

MINISTÉRIO DA DEFESA
EXÉRCITO BRASILEIRO
DEPARTAMENTO DE CIÊNCIA E TECNOLOGIA
INSTITUTO MILITAR DE ENGENHARIA
CURSO DE MESTRADO EM SISTEMAS E COMPUTAÇÃO

FABIO HENRIQUE SILVA

AUTO-BALANCEAMENTO DE CARGA EM REDES DE
COMPUTADORES UTILIZANDO PROTOCOLOS DE
ROTEAMENTO DO TIPO ESTADO DE ENLACE

Rio de Janeiro
2010

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

INSTITUTO MILITAR DE ENGENHARIA

FABIO HENRIQUE SILVA

**AUTO-BALANCEAMENTO DE CARGA EM REDES DE
COMPUTADORES UTILIZANDO PROTOCOLOS DE ROTEAMENTO
DO TIPO ESTADO DE ENLACE**

Dissertação de Mestrado apresentada ao Curso de Mestrado em Sistemas e Computação do Instituto Militar de Engenharia, como requisito parcial para obtenção do título de Mestre em Sistemas e Computação.

Orientador: Prof. Ronaldo Moreira Salles - Ph.D.

Rio de Janeiro
2010

c2010

INSTITUTO MILITAR DE ENGENHARIA
Praça General Tibúrcio, 80-Praia Vermelha
Rio de Janeiro-RJ CEP 22290-270

Este exemplar é de propriedade do Instituto Militar de Engenharia, que poderá incluí-lo em base de dados, armazenar em computador, microfilmear ou adotar qualquer forma de arquivamento.

É permitida a menção, reprodução parcial ou integral e a transmissão entre bibliotecas deste trabalho, sem modificação de seu texto, em qualquer meio que esteja ou venha a ser fixado, para pesquisa acadêmica, comentários e citações, desde que sem finalidade comercial e que seja feita a referência bibliográfica completa.

Os conceitos expressos neste trabalho são de responsabilidade do autor e do orientador.

S586 Silva, Fabio Henrique
Auto-balanceamento de Carga em Redes de Computadores utilizando Protocolos de Roteamento do tipo Estado de Enlace/ Fabio Henrique Silva.
– Rio de Janeiro: Instituto Militar de Engenharia, 2010.
97 p.: il., graf., tab.

Dissertação (mestrado) – Instituto Militar de Engenharia – Rio de Janeiro, 2010.

1. Redes de Computadores. 2. Protocolos de Roteamento. I. Título. II. Instituto Militar de Engenharia.

CDD 004.6

INSTITUTO MILITAR DE ENGENHARIA

FABIO HENRIQUE SILVA

**AUTO-BALANCEAMENTO DE CARGA EM REDES DE
COMPUTADORES UTILIZANDO PROTOCOLOS DE ROTEAMENTO
DO TIPO ESTADO DE ENLACE**

Dissertação de Mestrado apresentada ao Curso de Mestrado em Sistemas e Computação do Instituto Militar de Engenharia, como requisito parcial para obtenção do título de Mestre em Sistemas e Computação.

Orientador: Prof. Ronaldo Moreira Salles - Ph.D.

Aprovada em 16 de agosto de 2010 pela seguinte Banca Examinadora:

Prof. Ronaldo Moreira Salles - Ph.D. do IME - Presidente

Prof. Raquel Coelho Gomes Pinto - D.Sc, do IME

Prof. Sidney Cunha de Lucena - D.Sc, da UNIRIO

Rio de Janeiro
2010

AGRADECIMENTOS

Aos amigos que deram força e contribuíram para a construção da conquista de realizar o curso de mestrado.

A todos que durante a realização do curso ajudaram para que o sucesso fosse alcançado, ao professor Ronaldo Salles pela ajuda, paciência e confiança depositadas, aos professores Sérgio Cardoso, David Moura e Péricles Vianna pelo apoio.

Aos meus pais e amigos que torceram com ações e pensamentos positivos.

A Deus, pela vida, por tudo.

“A alegria que se tem em pensar e aprender faz-nos pensar e aprender ainda mais.”

Aristóteles

SUMÁRIO

LISTA DE ILUSTRAÇÕES	8
LISTA DE TABELAS	10
LISTA DE ABREVIATURAS	11
1 INTRODUÇÃO	14
1.1 Motivação e Problema	14
1.2 Objetivo da Dissertação	16
1.3 Organização da Dissertação	19
2 CONCEITOS PRELIMINARES	20
2.1 Protocolos de Roteamento	20
2.2 Gerenciamento de rede	22
2.3 Contexto Atual da Rede	25
2.4 Engenharia de Tráfego	25
2.5 Balanceamento de Carga	27
3 TRABALHOS RELACIONADOS	30
3.1 Aplicação do protocolo OSPF em Engenharia de Tráfego	30
3.2 Sistemas Autônomos Auto-Balanceados	33
4 ARQUITETURA PROPOSTA	39
4.1 Taxonomia do Sistema proposto na Engenharia de Tráfego	39
4.2 Contexto do Sistema	41
4.3 Implementação do sistema	42
4.4 Módulos de Atuação	44
4.4.1 Módulo de Análise de Redes	45
4.4.2 Módulo Tomador de Decisão	46
4.4.3 Módulo Mudança de Parâmetros	48
4.5 Heurísticas adotadas	48
5 RESULTADOS	53
5.1 Ambiente de Experimento	53

5.2	Experimento 1 e Validação da proposta	56
5.2.1	Análise teórica	56
5.2.2	Resultados da Análise Teórica	62
5.2.3	Análise prática	67
5.2.4	Resultados da Análise Prática	75
5.3	Experimento 2	80
5.4	Experimento 3	85
6	CONCLUSÃO	91
6.1	Considerações finais	91
6.2	Trabalhos futuros	93
7	REFERÊNCIAS BIBLIOGRÁFICAS	95

LISTA DE ILUSTRAÇÕES

FIG.2.1	Árvore MIB-2	23
FIG.3.1	Custo atribuído aos enlaces	32
FIG.3.2	Ciclo de Trabalho do SAAB - Esquema original	34
FIG.3.3	Topologia de rede	36
FIG.3.4	Modelagem do Peso do Enlace E como uma Barreira	37
FIG.4.1	Ciclo de Trabalho do SAAB - Esquema proposto	42
FIG.4.2	Sistema em funcionamento	43
FIG.4.3	Módulos de atuação	44
FIG.5.1	Topologia de rede dos experimentos	54
FIG.5.2	Topologia de teste para a validação da solução proposta	57
FIG.5.3	Cenário de teste para a validação da solução proposta	59
FIG.5.4	Modelagem do Peso do Enlace E como uma Barreira	61
FIG.5.5	Detecção de sobrecarga e execução do algoritmo	70
FIG.5.6	Modelagem do Peso do Enlace E como uma Barreira na análise prática	72
FIG.5.7	Comparação da taxa de ocupação do enlace afetado com as taxas de ocupação dos enlaces da nova rota escolhida para a demanda modelada	73
FIG.5.8	Descarte de pacotes na saída da interface ocorrido durante o exper- imento 1 na heurística Cisco	77
FIG.5.9	Descarte de pacotes nas saídas das interfaces ocorrido durante o experimento 1 na heurística Cisco modificado	77
FIG.5.10	Descarte de pacotes na saída da interface ocorrido durante o exper- imento 1 na heurística SAAB	77
FIG.5.11	Gráfico do desvio padrão entre as heurísticas no experimento 1 re- tirados os enlaces não participantes	79
FIG.5.12	Topologia de rede do experimento 2	81
FIG.5.13	Descarte de pacotes na saída da interface ocorrido durante o exper- imento 2 na heurística Cisco	83

FIG.5.14	Descarte de pacotes na saída da interface ocorrido durante o experimento 2 na heurística Cisco modificado	83
FIG.5.15	Descarte de pacotes na saída da interface ocorrido durante o experimento 2 na heurística SAAB	83
FIG.5.16	Gráfico do desvio padrão entre as heurísticas no experimento 2 retirados os enlaces não participantes	84
FIG.5.17	Topologia de rede do experimento 3	86
FIG.5.18	Gráfico do desvio padrão entre as heurísticas no experimento 3 incluídos todos os enlaces	88

LISTA DE TABELAS

TAB.3.1	Diferença entre os pesos dos caminhos Γ_{notE} e Γ_E	36
TAB.5.1	Lista de eventos do teste de validação	56
TAB.5.2	Comparação da taxa de ocupação dos enlaces do caminho a ser seguido	61
TAB.5.3	Tempos de congestionamento nos enlaces após o experimento 1	76
TAB.5.4	Média dos desvios padrão no experimento de validação retirados os enlaces não participantes	79
TAB.5.5	Lista de eventos do experimento 2	80
TAB.5.6	Tempos de congestionamento nos enlaces após o experimento 2	82
TAB.5.7	Média dos desvios padrão no experimento 2 retirados os enlaces não participantes	84
TAB.5.8	Lista de eventos do experimento 3	87
TAB.5.9	Tempos de congestionamento nos enlaces após o experimento 3	88
TAB.5.10	Média dos desvios padrão no experimento 3 incluídos todos os enlaces	89

LISTA DE ABREVIATURAS

ABREVIATURAS

AS	-	<i>Autonomous System</i>
IGP	-	<i>Interior Gateway Protocol</i>
IP	-	<i>Internet Protocol</i>
LS	-	<i>Link State</i>
MIB	-	<i>Management Information Base</i>
OSPF	-	<i>Open Shortest Path First</i>
SAAB	-	<i>Sistema Autônomo Auto-Balanceado</i>
SNMP	-	<i>Simple Network Management Protocol</i>
TCP	-	<i>Transmission Control Protocol</i>
TE	-	<i>Traffic Engineering</i>
UDP	-	<i>User Datagram Protocol</i>

RESUMO

O desempenho da rede é visto pelos usuários finais de serviços de rede como o ponto mais importante a ser tratado. A aplicação de conceitos de Engenharia de Tráfego ajuda a melhorar as propriedades emergentes da rede visíveis aos usuários finais. Um desses conceitos é denominado Balanceamento de Carga, que visa minimizar o congestionamento e otimizar a alocação eficiente de recursos. Alguns estudos e tecnologias buscam a otimização da rede através da alteração dos caminhos por onde os pacotes de dados trafegam. Uma das propostas existentes inclui técnicas de implementação do Auto-balanceamento de carga em um Sistema Autônomo que utilize o protocolo de estado de enlace OSPF. O presente trabalho propõe a construção de um Sistema Autônomo Auto-Balanceado a se realizar através da construção de uma ferramenta que analisa os fluxos de dados nos enlaces, e ao constatar a ocorrência de sobrecarga, aciona um algoritmo baseado em heurísticas de cálculo de alteração do custo do enlace modelado. Resultados experimentais realizados em testes de laboratório mostram a viabilidade e o desempenho da proposta.

ABSTRACT

The network performance is seen by end users of network services as the most important point to be treated. Applying concepts of traffic engineering helps to improve the emergent properties of the network visible to end users. One of these concepts is called Load Balancing, which seeks to minimize congestion and optimize the efficient resources allocation. Some studies and technology seek to optimize the network by changing the paths through which data packets travel. One existing proposal includes techniques for implementing the Self-load balancing in an autonomous system that uses link state protocol OSPF. This work proposes the construction of Self-Balanced Autonomous System to be held by building a tool that analyzes the data flows in the links and, to establish the occurrence of overload, triggers an algorithm based on the calculation of changes in modeled link cost. Experimental results achieved in laboratory tests show the feasibility and performance of the proposal.

1 INTRODUÇÃO

1.1 MOTIVAÇÃO E PROBLEMA

A Internet evoluiu rapidamente para uma infra-estrutura de comunicações muito crítica, suportando atividades econômicas, educacionais e sociais significativas. Simultaneamente, a prestação de serviços de comunicações via Internet tem se tornado muito competitiva, e os usuários finais estão exigindo um serviço de qualidade muito alto de seus prestadores de serviços. Por conseguinte, a otimização de desempenho de redes IP de grande escala, especialmente *backbones* públicos da internet, tornou-se um problema importante. Requisitos de desempenho de rede são multidimensionais, complexos e por vezes contraditórios, tornando o problema de engenharia de tráfego muito desafiador.

Redes IP variam em tamanho, desde pequenos grupos de roteadores situados dentro de um determinado local, até milhares de roteadores interconectados, *switches* e outros componentes distribuídos por todo o mundo. Os objetivos de otimização da engenharia de tráfego na Internet pode mudar com o tempo, com a imposição de novas exigências, o surgimento de novas tecnologias, ou novas idéias que surgem sobre os problemas subjacentes. Além disso, as redes podem ter objetivos de otimização diferentes, dependendo do seu modelo de negócios, capacidades e limitações operacionais. Os aspectos de otimização da engenharia de tráfego ultimamente se interessam com o controle de toda a rede, independentemente dos objetivos específicos de otimização em qualquer ambiente particular (AWDUCHE, 2002).

A otimização do desempenho de redes operacionais é fundamentalmente um problema de controle. No modelo de processo de engenharia de tráfego, o engenheiro de tráfego, ou um autômato adequado, age como o controlador em um sistema de controle adaptativo de *feedback*. Este sistema inclui um conjunto de elementos de rede interligada, um sistema de monitoramento de desempenho de rede, e um conjunto de ferramentas de gerenciamento de configuração de rede. O engenheiro de tráfego formula uma política de controle, observa o estado da rede através do sistema de monitoramento, caracteriza o tráfego, e aplica as ações de controle para conduzir a rede a um estado desejado, em conformidade com a política de controle. Isso pode ser feito de forma reativa, tomando medidas em resposta ao

estado atual da rede, ou pró-ativa usando técnicas de previsão para antecipar tendências futuras e aplicando medidas para evitar os indesejáveis estados futuros previstos. O nível de intervenção manual envolvido no processo de engenharia de tráfego deve ser minimizado sempre que possível. Isto pode ser conseguido através da automatização de aspectos das ações de controle acima descrito, de forma distribuída e escalável (AWDUCHE, 1999).

A capacidade de controle oferecida pelo funcionamento padrão dos atuais protocolos *Interior Gateway* da Internet não são adequados para Engenharia de Tráfego. Isso torna difícil a efetivação de políticas eficazes para resolver os problemas de desempenho da rede. Na verdade, IGP's baseados em algoritmos de caminho mais curto contribuem significativamente para os problemas de congestionamento em Sistemas Autônomos na Internet. Algoritmos de caminho mais curto geralmente fazem o balanceamento de carga baseado em uma simples métrica aditiva. Estes protocolos não consideram a largura de banda e as características de tráfego nas decisões de roteamento. Consequentemente, o congestionamento ocorre com frequência quando (AWDUCHE, 1999):

- a) Os caminhos mais curtos de múltiplos fluxos de tráfego convergem para *links* específicos ou interfaces do roteador, ou
- b) Um determinado fluxo de tráfego é encaminhado através de um *link* ou interface do roteador que não possui banda suficiente para acomodar o respectivo fluxo de tráfego.

Um exemplo de tentativa de solucionar essa questão em Engenharia de Tráfego é representada pelo MPLS (*Multiprotocol Label Switching*). O MPLS integra um *framework* de comutação de rótulos na camada de rede de roteamento. A idéia básica consiste em atribuir rótulos de comprimento fixo curto para pacotes que ingressam em uma nuvem MPLS (com base no conceito de encaminhamento por classes de equivalência). Durante todo o interior do domínio MPLS, os rótulos atribuídos aos pacotes são usados para tomar decisões de encaminhamento (normalmente, sem recorrer aos cabeçalhos do pacote original). Um conjunto de poderosas construções para resolver muitas questões críticas nos emergentes serviços diferenciados da Internet podem ser criados a partir deste paradigma relativamente simples (AWDUCHE, 1999).

Em contrapartida, a utilização de protocolos de roteamento de estado de enlace, particularmente o OSPF, no balanceamento da carga em um sistema autônomo, é um tema atual e bastante pesquisado na comunidade científico-acadêmica (VIANNA, 2008). Nesse

sentido, a principal motivação deste trabalho está em: propor a implementação prática de uma solução para automatização do processo de balanceamento de carga em redes IP que utilizem o protocolo OSPF, utilizando ferramentas acessíveis, de maneira a contornar os problemas de congestionamento causados pelo funcionamento padrão dos algoritmos de caminho mais curto; e implementar heurísticas de configuração do protocolo de estado de enlace para automatizar o balanceamento de carga, e assim buscar a melhoria do desempenho de sistemas autônomos de redes de computadores.

1.2 OBJETIVO DA DISSERTAÇÃO

O presente trabalho tem por objetivo principal propor a construção de um sistema para aplicação de Engenharia de Tráfego em uma rede de computadores, através da técnica de Auto-Balanceamento de Carga, por meio do monitoramento dos fluxos de dados passantes nas interfaces dos nós, e execução de heurísticas de cálculo e alteração do custo do enlace em um Sistema Autônomo que utilize protocolo de estado de enlace, em evidência o OSPF, durante a ocorrência de eventos que desencadeiem a sobrecarga de dados nos enlaces da topologia.

O OSPF foi escolhido por ser o protocolo de roteamento de estado de enlace mais usado na internet (FORTZ, 2000). Como a técnica utilizada será automatizada, ela é denominada Auto-balanceamento de Carga.

Como o tráfego de pacotes em uma rede de computadores pode mudar com frequência, é interessante que, na eventualidade de ocorrência de sobrecarga de dados em um ou mais enlaces específicos, os caminhos por onde estes pacotes trafegam possam se adaptar ao momento na medida do possível. Ou seja, que a topologia lógica possa ser alterada dinamicamente em busca da melhoria da disponibilidade para alocação de recursos em consequência da diminuição de descarte de pacotes, flutuação e atrasos. Existem estudos e tecnologias desenvolvidas neste assunto. Por exemplo, o OSPF-TE (KATZ, 2003) é um método que adiciona capacidades de Engenharia de Tráfego à topologia que usa o OSPF, distribuindo informações como largura de banda e restrições administrativas com o uso de *Opaque LSA* (COLTUN, 1998). Um outro método de realizar essa alteração da topologia lógica, e que será tratado por este trabalho, está na implementação do balanceamento de carga em um Sistema Autônomo que utilize OSPF, através da manipulação dos valores de custos dos enlaces sem a intervenção do administrador de rede (FORTZ, 2000) (FORTZ, 2002) (SALLES, 2008). Entretanto, os valores escolhidos não podem ser valores quais-

quer. Isso quer dizer que, não basta alterar o valor de custo do enlace para um número simplesmente diferente do valor corrente, pois isso poderá causar efeitos indesejáveis no desempenho.

A importância da observação quanto às propriedades emergentes das redes requer que cuidados especiais devam ser tomados na escolha de medidas para otimizar o desempenho da rede. Otimizando as medidas erradas, pode-se alcançar determinados objetivos locais, mas, por outro lado, pode-se ter consequências desastrosas sobre as propriedades emergentes da rede e, conseqüentemente, sobre a qualidade do serviço percebida pelos usuários finais de serviços de rede (AWDUCHE, 2002).

Dentro do objetivo principal, destacam-se os objetivos específicos deste trabalho, que são os seguintes:

- a) Estudar técnicas de Balanceamento de Carga usando o protocolo de Estado de Enlace OSPF;
- b) Implementar e verificar a eficácia da ferramenta proposta utilizando uma arquitetura viável e eficiente;
- c) Avaliar parâmetros de desempenho para verificar e comparar as heurísticas utilizadas.

Cabe aqui evidenciar os objetivos específicos em maiores detalhes. O primeiro objetivo consiste em pesquisar determinadas heurísticas e criar algoritmos, para que a melhor adequação da matriz de roteamento, em relação ao funcionamento padrão dos protocolos *Link State*, seja alcançada através de uma conveniente reconfiguração dos pesos dos enlaces. A correção automática da matriz de roteamento, via conveniente reconfiguração dos pesos, gera a capacidade de auto-balanceamento da rede, ou seja, dispensa a atuação do gerente da rede para reconfiguração da matriz de roteamento em situações de falha nos roteadores, falha nos enlaces ou em situações de modificação da matriz de tráfego. Aqui, o desempenho de heurísticas para solução do problema da adequada configuração dos pesos dos enlaces em uma rede OSPF deve ser otimizado, considerando a matriz de tráfego estática ou quase-estática. Os custos associados aos enlaces dependem da topologia da rede, em virtude da capacidade de acomodação de tráfego dos enlaces, mas, em primeiro instante, não levando em conta as características de tráfego.

Assim, a utilização do protocolo OSPF em engenharia de tráfego requer a solução do problema da configuração dos pesos dos enlaces. Observa-se na comunidade científico-acadêmica uma considerável demanda por pesquisas sobre heurísticas que produzam bons resultados na solução do problema da configuração dos pesos dos enlaces OSPF (FORTZ, 2002) (SALLES, 2007).

O segundo objetivo específico propõe a criação de uma solução viável e eficiente, dos pontos de vista funcional e econômico, para a efetivação do Auto-balanceamento de carga usando o protocolo de estado de enlace OSPF. Por solução viável e eficiente, entende-se que já existem *softwares* que realizam partes dos processos necessários para o funcionamento do Sistema Autônomo Auto-Balanceado, fazendo-se necessário então realizar as modificações e configurações requeridas para que esses aplicativos trabalhem integrados.

Esse sistema é denominado Sistema Autônomo Auto-Balanceado, termo definido por (VIANNA, 2008): trata-se de um sistema autônomo capaz de dinamicamente alterar os caminhos dos fluxos para distribuir, da forma mais equânime possível, a carga na rede. Eles executam a alteração dos caminhos sem a intervenção do administrador da rede. Esses sistemas podem ter duas abordagens: distribuída e centralizada. Na primeira, os roteadores, de forma distribuída, aumentam ou diminuem dinamicamente os custos dos seus enlaces conforme a variação do nível de ocupação dos enlaces. Na segunda, um determinado nó da rede faz o papel do administrador da rede, sendo necessário estabelecer um protocolo de comunicação entre este nó central e o restante dos nós da rede. Os nós da rede informam o nível de ocupação de seus enlaces para o nó central, e este decide os novos custos a serem configurados nos enlaces dos demais nós.

Neste trabalho, a proposta é adotar a abordagem centralizada, devido à possibilidade de integrar *softwares* e ferramentas de desenvolvimento existentes com recursos computacionais bastante acessíveis. Assim um *host* central, pertencente ao domínio de roteamento de estado de enlace, realiza a coleta das informações de tráfego e a atuação em caso de sobrecarga.

O terceiro e último objetivo está na avaliação do comportamento da solução proposta no cumprimento do objetivo principal. Cabe salientar que a análise estará focada na reconfiguração dos pesos e no reflexo que essa ação exercerá em uma melhor distribuição da matriz de tráfego na topologia estudada.

1.3 ORGANIZAÇÃO DA DISSERTAÇÃO

Esta dissertação está organizada da seguinte maneira: no capítulo 2, será feita uma revisão bibliográfica, necessária para o entendimento do contexto teórico que envolve este trabalho. Os assuntos expostos incluem protocolos de roteamento, gerência de rede e engenharia de tráfego relacionada ao balanceamento de carga. No capítulo 3, encontram-se breves descrições de alguns trabalhos provenientes de pesquisas realizadas na aplicação de protocolos de estado de enlace no balanceamento de carga de redes de computadores. Também encontra-se a descrição mais detalhada da pesquisa sobre Sistemas Autônomos Auto-balanceados. O capítulo 4 expõe a metodologia empregada, iniciando com a classificação do sistema no âmbito da engenharia de tráfego, passando pela forma de implementação escolhida para este trabalho, pela explicação do funcionamento do sistema proposto, e terminando com o detalhamento das heurísticas estudadas. Passa-se ao capítulo 5, onde é apresentado o ambiente de testes e o resultado das simulações. Também é realizada a validação da solução comparando o estudo teórico com a verificação do funcionamento prático. O capítulo 6 encerra o trabalho com as conclusões.

2 CONCEITOS PRELIMINARES

2.1 PROTOCOLOS DE ROTEAMENTO

A principal função da camada de rede é rotear pacotes da máquina de origem para a máquina de destino. Existe a distinção entre roteamento, que é a tomada de decisão sobre quais rotas utilizar, e encaminhamento, o que acontece quando um pacote chega, procurando a linha de saída que será usada nas tabelas de roteamento. O roteamento pode ser dividido em dois tipos: Roteamento Estático, onde as tabelas de roteamento são preenchidas pelo administrador da rede em cada roteador, ou seja, os endereços IP dos nós que formam a topologia já são preestabelecidos; e Roteamento Dinâmico, no qual os roteadores trocam mensagens sobre as redes que conhecem e assim montam a tabela de roteamento dinamicamente, sem a intervenção do administrador de rede. Os algoritmos de roteamento atuam justamente no roteamento dinâmico de pacotes. Muitos algoritmos de roteamento são usados nas redes de computadores. Os algoritmos dinâmicos incluem o algoritmo de vetor de distância e o algoritmo de estado de enlace. A maioria das redes utilizam um desses algoritmos (TANEMBAUM, 2003).

Os protocolos que utilizam roteamento dinâmico desempenham duas funções importantes. Primeiro, eles calculam um conjunto de caminhos mais curtos. Segundo, eles respondem a falhas de rede ou mudanças de topologias atualizando continuamente as informações de roteamento. Os *hosts* e roteadores que fazem uso dos protocolos de roteamento utilizam as tabelas de roteamento. Cada entrada na tabela de roteamento especifica a parte da rede de um endereço de destino e fornece o endereço da próxima máquina em um caminho usado para alcançar essa rede, além de também especificar uma interface usada para o próximo salto (COMER, 2006).

O elemento que mantém a Internet unida é o protocolo da camada de rede, o IP (Internet Protocol). Sua tarefa é fornecer a melhor forma possível (ou seja, sem garantias) de transportar datagramas da origem para o destino, independente dessas máquinas estarem na mesma rede ou de haver outras redes entre elas (TANEMBAUM, 2003). Uma rede IP é organizada em diversos Sistemas Autônomos (*Autonomous System - AS*), consistindo em um grupo de redes e roteadores controlados por uma única unidade administrativa.

Cada AS é responsável pelo seu roteamento interno, podendo usar qualquer algoritmo. A Internet é um conjunto de sistemas autônomos interligados. Os protocolos de roteamento são classificados em IGP e EGP (MOY, 1998b). O *Interior Gateway Protocol* (IGP) estabelece rotas no interior do AS, ou Intra-AS. Exemplos de protocolos Intra-AS são RIP, RIP-2, OSPF. Já o *Exterior Gateway Protocol* (EGP) estabelece rotas entre Sistemas Autônomos, ou Inter-AS, tendo como exemplo o protocolo BGP (BERTSEKAS, 1992). O presente trabalho terá como foco as tecnologias abordadas no interior de um AS, ou seja, os protocolos IGP.

Como citado, os protocolos de roteamento dinâmicos intra-AS se dividem em dois métodos. O primeiro é o Vetor de Distância, consistindo de um algoritmo que possui uma abordagem mais simples, na qual cada roteador contido no Sistema Autônomo (AS) divulga a sua tabela de roteamento completa, que será espalhada e conhecida pelos demais roteadores, comparando com as tabelas já existentes e a seguir inserindo as que possuem a menor distância administrativa. Essa distância de um roteador até cada um de seus vizinhos poderá utilizar várias métricas, como por exemplo, a quantidade de saltos de uma determinada entidade a outra (no caso do RIP), comprimento da fila, e o retardo (TANEMBAUM, 2003).

O segundo método é o Estado de Enlace. Nele os roteadores trocam informações entre si considerando, além do endereçamento, o estado de cada enlace da rede, tendo como principal métrica a largura de banda do enlace, se adaptando melhor às mudanças da topologia.

Hoje em dia a tecnologia de estado de enlace é mais utilizada do que a tecnologia vetor de distância nas redes computacionais (MOY, 1998b), como por exemplo a Internet, pelo principal fato de possuir um menor tempo de convergência, ou seja, uma eventual alteração na topologia de uma rede vetor distância, como a falha em um enlace ou em um roteador, leva mais tempo para ser conhecida e ser refletida nas tabelas de roteamento do que em uma rede de estado de enlace. O algoritmo de Estado de Enlace, ou *Shortest Path First* (SPF) exige que cada roteador participante receba ou calcule informações de topologia. Em vez de enviar mensagens que contenham listas de destinos, um roteador participante em um algoritmo SPF desempenha duas funções. Primeiro, ele testa ativamente o estado de todos os roteadores vizinhos. Em termos de rede física, dois roteadores são vizinhos se eles compartilharem um enlace; em termos de rede lógica, dois vizinhos se conectam a uma rede comum. Segundo, um roteador periodicamente propaga as informações do

estado de enlace para os outros roteadores (COMER, 2006).

Para encorajar a adoção da tecnologia de estado de enlace, um grupo de trabalho do IETF (*Internet Engineering Task Force*) projetou um IGP que usa o algoritmo de estado de enlace, chamado Open SPF (OSPF). É um padrão aberto que qualquer um pode implementar (COMER, 2006).

A base de dados que retrata a topologia da rede no OSPF chama-se *link-state database*. O *link-state database* é um conjunto de *link-state advertisements*, ou LSAs. Assim que um roteador torna-se ativo na rede, ele gera uma LSA em *broadcast* para todos os demais roteadores. Uma LSA informa basicamente com quem este roteador se conecta diretamente e os custos dessas conexões. Cada roteador gera uma LSA para todos os roteadores e também recebe LSAs de todos os roteadores. Os roteadores recebem as LSAs e formam o *link-state database*. A série de LSAs que um roteador recebe permite que ele tenha uma visão da topologia de rede, ou seja, a forma pela qual os roteadores estão dispostos na rede e os custos de todos os enlaces da rede. Com esta base de dados, o roteador tem como calcular os caminhos mais curtos para os outros roteadores da rede e, com isso, construir a tabela de roteamento (VIANNA, 2008). As rotas são calculadas e recalculadas pelo roteador aplicando o conhecido *algoritmo de caminho mais curto* de Dijkstra. O algoritmo de Dijkstra calcula os caminhos mais curtos até todos os destinos de uma única origem (COMER, 2006). As LSAs no *link-state database* podem ser apagadas ou corrompidas. Por esta razão, os protocolos de roteamento de estado de enlace têm um parâmetro de configuração que é o tempo de *refresh*, ou tempo de atualização, das LSAs (VIANNA, 2008).

2.2 GERENCIAMENTO DE REDE

Além dos protocolos que fornecem serviços em nível de rede e programas aplicativos que utilizam esses serviços, é necessário um subsistema que permita a um gerente depurar problemas, controlar roteamento e localizar computadores que violam padrões de protocolos. A essas atividades chamamos *gerenciamento de rede*. Em uma rede TCP/IP, o gerente precisa examinar e controlar roteadores e outros dispositivos de rede. Como esses dispositivos se conectam a redes arbitrárias, os protocolos para o gerenciamento de rede operam no nível de aplicação e se comunicam usando protocolos em nível de transporte TCP/IP. O padrão TCP/IP para gerenciamento de rede é o *Simple Network Management Protocol* (SNMP). O protocolo evoluiu através de três gerações. A versão mais atual é a

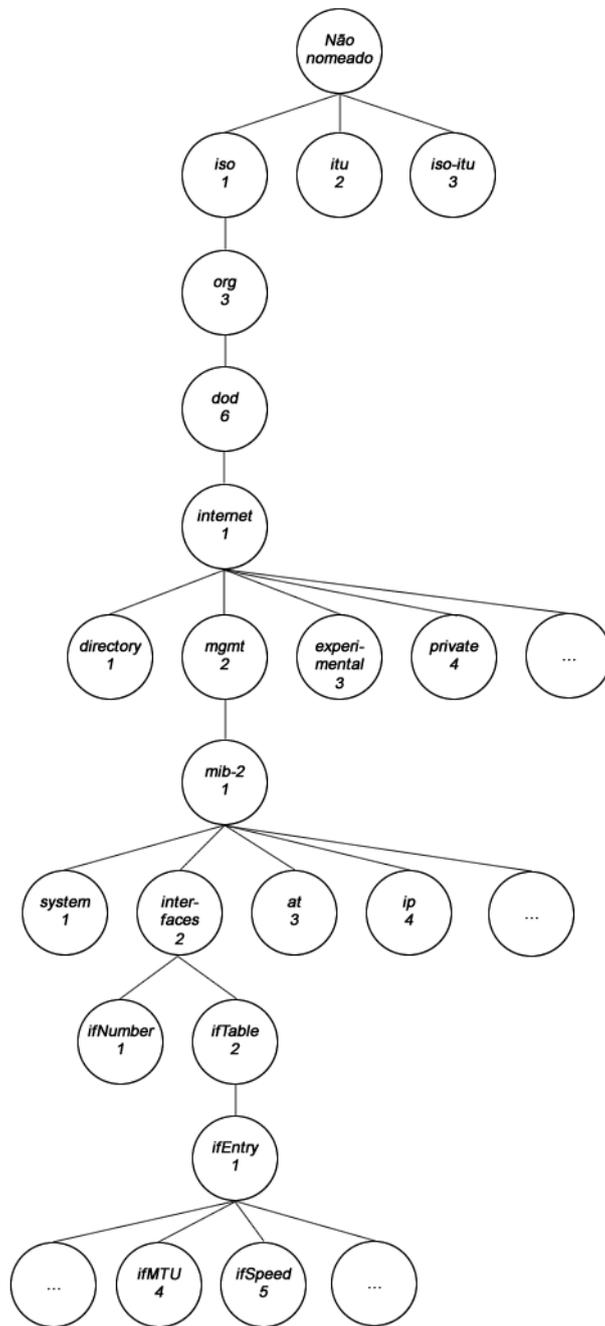


FIG. 2.1: Árvore MIB-2

SNMPv3, e as anteriores são SNMPv1 e SNMPv2. As mudanças entre elas foram pequenas, todas as três versões usam a mesma estrutura geral, e muitos recursos são compatíveis com as versões anteriores. Além de especificar detalhes como o formato de mensagem e o uso dos protocolos de transporte, o padrão SNMP define o conjunto de operações e o significado de cada um (COMER, 2006).

Um *framework* SNMP é dividido em três partes: um gerente SNMP, um agente SNMP e um MIB. O gerente SNMP é o sistema usado para controlar e monitorar as atividades dos *hosts* da rede usando SNMP. O termo mais comum usado para denominar um sistema de gerência é chamado *Network Management System* (NMS). O agente SNMP é um *software* inserido no dispositivo gerenciado que mantém as informações sobre o dispositivo e reporta essas informações ao sistema de gerência (CISCO, 2009). Já o *Management Information Base* (MIB) é um padrão independente do protocolo de gerenciamento de rede que especifica os detalhes para cada tipo de dispositivo. Esse padrão especifica os itens de dados que um dispositivo gerenciado precisa manter, as operações permitidas em cada um e o seu significado (COMER, 2006).

O IETF (*Internet Engineering Task Force*) permite a publicação dos documentos MIB individualmente, que especifiquem as variáveis para um tipo de dispositivo específico. Existe um padrão separado que especifica um conjunto de regras usadas para definir e identificar variáveis MIB, conhecidas como a especificação *Structure of Management Information* (SMI). O padrão SMI especifica que todas as variáveis MIB precisam ser definidas e referenciadas usando a *Abstract Syntax Notation 1* (ASN.1), da ISO. Trata-se de uma linguagem formal que possui dois recursos principais: uma notação usada em documentos lidos por humanos e uma representação codificada compacta das mesmas informações usadas nos protocolos de comunicação. Os nomes usados para variáveis MIB são tirados do *domínio do identificador de objeto* (OID), administrado pela ISO e ITU. Sua idéia central é fornecer um *namespace* em que todos os objetos possíveis possam ser designados. A raiz da hierarquia do OID não é nomeada, mas possui três descendentes diretos gerenciados por: ISO, ITU, e em conjunto por ISO e ITU. Aos descendentes, são atribuídas *strings* de texto curtas e inteiros que as identificam (COMER, 2006). A representação de uma árvore MIB pode ser vista na Figura 2.1.

Por exemplo, a variável MIB nomeada *ifSpeed*, que fornece uma estimativa de largura de banda atual da interface do roteador em bps, tem o seu nome representado por

```
iso . org . dod . internet . mgmt . mib-2 . interfaces . ifTable . ifEntry . ifSpeed
```

e a sua representação numérica correspondente é representada por:

1 . 3 . 6 . 1 . 2 . 1 . 2 . 2 . 1 . 5

O caminho percorrido na árvore MIB até a variável *ifSpeed* pode ser conferido na Figura 2.1.

2.3 CONTEXTO ATUAL DA REDE

Tanto a tecnologia TCP/IP quanto a internet não param de evoluir. Novos protocolos estão sendo propostos, e antigos estão sendo revisados. A demanda mais significativa na atual tecnologia de conectividade não surge de conexões de rede adicionais, mas sim de tráfego adicional. À medida que surgem novos usuários e novas aplicações na Internet, os padrões de tráfego mudam. Por exemplo, quando os usuários começaram a buscar informações usando a web, o tráfego aumentou drasticamente. Mais adiante, os padrões de tráfego mudaram novamente quando o compartilhamento de arquivos passou a ser usado com intensidade por grande parte dos usuários. E mais mudanças estão ocorrendo ao passo que a Internet é usada para serviços de telefonia e vídeo (COMER, 2006).

A Internet contemporânea possui três características significantes: (1) Fornece serviços em tempo real, (2) traz consigo missões críticas, e (3) seu ambiente operacional é muito dinâmico. As características dinâmicas de redes IP podem ser atribuídas, em parte, às flutuações na demanda, à interação entre vários protocolos e processos da rede, à rápida evolução da tecnologia que demanda a constante inclusão de novos recursos e elementos de rede, e às transitórias e permanentes deficiências que ocorrem no sistema (AWDUCHE, 2002).

2.4 ENGENHARIA DE TRÁFEGO

A Engenharia de Tráfego engloba a aplicação de tecnologia e princípios científicos para a medição, caracterização, modelagem e controle do tráfego da Internet (AWDUCHE, 1999).

Melhorar o desempenho de uma rede operacional, tanto o tráfego quanto os recursos disponíveis, são os objetivos principais da Engenharia de Tráfego da Internet. Isso é conseguido pelo endereçamento de tráfego orientado aos requisitos de desempenho, enquanto utiliza recursos ligados à economia e confiabilidade da rede. Medição de tráfego

orientado ao desempenho incluem atraso, variação de atraso, perda de pacotes e taxa de transferência (AWDUCHE, 2002).

Combater o congestionamento, a um custo razoável, é um dos objetivos principais da Engenharia de Tráfego da Internet.

Pacotes de dados competem pela utilização dos recursos da rede a medida em que são transmitidos. Um recurso de rede é considerado congestionado se a taxa de chegada de pacotes exceder a capacidade de serviço do recurso em um dado intervalo de tempo. O congestionamento pode resultar em alguns pacotes de chegada sendo atrasados ou até mesmo descartados. O congestionamento aumenta os atrasos, a variação de atraso, a perda de pacotes, e reduz a previsibilidade dos serviços de rede. Claramente, o congestionamento é um fenômeno altamente indesejável (AWDUCHE, 2002).

O contexto de solução para a Engenharia de Tráfego da Internet envolve a análise e avaliação de alternativas, e escolha entre cursos alternativos de ação. Geralmente, o contexto da solução se baseia em fazer inferências razoáveis sobre o estado atual ou futuro da rede e, posteriormente, a tomada de decisões adequadas, que podem envolver uma preferência entre conjuntos alternativos de ação. Mais especificamente, o contexto de solução demanda estimativas razoáveis de volume de tráfego, a caracterização do estado da rede, derivando em soluções para problemas de engenharia de tráfego que podem ser implícita ou explicitamente formuladas, e, possivelmente, instanciar um conjunto de ações de controle. As ações de controle podem envolver a manipulação de parâmetros associados com o roteamento, o controle sobre a aquisição da capacidade táctica e o controle sobre as funções de gestão do tráfego.

A lista a seguir descreve instrumentos que podem ser aplicáveis ao contexto de solução da Engenharia de Tráfego na Internet:

- a) Um conjunto de políticas, objetivos, e requisitos (que podem ser dependentes do contexto) para análise e otimização do desempenho da rede;
- b) Uma coleção de ferramentas e mecanismos para medição, caracterização, modelagem e controle do tráfego Internet e controle sobre a atribuição e alocação de recursos da rede, bem como o controle sobre o mapeamento ou distribuição de tráfego, considerando a infra-estrutura;
- c) Um conjunto de restrições no ambiente operacional, nos protocolos de rede, e no próprio sistema de engenharia de tráfego;

- d) Um conjunto de técnicas quantitativas e qualitativas, e metodologias para abstração, formulação e solução dos problemas de engenharia de tráfego;
- e) Um conjunto de parâmetros de controle administrativo que poderá ser manipulado através de um sistema de Gerenciamento de Configuração (*Configuration Management* - CM). O próprio sistema CM poderá incluir um subsistema de controle da configuração, um repositório de configuração, um subsistema de cálculo da configuração, e um subsistema de auditoria da configuração;
- f) Um conjunto de guias para avaliação, otimização e incremento do desempenho da rede.

2.5 BALANCEAMENTO DE CARGA

A natureza da Engenharia de Tráfego é efetivamente uma otimização de rotas para o reforço da capacidade de serviço de rede sem causar congestionamento. Ao fazê-lo, os objetivos típicos da Engenharia de Tráfego incluem o equilíbrio da distribuição de carga e a minimização do consumo de largura de banda na rede (WANG, 2008).

Muitas vezes, o congestionamento resultante da alocação de recursos ineficiente pode ser reduzido através da adoção de políticas de Balanceamento de Carga. O objetivo dessas estratégias é minimizar o congestionamento ao máximo ou, em alternativa, para minimizar a utilização máxima dos recursos, através da alocação eficiente de recursos. Quando o congestionamento é minimizado através da alocação eficiente dos recursos, diminui a perda de pacotes e o atraso dos pacotes, e aumenta o rendimento agregado. Dessa forma, a percepção da qualidade do serviço de rede experimentado por usuários finais torna-se significativamente reforçada.

Claramente, o Balanceamento de Carga é uma importante política de otimização de desempenho da rede. No entanto, a capacidade prevista de Engenharia de Tráfego deverá ser flexível o suficiente para que administradores de rede possam implementar outras políticas que levem em conta a estrutura de custos existentes e o modelo de utilidade ou de receita (AWDUCHE, 1999).

O Balanceamento de Carga e a Divisão de Carga não são a mesma coisa. Seja (O, D) uma notação que representa o nó de origem e o nó de destino das demandas. A Divisão de Carga provê meios para, aleatoriamente, distribuir os fluxos de tráfego (*, D) e (O, D) através do encaminhamento por múltiplos caminhos com custos iguais, sem

necessariamente resultar no tráfego com carga balanceada. Devido à aleatoriedade da distribuição dos fluxos (*, D) e (O, D), os métodos utilizados para a divisão da carga tentam distribuir um montante de fluxos de dados igual em cada um dos RPF (*Reverse Path Forwarding*) não pela contagem de fluxos, mas sim realizando uma pseudo-decisão. Esses métodos são coletivamente referidos como métodos *multicast* de Divisão de Carga. Esses métodos então resultam em um melhor compartilhamento da carga em redes em que existem muitos fluxos de que utilizem aproximadamente a mesma quantidade de carga (CISCO, 2007b).

Em contrapartida, políticas de provisionamento de serviços e de condicionamento de tráfego são suficientemente dissociados dos comportamentos de encaminhamento no interior da rede, para permitir a implementação de uma ampla variedade de recursos de rede, tendo espaço para futura expansão (BLAKE, 1998).

Quando algum nó ou *link* falhar, o roteamento adaptativo adotado encontra um caminho alternativo. Protocolos que usam mensagens de atualização periódica, como por exemplo o RSVP, automaticamente fazem um pedido de reserva de recursos junto ao novo caminho. Evidentemente, essa reserva pode falhar, porque não há capacidade suficiente disponível no novo caminho. Este é um problema de provisionamento e de engenharia de rede, que não pode ser resolvido por protocolos de roteamento ou de configuração (BRADEN, 1994).

Em certos casos, as políticas de Engenharia de Tráfego não solucionam a questão da alocação eficiente de recursos, pois nesse caso o problema pode estar no provisionamento de carga, que se relaciona com os investimentos realizados em infra-estrutura. Para os prestadores de serviços de Internet (ISPs), o provisionamento de carga e a garantia de serviços QoS são questões importantes. Especialmente quando enfrentam uma forte concorrência, os ISPs devem fornecer níveis de serviço satisfatórios, porém os custos médios de provisionamento devem ser de ordem econômica. Um provedor pode enfrentar vários cenários em que a largura de banda é necessária, incluindo o crescimento de assinantes, e o crescimento das demandas do usuário para aplicações de Internet, tais como VoIP (*Voice over Internet Protocol*) e o tráfego de outros dados. Devido à falta de instrumentos de planejamento adequado, os ISPs por vezes simplesmente dobram a capacidade de banda de rede e, portanto, suas despesas de investimento. Esta prática, amplificada por projeções demasiadamente otimistas de crescimento de largura de banda, levam a um impacto econômico catastrófico nos próximos anos. Algumas empresas têm mais largura de

banda do que o mercado pode consumir. Em outros cenários, os ISPs querem aumentar os lucros de suas redes de comunicação existentes, adicionando usuários e outros serviços, mas eles não têm certeza de como manter a qualidade de serviço. A qualidade de serviço está relacionada à satisfação dos usuários, e pode ter um impacto econômico quando os usuários insatisfeitos mudarem para outro ISP que ofereça um melhor serviço. Alguns outros cenários também existem, por exemplo, um ISP pode precisar estabelecer preços com base em vários níveis de qualidade de serviço (ALMHANA, 2006).

3 TRABALHOS RELACIONADOS

3.1 APLICAÇÃO DO PROTOCOLO OSPF EM ENGENHARIA DE TRÁFEGO

Esta dissertação se relaciona com trabalhos que tratam do problema da adequada configuração dos custos dos enlaces, a partir de uma topologia de rede e uma matriz de tráfego estática ou quase estática. Esse problema deve ser considerado nas pesquisas que trabalham em cima dos valores de custos, para a utilização do protocolo de estado de enlace OSPF como ferramenta de Engenharia de Tráfego. No caso dos trabalhos que pesquisaram o uso do OSPF como ferramenta de Engenharia de Tráfego, deve-se atentar para o fato dele ser o protocolo de roteamento mais usado na Internet (FORTZ, 2000).

Em (MOY, 1998b), é dito que o OSPF não é responsivo às sobrecargas na rede, e que caso a tentativa de realizar essa responsividade for feita seguindo o funcionamento padrão do protocolo, tornará a escolha de novos caminhos em resposta às sobrecargas difícil de ser realizada no contexto distribuído: o roteamento ficará instável, pois todos os roteadores escolherão inicialmente o melhor caminho e moverão os fluxos em massa para o caminho secundário mais aliviado. Isso causará uma nova sobrecarga, e assim os roteadores ficarão sempre movendo os fluxos de volta.

Um aspecto considerado no problema da adequada configuração dos pesos dos enlaces é a redução do espaço de busca das heurísticas na resolução do problema, tratada em (MOY, 1998a). Parte-se da premissa que o peso do enlace do nó X para o nó Y seja igual ao peso de Y para X. No OSPF, destina-se 16 bits para configuração dos custos dos enlaces podendo variar de 1 a 65.535 . Dado uma tabela de roteamento gerada por um sistema de pesos, se esse sistema de pesos for multiplicado por um número natural k maior que 1, então os caminhos mais curtos continuarão sendo os mais curtos, implicando em que a matriz de roteamento permaneça inalterada. O espaço de busca deve limitar-se a vetores de pesos que não possuam divisores comuns a todos os pesos, à exceção do divisor 1. Um determinado vetor de pesos deverá então satisfazer esta condição para pertencer ao espaço de busca, qualquer que seja a topologia de rede e qualquer que seja a matriz de tráfego.

Outro trabalho que trata do aspecto considerado no problema da adequada con-

figuração dos pesos dos enlaces está em (SALLES, 2007), onde afirma-se que uma das dificuldades centrais do OSPF para engenharia de tráfego deve-se à utilização do paradigma de encaminhamento baseado no destino, pois a origem do tráfego não tem controle e nem pode influenciar sobre o procedimento de seleção do caminho, e assim a decisão de roteamento é efetuada em cada roteador. O algoritmo baseado no menor custo seleciona os caminhos mais curtos para preencher a tabela de roteamento. Caso os caminhos mais curtos estejam sobrecarregados, outros caminhos de maior comprimento não serão utilizados mesmo se estiverem subutilizados. O trabalho relacionado traz como proposta desenvolver soluções heurísticas para resolver o problema do roteamento ótimo de menor caminho ou problema da otimização do roteamento OSPF, consistindo em encontrar os pesos ótimos para os enlaces de uma rede de comunicação que possibilite ao roteamento OSPF minimizar uma certa função de custo global.

Outras pesquisas já colocam em evidência outros aspectos além da adequada configuração dos pesos dos enlaces. O OSPF-TE é uma extensão do OSPF que adiciona capacidades de engenharia de tráfego ao protocolo. Essa extensão utiliza uma classe de *Link State Advertisements* (LSA), denominada *Opaque LSAs*, que provê um mecanismo generalizado para permitir a futura extensibilidade do OSPF. Um *Opaque LSA* consiste de um LSA padrão, seguido de uma informação específica de algum aplicativo. O campo “informação” pode ser utilizado diretamente pelo OSPF ou por outras aplicações. Os mecanismos de inundação de *Link-state Database* do OSPF padrão são usados para distribuir *Opaque LSA* para toda ou para parte de toda a topologia OSPF (COLTUN, 1998). Existem três tipos de *Opaque LSA*, cada um com um escopo diferente de inundação. O OSPF-TE faz uso do tipo 10, que está no escopo de inundação de intra-área, definindo assim um novo LSA, chamado *Traffic Engineering LSA*, que abrange roteadores, *links* ponto-a-ponto e conexões para redes de multi-aceso. Assim, as informações de engenharia de tráfego podem ser divulgadas pelos roteadores que possuem a extensão habilitada, e incluem largura de banda máxima do *link*, tipo de *link*, endereço IP do roteador que envia o *Traffic Engineering LSA*, entre outras (KATZ, 2003).

Uma metodologia de distribuição de custos importante que será utilizada nessa dissertação é a recomendação da fabricante de roteadores Cisco para que os custos dos enlaces em uma rede OSPF sejam inversamente proporcionais às suas capacidades (THOMAS, 1998). Vale lembrar que a Cisco é a maior fabricante de roteadores do mundo (CISCO, 2010). Como a largura de banda é utilizada pelo OSPF como métrica para o roteamento de

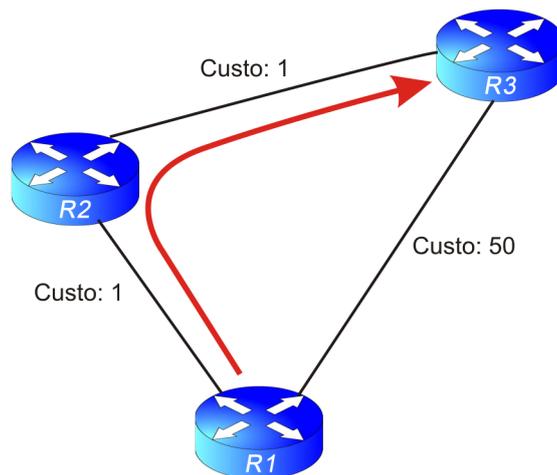


FIG. 3.1: Custo atribuído aos enlaces

pacotes através da definição de um custo associado às interfaces dos roteadores em um enlace, a fabricante recomenda como regra geral o uso do custo igual a $C = 10^8 / \textit{bandwidth}$, significando que, quanto mais alta for a capacidade de largura de banda de um enlace, o seu custo associado será proporcionalmente menor, aumentando assim a probabilidade de passagem dos fluxos por este enlace. Analogamente, quanto menor for a capacidade de um determinado enlace, maior será o seu custo associado, o que conseqüentemente fará com que diminua a probabilidade de passagem de fluxos por esse enlace. Os trabalhos relacionados mostram que o valor do custo de enlace recomendado pela Cisco para o uso geral é uma heurística simples, pois não leva em consideração o custo associado às demandas passantes e nem a taxa de ocupação dos enlaces. O tráfego de pacotes IP, ao atingir um determinado limite de fluxo, tornará a rede não balanceada.

Considere a Figura 3.1. Se, por exemplo, um fluxo de 95% da capacidade do enlace for direcionado do roteador R1 para o roteador R3 através da rota $R1 \rightarrow R2 \rightarrow R3$, cujo custo associado possui o valor de 2, a outra rota, $R1 \rightarrow R3$, continuará a não ser utilizada, pois possui um custo maior associado a ela, que no caso é de 50. Diz-se, assim, que a rede não está devidamente balanceada, pois existe um enlace capaz de acomodar o todo ou parte do fluxo direcionado à R3, mesmo considerando que ele possui um custo maior, ou ainda uma capacidade menor.

Logo, se faz necessário a adoção de métodos que façam a realização do Balanceamento de Carga, que é importante, sobretudo, para evitar a queda do desempenho na alocação de recursos da rede, tão essencial em determinados serviços de rede de computadores.

3.2 SISTEMAS AUTÔNOMOS AUTO-BALANCEADOS

No trabalho realizado por (VIANNA, 2008), o foco da pesquisa é a reconfiguração dinâmica dos pesos dos enlaces visando o balanceamento de carga, deixando de lado a configuração estática abordada em outros trabalhos. Ele propõe a implementação do Sistema SAAB (Sistema Autônomo Auto-Balanceado) para adoção em uma abordagem distribuída, i.e., que os algoritmos para o balanceamento de carga sejam executados em todos os roteadores de um determinado Sistema Autônomo, permitindo a adaptação dinâmica às variações no tráfego e na topologia da rede. Os custos dos enlaces poderiam ser alterados pelo administrador da rede, porém esta tarefa passa a ser inviável a partir do momento em que a topologia e a intensidade da mudança e tráfego de dados atingem determinado grau de complexidade.

Existem duas abordagens de construção possíveis do sistema: centralizada e distribuída. Na abordagem centralizada, existe uma simplicidade inicial, porém algumas dificuldades ocorrem em sua implementação, principalmente no que tange à fragilidade, pelo fato de que a eventual ocorrência de falha no nó central, ou na comunicação entre este nó e os demais, resultará na inoperância do auto-balanceamento, e a diminuição da banda disponível na rede devido a troca de mensagens entre os nós. Já na abordagem distribuída, cada roteador, devidamente distribuído na topologia, altera dinamicamente os custos dos seus enlaces conforme a variação dos níveis de ocupação, entretanto sem troca de comunicação entre eles. Essa abordagem garante que não haja os gargalos que existem no projeto centralizado quanto à troca de mensagens, porém possui uma grande desvantagem de gerar instabilidade na rede, pois os caminhos dos fluxos se alteram constantemente, fazendo com que os pacotes se percam na rede.

A proposta original do sistema SAAB é um sistema distribuído, instanciado em cada um dos roteadores da topologia do AS (Sistema Autônomo). Ele segue o ciclo de trabalho mostrado na Figura 3.2, e tem como principais características o seu caráter dinâmico, adaptável às variações no tráfego e na topologia da rede.

Segundo (VIANNA, 2008), o SAAB envolve os seguintes processos e respectivos métodos utilizados em cada um dos processos:

- a) Verificação do nível de ocupação dos enlaces. Neste processo, cada roteador descobre o nível de ocupação dos seus enlaces, realizada através da contagem dos pacotes que partem e que chegam por cada enlace. O roteador deve conhecer *a priori* a

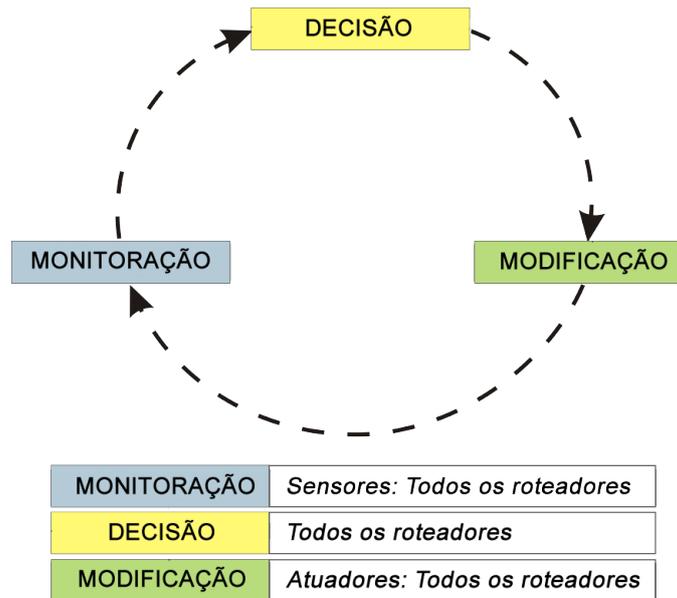


FIG. 3.2: Ciclo de Trabalho do SAAB - Esquema original

capacidade dos enlaces. A razão entre os fluxos que chegam e que partem dos enlaces e a capacidade dos enlaces fornece o nível de ocupação dos enlaces.

b) Divulgação do nível de ocupação dos enlaces.

Neste processo, cada roteador informa aos demais roteadores da rede o nível de ocupação dos seus enlaces, utilizando para isso um método que insere nos custos dos enlaces a informação referente à ocupação desses enlaces, sem que isso interfira na tabela de roteamento, dado que os protocolos de roteamento de estado do enlace já se encarregam da divulgação dos custos dos enlaces para todos os roteadores.

c) Escolha do roteador encarregado de redistribuir a carga.

Durante este processo, os roteadores, com base no nível de ocupação dos enlaces e na topologia da rede, descobrem o roteador que tem direito a alterar os pesos de seus enlaces, que seguirá os seguintes critérios em ordem de preferência, passando-se para o próximo em caso de empate: Roteador que esteja habilitado a melhorar a distribuição da carga → Nível de ocupação do enlace mais sobrecarregado do roteador → Quantidade de enlaces habilitados para alteração dos pesos que o roteador possui → Número IP mais alto. A alteração nos pesos dos enlaces cabe apenas a um roteador por vez, a fim de evitar instabilidade na rede.

d) Alteração no peso dos enlaces.

Neste processo, o roteador que tem direito a alterar os pesos de seus enlaces procede a alteração, caso julgue pertinente conforme o momento atual da rede, seguindo uma heurística de cálculo do valor do custo.

Conforme já colocado no capítulo 1, os valores de custo escolhidos para a alteração nos enlaces não podem ser valores quaisquer. A teoria do caminho mais curto usa os custos para o cálculo das rotas a serem utilizadas por determinados fluxos de dados. Esses fluxos, ou demandas, são transferidas sempre de nós para nós, entretanto o protocolo de estado de enlace não conhece a origem e o destino, ou seja, a matriz de tráfego. O protocolo de estado de enlace trabalha levando em conta a largura de banda do *link*, quanto menor a largura de banda, maior deverá ser o custo atribuído, e quanto maior a largura de banda, menor deverá ser o valor de custo. Caso o valor de custo seja alterado para um valor incompatível com o momento, poderá causar efeitos indesejáveis, pois a topologia lógica poderá ser mudada para que passe a acomodar demandas que não deveriam sofrer alteração no caminho percorrido.

A heurística considera que um enlace E pode ser modelado como uma chave, supondo que os demais enlaces permaneçam com os pesos inalterados. Dependendo da posição da chave, os fluxos passam ou não pelo enlace que está sendo modelado. A posição da chave é o peso do enlace. Conforme o valor do peso do enlace, os fluxos passam por esse enlace (peso baixo) e, a medida que o peso aumenta, os fluxos deixam de passar por esse enlace. O peso do enlace age como uma barreira para a passagem dos fluxos pelo enlace.

A modelagem do enlace E é feita da maneira descrita a seguir, e sua explicação é importante no contexto do presente trabalho. Considerar as seguintes notações:

- $P(E)$, peso do enlace E
- Γ_E , o caminho mais curto da demanda (O, D) que obrigatoriamente inclua o enlace E , se existir;
- Γ_{notE} , o caminho mais curto da demanda (O, D) que obrigatoriamente não inclua o enlace E , se existir;
- $P(\Gamma_E)$, o peso do caminho Γ_E , que é a soma dos pesos de todos os enlaces que formam Γ_E ;
- $P'(\Gamma_E)$, o peso do caminho Γ_E menos o peso do enlace E ;

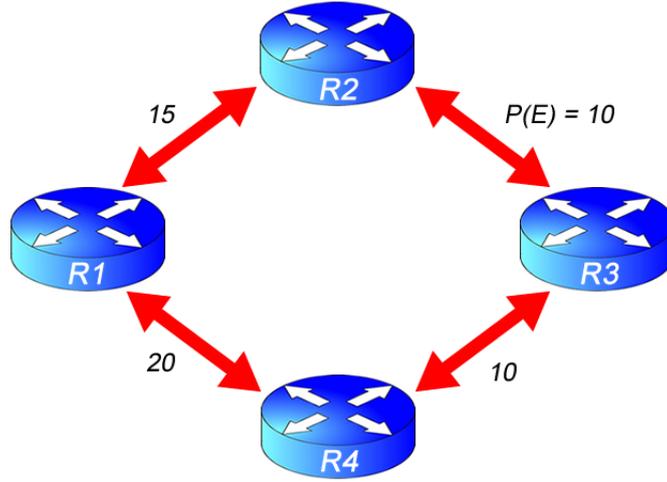


FIG. 3.3: Topologia de rede

- $P(\Gamma_{notE})$, o peso do caminho Γ_{notE} .

Como $P(\Gamma_E) = P'(\Gamma_E) + P(E)$, logo: $P'(\Gamma_E) = P(\Gamma_E) - P(E)$.

Se $P'(\Gamma_E) + P(E) < P(\Gamma_{notE})$, então o caminho que pertence à demanda (O, D) incluirá o enlace E. Assim, o fluxo (O, D) passará pelo enlace E se $P(\Gamma_{notE}) - P'(\Gamma_E) > P(E)$.

Da mesma maneira, se $P'(\Gamma_E) + P(E) > P(\Gamma_{notE})$, então o menor caminho da demanda (O, D) não incluirá o enlace E. Logo, o fluxo (O, D) não passará pelo enlace E se $P(\Gamma_{notE}) - P'(\Gamma_E) < P(E)$.

Demanda	$P(\Gamma_{notE})$	$P'(\Gamma_E)$	$P(\Gamma_{notE}) - P'(\Gamma_E)$
$R1 \rightarrow R2$	15	30	-15
$R1 \rightarrow R3$	30	15	15
$R1 \rightarrow R4$	20	25	-5
$R2 \rightarrow R3$	45	0	45
$R2 \rightarrow R4$	35	10	15
$R3 \rightarrow R4$	10	35	-25

TAB. 3.1: Diferença entre os pesos dos caminhos Γ_{notE} e Γ_E

O exemplo a seguir, retirado de (VIANNA, 2008), buscará facilitar a compreensão da heurística. Considerar a topologia de rede mostrada na Figura 3.3. Na Tabela 3.1, é mostrada a diferença entre os pesos dos caminhos Γ_{notE} e Γ_E , referentes ao enlace E, (R2, R3). Para o entendimento da modelagem do peso do enlace E como uma barreira para os fluxos, será utilizada a Figura 3.4.

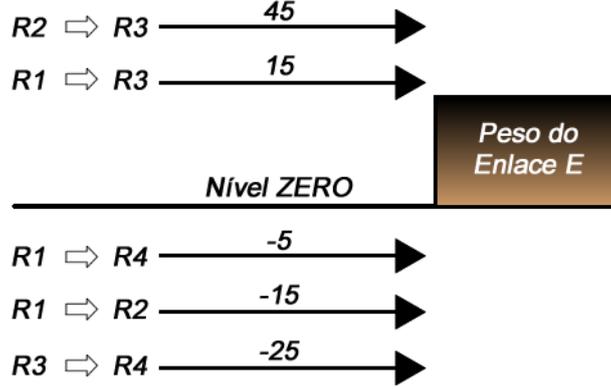


FIG. 3.4: Modelagem do Peso do Enlace E como uma Barreira

O peso $P(E)$ atua como uma barreira para os fluxos. Caso $P(\Gamma_{notE}) - P'(\Gamma_E)$ esteja acima da barreira $P(E)$, o fluxo (O, D) passará pelo enlace E , caso contrário, o fluxo não passará pelo enlace. Essa diferença, $P(\Gamma_{notE}) - P'(\Gamma_E)$, representa uma altura para os fluxos, enquanto que $P(E)$ representa uma altura para a barreira do enlace E .

Continuando o exemplo, mantendo-se constantes, com exceção do enlace E , os valores dos pesos dos enlaces da topologia mostrada na Figura 3.3 (lembrando que os pesos dos enlaces são números naturais), verifica-se que:

- Se $0 < P(E) < 15$, então os fluxos $R1 \rightarrow R3$ e $R2 \rightarrow R3$ passarão pelo enlace E .
- Se $15 < P(E) < 45$, então somente o fluxo $R2 \rightarrow R3$ passará pelo enlace E .
- Se $P(E) > 45$, então nenhum fluxo passará pelo enlace E .
- Os fluxos $R1 \rightarrow R4$, $R1 \rightarrow R2$ e $R3 \rightarrow R4$ não passarão pelo enlace E , qualquer que seja o $P(E)$.
- É impossível alterar o peso do enlace E de forma a permitir que o fluxo $R1 \rightarrow R3$ passe e o fluxo $R2 \rightarrow R3$ não passe pelo enlace E .

Conforme a Figura 3.4, o fluxo $R1 \rightarrow R3$ passa pelo enlace E . O roteador $R2$ compara o nível de ocupação dos enlaces do caminho do fluxo $R1 \rightarrow R3$ atual com o nível de ocupação dos enlaces do caminho deste fluxo com $C(E) = 16$. Ao verificar que o aumento do peso do enlace E minimiza o desvio padrão da ocupação

dos enlaces na rede, então o roteador R2, utilizando a heurística proposta pelo SAAB, aumentará o peso do enlace E para 16.

4 ARQUITETURA PROPOSTA

4.1 TAXONOMIA DO SISTEMA PROPOSTO NA ENGENHARIA DE TRÁFEGO

De acordo com (AWDUCHE, 2002), a taxonomia para os sistemas de Engenharia de Tráfego pode ser construída baseada em diferentes pontos de vistas, conforme listados a seguir:

- Dependente do tempo vs Dependente do estado vs Dependente do evento
- *Offline* vs *Online*
- Centralizado vs Distribuído
- Informação Local vs Global
- Prescritivo vs Descritivo
- Circuito aberto vs Circuito fechado
- Tático vs Estratégico

Com base nisso, a solução proposta neste trabalho possui a seguinte classificação:

- a) Dependente do Estado: Engenharia de Tráfego dependente do estado adapta o planejamento do roteamento de pacotes baseado no estado atual da rede. O estado atual de uma rede fornece informações adicionais sobre variações no tráfego real (i.e., perturbações ocorridas em variações periódicas) que não poderiam ser previstos com base nas informações históricas. O roteamento baseado em restrições é um exemplo de TE dependente do estado operando em uma escala de tempo relativamente longa. Um exemplo de operação em uma escala de tempo relativamente curto é o algoritmo de balanceamento de carga descrito em (ELWALID, 2001).
- b) Tempo real (*Online*): A computação *Online* é requisitada quando o planejamento de roteamento deve se adaptar às condições de mudanças na rede, tal como os algoritmos dependentes do estado. Ao contrário da computação *Offline* (que pode exigir

esforço computacional), a computação *Online* é voltada para cálculos relativamente simples e rápidos para a seleção de rotas, ajuste fino da alocação de recursos, e execução do balanceamento de carga.

- c) Centralizado: O controle centralizado possui uma autoridade central que determina os planos de roteamento ou talvez outros parâmetros de controle de TE em favor da cada roteador. A autoridade central coleta a informação de estado da rede periodicamente de todos os roteadores e retorna a informação de roteamento para os roteadores. O ciclo de atualização do roteamento é um parâmetro crítico que impacta diretamente no desempenho da rede que está sendo controlada. O controle centralizado pode necessitar de alto poder de processamento e alta largura de banda dos canais de controle.
- d) Local: A informação local pertence ao estado de uma parcela do domínio. Exemplos incluem largura de banda e a taxa de perda de pacotes de um caminho em particular. Informação de estado local pode ser suficiente para certos casos de TEs distribuídos-controlados.
- e) Prescritivo: Engenharia de Tráfego Prescritiva avalia alternativas e recomenda um curso de ação. A TE Prescritiva pode ainda ser categorizada como corretiva ou preditiva. A engenharia de tráfego corretiva, que é o caso empregado neste trabalho, prescreve um curso de ação endereçada a uma anomalia existente ou prevista, enquanto que a preditiva prescreve um curso de ação para evoluir e melhorar o desempenho da rede mesmo quando as anomalias não são evidentes.
- f) Circuito fechado: O controle da engenharia de tráfego em circuito fechado é quando o controle da ação utiliza *feedback* de informações sobre o estado da rede. As informações de *feedback* podem ser na forma de histórico de informações ou medições atuais.
- g) Tático: TE Tático tem como objetivo tratar de problemas específicos de desempenho (como *hot-spots*) que ocorrem na rede do ponto de vista tático, sem considerar os imperativos estratégicos globais, enquanto que a TE do ponto de vista Estratégico aborda o problema a partir de uma perspectiva mais organizada e sistemática, levando em consideração as consequências imediatas e de longo prazo de ações e políticas específicas.

4.2 CONTEXTO DO SISTEMA

O presente trabalho abordará muitos dos conceitos originalmente propostos pelo sistema SAAB adotado em (VIANNA, 2008), implementando algumas modificações, bem como introduzindo algumas particularidades.

Uma das modificações está na colocação do sistema em funcionamento. No trabalho relacionado, o funcionamento foi demonstrado através de simulações realizadas com o uso do *software* NS-2, enquanto que no presente trabalho o funcionamento do sistema é realizado com experimentos práticos, ou seja, com o sistema atuando em uma topologia de rede real.

Conforme já classificado na Subseção 4.1, uma outra modificação está na abordagem de construção: enquanto que no sistema SAAB original a proposta de construção é distribuída, neste trabalho a proposta de construção será feita de maneira centralizada, ou seja, o nó central verifica os níveis de ocupação dos enlaces de um Sistema Autônomo, decide os momentos, as pertinências e em quais *links* deve atuar, calcula os novos valores de custo associados às rotas, e efetua as alterações nas interfaces dos roteadores.

Apesar das desvantagens citadas na Subseção 3.2, existem várias vantagens na adoção desse tipo de abordagem. Conforme explicado na subseção 1.2, um dos objetivos específicos deste trabalho é criar uma solução viável e eficiente para a efetivação do Auto-balanceamento de carga junto com o protocolo OSPF. Como já existem *softwares* que realizam partes dos processos do Sistema Autônomo Auto-Balanceado, o funcionamento da solução depende da adequada integração entre eles, obedecendo os requisitos definidos pelo ciclo de atuação do SAAB, conforme mostrado na Figura 4.1. Esses aplicativos são executados em um *host* central, inserido em uma topologia de rede controlada, ou seja, com todos os roteadores previamente conhecidos em um AS (sistema autônomo). A implementação de uma solução distribuída demandaria um maior tempo e esforço técnico para o seu estudo e construção, por tratar-se, a princípio, de uma solução de implementação mais complexa do que a solução centralizada. Se for considerado o caso dos roteadores, cada fabricante possui a sua própria aplicação sendo executada nos equipamentos, enquanto que a solução centralizada pode fazer uso de sistemas operacionais e aplicativos comuns à computadores pessoais.

Durante o estudo do sistema proposto, considerar como sinônimos: os termos fluxo e demanda, rota e caminho. Considerar (O, D) uma notação que representa o nó de origem

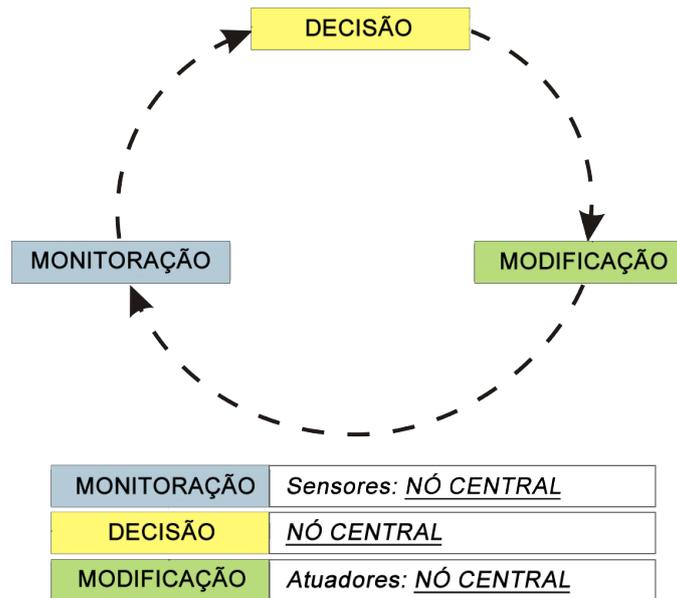


FIG. 4.1: Ciclo de Trabalho do SAAB - Esquema proposto

e o nó de destino das demandas.

4.3 IMPLEMENTAÇÃO DO SISTEMA

A implementação do sistema foi realizada da seguinte maneira: todos os computadores utilizados foram PCs que tinham em sua configuração interface de rede 10/100Mbps, processadores fabricados a partir de aproximadamente três anos, contando a partir do período de elaboração do trabalho, memória RAM acima de 1GB. Os sistemas operacionais utilizados foram o Microsoft Windows XP/Windows 7, e o Linux Ubuntu 9.10. Este último foi virtualizado dentro do Windows através de uma máquina virtual construída no VirtualBox. A opção de uso da distribuição sob uma máquina virtual se deve ao fato das diversas facilidades encontradas, tais como o *back-up* da imagem do disco virtual, a possibilidade de regressão do sistema em caso de algum problema, a portabilidade para diversas máquinas, o uso em conjunto com outro sistema operacional, como o Windows, possibilitando usar em conjunto outras ferramentas de atuação na rede disponíveis para o SO hospedeiro, a facilidade de integração com a rede, etc. O desempenho da solução nesse ambiente virtualizado pode ser questionado quanto a sua eficiência, quando comparada com a mesma solução sendo executada no Linux instalado na partição do disco rígido. Embora o desempenho não seja um objetivo primordial deste trabalho, ainda que necessário para o sucesso da finalidade proposta, a solução comportou-se de maneira satisfatória nos

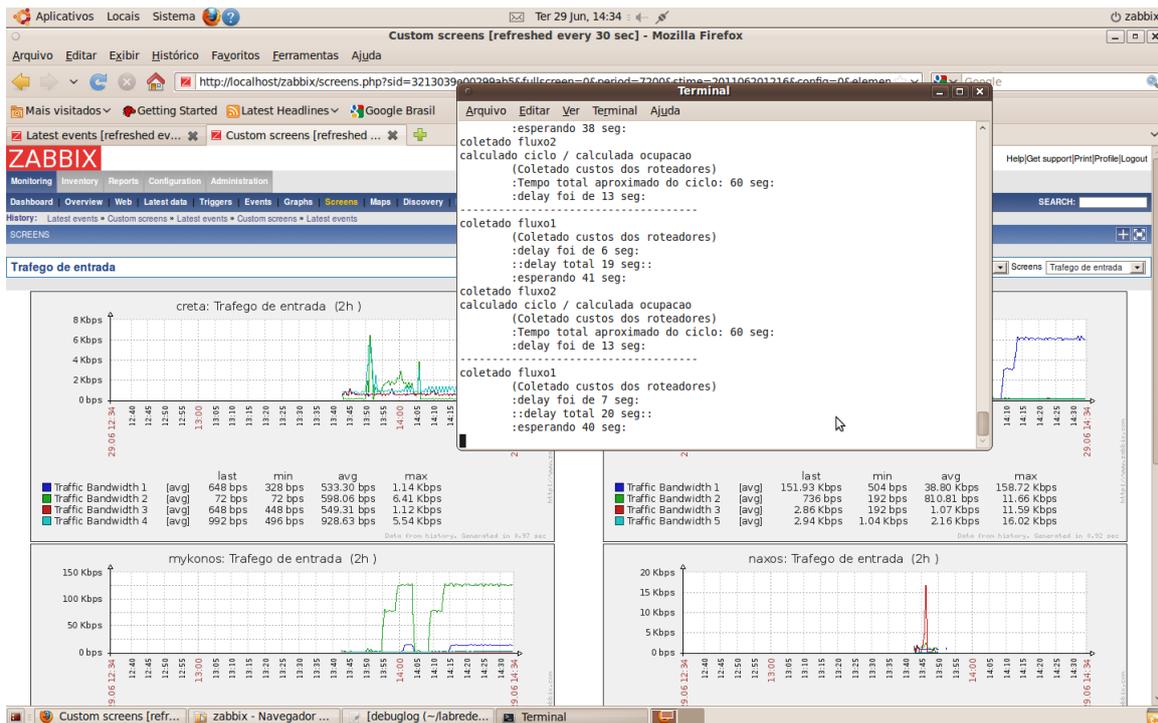


FIG. 4.2: Sistema em funcionamento

equipamentos utilizados.

No sistema operacional Linux, foram utilizadas as aplicações descritas a seguir.

- ZABBIX 1.8, que consiste de um sistema de monitoramento distribuído *open-source* capaz de monitorar a disponibilidade e performance de servidores, dispositivos de rede e aplicações. Entre suas vantagens de uso, podem ser citados: dados de monitoramento e gráficos em tempo real na mesma ferramenta; interface *web* ágil, definição de alarmes de ocorrência de problemas definidas através de expressões lógicas denominadas *triggers*; monitoramento usando protocolos de gerência, como o SNMP; estensibilidade de ações através de scripts chamadas pelas *actions*; dentre outras vantagens oferecidas para várias finalidades de uso;
- Ferramentas de rede telnet, ping, e Snmpwalk, para consultas SNMP aos elementos da árvore MIB através das OIDs;
- PHP CLI, executa códigos PHP no terminal.

Ainda sob o Linux, foi utilizada a linguagem Shellscript para a confecção dos aplicativos de coleta de valores dos objetos SNMP, das heurísticas de decisão e alteração de

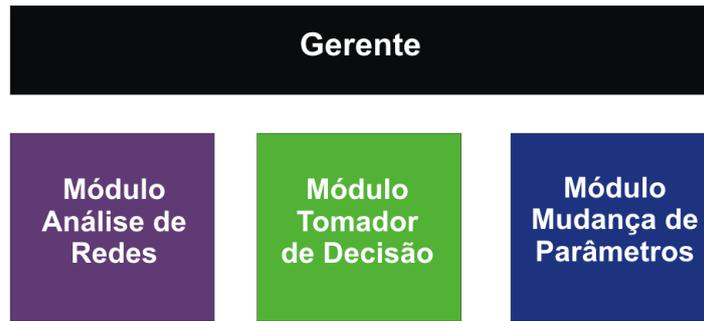


FIG. 4.3: Módulos de atuação

custos OSPF, e demais configurações da solução. A Figura 4.2 ilustra o sistema em execução no nó central.

No sistema operacional Windows, as seguintes aplicações foram usadas:

- WAN Killer, um gerador de pacotes IP sobre UDP ou TCP. Outro aplicativo do gênero pesquisado foi o Mausezahn, disponível para Linux;
- Sun Virtual Box 3.x, *software* para virtualização de diversos sistemas operacionais. Um outro aplicativo similar utilizado durante a pesquisa foi o VMWare Server 2.x, também disponível para Windows.

O ambiente do Sistema Autônomo da rede usado nos testes engloba seis roteadores Cisco, um deles sendo o *gateway* de acesso a Internet, e os *hosts*, compostos de computadores que participam das redes dos roteadores.

4.4 MÓDULOS DE ATUAÇÃO

O sistema possui um gerente, o nó central, que controla os eventos desejados que ocorrem no Sistema Autônomo. Se possível, o gerente deverá ser conectado a um enlace da topologia que possua uma menor vazão de dados, a fim de minimizar congestionamentos que possam atrapalhar o monitoramento e as ações sobre os *hosts*. Funcionalmente, esse gerente é composto por três módulos de atuação conceituais, a saber: Módulo de Análise de Redes, Módulo Tomador de Decisão e Módulo Mudança de Parâmetros. Esses módulos estão representados na figura 4.3, e seguem o esquema proposto pelo ciclo de atuação do SAAB, adaptado este trabalho.

4.4.1 MÓDULO DE ANÁLISE DE REDES

O Módulo de Análise de Redes realizará o monitoramento em tempo real de uma rede pertencente a um Sistema Autônomo e, em caso da ocorrência de algum evento que caracterize o congestionamento de pacotes de dados, irá passar as informações coletadas ao Módulo Tomador de Decisão. Isso será feito através da verificação da taxa de utilização na entrada das interfaces dos roteadores da rede entre os diversos enlaces.

Todos os roteadores devem ser previamente cadastrados no ZABBIX através do endereço IP. O nó central conhece a capacidade de todos os enlaces monitorados, e realizará o monitoramento em tempo real do tráfego passante nas interfaces dos roteadores cadastrados. A verificação do nível de ocupação dos enlaces será realizada usando a estratégia de contagem dos bytes dos pacotes que chegam por cada enlace para o cálculo da taxa de ocupação, ou seja, o tráfego *One-way*. Enquanto o autor do Sistema SAAB original propôs que essa taxa de ocupação fosse dada pela razão entre os fluxos que chegam e que partem dos enlaces, aqui nesse trabalho a medição será realizada a partir da entrada dos pacotes nas interfaces dos roteadores. Em (MAHDAVI, 1999), é dito que, para algumas aplicações, a conectividade bidirecional é considerada mais pertinente que a conectividade unidirecional, por exemplo em qualquer conexão TCP, enquanto que, para outras aplicações, a conectividade unidirecional pode possuir um uso mais interessante, como, por exemplo, para testar se um *firewall* filtra corretamente um *ping of death*. Sendo assim, a coleta dos fluxos de dados será feita unidirecionalmente na entrada das interfaces de rede, já que o congestionamento ocorre quando a taxa de chegada de pacotes excede a capacidade de serviço do recurso em um dado intervalo de tempo.

Para cada roteador cadastrado no ZABBIX, devem ser configurados os *items*. Um *item* se refere a um item individual que é monitorado em um *host*, tais como a média do tráfego ou o tempo de resposta, obtido pelo ZABBIX através de um agente SNMP ou outros meios (ZABBIX, 2008). O monitoramento dos pacotes de chegada foi realizado através de um *item* que utiliza o protocolo de gerência de rede SNMP, com o uso do objeto *ifInOctets*, que fornece o valor de dados bruto, em bytes, que entraram em uma determinada interface identificada pelo seu respectivo OID. Outros *items* cadastrados na ferramenta para a solução proposta foram o descarte de pacotes de entrada e saída nas interfaces (*ifInDiscards* e *ifOutDiscards*), e a medição de custos OSPF (*ospfMetricValue*). Além disso, a configuração de *ifInOctets* foi modificada no ZABBIX para que forneça a variação de dados de entrada que ocorreu em cada ciclo de consulta aos roteadores, ou

seja, o tráfego atual que está passando nas interfaces.

Ainda para cada roteador, foram criados os *triggers* no ZABBIX. Um *trigger* é usado para definir restrições sobre os *items*, e fornecer notificações quando essas restrições são superadas. Por exemplo, pode-se querer monitorar a média da carga em um determinado *host*, e então desejar saber quando essa carga média exceder em 1.0. *Triggers* são muito flexíveis e permitem múltiplas restrições (ZABBIX, 2008). Neste trabalho, os *triggers* foram utilizados para criar os alarmes de que a taxa de ocupação de um determinado enlace está acima do limite definido para o limite entre a situação de não-congestionamento e a de congestionamento, utilizando como base o item *ifInOctets* modificado para variação de banda passante. Quando isso ocorre, o *trigger* avisa o incidente, e o aplicativo então passará a informação do estado para o Módulo Tomador de Decisão.

Um exemplo ilustrativo de como é um *trigger* configurado para a solução proposta é mostrado a seguir:

```
router1 : ifInOctets.if1.last(0) > 1900000
```

Nesse exemplo, o status do *trigger* passará de “OK” para “Problem” toda vez que o fluxo de dados de entrada em uma determinada interface estiver acima de 90% da capacidade de um link serial com *bandwidth* e *clock rate* de 2000Kbps.

4.4.2 MÓDULO TOMADOR DE DECISÃO

O Módulo Tomador de Decisão agirá com base nos dados colhidos pelo Módulo Análise de Rede, decidindo se vale a pena mudar o custo do enlace afetado, e qual o valor de custo a ser associado a esse enlace, quando a ocupação atingir o patamar pré-estabelecido no *trigger* da interface relacionada ao enlace. Quando o Módulo Tomador de Decisão for acionado, significa que está ocorrendo uma sobrecarga em algum enlace, e o ZABBIX irá executar uma ação pré-configurada, ou *action*. Uma *action* é uma resposta tomada quando um gatilho, ou *trigger*, foi acionado, no caso, indicando que o patamar de taxa de ocupação foi excedido. As *actions* podem ser configuradas para enviar mensagens para grupos de usuários específicos definidos no ZABBIX, com base em suas configurações de tipo de mídia, ou executar comandos remotos (ZABBIX, 2008). No caso da solução proposta, será executado um comando remoto na forma de códigos, ou *scripts*, escritos na linguagem *shellscript* do Linux, que irão atuar tanto no Módulo Tomador de Decisão, quanto no Módulo Mudança de Parâmetros.

Paralelo ao funcionamento do sistema, um *script* estará periodicamente realizando a coleta e armazenando, em arquivo, as informações do tráfego atual que passa em cada uma das interfaces dos roteadores. Estas informações coletadas serão utilizadas no processo de cálculo do custo a ser inserido no enlace afetado pelo congestionamento.

Existe uma *action* correspondente para cada um dos *triggers* que monitoram as interfaces dos roteadores da topologia que compõem o AS. Cada *action* associa também o nome do *host* e o valor condição desejada do *trigger*. Por exemplo, suponha que a *action* deverá ser acionada toda vez que o *trigger* indicar que o tráfego está situado acima dos 90% da capacidade de um *link*. Assim, a configuração das condições deverá ser criada da seguinte forma:

Trigger value = "PROBLEM"

*Trigger = "Bandwidth Utilization on router1 in Range Measure 3 of Interface 2 is too
High on the ifIn"
Host = "router1"*

Essas condições, unidas pelos operadores lógicos AND e OR (para condições de grupos idênticos), acionarão o comando remoto quando estiverem totalmente validadas. Simultaneamente, a informação sobre o identificador da interface sobrecarregada é gravada em arquivo.

Como vários enlaces podem estar congestionados ao mesmo tempo, diversas instâncias do *script* são chamadas. Para garantir que apenas uma instância seja executada, o *script* bloqueia novas chamadas no momento que a primeira solicitação de execução tiver sido aceita.

Em linhas gerais, o primeiro procedimento que o *script* realiza é coletar todas as interfaces correspondentes aos enlaces sobrecarregados do arquivo, e selecionar o enlace com o maior valor de tráfego, que será então denominado como o enlace afetado. Lembrando que esse valor de tráfego é o bruto, expresso em bps, entretanto a configuração do *trigger* já leva em conta a capacidade do enlace, assim o enlace escolhido está realmente com a maior taxa de ocupação dentre os demais *links* indicados.

A seguir, inicia-se o processo de definição do valor do novo custo a ser associado ao enlace afetado. A solução proposta possui a vantagem de permitir a implementação de diversas heurísticas para o cálculo e alteração de custo, ou ainda demais heurísticas que se encaixem no contexto, bastando para isso adaptar a ferramenta ZABBIX para que

invoque o respectivo comando remoto. Isso significa que a ferramenta oferece o suporte para determinadas rotinas de auto-balanceamento de carga, no caso deste trabalho, mas a sua eficiência e sucesso depende exclusivamente de como o *script* foi elaborado, relacionado à heurística utilizada.

O último processo deste módulo é verificar se compensa mudar o custo do enlace afetado, ou seja, se a mudança realmente poderá trazer benefícios visando o balanceamento da carga. Este processo age comparando a taxa de ocupação do enlace afetado com a taxa de ocupação de cada um dos enlaces que compõem o novo caminho escolhido, calculado pelo algoritmo de Dijkstra, sendo aquele que não inclui o enlace E , inserido na demanda associada ao custo modelado escolhido que, conforme já explicitado, é o custo com o valor imediatamente maior que o valor do enlace E . Caso seja encontrado um enlace da nova rota pretendida com a taxa de ocupação maior que a taxa de ocupação do enlace E , o programa é encerrado e nenhuma modificação de custo é realizada. Caso todos os enlaces da nova rota pretendida estejam com as suas taxas de ocupação menores que a do enlace afetado, o sistema então prossegue para o Módulo Mudança de Parâmetros.

4.4.3 MÓDULO MUDANÇA DE PARÂMETROS

O Módulo Mudança de Parâmetros realizará a alteração no custo OSPF da interface do roteador correspondente ao enlace afetado pela sobrecarga, após a decisão tomada pelo Módulo Tomador de Decisão. O *script* se conectará ao roteador através de um comando Telnet e assim irá alterar o custo da interface para o valor calculado nas heurísticas de cálculo de custo. O próprio protocolo de estado de enlace se encarregará de divulgar a informação do novo peso do enlace para todos os roteadores, alterando a topologia lógica de rede se for o caso. O ciclo de trabalho do SAAB então se encerra, e está pronto para a execução de um novo ciclo.

4.5 HEURÍSTICAS ADOTADAS

O ZABBIX foi o *software* escolhido neste trabalho para atuar no contexto do Módulo de Análise de Redes, enquanto que o funcionamento em código dos algoritmos de cálculo e mudança do custo OSPF está no contexto do Módulo Tomador de Decisão e do Módulo Mudança de Parâmetros. Essa estrutura permite abrir dois grandes focos de funcionamento: o primeiro está no *software* que fará o monitoramento da situação dos enlaces da

rede, que poderá ser qualquer um que contenha os requisitos necessários ao funcionamento da solução, basicamente cadastro e coleta de objetos SNMP e gatilhos de disparo de comandos remotos perante a ocorrência de um evento (sobrecarga no enlace); enquanto que o segundo foco está no comando remoto chamado pelo gatilho da ferramenta de monitoramento. Este comando poderá ser um ou mais aplicativos, ou códigos quaisquer, desde que prossigam com o cálculo e decisão do custo OSPF, e sejam compatíveis com o tipo de processamento de chamadas do *software* de monitoramento. Quanto à questão de qual será o processo a ser executado pelo comando remoto, caberá aos algoritmos envolvidos realizar a dinâmica desse processo. Isso quer dizer que a solução abre espaço para a implementação de inúmeras heurísticas, bem como inúmeras opções de desenvolvimento das mesmas.

Como já citado na seção anterior, os algoritmos que executam as heurísticas foram desenvolvidos na linguagem *shellscript*. De acordo com o exposto no Capítulo 3, o presente trabalho contribuirá com a implementação de duas heurísticas de cálculo de valor de custo, e posterior comparação entre elas e a mais a heurística adotada pela Cisco:

- Método original proposto para o SAAB (adaptada para a solução proposta): Essa heurística é baseada no método de cálculo e alteração de custo dos enlaces do ciclo de trabalho SAAB proposto por (VIANNA, 2008).

O primeiro passo descrito no método original é a realização da modelagem dos custos associados aos caminhos por onde passam os fluxos de dados. Para isso é necessário calcular, para todas as demandas (O, D) possíveis da topologia, o valor Γ_E , que é o caminho mais curto da demanda (O, D) que obrigatoriamente inclua o enlace E , e o valor Γ_{notE} , que é o caminho mais curto da demanda (O, D) que obrigatoriamente NÃO inclua o enlace E . Esses valores são obtidos através da execução do algoritmo de Dijkstra dentro do *script*, que simulará em *off-line*, ou seja, ainda sem atuar nos roteadores, todas as melhores rotas para todas as demandas possíveis na topologia. Para a obtenção dos valores Γ_E , são selecionadas todas as demandas cujo caminho calculado pelo algoritmo de Dijkstra contenha o enlace afetado pela sobrecarga, e logo após são calculados os valores de $P'(\Gamma_E)$, ou seja, o peso do caminho Γ_E menos o peso do enlace E , isto é, $P'(\Gamma_E) = P(\Gamma_E) - P(E)$. Já para a coleta de Γ_{notE} , é atribuído ao enlace afetado E um custo que corresponde ao infinito, ou seja, esse enlace não fará mais parte de nenhuma rota. Após o cálculo de Γ_{notE} , são selecionadas todas as demandas obtidas no cálculo de Γ_E , isso quer dizer que, para

todas as demandas que tinham o enlace E constando em suas rotas, foi calculada uma nova melhor rota para as respectivas demandas, mas que não contenham o enlace E em suas rotas.

Com isso, a modelagem dos pesos será obtida pelo cálculo do “Peso do menor caminho que não inclua o enlace afetado E menos o peso do menor caminho que inclua o enlace afetado E , porém com o enlace E retirado do cálculo”, ou seja, $P(\Gamma_{notE}) - P'(\Gamma_E)$. Com a modelagem de pesos construída, podem-se conhecer quais os fluxos de dados serão afetados com a mudança de peso do enlace E .

Agora o sistema passa para o processo de escolha do valor de custo calculado na modelagem dos pesos, associado à sua respectiva demanda, que será usado para a efetivação da mudança de custo. Esse custo será aquele que tiver o valor imediatamente maior que o valor de custo do enlace afetado E , conforme foi demonstrado na Figura 3.4. A esse valor, será somado o valor 1, ocasionando a formação de uma barreira para a(s) demanda(s) modelado(s) que seguia(m) pelo caminho originalmente escolhido pelo protocolo de estado de enlace, mas que contém o enlace afetado. Desse modo, o protocolo escolherá a melhor rota calculada que não contenha o enlace E para o redirecionamento dos fluxos de dados adequados.

Além disso, o método SAAB utilizado nesse trabalho foi adaptado com a introdução de duas funcionalidades em relação ao método original: (1) a execução do comparativo simples entre as taxas de ocupação dos enlaces do novo caminho com a taxa de ocupação do enlace afetado E , pois, caso algum dos enlaces do novo caminho modelado apresentar taxa de ocupação maior, significa que ele possui uma tendência significativa a ficar sobrecarregado quando a nova demanda redirecionada for acomodada; e (2) a verificação da necessidade de realizar a divisão do vetor de pesos por um número natural. No método SAAB, os custos sempre serão aumentados, nunca diminuídos. Acontece que, em um determinado instante, a configuração dos pesos poderá estar com valores demasiados altos. Quanto maior o custo, menos fluxos tendem a passar pelo enlace. (MOY, 1998b) diz duas características importantes sobre custos OSPF: primeiro, é preferível configurar os enlaces com os menores custos possíveis; segundo, o custo de cada *link* deve ficar em uma faixa de valores compreendida entre 1 e 65.535. Assim, em um determinado instante, esses valores devem ser reduzidos, atentando para que essa ação reduza todos os valores

simultaneamente. Se todos os valores forem divididos por um número natural k , no qual k é uma constante comum a todos os números, então o novo vetor de pesos ficará proporcionalmente igual ao vetor anterior. Por exemplo, seja o vetor de pesos [3, 12, 36, 23]. Se todos eles forem divididos número natural $k = 3$, o novo vetor será [1, 4, 12, 7]. O peso com valor 23 não é divisível por 3, então, nesse caso, o resultado que vale é o número inteiro resultante da divisão arredondado para baixo. No presente trabalho, o cálculo será realizado quando algum valor de enlace ultrapassar um limite de peso preestabelecido: todos os pesos serão divididos pela constante k , previamente definida, igual 2 (dois), no momento em que o peso de algum enlace ultrapassar o limite de 10.000. Esses valores foram escolhidos por conveniência. De fato, a constante não é necessariamente divisível por todos os valores, revelando-se uma probabilidade, não estudada no escopo deste trabalho, de alteração da matriz de tráfego.

- Método Cisco modificado: Essa proposta tem o objetivo de introduzir mais uma heurística para o estudo de balanceamento de carga utilizando o protocolo de estado de enlace OSPF, e também fazer uma comparação de funcionamento e desempenho com a heurística originalmente proposta para o SAAB. O algoritmo envolvido é bem mais simples que o algoritmo SAAB. Nenhuma modelagem de custos é feita, e o cálculo do custo é realizado de maneira semelhante ao cálculo adotado pela Cisco para a atribuição dos custos, que é $C = 10^8 / bandwidth$, com a diferença que o *bandwidth* não será a capacidade total do enlace, e sim o disponível no momento. Por exemplo, se um *link* de 1000Kbps estiver com 70% da sua capacidade, ou 700000 bps, a banda disponível será de 30%, ou 300000 bps. O valor do custo será calculado seguindo a expressão $C = 10^8 / available\ bandwidth$. Desse modo, de acordo com o exemplo, o novo valor de custo será:

$$C = \frac{100000000}{300000} \approx 333$$

Da mesma maneira que o método SAAB, esse método verifica a necessidade de realização da divisão do vetor de pesos dos enlaces por um número natural ao ultrapassar um valor previamente estabelecido. A finalidade do funcionamento do método Cisco modificado é forçar o aumento do custo proporcional à diminuição da largura de banda disponível de um determinado enlace. Quanto menor a largura

de banda disponível, maior será o custo associado ao *link*. Maiores detalhes serão mostrados nas análises teórica e prática.

5 RESULTADOS

5.1 AMBIENTE DE EXPERIMENTO

O ambiente de testes utilizado foi composto pelos equipamentos existentes no Laboratório de Redes (Labredes) do Instituto Militar de Engenharia - IME, cuja descrição encontra-se a seguir.

- 6 roteadores:
 - Modelos: Cisco 1700 e 2600;
 - Sistemas operacionais: IOS C1700 Software (C1700-SV3Y-M) Version 12.1(1) RELEASE SOFTWARE (fc1); IOS C2600 Software (C2600-I-M) Version 12.2(17) RELEASE SOFTWARE (fc3).

- 5 Hosts:
 - Todos os computadores usados foram PCs equipados com interface de rede *FastEthernet*, SO Linux e/ou Windows, nas versões já citadas.

- Demais equipamentos:
 - *Switches* de rede Cisco modelo Catalyst 2950;
 - Cabos de rede para interligação dos equipamentos UTP Cat. 5e.

A Figura 5.1 mostra a topologia de rede em questão.

Os parâmetros de desempenho escolhidos para a obtenção dos resultados dos experimentos foram:

- a) Medida de congestionamento: Será representada pela quantidade de enlaces, quantidade de tempo e máximo valor de tempo que os enlaces que sofreram sobrecarga no tráfego apresentaram a taxa de ocupação maior que o valor de custo definido para representar a sobrecarga, dentro do intervalo de tempo do experimento. O cenário de experimento que utilize a heurística que ocasione a maior quantidade de enlaces

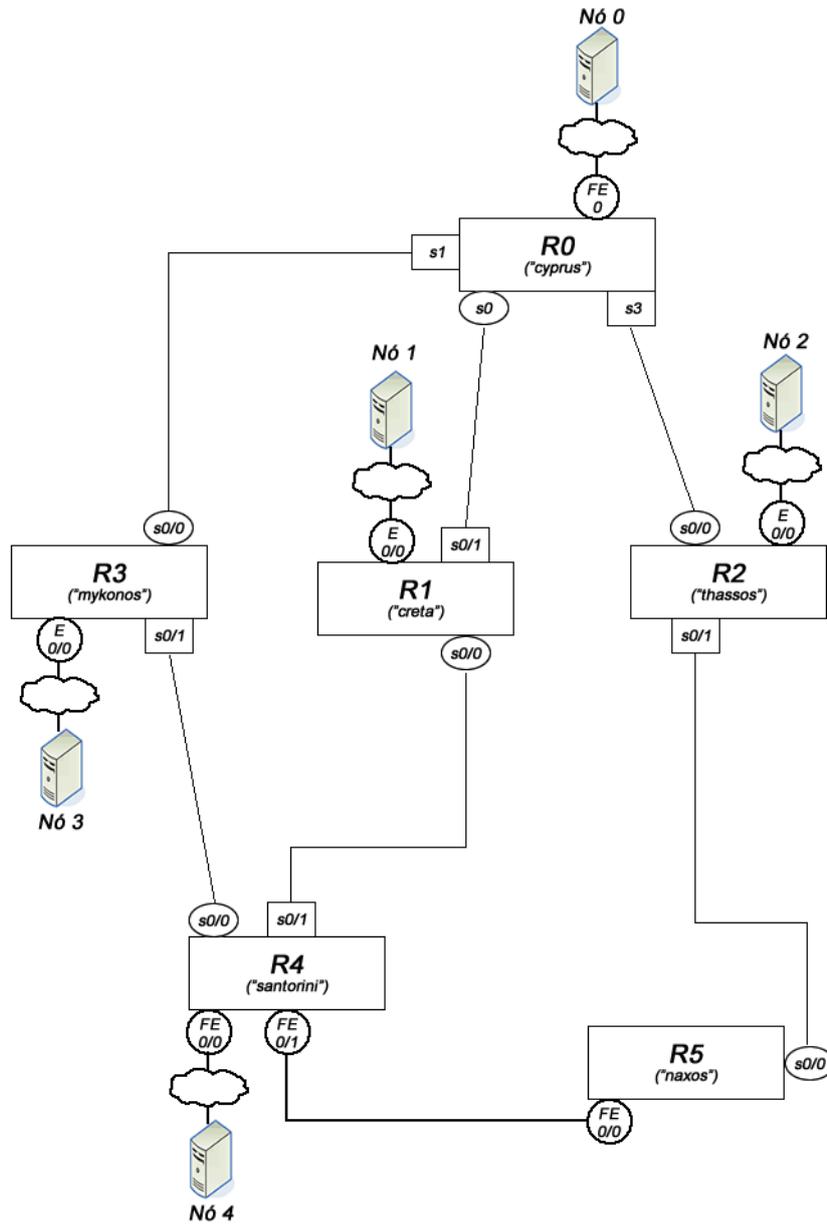


FIG. 5.1: Topología de red de los experimentos

apresentando os maiores valores da taxa de ocupação acima do limite considerado como sobrecarga, por mais tempo, é dito o mais congestionado em comparação aos demais cenários com o uso de outras heurísticas. Quanto maiores os valores encontrados, mais congestionados estarão os respectivos enlaces, e pior estará a situação da rede.

- b) Medida de desempenho para o usuário: Obtida através das taxas de descarte na saída das interfaces dos enlaces da topologia, quando há o congestionamento no enlace da rede. O descarte de pacotes ocorre quando a taxa de pacotes roteados para uma determinada interface é maior do que a capacidade suportada na fila de saída da mesma, e então a interface começa a descartar esses pacotes não acomodados pela fila. Quanto maior o valor do descarte, pior será o desempenho da rede, o que afetará a experiência do usuário na utilização dos aplicativos que usam a rede.
- c) Medida de balanceamento de carga: Obtida através do cálculo do desvio padrão das taxas de ocupação na entrada das interfaces dos enlaces da topologia em cada instante definido para o experimento, e o posterior cálculo da média dos desvios, para as heurísticas utilizadas. O desvio padrão da taxa de ocupação dos enlaces indica o grau de dispersão da taxa de ocupação em torno da média. Quanto menor o desvio padrão, melhor será o balanceamento da carga na rede, e vice-versa. A comparação dos valores do desvio padrão obtidos no experimento para cada um dos algoritmos utilizados será função do tempo. Já a média dos desvios é a soma de todos os desvios padrões gerados pelo algoritmo durante o experimento dividido pelo tempo total do experimento. A soma de todos os desvios padrões gerados no experimento corresponde à integral da função do desvio padrão durante o tempo do teste. Portanto, o desvio padrão médio é a razão entre a integral da função desvio padrão e o tempo do teste. A integral da função desvio padrão representa a área entre a curva do desvio padrão e o eixo das abscissas que é o eixo do tempo. O algoritmo que apresentar menor desvio padrão médio é dito mais eficiente no balanceamento da carga (VIANNA, 2008).

Quando comparadas as médias entre os resultados dos cenários de testes, quanto maior o valor, pior será o balanceamento de carga na rede.

5.2 EXPERIMENTO 1 E VALIDAÇÃO DA PROPOSTA

O primeiro experimento consistiu em submeter a solução desenvolvida em um cenário de teste controlado, com a finalidade de comprovar o seu adequado funcionamento, perante o comportamento teórico modelado e conhecido, no decorrer dos eventos da rede que caracterizam a sobrecarga na rede e propiciam a aplicação do auto-balanceamento de carga.

Nesse e nos demais experimentos aqui realizados, os custos de todos os enlaces foram inicialmente configurados com o valor de custo recomendado pela Cisco (inverso da capacidade do enlace), bem como as demandas foram configuradas para serem disparadas de uma origem para um destino, cuja notação é (O, D) ou $O \rightarrow D$. As próximas subseções apresentarão o estudo detalhado e a posterior comparação entre o comportamento teórico e o comportamento prático do cenário elaborado.

5.2.1 ANÁLISE TEÓRICA

O cenário do primeiro experimento segue a topologia mostrada na Figura 5.2, e os fluxos mostrados na Figura 5.3 e na Tabela 5.1.

Instantes(min)	Evento
0	Iniciar coleta de dados
1	Configurar demanda: Nó #4 \rightarrow Nó #3; 85% de 1000kbps [850Kbps] Configurar demanda: Nó #3 \rightarrow Nó #0; 75% de 1000kbps [750Kbps]
5	Configurar demanda: Nó #2 \rightarrow Nó #0; 75% de 2000Kbps [1500Kbps] Configurar demanda: Nó #4 \rightarrow Nó #2; 85% de 2000Kbps [1700Kbps]
10	Configurar demanda: Nó #4 \rightarrow Nó #0; 20% de 2000Kbps [400Kbps]

TAB. 5.1: Lista de eventos do teste de validação

Os enlaces da topologia são inicialmente configurados como o mostrado na figura 5.2: Os enlaces seriais $R0 \Leftrightarrow R3$, $R4 \Leftrightarrow R3$ e $R1 \Leftrightarrow R0$ possuem suas larguras de banda iguais a 1000Kbps, conseqüentemente com custo associado aos *links* igual a 100, seguindo a regra

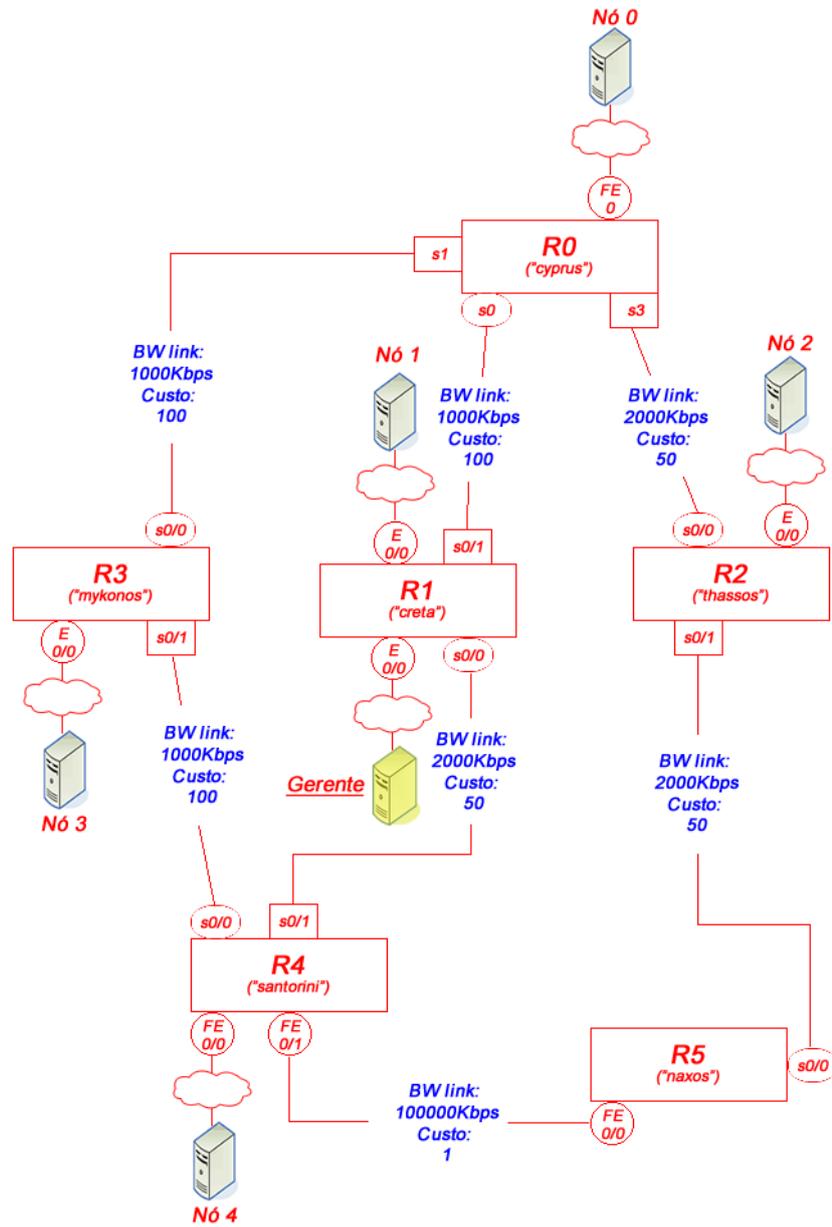


FIG. 5.2: Topologia de teste para a validação da solução proposta

da Cisco que é, lembrando, igual a $C = 10^8/\text{bandwidth}$, enquanto que os enlaces seriais $R4 \Leftrightarrow R1$, $R2 \Leftrightarrow R0$ e $R5 \Leftrightarrow R2$ possuem suas larguras de banda iguais a 2000Kbps e custo 50. Por fim, o enlace *FastEthernet* $R4 \Leftrightarrow R5$ possui largura de banda 100000Kbps com consequente custo associado igual a 1.

Os nós constituíram-se de PCs conectados na rede. Os fluxos foram enviados de *host* para *host*, a princípio seguindo os caminhos resultantes do algoritmo de Dijkstra, que obtém o melhor caminho a ser seguido pelos fluxos passantes na topologia configurada com o protocolo OSPF, baseado nos custos dos enlaces recomendados pela Cisco, associados aos caminhos. Por exemplo, a demanda (Nó #4, Nó #2) seguirá o caminho $R4 \rightarrow R5 \rightarrow R2$, pois de acordo com o cálculo do melhor caminho realizado pelo algoritmo de Dijkstra, em comparação com os outros caminhos possíveis, essa é a rota com o menor custo associado, já que $1 + 50 = 51$. Os outros caminhos possíveis, nessa circunstância, são $R4 \rightarrow R3 \rightarrow R0 \rightarrow R2$, com custo $100 + 100 + 50 = 250$, e $R4 \rightarrow R1 \rightarrow R0 \rightarrow R2$, com custo $50 + 100 + 50 = 200$.

Os enlaces que terão os custos modificados e considerados no cálculo do novo valor são aqueles cujas interfaces fazem a conexão entre roteadores. O custo das interfaces que conectam os roteadores aos *hosts* não afetam as demandas, pois são enlaces que se ligam a nós folha, possuindo apenas um caminho para o tráfego das demandas.

O objetivo deste experimento é: realizar o auto-balanceamento de carga quando um determinado enlace estiver com a taxa de ocupação na entrada da interface maior que 90% da sua capacidade. Os instantes 1 min e 5 min inicializam demandas nó a nó que não atingem esse percentual, e consequentemente nada acontece no sentido de alterar o custo de algum enlace. O instante 10 min inicializa a demanda Nó #4 \rightarrow Nó #0 com taxa de ocupação de 20%, que encontra as demandas Nó #4 \rightarrow Nó #2 e Nó #2 \rightarrow Nó #0, com 85% e 75% de taxa de ocupação, respectivamente, nos seus enlaces seriais (como o enlace $R4 \Leftrightarrow R5$ é *FastEthernet*, terá sua taxa de ocupação irrelevante para esse estudo de caso). Logo, os respectivos enlaces ficarão com 100% (105% somados) e 95%, o que aciona o sistema de cálculo e mudança de custo.

Como o enlace mais afetado é o (R5, R2), a alteração do custo será feita em cima deste enlace, através da utilização de alguma das heurísticas de mudança de custo propostas. Dessa maneira, é respeitada a proposta feita por (VIANNA, 2008): alterar o custo de apenas um enlace por vez. Deseja-se que a demanda Nó #4 \rightarrow Nó #0 mude o seu caminho, seguindo então pela nova rota $R4 \rightarrow R1 \rightarrow R0$, que está com os seus enlaces

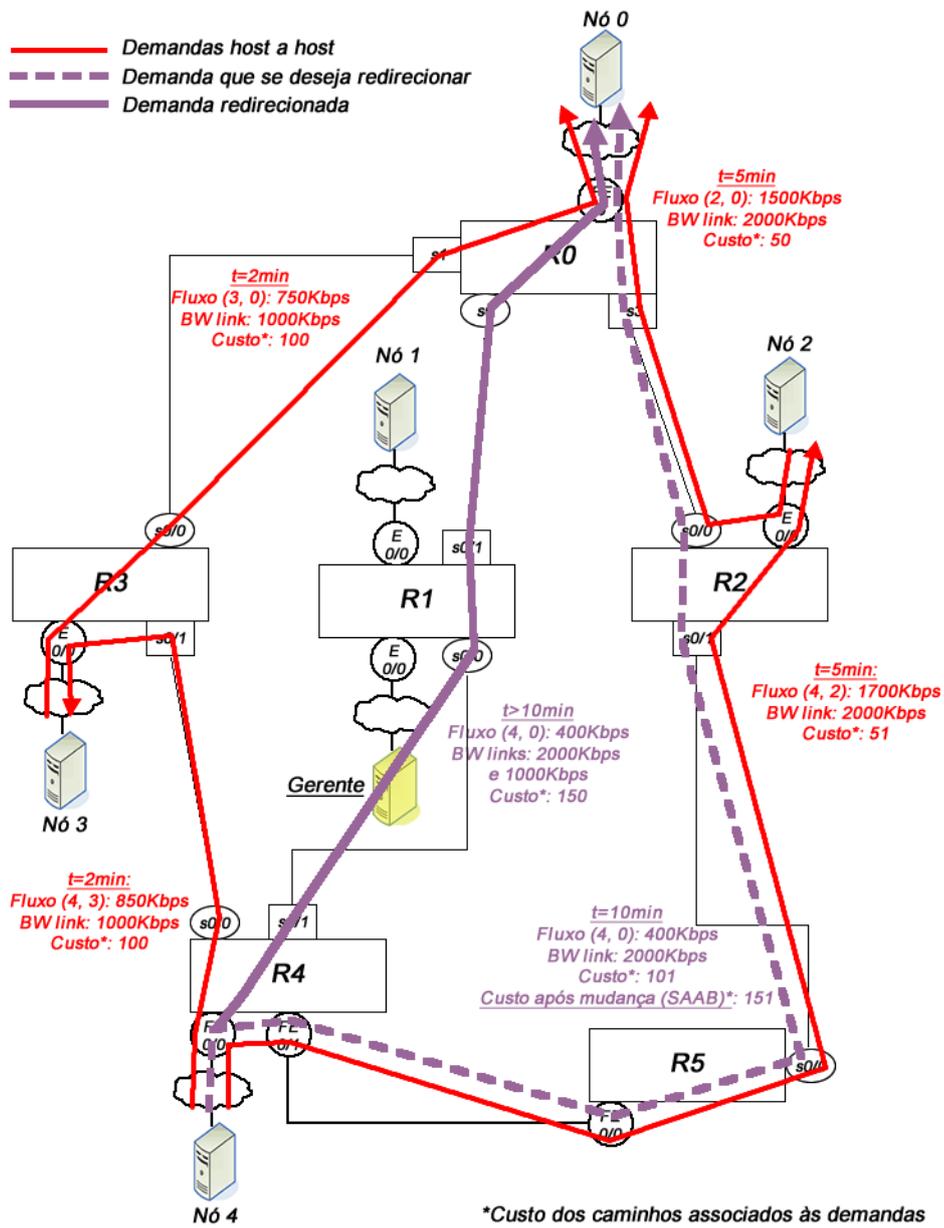


FIG. 5.3: Cenário de teste para a validação da solução proposta

totalmente ociosos quanto à sua taxa de ocupação, ao invés do caminho inicialmente escolhido $R4 \rightarrow R5 \rightarrow R2 \rightarrow R0$, que está sobrecarregado, inclusive com a ocorrência de descarte de pacotes de dados.

Segue inicialmente a demonstração do cálculo seguindo a heurística do método original proposto para o SAAB, adaptada para a solução proposta. Neste exemplo, a origem e o destino de todos os fluxos já são conhecidos, logo o cálculo levará este fator em conta. O primeiro passo é a realização do cálculo da modelagem dos custos associados às demandas que estão passando pelo enlace afetado E , para então averiguar qual demanda realmente terá seu caminho alterado na topologia. Isso é realizado com o cálculo de Γ_E , que é o caminho mais curto das demandas $Nó \#4 \rightarrow Nó \#0$ e $Nó \#4 \rightarrow Nó \#2$ que obrigatoriamente incluam o enlace E , e o valor Γ_{notE} , que é o caminho mais curto das demandas $Nó \#4 \rightarrow Nó \#0$ e $Nó \#4 \rightarrow Nó \#2$ que obrigatoriamente não incluam o enlace E , sucedido do cálculo da modelagem do peso do enlace E como uma barreira para os fluxos, que é constituído por $P(\Gamma_{notE}) - P'(\Gamma_E)$. Outras possíveis demandas que podem passar pelo enlace E foram suprimidas aqui.

- Cálculo de Γ_E :

- Demanda (4, 0): Melhor caminho com o enlace $E : R4 \rightarrow R5 \rightarrow R2 \rightarrow R0 \implies 1 + 50 + 50 = 101$

Tirando o enlace E : $101 - 50 = 51$

- Demanda (4, 2): Melhor caminho com o enlace $E : R4 \rightarrow R5 \rightarrow R2 \implies 1 + 50 = 51$

Tirando o enlace E : $51 - 50 = 1$

- Cálculo de Γ_{notE} :

- Demanda (4, 0): Melhor caminho sem o enlace $E : R4 \rightarrow R1 \rightarrow R0 \implies 50 + 100 = 150$

- Demanda (4, 2): Melhor caminho sem o enlace $E : R4 \rightarrow R1 \rightarrow R0 \rightarrow R2 \implies 50 + 100 + 50 = 200$

- Modelagem dos caminhos $P(\Gamma_{notE}) - P'(\Gamma_E)$:

- Demanda (4, 0): $150 - 51 = 99$

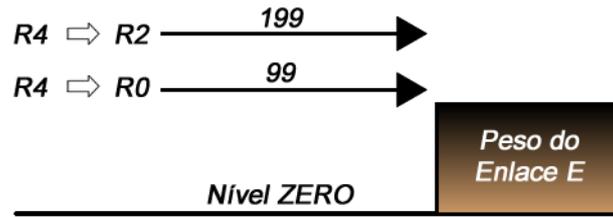


FIG. 5.4: Modelagem do Peso do Enlace E como uma Barreira

– Demanda (4, 2): $200 - 1 = 199$

De acordo com a modelagem dos pesos do enlace E como uma barreira para os fluxos mostrada na Figura 5.4, o primeiro custo modelado, associado à demanda, que possui o custo maior que o do enlace afetado é $(R4, R0)$, logo essa é a demanda que sofrerá a alteração no caminho a ser seguido. Essa mudança ocorre quando o custo do enlace E for alterado para o custo modelado mais um, ou seja, $99 + 1 = 100$. Com isso, o enlace afetado funcionará como uma barreira para os fluxos, pois o custo associado ao caminho $R4 \rightarrow R5 \rightarrow R2 \rightarrow R0$ (151) passará a ser maior que o do novo melhor caminho a ser seguido, que é o $R4 \rightarrow R1 \rightarrow R0$ (150), já que outro caminho possível também tem o custo maior, $R4 \rightarrow R3 \rightarrow R0$ (200).

O último passo é comparar as taxas de ocupação dos enlaces que compõem o novo caminho com a taxa de ocupação do enlace afetado E . Isso é importante porque, se o redirecionamento do fluxo for acomodado por um enlace que já estiver com o valor de sua ocupação na entrada maior que a do enlace afetado, poderá ocasionar uma nova sobrecarga, inclusive com descarte de dados.

(4, 1): 0%
(1, 0): 0%

TAB. 5.2: Comparação da taxa de ocupação dos enlaces do caminho a ser seguido

Pela análise dos valores mostrados na Tabela 5.2, não existe nenhum enlace pertencente ao melhor caminho proposto, $R4 \rightarrow R1 \rightarrow R0$, com a taxa de ocupação superior ao enlace E . Os enlaces, inclusive, encontram-se ociosos (sem a passagem de nenhuma demanda). Assim, as ações pertencentes ao Módulo de Análise de Redes e ao Módulo Tomador de Decisão foram concluídas com êxito, e o processo pode finalmente migrar para o Módulo Mudança de Parâmetros, realizando a alteração do custo nas interfaces dos roteadores que compõem o enlace E .

5.2.2 RESULTADOS DA ANÁLISE TEÓRICA

Após a mudança efetiva, os parâmetros de desempenho podem ser calculados para a composição do resultado.

- a) Medida de congestionamento (quanto menor, melhor): Para este cálculo teórico, será feita a aproximação do tempo na qual os enlaces sobrecarregados estão com a taxa de ocupação maior que o valor de custo definido para representar a sobrecarga, que neste trabalho é de 90%, dentro do intervalo de tempo do experimento, no caso 30 minutos. Foram considerados na colocação dos valores de taxa de ocupação a tolerância de tempo de 1 minuto para atingir o valor nominal após o fluxo ter sido disparado e após a mudança de custo ter sido efetivada, e 3 minutos para ocorrência da mudança do custo após o início da sobrecarga do enlace e a sua detecção pelo nó gerente.

Como a demanda (4, 0) será disparada após o instante 10min, seguindo pelo caminho $R4 \rightarrow R5 \rightarrow R2 \rightarrow R0$, conforme já demonstrado, os congestionamentos deverão ocorrer primeiramente no enlace $R5 \rightarrow R2$, com taxa de ocupação 100% (105% combinado) a 2000Kbps de *bandwidth*, e logo após no enlace $R2 \rightarrow R0$, com taxa de ocupação 95% a 2000Kbps de *bandwidth*. Como o enlace $R4 \rightarrow R5$ possui um *bandwidth* de 100000Kbps, este não sofrerá sobrecarga com os valores de taxa de ocupação resultantes das demandas (4, 5) e (5, 2) disparadas, passantes por esse enlace com taxa de ocupação de 1%. Após o término do teste, os seguintes enlaces e valores de tempo e máximo valor de tempo de congestionamento dentre todos os enlaces, acima de 90%, deverão ser encontrados:

- Cisco:
 - Enlace $R5 \rightarrow R2$: aprox. 20min
 - Enlace $R2 \rightarrow R0$: aprox. 19min a 20min
 - Tempo máximo:** 20min
- SAAB:
 - Enlace $R5 \rightarrow R2$: aprox. 3min
 - Enlace $R2 \rightarrow R0$: aprox. 2min a 3min
 - Tempo máximo:** 3min

Pelos valores colocados, é visto que a heurística SAAB leva vantagem sobre a Cisco, pois enquanto a primeira apresenta o tempo de congestionamento proporcional ao tempo de execução de todos os módulos de atuação, desde a detecção, passando pelo cálculo do valor de custo até a posterior alteração nas interfaces dos roteadores do enlace afetado, a segunda possui o tempo de congestionamento proporcional ao tempo restante até o término do experimento, após a demanda (4, 0) ter sido disparada.

b) Medida de desempenho para o usuário (quanto menor, melhor): Neste teste, é esperado que a solução proposta reduza o descarte de pacotes a zero, assim que este evento começar a acontecer, em virtude da demanda (4, 0) de 20% ter sido disparada, pois irá ao encontro da demanda (5, 2) de 85%, somando 105% de taxa de utilização nominal. Como não é possível ter um percentual de taxa de utilização acima de 100%, começará a haver descarte de pacotes de aproximadamente $105 - 100 = 5\%$, considerando a variação da passagem do tráfego de dados, na saída da interface do roteador 2. A quantidade de pacotes descartados segue o seguinte cálculo:

Seja,

- Tamanho do pacote IP disparado: 1500 bytes, ou $1500 \times 8 = 12000 \text{ bits}$
- *Bandwidth* do enlace afetado: 2000 Kbps, ou 2000000 *bps*

A quantidade de pacotes total *one-way*, que pode passar pelo enlace, é:

$$2000000/12000 \approx 167 \text{ pacotes}$$

Fazendo a regra de três:

$$\begin{array}{l} 100\% \text{ --- } 167 \text{ pacotes} \\ 105\% \text{ --- } x \text{ pacotes} \end{array} \Rightarrow x \approx 175 \text{ pacotes}$$

Assim, $175 - 167 \approx 8 \text{ pacotes}$, serão descartados na interface de saída do roteador 5.

No teste realizado com a heurística Cisco, este descarte ocorrerá continuamente, pois nenhum custo será modificado a ponto de desviar a demanda que ocasionou o evento. Já na heurística SAAB, o descarte ocorrerá até que haja a mudança do custo do enlace $R5 \Leftrightarrow R2$.

c) Medida de balanceamento de carga (quanto menor, melhor): O cálculo do desvio padrão para todos os instantes do teste não é necessário na análise teórica. Já que o experimento foi realizado com as demandas em tráfego constante (*Constant Bit Rate* - CBR), o que importa nesse caso é a obtenção do desvio padrão da taxa de ocupação dos enlaces da topologia em qualquer um dos instantes do intervalo considerado, compreendido entre 15min e 30min, após a alteração do custo do enlace afetado pela sobrecarga. As heurísticas que trabalham com alteração do custo agirão após o disparo da última demanda preparada para o experimento. Já o cenário que utiliza a heurística Cisco não terá alterações nos valores de custo, assim o mesmo cálculo do desvio padrão pode ser feito em qualquer um dos instantes que sucedem o disparo do último fluxo programado no experimento, mas para efeitos de cálculo também será considerado o intervalo entre 15min e 30min.

A equação do desvio padrão é a seguinte:

$$\sigma = \sqrt{\sum \frac{(x_i - \bar{x})^2}{(n-1)}}$$

- Cálculo de σ para a heurística Cisco, no intervalo entre 15 min e 30 min:

- Taxas de ocupação dos enlaces no intervalo.

$$R4 \Rightarrow R5: 1\%$$

$$R5 \Rightarrow R2: 100\%$$

$$R2 \Rightarrow R0: 95\%$$

$$R4 \Rightarrow R3: 85\%$$

$$R3 \Rightarrow R0: 75\%$$

$$R4 \Rightarrow R1: 0\%$$

$$R1 \Rightarrow R0: 0\%$$

- Cálculo da média das ocupações (\bar{x}):

$$\bar{x} = \frac{(1+100+95+85+75+0+0)}{7} = 50,85$$

- Cálculo de cada uma das ocupações menos a média ao quadrado ($(x_i - \bar{x})^2$):

$$(1 - 50,85)^2 = 2485,02$$

$$(100 - 50,85)^2 = 2415,72$$

$$(95 - 50,85)^2 = 1949,22$$

$$(85 - 50,85)^2 = 1166,22$$

$$(0 - 50,85)^2 = 2585,72$$

$$(75 - 50,85)^2 = 583,22$$

– Cálculo de desvio padrão (σ):

$$\sigma = \sqrt{\frac{2485,02+2415,72+1949,22+1166,22+583,22+2585,72}{7-1}} \approx \sqrt{2295,14} \approx 47,9$$

• Cálculo de σ para a heurística SAAB original, no intervalo entre 15min e 30min:

– Taxas de ocupação dos enlaces no intervalo.

$$R4 \Rightarrow R5: 1\%$$

$$R5 \Rightarrow R2: 85\%$$

$$R2 \Rightarrow R0: 75\%$$

$$R4 \Rightarrow R3: 85\%$$

$$R3 \Rightarrow R0: 75\%$$

$$R4 \Rightarrow R1: 20\%$$

$$R1 \Rightarrow R0: 40\%$$

– Cálculo da média das ocupações (\bar{x}):

$$\bar{x} = \frac{(1+85+75+85+75+20+40)}{7} = 54,42$$

– Cálculo de cada uma das ocupações menos a média ao quadrado ($(x_i - \bar{x})^2$):

$$(1 - 54,42)^2 = 2853,69$$

$$(85 - 54,42)^2 = 935,13$$

$$(75 - 54,42)^2 = 423,53$$

$$(20 - 54,42)^2 = 1184,73$$

$$(40 - 54,42)^2 = 207,93$$

– Cálculo de desvio padrão (σ):

$$\sigma = \sqrt{\frac{2853,69+935,13+423,53+935,13+423,53+1184,73+207,93}{7-1}} \approx \sqrt{1160,61} \approx 34,06$$

Segue agora a demonstração do cálculo seguindo a heurística Cisco modificado. Aqui o processo é mais simples que o processo de cálculo da heurística SAAB. Quando o enlace (5, 2) estiver sobrecarregado, a sua taxa de ocupação estará muito próxima ou igual ao limite de acomodação de dados da largura de banda total do enlace. Conseqüentemente, a largura de banda **disponível** estará tendendo ao mínimo, até chegar ao valor zero. Neste experimento, a taxa de ocupação no enlace afetado E coletada no ciclo de medição

teoricamente estará em 100%, logo restará 0% de banda disponível e 0 Kb de dados coletados no ciclo. Se o cálculo for realizado seguindo a expressão do método, tem-se:

$$C = \frac{100000000}{0}$$

Como não é possível dividir por zero, o método propõe, nesse caso, adotar o valor de taxa de ocupação de 99%. Antes de calcular o custo, deve-se converter a taxa de ocupação percentual na largura de banda passante em bps coletada no ciclo, conforme a expressão:

$$\text{poll bandwidth} = \frac{\text{poll ocup} \times \text{link bandwidth}}{100}$$

Logo,

$$\text{poll bandwidth} = \frac{99 \times 2000000}{100} = 1980000 \text{bps}$$

E o cálculo do custo será:

- Largura de banda comprometida: 1998000 \Rightarrow Largura de banda disponível: 2000000 – 1998000 = 20000

$$C = \frac{100000000}{20000} \approx 5000$$

Apesar desse processo ser mais simples que o SAAB, realizar o cálculo teórico com precisão é mais complicado. Cabe lembrar que a taxa de ocupação na prática pode variar conforme o funcionamento normal dos equipamentos, assim os valores aqui expressos foram escolhidos por consideração. Como se pode ver, o cálculo para esse método não leva em conta a modelagem de custos dos enlaces associados às demandas, como é feito no método SAAB. A vantagem desse método está na simplicidade do cálculo, em contrapartida, o custo é associado apenas à situação do enlace afetado, não levando em conta a situação dos demais enlaces da topologia envolvidos no contexto dos fluxos passantes e dos seus custos associados. Isso pode levar à alterações de custos para números demasiados elevados, conforme visto no exemplo demonstrado, e em determinado momento, a redistribuição dos fluxos poderá não levar ao balanceamento de carga desejado, enquanto o SAAB permite conhecer precisamente a redistribuição de uma ou mais demanda, e por qual(is) caminho(s) ela(s) seguir(ão).

Evidentemente que, conforme a explicação do funcionamento dos custos associados às demandas passantes nos enlaces, vista na teoria do método SAAB, esse valor de custo calculado fará com que não apenas a demanda (4, 0), com custo 99, seja redistribuída para o caminho $R4 \rightarrow R1 \rightarrow R0$, como também a demanda (4, 2), com custo 199. A consequência disso é que haverá uma nova sobrecarga de dados nos enlaces (4, 1) e (1, 0), pois a taxa de ocupação redistribuída foi de 85% do *link* (4, 2) e 20% do *link* (4, 0), que somados resulta em 105%.

Assim, o módulo análise de dados, configurado com o *trigger* a ser acionado quando o tráfego ultrapassar 90% da largura de banda de um enlace, terá que realizar uma nova chamada ao módulo tomador de decisão. Essa aleatoriedade no cálculo dos custos a refletir no redirecionamento das demandas é justificada pela simplicidade do cálculo já comentada. Devido a esse fato, a análise dos resultados será realizada na análise prática.

5.2.3 ANÁLISE PRÁTICA

Uma vez realizada a análise teórica do cenário de teste proposto para a validação da solução desenvolvida, deseja-se verificar se esta possui o funcionamento prático equivalente ao funcionamento teórico. Isso quer dizer que, na prática, todos os passos efetuados pelos três módulos pertencentes ao sistema deverão ser idênticos aos passos demonstrados na Subseção 5.3.1.

A preparação do teste foi constituída pelas seguintes providências:

- 5 (cinco) computadores *hosts* conectados à topologia para o envio e recebimento das demandas: cada um dos cinco *hosts*, com seus endereços IP devidamente configurados, estão conectados a cada um dos roteadores que provêem acesso às suas respectivas redes (R0, R1, R2, R3, R4);
- 1 (um) computador *host*, com seu endereço IP devidamente configurado, conectado à topologia para a execução do gerente do sistema (nó central);
- No nó central, execução do *script* de *reset* dos valores de custo, de *bandwidth*, e de *clock rate* dos roteadores, dos *logs* e das demais configurações dos *scripts*, para os valores desejados para o experimento em questão;
- Escolha da heurística de mudança de custo desejada para o experimento;

- Preparação do aplicativo de disparo dos tráfegos ponto a ponto com a configuração dos endereços IP dos destinos e a vazão de dados desejada. O tráfego é gerado ajustando o percentual de vazão de dados desejado em função da largura de banda informada, que conseqüentemente define uma quantidade de pacotes enviados. O tipo de pacote utilizado nos testes foi o UDP sobre o IP;
- Inicialização do *software* ZABBIX para acompanhamento dos eventos e coleta dos gráficos de fluxos de dados, descarte de pacotes, e custos das interfaces;
- Ativação, no *software* ZABBIX, das *actions* desejadas para execução no caso do alarme emitido pelos *triggers*. Neste exemplo, foram ativadas as ações que são chamadas no momento em que o tráfego nas interfaces ultrapasse 90% do fluxo real de dados dos enlaces configurados com a largura de banda de 2000Kbps, ou seja, quando uma determinada interface ultrapassar o valor de 1900Kbps, o *trigger* passará do status “OK” para “Problem”, e fará com que a *action* correspondente ao *host* e à interface afetada chame o *script* de definição e alteração de custos.

No instante 0, foi iniciada a execução da aplicação feita em *script* de coleta periódica do tráfego de entrada, taxa de ocupação e valores de custo das interfaces dos roteadores envolvidos, denominada IFINOCKETS.sh. Este aplicativo coleta as informações diretamente dos roteadores via SNMP, através das variáveis MIB *ifInOctets* e *ospfMetricValue*. A leitura de *ifInOctets* fornece a quantidade bruta de octetos que já passaram pela interface do roteador. Para a obtenção do tráfego de entrada em um determinado intervalo de tempo, a aplicação foi dividida em dois *scripts*, onde o primeiro realiza a coleta dos valores em um tempo denominado 1, aguarda um intervalo de tempo x definido pelo usuário, e chama o segundo, que realiza a nova coleta no tempo 2. A diferença entre a última e a primeira coleta fornece o tráfego de entrada no intervalo de tempo. Após essa coleta, o aplicativo calcula a taxa de ocupação de entrada na interface, que é o percentual de uso em função da largura de banda configurada para a interface pertencente ao enlace. Para este cálculo, considerar:

- $\Delta ifInOctets$: A diferença (ou Δ) entre o intervalo de duas coletas dos valores do objeto SNMP *ifInOctets*, que representa a contagem de octetos que entraram;
- *ifSpeed*: A velocidade da interface, como reportado no objeto SNMP *ifSpeed*

O cálculo é realizado através da seguinte equação (CISCO, 2007a):

$$Utilizacao\ na\ entrada = \frac{\Delta ifInOctets \times 8 \times 100}{(numero\ de\ segundos\ em\ \Delta) \times ifSpeed}$$

A configuração e coleta dos valores do objeto *ifSpeed* previamente definidos é realizada durante a execução do aplicativo de *reset* das configurações da ferramenta.

Dois detalhes importantes a respeito da aplicação de coleta de dados: o primeiro é a respeito do atraso que ocorre no momento da coleta das informações, pois o nó central envia a solicitação de leitura dos valores dos objetos SNMP e aguarda o retorno dos mesmos. Esse período entre solicitação e resposta pode variar dependendo de algumas situações, tais como o congestionamento nos *links*, a sobrecarga da CPU no nó central, etc. Devido a este fato, o *script* é dotado de um controle de compensação dos atrasos nas leituras, para que o intervalo real de leitura praticamente não se diferencie do intervalo de leitura definido pelo usuário. Para este teste, o intervalo de tempo desejado para a coleta dos dados foi de 60 segundos. O segundo detalhe está na impossibilidade de coleta, quando algum *link* cair ou ficar sobrecarregado. O *script* age nesse caso comparando o tempo gasto na coleta com o tempo de leitura definido pelo usuário. Se durante a leitura realizada no tempo 1 o tempo de leitura exceder o tempo pré-definido, os valores da última coleta bem sucedida de taxa de ocupação e demanda de dados serão repetidos, e a aplicação retorna para a medição do tempo 1. Já no tempo 2, o cálculo será realizado mesmo no caso de algum valor não tiver sido coletado (esse ficará de fora após o *timeout*), e a aplicação reajusta a periodicidade do tempo de coleta baseada no atraso da medição.

No instante 1, duas demandas foram disparadas, Nó #4 → Nó #3, 85% de 1000kbps (850Kbps); e Nó #3 → Nó #0, 75% de 1000kbps (750Kbps). Conforme esperado, elas seguiram pelos caminhos $R4 \rightarrow R3$ e $R3 \rightarrow R0$, respectivamente. O mesmo ocorreu no instante 5, com as demandas Nó #2 → Nó #0, 75% de 2000Kbps (1500Kbps); e Nó #4 → Nó #2, 85% de 2000Kbps (1700Kbps), que seguiram, respectivamente, pelos caminhos $R2 \rightarrow R0$ e $R4 \rightarrow R5 \rightarrow R2$.

Ao chegar no instante 10, foi disparada a demanda Nó #4 → Nó #0, 20% de 2000Kbps (400Kbps). De acordo com o passo demonstrado na análise teórica, o caminho escolhido pelo protocolo OSPF, devido à configuração prévia dos custos dos enlaces seguindo a recomendação Cisco, foi o $R4 \rightarrow R5 \rightarrow R2 \rightarrow R0$. Isso ocasionou a sobrecarga total de dados no enlace (5, 2) e a chamada das *actions* para os enlaces (5, 2) e (2, 0), por ultrapassarem o limite definido de 90% do tráfego de chegada.

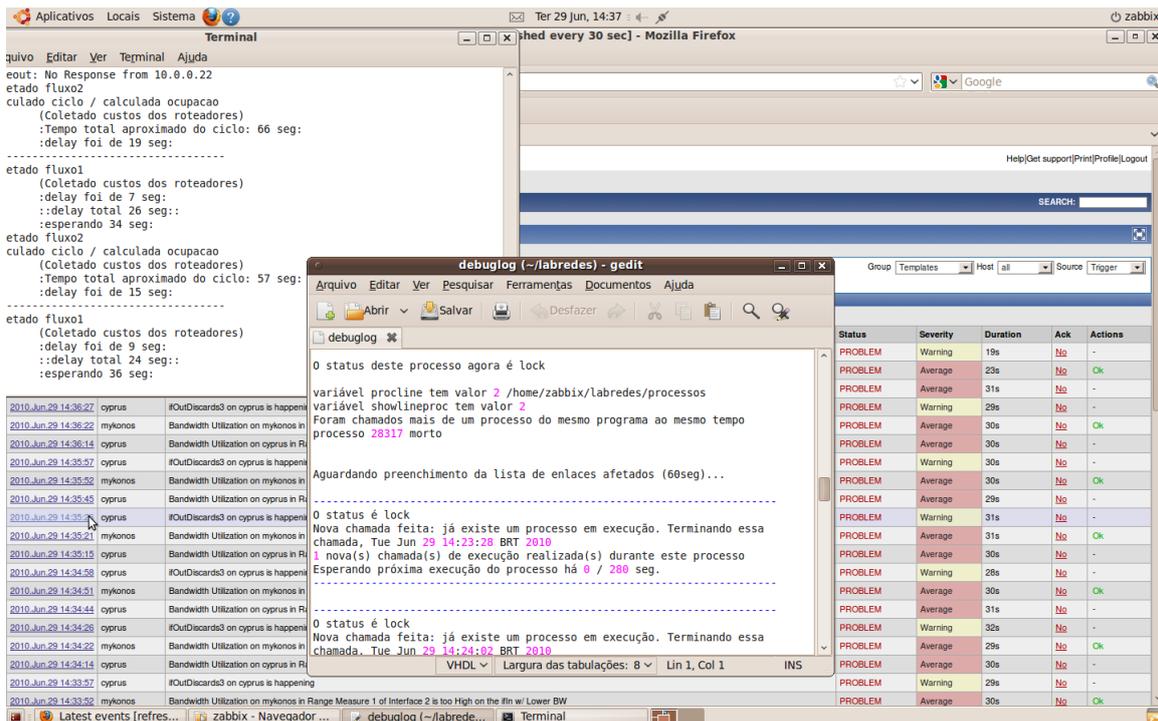


FIG. 5.5: Detecção de sobrecarga e execução do algoritmo

A partir desse ponto, segue a verificação do ocorrido quando a ferramenta de monitoramento ZABBIX fez a chamada ao primeiro *script* do ciclo de trabalho SAAB, denominado *ordenar.sh*, que possui a incumbência de eleger o enlace mais afetado pela sobrecarga de dados. A Figura 5.5 ilustra a ferramenta em ação nesse instante: a ação dos *triggers*, o *script* de coleta periódica de dados e o *log* das ações do *script* de cálculo e mudança do custo.

O primeiro passo do *script* é “trancar” o *script*, de modo a não permitir novas execuções do mesmo código enquanto todo o processo de decisão e atuação não tiver sido concluído. Isso ocorre porque as *actions* do ZABBIX não param de agir, enquanto o custo não for alterado, e ocasionar a parada dos alarmes dos *triggers*. Após o trancamento, é esperado um tempo definido pelo usuário, que aqui foi de 60 segundos, para a coleta de todos os enlaces afetados pela sobrecarga definida de 90% da largura de banda acusados pelo ZABBIX. O *script* então prosseguiu comparando todos os enlaces afetados entre si, e aquele que teve a maior taxa de ocupação (cujo valor é resgatado da coleta realizada pelo *script* de coleta periódica do tráfego de entrada, taxa de ocupação e valores de custo das interfaces) foi elegido como o enlace afetado E , que foi justamente o (5, 2).

Prosseguindo com o processo, o *script* de ordenação chamou um novo *script*, deno-

minado `modulo_mudanca.sh`. Ele realiza as tarefas encaixadas no contexto do Módulo Tomador de Decisão e do Módulo Mudança de Parâmetros, e usa uma das heurísticas desejadas pelo usuário. A primeira etapa seguida pela heurística do “Método original proposto para o SAAB (adaptada para a solução proposta)”, é a obtenção do $P(\Gamma_{notE})$ e do $P'(\Gamma_E)$. Na análise teórica, os melhores caminhos, e consequentemente os pesos associados a eles, podem ser obtidos com relativa facilidade, pois tem-se conhecimento da origem e destino das demandas. Porém, essa solução prática de trabalho seguindo o ciclo SAAB analisa constantemente inúmeras demandas passantes na topologia, sem saber a origem e o destino de cada uma delas. Para que esse conhecimento possa ser obtido em tempo real, a heurística de decisão deverá fazer estimativas baseadas na matriz de tráfego, utilizando os conceitos descritos pela ramo conhecido como Tomografia de Redes (COATES, 2002). Sendo assim, a heurística proposta obtém os melhores caminhos executando o algoritmo de Dijkstra, conforme as particularidades descritas a seguir:

- a) Os enlaces entre roteadores da topologia foram previamente cadastrados nos arquivos de configuração. Observando a Figura 5.2, apenas os enlaces entre os roteadores foram incluídos, já que os enlaces ligados aos *hosts*, por serem nós folha, não são relevantes para a questão do balanceamento de carga;
- b) O *script* de coleta periódica (`IFINOCETTS.sh`) obtém periodicamente o custo atual das interfaces dos roteadores;
- c) De posse dos enlaces e dos custos, o algoritmo de Dijkstra calcula, para cada um dos roteadores, os melhores caminhos de todas as combinações de demandas possíveis para a topologia estudada. Por exemplo, para o roteador $R3$, são calculados os melhores caminhos para as demandas $R3 \rightarrow R0$, $R3 \rightarrow R1$, $R3 \rightarrow R2$, $R3 \rightarrow R4$ e $R3 \rightarrow R5$;
- d) O primeiro cálculo que o algoritmo de Dijkstra realiza no processo é o da obtenção $P'(\Gamma_E)$, onde os enlaces cadastrados e seus custos são passados para o cálculo no arquivo `dijkstra.php` (GISWIKI, 2010). Note-se que os custos são associados aos enlaces, logo o mesmo valor de custo é atribuído às duas interfaces que constituem o enlace;
- e) Após o cálculo dos melhores caminhos para todas as demandas possíveis, são selecionadas todas as demandas cujo caminho calculado pelo algoritmo de Dijkstra

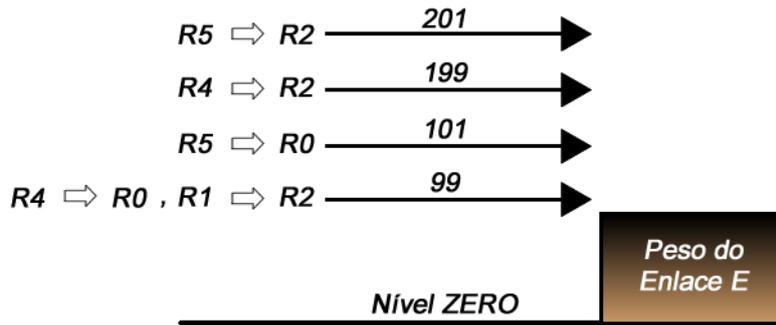


FIG. 5.6: Modelagem do Peso do Enlace E como uma Barreira na análise prática

contenha o enlace afetado E (5, 2). Nesse teste, partindo-se do princípio que não se conhece a origem e o destino da demanda que está causando o congestionamento, foram encontradas as seguintes possíveis demandas: (1, 2) = caminho (1-4-**5-2**), custo 101; (4, 0) = caminho (4-**5-2-0**), custo 101; (4, 2) = caminho (4-**5-2**), custo 51; (5, 0) = caminho (**5-2-0**), custo 100; (5, 2) = caminho (**5-2**), custo 50. Fazendo a subtração dos custos obtidos com o custo do enlace E (5, 2), que é 50, são obtidos os seguintes custos na ordem das demandas apresentadas: 51, 51, 1, 50 e 0.

- f) A seguir, o processo prossegue utilizando o algoritmo de Dijkstra novamente, desta vez para o cálculo do $P(\Gamma_{notE})$. A diferença é que o valor de custo atribuído pelo *script* para o enlace afetado E e passado para o arquivo `dijkstra.php` é igual a “ I ”, ou infinito. Esse valor indica que o enlace possui um valor de custo tão alto, que não existirá nenhum caminho que passe por ele, e naturalmente fará com que o algoritmo de Dijkstra não o inclua nos cálculos dos melhores caminhos para as demandas possíveis.
- g) Após o cálculo, são selecionadas todas as demandas obtidas originalmente no cálculo que tinham o enlace E constando em suas rotas, que são: (1, 2) = caminho (1-0-2), custo 150; (4, 0) = caminho (4-1-0), custo 150; (4, 2) = caminho (4-1-0-2), custo 200; (5, 0) = caminho (5-4-1-0), custo 151; (5, 2) = caminho (5-4-1-0-2), custo 201.

Agora o processo possui todos os valores necessários para realização da modelagem dos custos associados aos fluxos de dados. Após o cálculo da diferença, chegou-se a modelagem mostrada na Figura 5.6.

O valor de custo escolhido será aquele com valor de custo associado à demanda imediatamente superior ao custo do enlace E . Novamente de acordo com a Figura 5.6, o custo

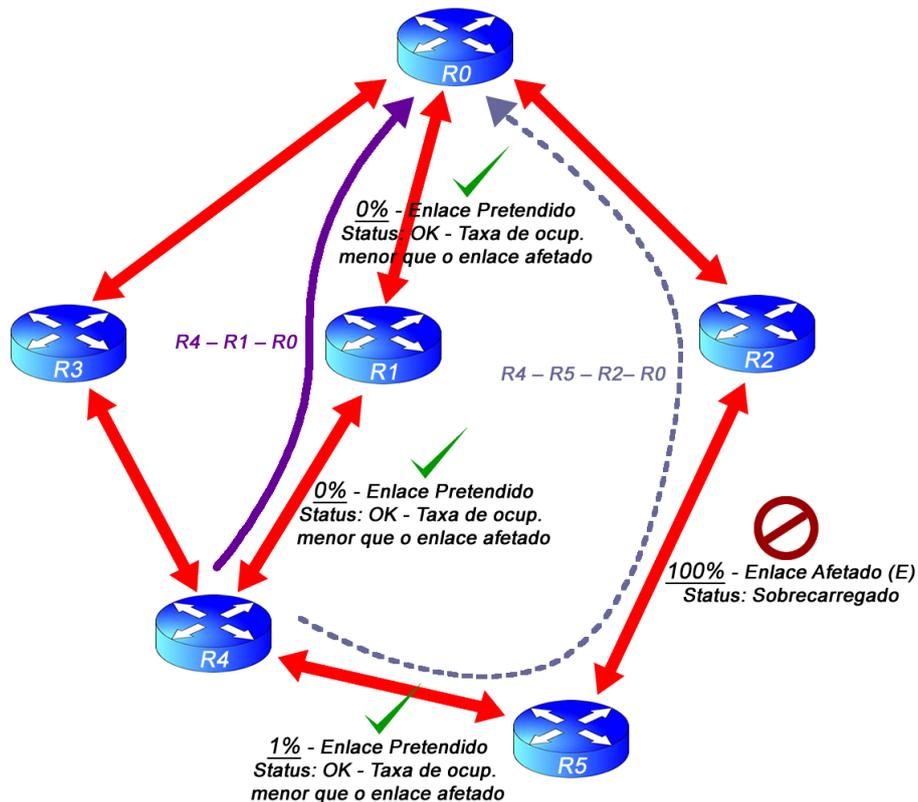


FIG. 5.7: Comparação da taxa de ocupação do enlace afetado com as taxas de ocupação dos enlaces da nova rota escolhida para a demanda modelada

que atende a essa condição, e será usado como base de cálculo para a mudança, é 99, situado imediatamente acima do peso do enlace afetado (2, 5), que é 50. Isso ocorreu exatamente como o esperado, conforme visto na análise teórica, só que aqui o peso está associado às demandas (4, 0) e (1, 2).

O processo agora passou para a comparação da taxa de ocupação do enlace afetado com as taxas de ocupação de entrada de cada um dos enlaces pertencentes à nova rota escolhida para a demanda modelada. Como se sabe, a demanda que se deseja redirecionar é a (4, 0), que seguirá pelo caminho (4-1-0). Entretanto, o *script* fez a escolha da demanda (1, 2) como sendo o fluxo a ser redirecionado. Essa é uma limitação da heurística adotada, que afetará a comparação das taxas de ocupação, mas não afeta a demanda real passante nos nós. Isso ocorreu porque duas demandas foram modeladas com o mesmo custo de limiar de barreira, e como a demanda (1, 2), na execução do código, veio disposta primeiro que a demanda (4, 0), foi a demanda escolhida. Novamente cabe frisar que não é possível, com a heurística adotada, conhecer a origem e o destino dos fluxos.

Como a demanda escolhida foi a (1, 2), seu melhor caminho escolhido foi (1-0-2).

Realizada a comparação das taxas de ocupação, verificou-se que o enlace afetado E (5, 2) estava com 100% de ocupação na sua entrada, enquanto que os enlaces (1, 0) e (0, 2) estavam ambos com 0% de ocupação. Como nenhum dos enlaces da nova rota estavam com suas ocupações acima da ocupação do enlace afetado E , o processo prosseguiu com sua execução, caso contrário o processo teria sido abortado. Se a demanda desejada tivesse sido escolhida, os enlaces verificados seriam o (4, 1) e o (1, 0), ambos também com 0% de ocupação, fato que confirma a validação do processo, conforme mostrado na Figura 5.7.

Uma observação a ser feita é que a comparação dos enlaces são realizadas obedecendo o sentido do fluxo, já que os *links* são *full-duplex*. Assim o enlace (0, 2), por exemplo, possuía 0% de ocupação na entrada da interface do roteador 2, enquanto que o enlace (2, 0) possuía 95% de taxa de ocupação na entrada da interface do roteador 0.

Finalmente a mudança pode ser realizada. O *script* do método original SAAB realiza, através do aplicativo Telnet, o acesso autorizado às interfaces dos dois roteadores que compõem o enlace $R5 \Leftrightarrow R2$, e executa o comando de alteração do custo OSPF para o valor modelado adicionando o valor 1, ou seja, $99 + 1 = 100$. A barreira do fluxo (4, 0) foi formada, e o *script* aguardou o tempo definido pelo usuário de 180 segundos, para a efetivação do redirecionamento dessa demanda, e por fim liberou o processo.

Agora novas modificações nos custos só poderão ser feitas caso o enlace afetado seja diferente do enlace alterado no último ciclo, ou que o enlace afetado seja o mesmo do enlace alterado por último, porém com a indicação da faixa de taxa de ocupação diferente da indicação marcada na última mudança. Por exemplo, se o enlace afetado fosse novamente o (5, 2), mas a indicação de faixa de ocupação fosse 95%, diferente da última que foi 90%, o ciclo seguiria o seu processo natural. Esse procedimento é importante para evitar que o mesmo enlace afetado seja modelado continuamente, e seu custo seja alterado sem necessidade.

O ciclo de trabalho SAAB foi encerrado com sucesso, e está pronto agora para um novo ciclo, assim como a análise prática conclui que a solução funcionou dentro do esperado na teoria.

Já a realização do experimento executado usando a heurística Cisco modificado, utilizando o mesmo cenário de testes, aproveitou os *scripts* utilizados para a heurística SAAB, devidamente configurados para chamar os procedimentos referentes ao cálculo desejado.

O experimento funcionou da seguinte maneira: Após o instante 10 min, a ferramenta de monitoramento ZABBIX fez a chamada ao *script* ordenar.sh, que elegeu o enlace mais

afetado pela sobrecarga de dados da mesma forma como foi na heurística SAAB. Tendo eleito o enlace (5, 2), ele chamou o modulo_mudanca.sh. Todo o cálculo de modelagem de custos associados às demandas não surte nenhum efeito aqui. Quando o *script* chama o procedimento Cisco modificado, ele inicialmente coleta a taxa de ocupação do enlace afetado E , que foi de 87% no momento da coleta. Se esse valor estivesse igual a 100%, seria considerado para efeito de cálculo o valor de taxa de ocupação igual a 99%. A seguir, o resultado da conversão da taxa de ocupação em fluxo de dados passante foi 1.740.000 bps ($2.000.000bps \times 87\%$), conseqüentemente com o valor da largura de banda disponível 260.000 bps, obtida da subtração da capacidade do enlace (2000 Kbps) com o fluxo de dados passante ($2.000.000bps - 1.740.000bps$). Finalmente, o cálculo do custo foi realizado com a expressão $C = 10^8 / available_bandwidth$, com o resultado achado de 384. O valor de custo seria de 65.535 caso o cálculo resultasse em um valor acima desse. Por fim, o novo custo foi aplicado às interfaces do enlace afetado E através do comando *telnet*.

Após isso, foi constatado nos arquivos de *log* exatamente o que foi previsto na análise teórica: como o custo ficou bem acima de todos os custos associados às demandas passantes no enlace (5, 2), todas elas foram redirecionadas para o caminho livre $R4 \rightarrow R1 \rightarrow R0$, que ficou sobrecarregado. A partir daí, houve mais uma execução do processo para alteração do custo do enlace (4, 1), com taxa de ocupação medida de 74% e custo do enlace alterado para 192. Os custos dos caminhos associados às demandas então ficaram configurados da seguinte maneira: $R4 \rightarrow R5 \rightarrow R2 \rightarrow R0$ com custo 435, e $R4 \rightarrow R1 \rightarrow R0$ com custo 292. Após essa segunda alteração do valor de custo, o experimento foi encerrado pelo tempo. Não foi necessário acionar o mecanismo de divisão do vetor de pesos dos enlaces por um número natural ao ultrapassar um valor de custo previamente estabelecido.

5.2.4 RESULTADOS DA ANÁLISE PRÁTICA

Segue agora a avaliação dos resultados seguindo os parâmetros de desempenho adotados:

- a) Medida de congestionamento (quanto menor, melhor): A partir do *log* da taxa de ocupação gerado durante o teste, foi possível, com o auxílio de um *script*, realizar o cálculo da quantidade de tempo, em minutos, que os enlaces sobrecarregados apresentaram a taxa de ocupação maior que o valor de custo definido para a sobrecarga, ao longo do tempo do experimento. Com isso, após o último instante do teste (30 min), foram obtidos os resultados aproximados mostrados na Tabela 5.3.

	Cisco	Cisco modificado	SAAB
1, 0	-	5min	-
2, 0	19min	2min	3min
3, 0	-	7min	-
4, 1	-	5min	-
0, 1	-	-	-
0, 2	-	-	-
5, 2	20min	3min	3min
0, 3	-	-	-
4, 3	-	21min	-
3, 4	-	-	-
5, 4	-	-	-
1, 4	-	-	-
4, 5	-	-	-
2, 5	-	-	-
Máximo	20min	21min	3min

TAB. 5.3: Tempos de congestionamento nos enlaces após o experimento 1

Nota-se que os valores esperados pela análise teórica coincidiram com os valores obtidos na análise prática, para as heurísticas Cisco e SAAB. Comparando elas com a heurística Cisco modificado, que não teve os valores de análise de desempenho obtidos na análise teórica devido a impossibilidade de se obter resultados precisos, é verificado que a heurística que teve o melhor desempenho foi a SAAB. A heurística Cisco fez com que o congestionamento se concentrasse apenas nos enlaces que acomodaram as demandas, enquanto que a heurística Cisco modificado apresentou congestionamentos em diversos enlaces, o que reduz as opções de redirecionamento das demandas pelos demais caminhos escolhidos pelo protocolo de estado de enlace visando a minimização dos congestionamentos.

- b) Medida de desempenho para o usuário (quanto menor, melhor): Os gráficos de descarte de pacotes na saída das interfaces, gerados em tempo real pelo ZABBIX, revelaram exatamente o valor de descarte obtido no cálculo realizado na análise teórica, conforme exibido nas Figuras 5.8, 5.9 e 5.10. O gráfico das Figura 5.8 e 5.10 refere-se ao roteador R5, já os gráficos da Figura 5.9 referem-se, lendo de cima para baixo, aos roteadores R1, R0, R3, R5, R4 e R2.

Pela análise dos gráficos, verifica-se que as simulações efetuadas com a finalidade de verificar o funcionamento das heurísticas Cisco e Cisco modificado apresenta-

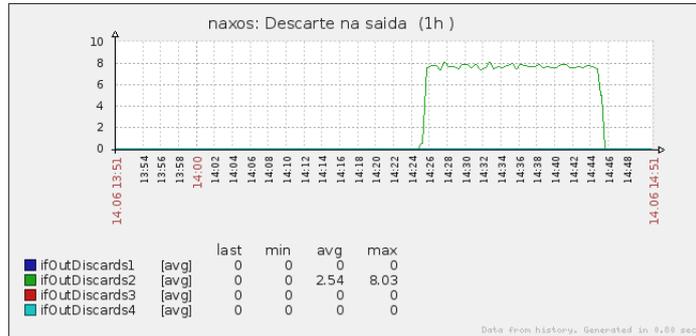


FIG. 5.8: Descarte de pacotes na saída da interface ocorrido durante o experimento 1 na heurística Cisco

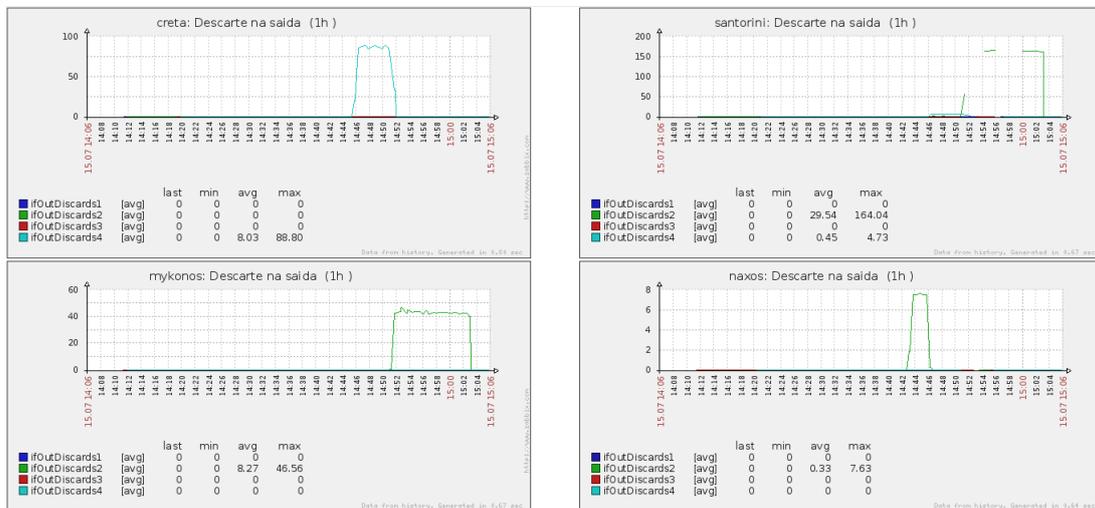


FIG. 5.9: Descarte de pacotes nas saídas das interfaces ocorrido durante o experimento 1 na heurística Cisco modificado

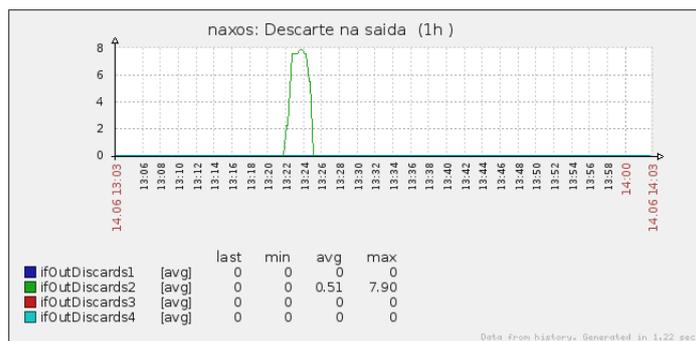


FIG. 5.10: Descarte de pacotes na saída da interface ocorrido durante o experimento 1 na heurística SAAB

ram descarte de pacotes, que tiveram início no momento em que a demanda (4, 0) foi disparada. No Cisco, como o custo não é modificado em nenhum momento, o descarte ocorreu em apenas uma interface do roteador correspondente à saída (5, 2), enquanto que no Cisco modificado, em virtude da troca de custos em mais de um instante, o descarte ocorreu em várias interfaces de saída. Inclusive a última modificação feita pelo Cisco modificado ocasionou a ocorrência de alguns *time-out* nos processos de coletas de informações. Já o processo executado pelo algoritmo SAAB alterou o custo do enlace apenas uma vez, de maneira a impedir a passagem do fluxo (4, 0), cujo valor de custo foi modelado de forma a desafogar o fluxo do enlace afetado E , atingido pela sobrecarga de dados. O descarte, nesse caso, aconteceu apenas durante o período em que o ciclo de trabalho do SAAB era executado no gerente do nó central antes da atuação no enlace.

- c) Medida de balanceamento de carga (quanto menor, melhor): O *script* de coleta periódica grava, a cada ciclo de coleta, os valores de ocupação das interfaces dos roteadores envolvidos no teste em um arquivo *log*. Após o término do teste, esse *log* é utilizado por um *script* construído com a finalidade de calcular o desvio padrão de todos os valores coletados, para cada intervalo de tempo definido na coleta. Como este intervalo foi de 60 segundos, o desvio padrão é calculado para cada intervalo de 1 minuto. Os resultados obtidos pelos *scripts* foram validados através da paridade com os valores obtidos pelo cálculo do desvio padrão na análise teórica. A diferença é que, no caso da análise prática, é possível fazer o cálculo para todo o intervalo do teste, enquanto que na análise teórica este cálculo é consideravelmente trabalhoso.

Após a retirada dos enlaces que não participaram da variação da taxa de ocupação após a mudança de custo do enlace E (5, 2), o *script* gerou os valores que estão representados no gráfico da Figura 5.11. Esse procedimento de retirada dos enlaces foi executado para que ficasse evidente a diferença entre os desvios padrão dos enlaces que, conhecidamente, fizeram parte dos caminhos por onde as demandas trafegaram. O conhecimento desses enlaces se deve ao fato do experimento ter sido construído com todo o planejamento dos custos, das demandas disparadas e das mudanças dos caminhos das demandas, executadas pelo protocolo de estado de enlace, em resposta às mudanças de valor de custo dos enlaces.

Na análise prática, já é possível calcular a média de todos os desvios coletados ao

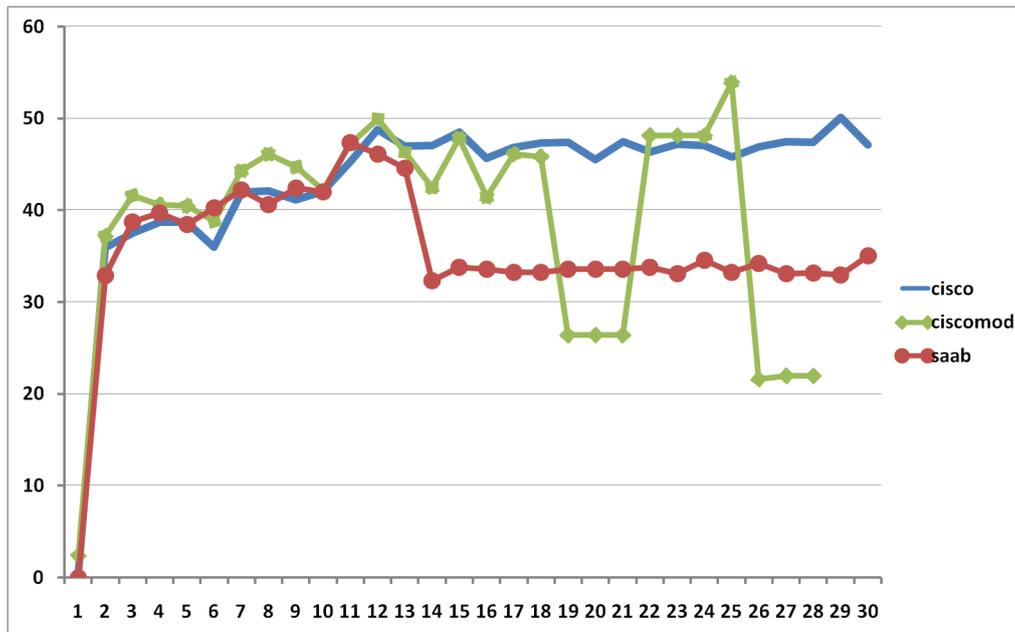


FIG. 5.11: Gráfico do desvio padrão entre as heurísticas no experimento 1 retirados os enlaces não participantes

longo do tempo. Os valores obtidos foram os exibidos na Tabela 5.4.

Cisco	43,17
Cisco modificado	38,88
SAAB	35,50

TAB. 5.4: Média dos desvios padrão no experimento de validação retirados os enlaces não participantes

Apesar do Cisco modificado apresentar o segundo melhor resultado, o comportamento dos valores no gráfico verificado foi relativamente instável devido às mudanças dos fluxos, enquanto que para a heurística SAAB é sabido que o valor do custo calculado é exatamente o valor necessário para que apenas a(s) demanda(s) necessária(s) sejam redirecionadas (apesar da limitação em não ser possível conhecer a matriz de tráfego). Assim, o comportamento dos valores do desvio padrão ao longo do tempo no gráfico tende à estabilidade após a atuação do gerente, e conclui-se que, neste teste, o melhor resultado da média dos desvios foi para a heurística SAAB.

5.3 EXPERIMENTO 2

O segundo experimento apresenta um cenário de teste adaptado da “Avaliação do teste 1”, apresentado em (VIANNA, 2008). Este teste, que será chamado aqui de “teste original”, teve os resultados decorrentes de simulação feita no *software* NS-2, realizado segundo a abordagem distribuída. As diferenças do presente trabalho, em relação ao teste original, estão nos valores das larguras de banda, embora mantidas as proporções entre o menor valor e o maior valor, no *status* dos roteadores e enlaces, que já se encontram ativados no momento do início do teste, e na origem e destino das demandas, de nós microcomputadores PC para roteadores. A Tabela 5.5 mostra os eventos do teste e a Figura 5.12 mostra a topologia utilizada.

Instantes(min)	Evento
0	Iniciar coleta de dados
1	Configurar demanda: Nó #0 \rightarrow R4; 60% de 128kbps [76Kbps]
5	Configurar demanda: Nó #0 \rightarrow R3; 60% de 128kbps [76Kbps]
7	Configurar demanda: Nó #3 \rightarrow R4; 10% de 128Kbps [12Kbps]
60	Configurar demanda: Nó #0 \rightarrow R3; 20% de 128Kbps [25Kbps]
70	Configurar demanda: Nó #3 \rightarrow R4; 30% de 128Kbps [38Kbps]

TAB. 5.5: Lista de eventos do experimento 2

O tempo de duração foi de 120 minutos. Os *triggers* foram configurados para acionarem as *actions* toda vez que a taxa de ocupação em uma determinada interface ultrapassasse 90% da capacidade do seu enlace. Após o teste, os resultados podem ser analisados seguindo os parâmetros de desempenho.

- a) Medida de congestionamento (quanto menor, melhor): O *log* da taxa de ocupação gerado durante o teste, revelou as médias das taxas de ocupação ao longo de todo o experimento para cada um dos enlaces envolvidos representadas na Tabela 5.6.

A heurística Cisco, por não atuar na mudança dos valores de custo dos enlaces, fez com que o enlace $R0 \rightarrow R3$ apresentasse sobrecarga durante todo o tempo seguinte ao disparo da demanda que causou o evento. Já a heurística Cisco modificado,

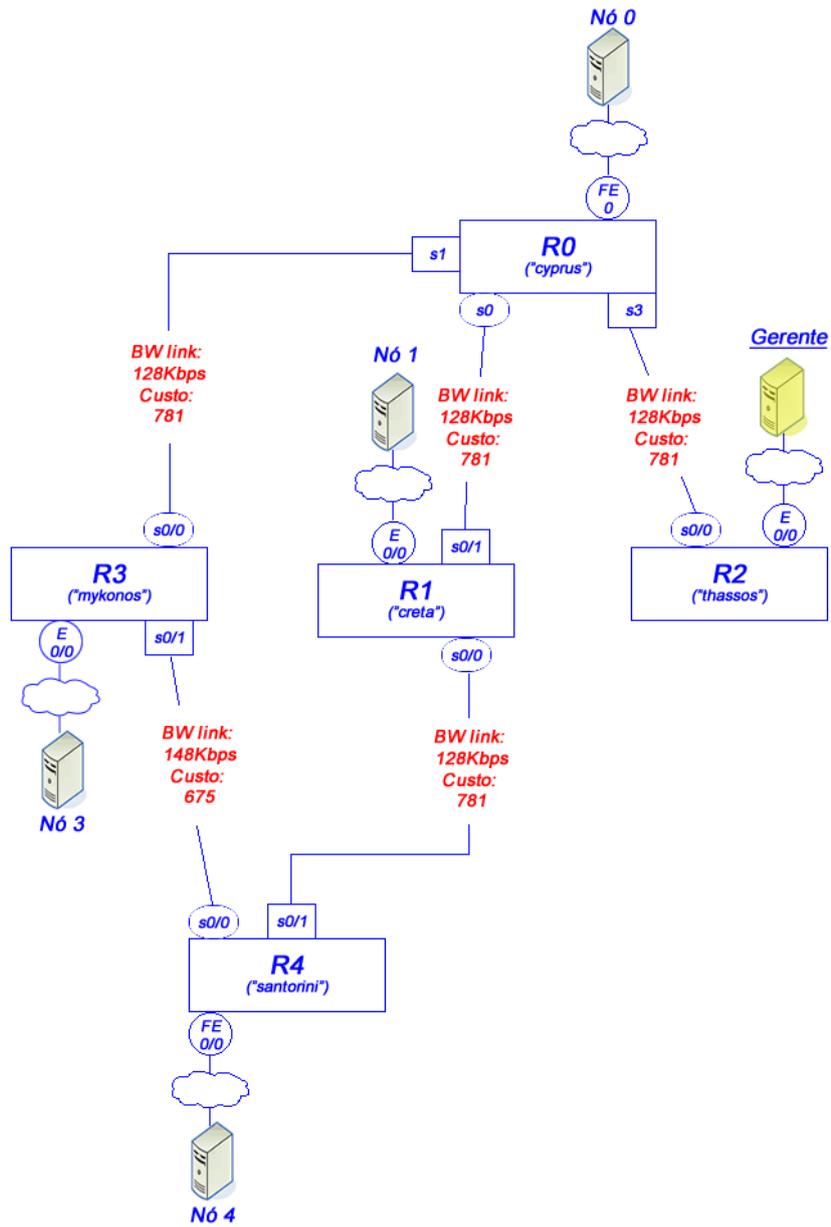


FIG. 5.12: Topologia de rede do experimento 2

	Cisco	Cisco modificado	SAAB
1, 0	-	-	-
2, 0	-	-	-
3, 0	-	-	-
4, 1	-	-	-
0, 1	-	21min	-
0, 2	-	-	-
0, 3	114min	92min	2min
4, 3	-	-	-
3, 4	-	-	-
1, 4	-	21min	-
Máximo	114min	92min	2min

TAB. 5.6: Tempos de congestionamento nos enlaces após o experimento 2

mesmo atuando na mudança dos valores de custo, não conseguiu minimizar o congestionamento tal como a heurística SAAB conseguiu, que por sua vez atuou no enlace afetado assim que a sobrecarga foi detectada, ajustando o valor do custo para que apenas a demanda necessária fosse redistribuída por outro caminho da topologia.

- b) Medida de desempenho para o usuário (quanto menor, melhor): Os gráficos de descarte de pacotes na saída das interfaces gerados pelo ZABBIX coletaram os valores mostrados nos gráficos das Figuras 5.13, 5.14 e 5.15, todos referentes ao roteador R0.

A análise dos gráficos revela que, enquanto as execuções dos algoritmos das heurísticas Cisco e Cisco modificado provocaram descartes de pacotes ao longo de todo o ensaio - incluindo a mudança de interface de ocorrência dos descartes durante a execução do teste utilizando Cisco modificado - uma única execução do algoritmo da heurística SAAB eliminou, até o término do experimento, o descarte que ocorria na interface do enlace afetado. Conclui-se assim que a heurística SAAB apresentou o melhor resultado dentro deste parâmetro de resultado.

- c) Medida de balanceamento de carga (quanto menor, melhor): O *shellscript* que realiza o cálculo do desvio padrão dos valores gravados em arquivo *log* coletados pelo *script* de coleta periódica dos valores de ocupação das interfaces, sem os enlaces que não participaram da variação da taxa de ocupação após a mudança de custo do enlace (por ser um teste com resultados previsíveis), retornou os valores convertidos

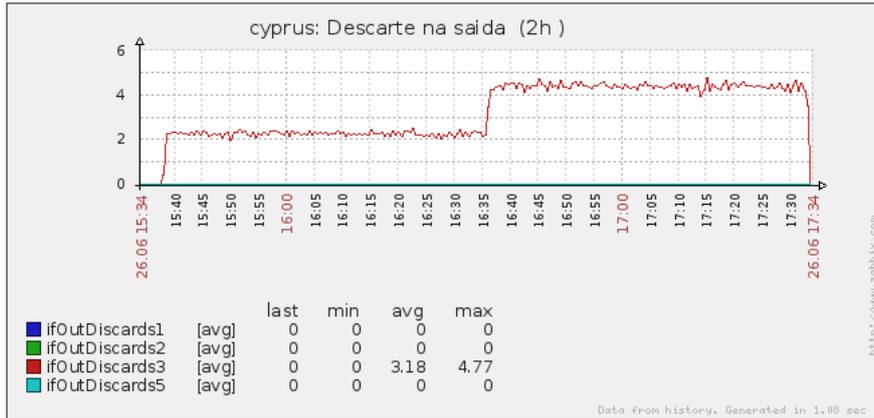


FIG. 5.13: Descarte de pacotes na saída da interface ocorrido durante o experimento 2 na heurística Cisco

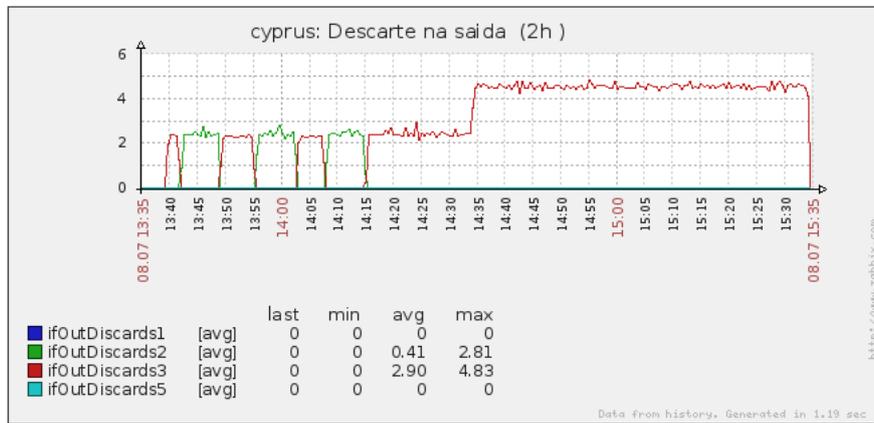


FIG. 5.14: Descarte de pacotes na saída da interface ocorrido durante o experimento 2 na heurística Cisco modificado

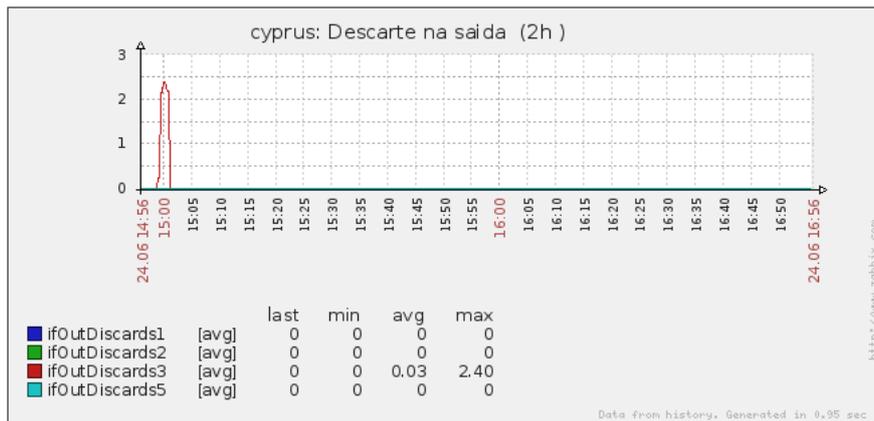


FIG. 5.15: Descarte de pacotes na saída da interface ocorrido durante o experimento 2 na heurística SAAB

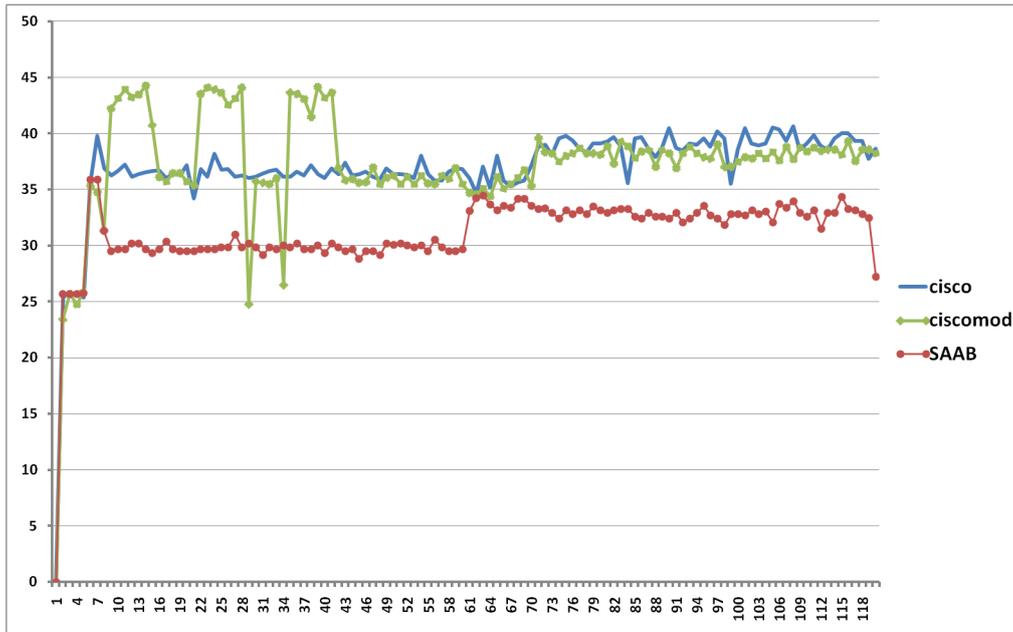


FIG. 5.16: Gráfico do desvio padrão entre as heurísticas no experimento 2 retirados os enlaces não participantes

no gráfico Desvio X Tempo da Figura 5.16.

A média de todos os desvios coletados ao longo do tempo foram os exibidos na Tabela 5.7.

Cisco	36,88
Cisco modificado	37,20
SAAB	31,08

TAB. 5.7: Média dos desvios padrão no experimento 2 retirados os enlaces não participantes

Mais uma vez, a análise dos valores no gráfico revelou que o comportamento da heurística Cisco modificado não teve a mesma estabilidade verificada na aplicação da heurística SAAB, à medida que as atuações do gerente foram feitas nos enlaces, e devido a esse fato ela foi a heurística que apresentou o pior desempenho na média dos desvios dessa vez. Foi verificado, no *log* dos eventos, que o algoritmo da heurística Cisco modificado precisou acionar o mecanismo de divisão do vetor de pesos dos enlaces por um número natural ao ultrapassar um valor de custo previamente estabelecido. O número natural usado neste teste foi igual a 2 (dois), e o valor de custo previamente estabelecido foi igual a 10.000 (dez mil). Com isso, os valores dos custos de todos os enlaces da topologia foram reduzidos para minimizar a diferença

de valores dos custos, e para impedir que o valor de um determinado enlace atinja valores muito altos, a ponto de chegar ao máximo permitido pelo protocolo OSPF de 65.535.

O melhor resultado constatado neste teste foi, mais uma vez, com a utilização da heurística SAAB.

5.4 EXPERIMENTO 3

O terceiro experimento apresenta um cenário de teste com todas as demandas (O, D) escolhidas de forma aleatória. Não foi realizado nenhum estudo prévio para saber o comportamento do tráfego de dados passante na topologia. A Figura 5.17 mostra a topologia utilizada e a Tabela 5.8 mostra os eventos do teste.

O tempo de duração foi de 30 minutos. Semelhantemente aos demais experimentos, os *scripts* foram chamados quando a taxa de ocupação fosse maior que 90% da capacidade do enlace. A finalidade principal deste experimento foi realizar a verificação do funcionamento do gerente, já que uma análise mais detalhada do desempenho da rede em resposta às alterações feitas pelas heurísticas, em um cenário randômico, demanda uma duração maior na coleta dos dados. Uma análise ainda mais aprofundada pode ser feita quanto maior for a complexidade da topologia. Os resultados obtidos, seguindo os parâmetros de desempenho definidos, são mostrados a seguir.

- a) Medida de congestionamento (quanto menor, melhor): A partir do *log* da taxa de ocupação gerado durante o teste, foi possível calcular o tempo que os enlaces sobrecarregados apresentaram a taxa de ocupação acima do custo de sobrecarga, cujos valores correspondentes ao final do teste estão exibidos na Tabela 5.9.

Dentro do cenário aleatório e do tempo transcorrido do experimento, a heurística Cisco que apresentou apenas um enlace sobrecarregado, devida a não alteração do valor do custo, o que pode ser vantajoso dependendo da circunstância; entretanto em termos de tempo decorrido, essa foi a heurística com pior desempenho. Já as heurísticas Cisco modificado e SAAB obtiveram desempenhos parecidos, com mais enlaces afetados, porém menos tempo de sobrecarga em cada um.

- b) Medida de desempenho para o usuário (quanto menor, melhor): Neste experimento, os gráficos de descarte de pacotes na saída das interfaces gerados pelo ZABBIX não registraram descartes de pacotes em nenhuma das heurísticas testadas.

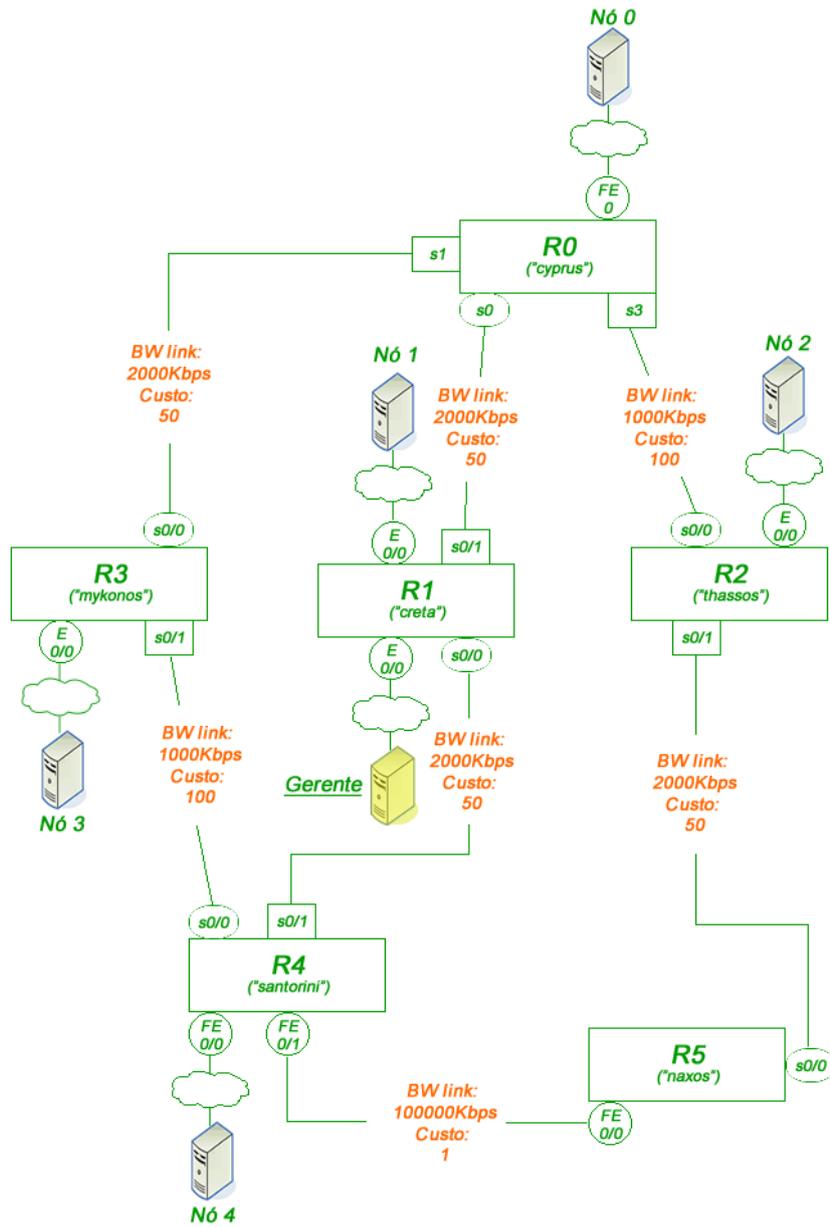


FIG. 5.17: Topologia de rede do experimento 3

Instantes(min)	Evento
0	Iniciar coleta de dados
1	Configurar demanda: Nó #2 $\rightarrow R4$; 40% de 2000kbps [800Kbps] Configurar demanda: Nó #0 $\rightarrow R3$; 5% de 2000kbps [100Kbps] Configurar demanda: Nó #1 $\rightarrow R4$; 70% de 2000kbps [1400Kbps]
5	Configurar demanda: Nó #3 $\rightarrow R4$; 36% de 2000kbps [720Kbps] Configurar demanda: Nó #4 $\rightarrow R5$; 30% de 2000kbps [600Kbps] Configurar demanda: Nó #0 $\rightarrow R1$; 84% de 2000kbps [1680Kbps]
10	Configurar demanda: Nó #4 $\rightarrow R5$; 51% de 2000kbps [1020Kbps] Configurar demanda: Nó #2 $\rightarrow R0$; 3% de 2000kbps [60Kbps] Configurar demanda: Nó #0 $\rightarrow R4$; 8% de 2000kbps [160Kbps]
15	Configurar demanda: Nó #4 $\rightarrow R5$; 9% de 2000kbps [180Kbps] Configurar demanda: Nó #2 $\rightarrow R3$; 4% de 2000kbps [80Kbps] Configurar demanda: Nó #3 $\rightarrow R5$; 8% de 2000kbps [160Kbps]

TAB. 5.8: Lista de eventos do experimento 3

	Cisco	Cisco modificado	SAAB
1, 0	-	-	-
2, 0	15min	2min	6min
3, 0	-	-	-
4, 1	-	-	-
0, 1	-	-	-
0, 2	-	-	-
5, 2	-	-	-
0, 3	-	-	-
4, 3	-	6min	3min
3, 4	-	1min	1min
5, 4	-	-	-
1, 4	-	2min	-
4, 5	-	-	-
2, 5	-	-	-
Máximo	15min	6min	6min

TAB. 5.9: Tempos de congestionamento nos enlaces após o experimento 3

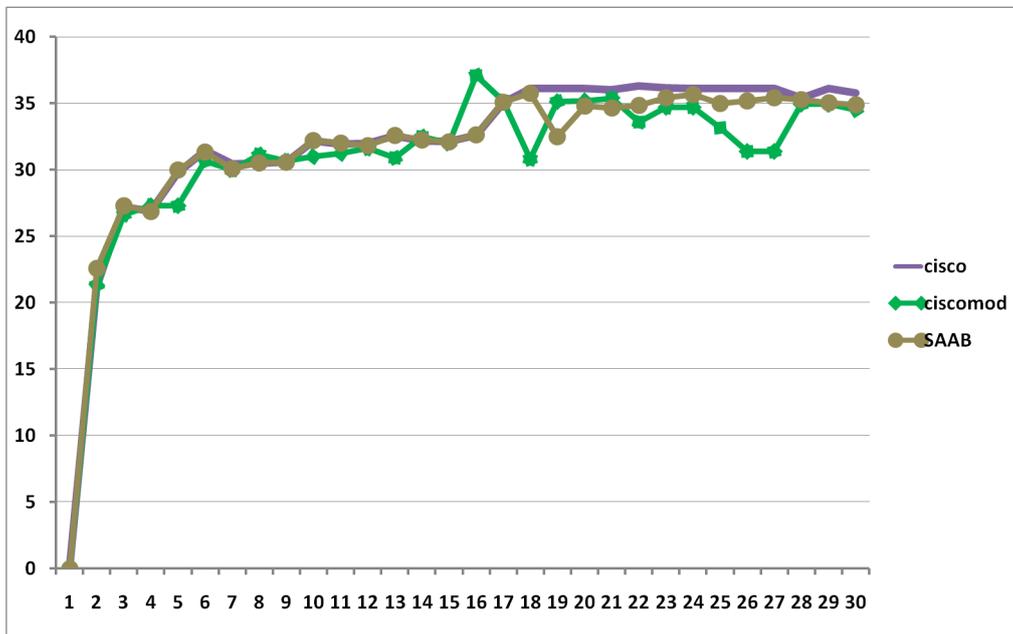


FIG. 5.18: Gráfico do desvio padrão entre as heurísticas no experimento 3 incluídos todos os enlaces

c) Medida de balanceamento de carga (quanto menor, melhor): O cálculo do desvio padrão realizado a partir dos valores de ocupação das interfaces gravados no *log* foi feito com a inclusão de todos os enlaces da topologia estudada, pelo motivo do caráter aleatório do experimento, ou seja, apenas o envio das demandas e o valor inicial dos custos foram planejados, não sendo conhecidos previamente quais enlaces seriam afetados e para quais valores de custos os enlaces seriam alterados, assim como o posterior redirecionamento das demandas. Os valores convertidos no gráfico Desvio X Tempo encontram-se na figura 5.18.

A média de todos os desvios coletados ao longo do tempo estão exibidos na tabela 5.10.

Pelo comportamento das curvas no gráfico, verificou-se que cada um dos algoritmos surtiu efeitos diferentes na variação das taxas de ocupação dos enlaces da topologia de rede usada no experimento. Todos os enlaces pertencentes à topologia foram utilizados no cálculo. Quanto à avaliação dos resultados, a variação entre os desvios foi bem pequena, com uma ligeira vantagem para a heurística Cisco modificado, seguida da heurística SAAB, e por último, a heurística Cisco. Esse resultado ocorreu devido ao esquema de redirecionamento dos fluxos executado pela heurística Cisco modificado, embora esse redirecionamento ocorra ignorando os custos associados aos caminhos por onde passam as demandas, o que pode gerar um comportamento mais instável (as vezes melhor, as vezes pior), ao contrário da heurística SAAB, que escolhe realmente aquelas demandas nas quais se desejam alterar os caminhos, e gera, por conseguinte, um comportamento mais estável.

Cisco	31,90
Cisco modificado	30,84
SAAB	31,47

TAB. 5.10: Média dos desvios padrão no experimento 3 incluídos todos os enlaces

Pode-se verificar que, embora a topologia utilizada no experimento seja pequena em relação à maioria dos AS, a solução conseguiu atuar no cenário de demandas aleatórias em um sistema autônomo, que é o cenário de utilização normal da rede. A consequência imediata e a longo prazo dessa atuação no auto-balanceamento de carga vai depender, principalmente, da matriz de tráfego existente no momento que a sobrecarga for detectada e das heurísticas de cálculo e mudança do valor de custo adotadas. Em todos os três

experimentos realizados, cada heurística utilizada provocou uma atuação diferente do protocolo de estado de enlace OSPF, ao alterar os caminhos por onde fluxos trafegam em virtude da mudança dos valores de custo dos enlaces.

6 CONCLUSÃO

6.1 CONSIDERAÇÕES FINAIS

É constante o desafio da otimização de recursos em redes de computadores. Tipos distintos de utilização demandam soluções específicas, adequadas para o problema estudado, que são tratadas pelo ramo da Engenharia de Tráfego.

O problema do congestionamento pode ser atacado com a técnica de Balanceamento de Carga. A correta alocação dos recursos possibilita otimizar a alocação de recursos, minimizando efeitos indesejáveis ao usuário final, como atrasos, flutuações e perda de dados. O esforço das pesquisas está em obter soluções eficientes e viáveis do ponto de vista econômico. Nesse sentido, este trabalho procurou implementar uma solução baseada em estudos que visam a utilização do protocolo de estado de enlace OSPF no Balanceamento de Carga em redes de computadores.

Baseada na técnica denominada Sistemas Autônomos Auto-Balanceados, que está relacionada com o trabalho realizado por (VIANNA, 2008), o presente trabalho contribui ao adaptar esta técnica, convenientemente simulada pelo trabalho relacionado em uma abordagem distribuída, em um sistema que atua no real funcionamento de uma topologia de rede. Consiste em monitorar constantemente a rede através de um nó central e, na ocorrência de eventos que caracterizem sobrecarga de dados nos enlaces, realizar o cálculo e a modificação do valor de custo no enlace modelado, para desviar o fluxo adequado por outra rota que apresente menor taxa de ocupação.

Com isso, as contribuições surgem a partir do enquadramento da solução nas classificações da Engenharia de Tráfego. Foi mostrada a integração das diversas ferramentas utilizadas, escolhidas com o intuito de cumprir o requisito da solução ser eficiente e acessível, requisito este que serviu de base para a escolha da abordagem centralizada no lugar da abordagem distribuída proposta por (VIANNA, 2008). A solução é dividida em Módulos de Atuação conceituais que são executados em um nó central, o gerente, que realiza quando solicitado o ciclo de trabalho SAAB. O Módulo de Análise de Redes monitora em tempo real a situação dos enlaces do Sistema Autônomo por meio do *software* de monitoramento que usa o protocolo de gerência de redes SNMP. Através da conveniente

configuração de gatilhos, é possível detectar a ocorrência de sobrecarga, e assim passar para o Módulo Tomador de Decisão. Esse módulo é chamado por uma ação do *software* de monitoramento, e consiste em um algoritmo escrito em linguagem *script* baseado em heurísticas de cálculo do novo valor de custo OSPF. Uma vez calculado o novo valor e verificada a conveniência, o Módulo Mudança de Parâmetro é chamado para efetivar a alteração do custo.

No Capítulo 5, foram mostradas as heurísticas escolhidas e os experimentos realizados para aferição do funcionamento da solução e verificação das consequências no Balanceamento de Carga, que foram: Cisco, que é a heurística padrão de atribuição de valor de custo aos enlaces considerando a largura de banda; SAAB, adaptada de (VIANNA, 2008), que considera a taxa de ocupação nos enlaces e o custo dos caminhos utilizados pelas demandas de dados; e Cisco modificado, que foi uma proposta realizada pelo presente trabalho, que leva em conta a taxa de ocupação atual e disponível, mas não considera as demandas passantes. Outras contribuições estão na implementação de novas funcionalidades nas heurísticas - comparação da taxa de ocupação do enlace afetado com as taxas de ocupação de entrada de cada um dos enlaces pertencentes à nova rota escolhida para a demanda modelada, e divisão do vetor de pesos dos enlaces por um número natural ao ultrapassar um valor de custo previamente estabelecido -, e funcionalidades necessárias para o funcionamento da solução centralizada em uma topologia real, tais como a realização do cálculo Dijkstra antes da atuação na rede, controle de atrasos de leitura dos objetos da MIB através do protocolo SNMP e controle de concorrência na chamada aos *scripts*.

Ainda no Capítulo 5 verificou-se, no decorrer dos experimentos feitos, que a ferramenta se comportou conforme o esperado dentro dos objetivos do trabalho, e que os parâmetros de desempenho adotados apontaram, de modo geral, que a heurística SAAB obteve o melhor resultado no Auto-balanceamento de Carga utilizando o protocolo OSPF, em comparação com as heurísticas Cisco e Cisco modificado. Esses parâmetros de desempenho foram Medida de congestionamento, Medida de desempenho para o usuário e Medida de balanceamento de carga.

Assim, acredita-se que este trabalho contribuiu no sentido de mostrar que é possível realizar o Balanceamento de Carga utilizando o protocolo OSPF, pois, dependendo do objetivo desejado na Engenharia de Tráfego, é possível aproveitar tecnologias já existentes bastando apenas adequar o funcionamento ao objetivo desejado, não sendo necessária a

pesquisa de novas tecnologias nesse caso.

6.2 TRABALHOS FUTUROS

Este trabalho abre caminho para novas formas de implementações práticas do Auto-Balanceamento de Carga com o uso de protocolos de estado de enlace. Com a verificação do comportamento real da solução, ficaram evidentes as vantagens, desvantagens e possibilidades de melhoria ou continuidade.

- Criação de novas heurísticas e/ou adaptação das heurísticas apresentadas de cálculo de valor de custo para a mesma ferramenta utilizada. A solução proposta pode ser categorizada no conceito de *framework* para atuação no auto-balanceamento de carga na rede, possibilitando a incorporação de novas heurísticas. Estas heurísticas podem explorar técnicas não vistas neste trabalho;
- Novas funcionalidades podem ser agregadas na execução dos *scripts*, visando, por exemplo, uma escalabilidade maior no cadastro de novos enlaces, alterações críticas da topologia e desempenho na execução, bem como a escrita em outras linguagens e a criação de interfaces com o usuário. Múltiplas execuções do algoritmo podem ser propostas;
- Exploração dos recursos não estudados do *software* de monitoramento, bem como a utilização de novos *softwares*;
- Implementação de mais de um gerente conectado à topologia, mas em pontos diferentes da rede, que troquem informações entre eles sobre os processos executados, afim de que trabalhem em cooperação e minimize a possibilidade de parada do gerente em caso de caída ou sobrecarga do enlace conectado;
- Execução da solução proposta em um estudo mais abrangente relacionado ao impacto do auto-balanceamento de carga em uma topologia maior;
- Introdução de estimativas de matriz de tráfego usando os conceitos da Tomografia de Redes, para descoberta em tempo real das demandas com conhecimento de origem e destino. Foi verificada uma limitação na escolha do novo caminho a ser seguido pela(s) demanda(s) redirecionada(s), pois a matriz de tráfego não é conhecida, e nem sempre a comparação da taxa de ocupação do enlace afetado pela sobrecarga é

em relação ao enlaces da nova rota proposta para o redirecionamento da demanda escolhida.

Saindo do paradigma centralizado, pode-se aproveitar determinados elementos vistos neste trabalho para estudar a viabilidade da implementação prática seguindo o paradigma distribuído, que é a proposta original do Sistema Autônomo Auto-Balanceado.

Uma outra idéia é implementar uma heurística que trabalhe em cima do método ECMP (*Equal Cost Multipaths*), combinando os métodos de Balanceamento de Carga com os métodos de Divisão de Carga, e assim verificando o desempenho dentro dos objetivos da Engenharia de Tráfego.

7 REFERÊNCIAS BIBLIOGRÁFICAS

- ALMHANA, J., LIU, Z., MCGORMAN, R. e LANTEIGNE, B. **Bandwidth provisioning and capacity planning tools**. Em *Communication Networks and Services Research Conference, 2006. CNSR 2006. Proceedings of the 4th Annual*, págs. 3 pp. –330, 24-25 2006.
- AWDUCHE, D., CHIU, A., ELWALID, A., WIDJAJA, I. e XIAO, X. **Overview and Principles of Internet Traffic Engineering**. Technical Report RFC 3272, Internet Engineering Task Force and Internet Engineering Steering Group, may 2002. URL <http://www.ietf.org/rfc/rfc3272.txt>. Online; acessado 28-Julho-2010.
- AWDUCHE, D., MALCOLM, J., AGOGBUA, J., O'DELL, M. e MCMANUS, J. **Requirements for Traffic Engineering Over MPLS**. Technical Report RFC 2702, Internet Engineering Task Force and Internet Engineering Steering Group, september 1999. URL <http://www.ietf.org/rfc/rfc2702.txt>. Online; acessado 28-Julho-2010.
- BERTSEKAS, R. G. D. *Data Networks*. Prentice-Hall, 2 edition, 1992.
- BLAKE, S., BLACK, D., CARLSON, M., DAVIES, E., WANG, Z. e WEISS, W. **An Architecture for Differentiated Services**. Technical Report RFC 2475, december 1998. URL <http://www.ietf.org/rfc/rfc2475.txt>. Online; acessado 28-Julho-2010.
- BRADEN, R., CLARK, D. e SHENKER, S. **Integrated Services in the Internet Architecture: an Overview**. Technical Report RFC 1633, June 1994. URL <http://www.ietf.org/rfc/rfc1633.txt>. Online; acessado 28-Julho-2010.
- CISCO. *How To Calculate Bandwidth Utilization Using SNMP*. Cisco Documentation, 2007a. URL http://www.cisco.com/application/pdf/paws/8141/calculate_bandwidth_snmp.pdf. Online; acessado 29-Julho-2010.
- CISCO. *Load Splitting IP Multicast Traffic over ECMP*. Cisco Documentation, 2007b. URL http://www.cisco.com/en/US/docs/ios/12_4t/ip_mcast/configuration/guide/mctlsplt.pdf. Online; acessado 29-Julho-2010.
- CISCO. *Configuring SNMP Support*. Cisco Documentation, 2009. URL http://www.cisco.com/en/US/docs/ios/12_2/configfun/configuration/guide/fcf014.pdf/. Online; acessado 29-Julho-2010.
- CISCO. *Cisco Corporate Overview*. Cisco Documentation, 2010. URL <http://newsroom.cisco.com/dlls/index.html>. Online; acessado 29-Julho-2010.
- COATES, A., HERO III, A., NOWAK, R. e YU, B. **Internet tomography**. *Signal Processing Magazine, IEEE*, 19(3):47–65, may 2002. ISSN 1053-5888.

- COLTUN, R. **The OSPF Opaque LSA Option**. Technical Report RFC 2370, july 1998. URL <http://www.ietf.org/rfc/rfc2370.txt>. Online; acessado 28-Julho-2010.
- COMER, D. E. **Interligação de Redes com TCP/IP**. Campus, 5 edition, 2006.
- ELWALID, A., JIN, C., LOW, S. e WIDJAJA, I. **MATE: MPLS adaptive traffic engineering**. Em *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, págs. 1300 –1309 vol.3, 2001.
- FORTZ, B., REXFORD, J. e THORUP, M. **Traffic engineering with traditional IP routing protocols**. *Communications Magazine, IEEE*, 40(10):118 – 124, oct. 2002. ISSN 0163-6804.
- FORTZ, B. e THORUP, M. **Internet traffic engineering by optimizing OSPF weights**. Em *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, págs. 519 – 528 vol.2, 2000.
- GISWIKI. **Dijkstra's algorithm**. Technical report, june 2010. URL http://en.giswiki.net/wiki/Dijkstra%27s_algorithm. Online; acessado 03-Julho-2010.
- KATZ, D., KOMPELLA, K. e YEUNG, D. **Traffic Engineering (TE) Extensions to OSPF Version 2**. Technical Report RFC 3630, september 2003. URL <http://www.ietf.org/rfc/rfc3630.txt>. Online; acessado 28-Julho-2010.
- MAHDAVI, J. e PAXSON, V. **IPPM Metrics for Measuring Connectivity**. Technical Report RFC 2498, september 1999. URL <http://www.ietf.org/rfc/rfc2498.txt>. Online; acessado 28-Julho-2010.
- MOY, J. **OSPF Version 2**. Technical Report RFC 2328, april 1998a. URL <http://www.ietf.org/rfc/rfc2328.txt>. Online; acessado 28-Julho-2010.
- MOY, J. T. **Anatomy of an Internet routing protocol**. Addison-Wesley, 1998b.
- SALLES, R. e ROLLA, V. **Efficient Routing Heuristics for Internet Traffic Engineering**. *Computer Communications*, págs. 1942–1952, June 2007.
- SALLES, R. e VIANNA, P. **Using Link-State Routing Protocols for Autonomic Network Management**. Em *LAACS'08*, September 2008.
- TANEMBAUM, A. S. **Redes de Computadores**. Campus, 4 edition, 2003.
- THOMAS, T. M. **OSPF network design solutions**. Cisco Press, 1998.
- VIANNA, P. J. V. **Aplicação da Tecnologia de Roteamento de Estado de Enlace em Projetos de Sistemas Autônomos Auto-balanceados para Engenharia de Tráfego**. Dissertação de Mestrado, Instituto Militar de Engenharia, 2008.

WANG, N., HO, K., PAVLOU, G. e HOWARTH, M. **An overview of routing optimization for internet traffic engineering.** *Communications Surveys Tutorials, IEEE*, 10(1):36 –56, first 2008. ISSN 1553-877X.

ZABBIX. **ZABBIX Manual 1.6.** ZABBIX SIA, 2008. URL www.zabbix.com/downloads/ZABBIX%20Manual%20v1.6.pdf. Online; acessado 12-Outubro-2009.

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)