decrease in | |
|----------------|------------------|-------------|
| | Ag_W | Av_WL |
| 1 | 32 | 20 |
| 2 | 46 | 39 |
| 3 | 32 | 29 |
| 4 | 39 | 32 |
| 5 | 38 | 39 |
| Average | 37.4 | 31.8 |

Tabela 18 - Desempenho de uma rede de 22 nós [17]

A Tabela 18 mostra o desempenho de uma rede de 22 nós. Dois parâmetros são seleccionados para mostrar o desempenho: o peso agregado (*Ag_W*) da ST e o comprimento do peso médio (*Av_WL*) dos percursos dentro da árvore. Para calcular o comprimento do peso médio é encontrado primeiro o comprimento do peso de todos os percursos entre pares de APs para a C-VLAN em particular. A média destes valores dá o valor final de *Av_WL*. Estes parâmetros dão claramente a ideia do atraso envolvido e o tamanho do domínio *broadcast*. Estes parâmetros são também comparados com o uso dos protocolos standard. A rede resultante tem 13 pontos de acesso e é recolhida uma matriz de desempenho para 5 C-VLANs.

| # of nodes | # of access points | # of C-VLANs | Average % of decrease | |
|------------|--------------------|--------------|-----------------------|-------|
| | | | Ag_W | Av_WL |
| 10 | 6 | 4 | 24.5 | 21.5 |
| 15 | 10 | 6 | 32.8 | 27.2 |
| 20 | 12 | 8 | 30.3 | 24.4 |
| 25 | 14 | 10 | 32.1 | 26.9 |
| 30 | 17 | 13 | 34.4 | 27.9 |

Tabela 19 - Matriz de desempenho para 5 redes diferentes [17]

Se em vez da rede anterior, tivermos 5 redes com diferente número de nós, i.e., com uma topologia diferente, e com diferente número de C-VLANs, e utilizar a mesma simulação para todas, obtém-se uma matriz de desempenho diferente para cada rede, como se pode observar na Tabela 19. Os parâmetros são seleccionados pela média para cada tipo de rede, sendo que os valores de *Ag_W* e *Av_WL* são valores médios.

Os resultados das simulações mostram que a abordagem dos autores dá, de uma forma consistente, uma média de desempenho de 20% (i.e. a média entre *Ag_W* e *Av_WL*) ou mais. Em alguns

casos, o aumento do desempenho foi inferior a 15% para algumas C-VLANs, mas em geral o desempenho da rede foi acima dos 20%

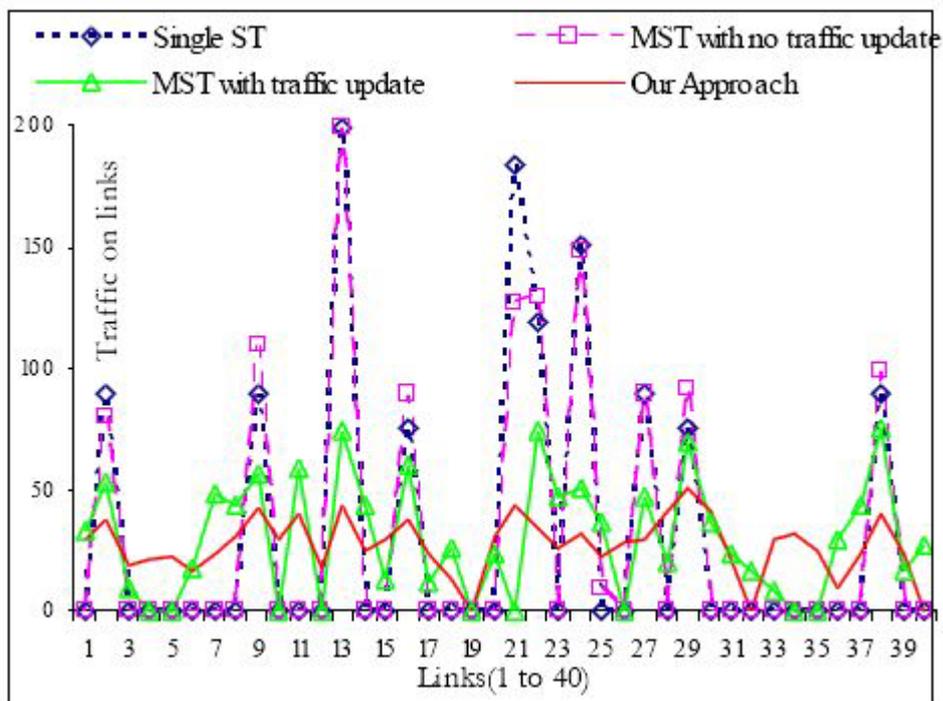


Figura 53 - Matriz desempenho para 5 redes diferentes [17]

A Figura 53 mostra a distribuição do tráfego numa rede com 12 nós ligados (40 ligações) com 100 Mbps de largura de banda baseada nas seguintes abordagens:

- Uma única ST para ligar todos os nós da rede. Esta ST é usada por todas as C-VLAN da rede.
- MSTs para as C-VLANs. No entanto, as capacidades das ligações não são actualizadas depois da construção das árvores, resultando em árvores bastante similares.
- MSTs são construídas e as capacidades das ligações são actualizadas depois de ter sido construídas para a C-VLAN. Isto é muito semelhante à abordagem deste artigo, excepto a procura do percurso mais curto e a atribuição na árvore que não se aplicam.
- A abordagem proposta pelo artigo.

Pelo gráfico da figura anterior, é claro que as técnicas 1 e 2 causam uma sobrecarga nas ligações e são usadas menos de 30% das ligações. Mesmo assim, são usadas MSTs na segunda abordagem, o resultado é próximo do uso de uma única ST. Uma vez que o tráfego não é reflectido num grafo depois de ser construída a ST, escolhendo diferentes nós raiz para cada C-VLAN não fará muita diferença para a estrutura das STs. No entanto a terceira técnica e a quarta abordagem são muito semelhantes. No algoritmo proposto, o tráfego máximo numa ligação é inferior a 50% da capacidade de ligação, e menos de 10% das

ligações não tiveram qualquer tráfego. Na terceira técnica, 25% das ligações não foram usadas e mais de 80% da capacidade de algumas ligações foram usadas.

3.5.2 Mobilidade

O crescimento rápido dos pedidos de serviços de comunicações pessoais com terminais móveis reforça a importância da implantação de tecnologia de redes de área metropolitana (*Metropolitan Area Network*, MAN) e a sua integração com sistemas *wireless* capazes de fornecer serviços de rede a milhões de utilizadores móveis. Estes pedidos de comunicações móveis não são uniformes em toda a rede pelo que a rede tem que ser flexível na sua configuração. Vários sistemas rádio podem integrar as MANs, tais como 3G, 4G e *wireless Local Area Networks* (LANs), fornecendo também serviços IP comuns beneficiando de cada acesso *wireless* ao mesmo tempo que suporta a mobilidade [19].

O suporte à mobilidade dos terminais é predominantemente endereçado em termos da camada IP. O mecanismo IP móvel (*mobile IP*) deve suportar a mobilidade terminal, mas precisa de uma pesada troca de mensagens, tais como *Binding Update* e *Return Routability*, para atribuir cuidadosamente um novo endereço e ligá-lo ao endereço residencial (*home address*) na transição do terminal móvel entre redes de acesso. Os pacotes IP passam pelos routers e deparam-se com fragmentações, esperas e pesquisas da *Route Table*, causando perda de pacotes e atrasos. Comparativamente, os routers são mais complexos do que os *switches* da camada 2.

A tecnologia Ethernet é candidata para a construção de uma MAN, tendo esta *wireless* LANs, devido à sua simplicidade e flexibilidade de manutenção de uma topologia de rede. A Ethernet móvel consiste na introdução de *Layer 2 Cache Learning* e mecanismos de *Handover* em *switches*. Contudo, a rede não garante a reconfiguração da rede de forma a atingir a flexibilidade necessária para suportar utilizadores móveis.

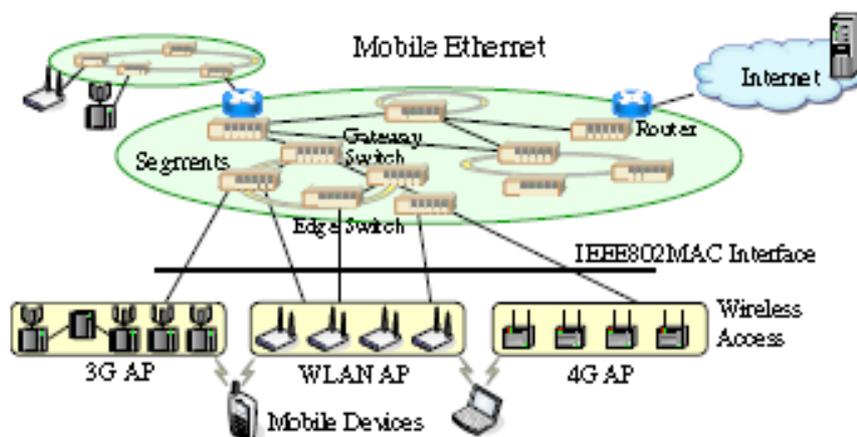


Figura 54 - Ethernet móvel e acesso wireless [19]

A Figura 54 descreve o modelo da Ethernet móvel, sendo uma das soluções que permite integrar variados acessos *wireless*. A Ethernet móvel é baseada na Ethernet WAN, onde cada mensagem é virtualmente transmitida na rede partilhada com endereços MAC e permite ligar vários tipos de sistemas de rádio, tais como 3G, 4G e *Wireless LANs*, seguindo uma interface MAC e autenticação da rede. Os terminais móveis escolhem, de entre as redes de acesso disponíveis, aquele que melhor se adequa baseado, por exemplo, na sua própria política e na informação disponibilizada pelas redes.

A Ethernet móvel organiza secções para obter uma boa escalabilidade da Ethernet. Desta forma, tem-se uma certa liberdade ao projectar uma estrutura de secções. As secções são ligadas através de redes de *Fast Core*. Para implementar a Ethernet móvel dentro da *Metropolitan Area* é imperial que haja escalabilidade nas redes do núcleo e nas secções. Isto deve-se ao facto de o Core (núcleo) ter que cobrir uma cidade e as suas áreas suburbanas, bem como cada secção tem que fazer a ligação entre centenas de pontos de acesso na área da cidade. Tecnologias com redes *Mesh* são adequadas para uma secção Ethernet uma vez que não possuem restrições topológicas. Contudo, não existe nenhum mecanismo de manutenção da ST numa rede *Mesh* de milhares de *switches* que satisfaça os requisitos para aplicações sensíveis ao tempo.

Uma rede Ethernet típica tem redundância nos percursos que lhe confere flexibilidade e fiabilidade. No entanto, geram-se ciclos criando uma sobrecarga na rede, como referido anteriormente. O STP resolve este problema mas o seu algoritmo demora algum tempo até convergir para uma topologia final em caso de falha ou alteração da rede. Como se sabe, o RSTP constrói uma árvore de suporte mais rapidamente. No artigo [19] é medido o tempo decorrido para completar a árvore de suporte em redes *Mesh* (com N nós) com a condição de que estas estão simultaneamente ligadas com parâmetros recomendados pelo RSTP.

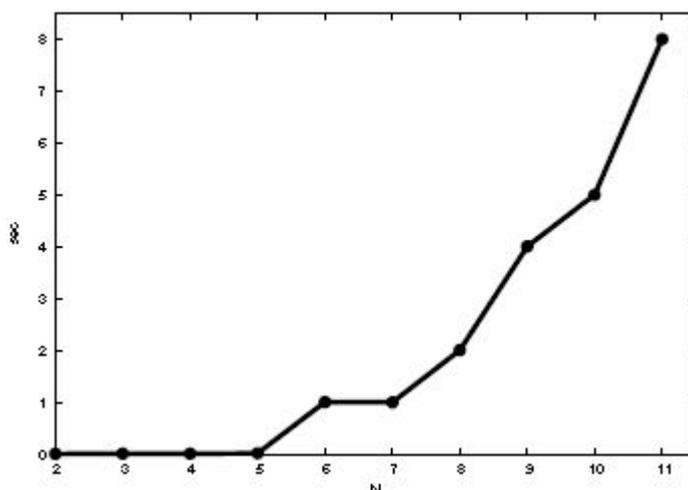


Figura 55 – Tempo de convergência RSTP numa rede *mesh* com N nós [19]

A Figura 55 mostra o que o tempo necessário para formar a árvore de suporte aumenta com o número de bridges sendo precisos 8 segundos para se formar uma rede Mesh com 11 nós. Isto acontece devido a alterações em partes específicas da topologia, o que implicam a reconstrução da árvore de suporte pelo RSTP. Isto faz com que um novo protocolo assente em STs escaláveis numa MAN de grande dimensão seja necessária.

Neste ponto, com o MSTP, já se sabe que são atribuídas várias VLANs a cada uma das MSTs, existindo menor volume de mensagens de controlo trocadas criando, desta forma, múltiplas regiões numa subnet IP.

Assume-se que uma rede MAN consiste em milhares de *switches* Ethernet com milhões de terminais móveis e que contém problemas inerentes ao uso do STP, relativamente à escalabilidade.

Os autores de [19] propõem um *Scalable Spanning Tree Protocol* (SSTP), que é uma extensão do MSTP, para organizar as regiões automaticamente com a limitação de um número de *switches* predefinido numa região. Propõem também uma redução do tráfego de sinalização e do tempo de convergência na construção da ST. O SSTP possui três funções:

1. Construir e manter automaticamente as regiões;
2. Construir e manter uma ST em cada região;
3. Construir e manter uma ST entre regiões.

O artigo refere que a função 1. foi desenvolvida e posteriormente integrada na 2. e na 3. usando o IEEE 802.1s MSTP.

São então definidos os parâmetros para o SSTP: *Region Size*, *Region ID* e *Prev Region ID*. O *Region Size* é o número de bridges que pertencem à mesma região; o *Region ID* é um valor único para cada região e é igual ao menor Bridge ID da região. Numa primeira fase, cada bridge tem um valor inicial de *Region Size* que é igual a 1, o *Region ID* é um valor aleatório que é atribuído pelas suas Bridge ID.

Posteriormente são criadas regiões fundindo regiões vizinhas. O SSTP agrupa duas regiões vizinhas quando a soma das *Region Size* das duas regiões não excede o máximo *Region Size* predefinido. A nova *Region ID* da região agrupada passa a ser a *Region ID* de menor valor entre as duas regiões originais. Como resultado da repetição desta sequência de acontecimentos, regiões cuja *Region Size* é menor do que o máximo *Region Size* são agrupadas.

Este protocolo usa mensagens SSTP para comunicar. Essas mensagens consistem em campos de informação que são: *type*, *Bridge ID*, *Region Size*, *Region ID* e *Prev Region ID*. Há dois tipos de mensagens *Hello* ou *Change* (cada tipo identificado pelo campo *type*). O campo *Bridge ID* indica o *Bridge ID* da bridge que envia a mensagem, enquanto o campo *Region Size* dá a informação do número de bridges existentes nessa região. O *Region ID* e o *Prev Region ID* indicam as *Region IDs* que a bridge tem agora e que tinha antes, respectivamente.

A mensagem *Hello* é uma mensagem do tipo *heart-beat* para perceber se duas regiões devem ser agrupadas. A mensagem *Change* notifica o agrupamento das duas regiões a todas as bridges envolvidas no processo.

Periodicamente as bridges trocam mensagens *Hello* para todas as bridges às quais estão ligadas num dado intervalo de tempo. Uma bridge que recebe uma mensagem *Hello* verifica se a sua *Region ID* é menor que a sua e se a soma das *Region Size* da mensagem e da bridge é igual ou inferior ao valor máximo do *Region Size*. Se for menor, o campo *Region ID* da bridge passa a ter o valor do da mensagem e o campo *Region Size* é incrementado do valor da mensagem. O *Region ID* é guardado como *Prev Region ID*. De seguida, a mensagem *Change* é enviada para todas as bridges que estão ligadas com a bridge em questão.

Quando uma bridge recebe uma mensagem *Change*, esta verifica se o valor do *Prev Region ID* da mensagem é igual ao seu Bridge ID. Se for, o *Bridge ID* da bridge é alterado para o valor do mesmo campo da mensagem e mais uma vez o valor do anterior passa guardado em *Prev Region ID* sendo enviada uma nova mensagem *Change* pela bridge para todas as bridges a que está ligada. Por outro lado, se o *Region ID* da mensagem for igual ao da bridge, o campo *Region Size* da bridge passa a ser igual ao da mensagem. Seguidamente é enviada a todas as bridges que estão ligadas à bridge em questão.

O artigo [19] refere também a necessidade de fazer com que o parâmetro *Region Size*, em cada bridge, se mantenha coerente quando uma bridge falha ou uma ligação é desactivada. Por exemplo, quando uma bridge falha depois das regiões terem sido criadas, as bridges cujo *Region Size* tem um valor superior ao da bridge que falhou, precisam de mudar o valor desse parâmetro uma vez que não representa a situação no momento. Assim, uma bridge deve verificar periodicamente a actividade das bridges ligadas a si que têm um valor inferior de *Region Size* comparado ao seu. Se nenhuma mensagem *Hello* for recebida de uma bridge, durante um período de tempo pré-definido, assume-se que essa bridge falhou. Esse tempo deve ser superior ao intervalo de tempo em que as mensagens *Hello* são enviadas.

Se o parâmetro *Region Size* ultrapassar o limite máximo, a bridge limpa os seus parâmetros e passa a guardar os que possuía inicialmente.

Quando uma bridge detecta uma mudança nos parâmetros SSTP, eles reflectem-se nos parâmetros IEEE 802.1s MSTP. O nome da configuração é atribuído através de uma mistura do *Region ID* da bridge com uma chave comum à rede. Posteriormente, a bridge envia um 802.1s BPDU de configuração para todas as bridges a que está ligada.

Para avaliar o desempenho e a eficiência do SSTP, os autores adoptaram um simulador, *MIRAI Simulation Framework* (MIRAI-SF).

Na simulação, são definidos $N \times N$ redes *Mesh*, constituídos por bridges que contém um valor aleatório como *Bridge ID*. O intervalo definido para o envio de mensagens *Hello* é de 1 segundo e o atraso de comunicação entre duas bridges é de 100 microssegundos.

A simulação foi feita com diferentes tamanhos de redes *Mesh* e de *Region Size* assumindo que todas as bridges são activadas ao mesmo tempo, o que constitui o pior cenário para o protocolo dado que vários agrupamentos vão acontecer em simultâneo.

Este artigo mostra a criação de regiões através do SSTP em 50 x 50 redes *Mesh* com o parâmetro *Region Size* a 10. O número de bridges em algumas regiões é superior ao valor do *Region Size*. Isto acontece porque o agrupamento de regiões é feito de forma independente. Existe a possibilidade de algumas regiões se tornarem bastante grandes, mas este é o compromisso que é feito para se poder ter a simplicidade deste protocolo e continuar a ser útil para o objectivo final, mesmo que o *Region Size* não represente exactamente o número de bridges da região.

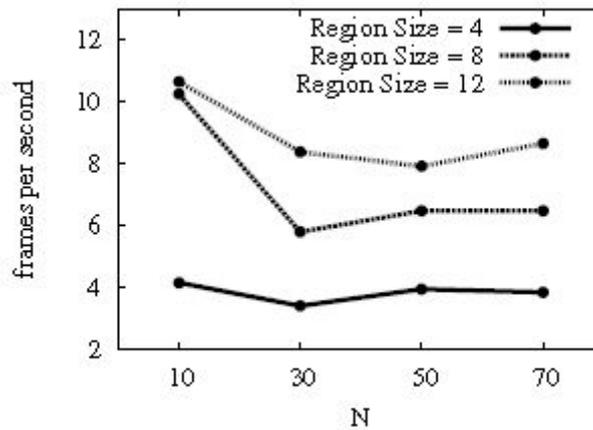


Figura 56 - Tramas SSTP para a configuração de redes [19]

A Figura 56 apresenta o número médio de tramas por bridge para uma configuração automática de uma região, excluindo as mensagens MSTP. O valor máximo é por volta das 11 tramas por segundo, o que é insignificante para uma rede de alta velocidade como a *Gigabit Ethernet*, uma vez que o tamanho das mensagens SSTP é menor do que 1kB. Assim, a largura de banda consumida pelo SSTP é no máximo 10kB/s. O tempo necessário para a construção de regiões neste caso é cerca de 1 a 2 segundos em qualquer dos tamanhos da rede *Mesh*, que comparativamente com o caso RSTP mostrado na Figura 55 é considerada pequena.

De seguida é analisado o caso em que uma bridge seleccionada aleatoriamente da rede é activada quando todas as outras bridges já formaram as respectivas regiões e a ST.

| maximum Region Size | 4 | 8 | 16 |
|---------------------|-----|-----|-----|
| $0 \leq T < 1$ | 89% | 95% | 95% |
| $1 \leq T < 2$ | 5% | 5% | 5% |
| $2 \leq T < 3$ | 3% | 0% | 0% |
| $3 \leq T < 4$ | 3% | 0% | 0% |

Tabela 20 - Tempo necessário para reconstruir a ST na adição de uma nova bridge [19]

A Tabela 20 mostra o tempo decorrido, em segundos, que é necessário para reconstruir a ST depois de se correr 100 vezes a simulação com vários parâmetros *Region Size*. Como se pode verificar, a ST é reconstruída na maior parte das vezes dentro de 1 segundo, que é mais rápido em relação ao RSTP para o mesmo tamanho de rede *Mesh*.

Para um gestor de rede é um fardo ter de configurar cada bridge com parâmetros de regiões MSTP numa rede MAN que contém várias bridges. O SSTP resolve o problema uma vez que os parâmetros MSTP são configurados automaticamente apenas substituindo as bridges existentes por bridges SSTP. Este protocolo funciona completamente independente do mecanismo de *Forwarding*, logo é compatível com IEEE 802.1s MSTP.

Apesar de o SSTP diminuir o tempo necessário para reconstruir uma ST, ligar todos os nós de uma cidade do tamanho de uma MAN com *Ethernet* é ainda limitado pelos serviços em tempo real, tais como tráfego de voz e vídeo. De qualquer forma os autores acreditam que a combinação de tecnologias *backbone* com o SSTP resulta numa MAN ideal.

É também importante evitar a concentração de tráfego na raiz da ST que é causado pela comunicação entre nós em lados diferentes da raiz. Note-se que o SSTP pode ser conjugado com outras propostas para o balanceamento de carga e QoS na *Ethernet* usando o MSTP [15].

CAPÍTULO 4 – CONCLUSÕES

A Ethernet cresceu desde as suas origens em ambientes domésticos e empresariais até abranger outro tipo de redes como as redes de acesso e redes metropolitanas (MANs). No entanto, a implementação da Ethernet em redes MAN requer actualizações consideráveis.

Nos artigos analisados foram endereçadas e discutidas inúmeras questões que se levantam quando é considerada a engenharia de tráfego de redes metropolitanas Ethernet. Foram discutidos vários esquemas e algoritmos eficientes para proporcionar aos clientes VLANs próprias de redes metro Ethernet. Endereçaram-se questões relevantes, tais como, o tamanho do VLAN ID, o balanceamento de carga, a minimização da largura de banda e a sobrevivência da rede. No primeiro artigo [7] analisado foi proposto um esquema de agrupamento que estende o espaço das etiquetas no domínio do operador e permite proporcionar um grande número de VLANs de forma eficiente. Posto isto, as questões de balanceamento de carga, MSTs e interacção entre grupos e fornecimento de largura de banda são discutidas e propostas soluções. Finalmente, é apresentada uma técnica que permite ao operador atribuir VLANs de acordo com os seus requisitos de sobrevivência. Os resultados das simulações mostraram que o alargamento do tamanho da etiqueta VID pode ser atingido usando heurísticas simples, o balanceamento da distribuição do tráfego pode ser obtido usando uma ferramenta eficiente e rápida e, a sobrevivência baseada na qualidade de serviço pode ser proporcionada aos clientes usando algoritmos de construção/atribuição de MSTs inteligentes.

As questões de construção de MSTs foram abordadas com cenários dinâmicos e estáticos. Nos cenários dinâmicos foram comparadas diversas métricas de custo de ligação propostas. As modificações introduzidas afectam sobretudo a forma de definir a métrica de custo de ligação, ao contrário das métricas de custo fixo, que são dadas pelo inverso da capacidade da ligação. São propostas funções de custo de ligação que dependem da largura de banda disponível. Foram analisados os desempenhos dos diversos algoritmos para várias topologias de rede assentes em diferentes cenários de tráfego. Os resultados demonstram que os algoritmos que usam uma métrica de custo de ligação dinâmica têm um melhor desempenho em termos de probabilidade de bloqueio, eficiência no uso dos recursos da rede e são também mais flexíveis.

Foram também endereçados problemas de como usar o MSTP para melhorar o balanceamento de carga e o *service disruption* nas redes Ethernet, com cenários estáticos. Aqui foram propostos algoritmos de atribuição de parâmetros MSTP que implementam um conjunto qualquer de STs que minimizam o número de ligações mudando os seus estados entre activo e *backup* quando ocorre uma única falha de ligação. Foi também proposta uma optimização do algoritmo que determina um conjunto de STs que optimiza o balanceamento de carga e o *service disruption*. Os resultados computacionais mostraram que a combinação dos dois algoritmos é bastante eficiente e consegue obter boas soluções com tempos de computação muito pequenos. O algoritmo de atribuição de parâmetros MSTP é um importante resultado por si só uma vez que

permite a determinação dos parâmetros MSTP apropriados para outro conjunto de STs que podem ser calculadas por outro método qualquer, possivelmente endereçando outros objectivos de optimização. Foi estudada a relação entre o balanceamento de carga e o *service disruption* assentes numa única falha de rede quando é adoptada uma ou mais regiões. Foi também mostrado que a abordagem com uma única região é sempre melhor quando o MSTP é configurado de forma óptima e que um pequeno número de STs adicionais podem obter resultados muito próximos do óptimo.

Introduziu-se também a problemática da divisão de regiões de uma rede. Foi então proposto um algoritmo de construção de regiões MST que proporciona engenharia de tráfego básica da Ethernet num domínio empresarial. O método proposto suporta a natureza dinâmica das VLANs onde nós VLAN podem ser adicionados, movidos ou removidos sem grande esforço. Este método proporciona também recursos suficientes na região para suportar protecção contra, pelo menos, falhas únicas. Ao longo das simulações, concluiu-se que esta abordagem pode ser usada para aumentar o desempenho do MSTP dividindo a rede em regiões com base nas VLANs. Este algoritmo é facilmente extensível de forma a basear-se em prioridades de tráfego, departamentos, acesso privilegiado e em requisitos de segurança.

Apontaram-se limitações significativas básicas do STP e do MSTP actualmente definidos pelas normas IEEE no que toca à qualidade de serviço. O MSTP apenas atinge o balanceamento de carga geográfico e não tem em consideração diversas características do tráfego que transporta usando técnicas de filas de prioridade simples. São apresentadas melhorias simples e altamente eficazes ao MSTP para atingir um elevado grau de QoS. Isto é conseguido mantendo em perspectiva as diferentes características dos vários tipos de tráfego. Os detalhes dos esquemas propostos tornam claro que é bastante viável e podem ser postos em prática com um esforço mínimo.

Tendo em vista a mobilidade, foi discutida a implementação da tecnologia Ethernet assente nas redes MAN, e as questões inerentes de escalabilidade. Foi proposto o *Scalable Spanning Tree Protocol* (SSTP), que trabalha com o MSTP, que configura os seus parâmetros automaticamente mantendo a compatibilidade com os protocolos existentes nos *switches*. Os resultados das simulações mostraram que o SSTP é rápido na reconstrução das STs comparando com o existente STP e tem um *overhead* pequeno na configuração das regiões.

REFERÊNCIAS

- [1] IEEE Standard 802.1D, "Media Access Control (MAC) Bridges", 1998
- [2] IEEE Standard 802.1Q, "Virtual Bridged Local Area Networks", 1998
- [3] IEEE Standard 802.1w, "Media Access Control (MAC) Bridges-Amendment 2: Rapid Reconfiguration", 2001
- [4] IEEE Standard 802.1s, "Virtual Bridged Local Area Networks-Amendment 3: Multiple Spanning Trees", 2002
- [5] Cisco Systems, "Understanding Rapid Spanning Tree Protocol (802.1w)", 2007-2008
- [6] Cisco Systems, "Understanding Multiple Spanning Tree Protocol (802.1s)", 2006-2007
- [7] Ali, M., Chiruvolu, G., Ge, A.: Traffic Engineering in Metro Ethernet. IEEE Network Vol. 19 No. 2 (2005) 10-17
- [8] Kolarov, A., Sengupta. B., Iwata, A.: Design Of Multiple Reverse Spanning Trees in Generation os Ethernet-VPNs. IEEE GLOBECOM'04, Dallas, USA Vol. 3 (2004) 1390-1395
- [9] Sharma, S., Gopalan, K., Nanda, S., Chiueh, T.: Viking: A Multi-Spanning-Tree Ethernet Architecture for Metropolitan Area and Cluster Networks. IEEE INFOCOM'04, Hong Kong, Vol. 4 (2004) 2283-2294
- [10] Chen, W., Jin, D., Zeng, L.: Design of Multiple Spanning Trees for Traffic Engineering in Metro Ethernet. International Conference on Communication Technology (ICCT'06), Tsinghua Univ., Beijing, (2006) 1-4
- [11] de Sousa, A.: Improving Load Balance and Resilience of Ethernet Carrier Networks with IEEE 802.1s Multiple Spanning Tree Protocol. 5th Int. Conference on Networking (ICN'06), Mauritius Islands (2006)
- [12] de Sousa, A., Soares, G.: Improving Load Balance of Ethernet Carrier Networks using IEEE 802.1s MSTP with Multiple Regions. 5th Int. IFIP-TC6 Networking Conference, LNCS, Vol. 3976, Springer (2006) 1250-1260
- [13] de Sousa, A., Soares, G.: Improving Load Balance and Minimizing Service Disruption on Ethernet Networks with IEEE 802.1s MSTP. Proc Workshop on IP QoS and Traffic Control, Lisbon, Portugal, Vol.1 (2007) 25-35
- [14] Padmaraj, M., Nair, S., Marchetti, M., Chiruvolu, G., Ali, M.: Traffic Engineering in Enterprise Ethernet with Multiple Spanning Tree Regions. Proc of System Communications (ICW'05), Montreal, Canada (2005) 261-266
- [15] Lim, Y., Yu, H., Das, S., Lee, S.-S., Gerla, M.: QoS-aware Multiple Spanning Tree Mechanism over a Bridge LAN Environment. IEEE GLOBECOM'03, San Francisco, USA Vol. 6 (2003) 3068-3072
- [16] Cinkler, T., Moldován, I., Kern, A., Lukovszki, C., Saltai, G.: Optimizing QoS Aware Ethernet Spanning Trees. 1st Int. Multimedia Services Access Networks (MSAN'05), (2005).

- [17] Padmaraj, M., Nair, S., Marchetti, M.: Metro Ethernet Traffic Engineering Based on Optimal Multiple Spanning Trees. 2nd IFIP Int. Conference on Wireless and Optical Communications Networks, Dallas, USA (2005)
- [18] Kern, A., Moldován, I., Cinkler, T.: Scalable Tree Optimization for QoS Ethernet. 11th IEEE Symposium on Computers and Communications (ISCC'06), Cagliari, Italy (2006)
- [19] Ishizu, K., Kuroda, M., Kamura, K.: SSTP: na 802.1s Extension to Support Scalable Spanning Tree for Mobile Metropolitan Area Network. IEEE GLOBECOM'04, Dallas, USA Vol. 3 (2004) 1500-1504

