

**UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS TECNOLÓGICAS
ENGENHARIA DE TELECOMUNICAÇÕES**

**OUTROS TRABALHOS EM:
www.projetedereDES.com.br**

**IMPLEMENTAÇÃO E MEDIÇÕES DE BALANCEAMENTO DE CARGA ATRAVÉS
DE ROTEAMENTO EM REDES IP**

SAMUEL SOARES DA CUNHA

**BLUMENAU
2010**

SAMUEL SOARES DA CUNHA

**IMPLEMENTAÇÃO E MEDIÇÕES DE BALANCEAMENTO DE CARGA ATRAVÉS
DE ROTEAMENTO EM REDES IP**

Trabalho de Conclusão de Curso apresentada ao Programa de Graduação em Engenharia de Telecomunicações do Centro de Ciências Tecnológicas da Universidade Regional de Blumenau, como requisito parcial para a obtenção do grau de Engenheiro de Telecomunicações.

Prof. Francisco Adell Péricas, Ms. – Orientador

**BLUMENAU
2010**

Aos meus pais

AGRADECIMENTOS

Especialmente aos meus pais que, durante toda a minha vida, de maneira incansável me apoiaram e deram todo o suporte para o meu crescimento intelectual e profissional.

Aos meus irmãos, Juliano, Cauê e Malu pelo eterno apoio dedicado a mim.

À minha namorada Karina que sempre esteve ao meu lado durante o processo de elaboração deste trabalho, com carinho e paciência.

Aos meus amigos e colegas de trabalho.

Ao Professor Francisco Adell Péricas pela excelente orientação fornecida durante a elaboração deste trabalho.

E aos professores e profissionais do Departamento de Engenharia Elétrica e de Telecomunicações pelo suporte dado.

RESUMO

O presente trabalho busca apresentar as principais técnicas de balanceamento de carga através de roteamento em redes IP, reunindo informações de fundamentação e de testes práticos dos protocolos de roteamento RIP (*Routing Internet Protocol*), OSPF (*Open Shortest Path First*), EIGRP (*Enhanced Interior Gateway Routing Protocol*) e BGP (*Border Gateway Protocol*), suas formas de configuração e respectivas técnicas de balanceamento de carga, a fim de servir como uma referência para o estudo e análise em modelos de implementação prática. Para o desenvolvimento deste trabalho optou-se por utilizar *hardwares* e *softwares* da Cisco Systems (ou baseados em modelos Cisco), escolha motivada principalmente pela grande disponibilidade de equipamentos da marca. Os resultados demonstraram importantes diferenças entre as técnicas, sendo que o conhecimento destas é extremamente importante na tomada de decisões em processos de implantação práticos.

Palavras Chaves: Roteamento, Balanceamento de Carga, RIP, OSPF, EIGRP, BGP, IP.

ABSTRACT

This study presents the main techniques of load balancing through routing in IP networks, gathering information from basic fundamentals and practical tests of the routing protocols RIP (Routing Internet Protocol), OSPF (Open Shortest Path First) EIGRP (Enhanced Interior Gateway Routing Protocol) and BGP (Border Gateway Protocol), their ways of setting and their methods of load balancing in order to serve as a reference for the study and analysis of models for practical implementation. To develop this work, the use hardware and software from Cisco Systems (or model-based Cisco) was chosen, choice motivated primarily by the wide availability of equipment of the brand. The results showed significant differences between the techniques, meaning that the knowledge of which is extremely important in the decision-making process of implementation.

Keywords: Routing, Load Balancing, RIP, OSPF, EIGRP, BGP, IP.

SUMÁRIO

LISTA DE FIGURAS	x
LISTA DE TABELAS	xi
LISTA DE ABREVIACÕES	xii
1 INTRODUÇÃO	14
1.1 JUSTIFICATIVA.....	14
1.2 OBJETIVO GERAL.....	15
1.3 OBJETIVOS ESPECÍFICOS.....	15
1.4 ESTRUTURA DO TRABALHO.....	16
2 REDES DE COMPUTADORES E INTERNET	17
2.1 INTERNET.....	17
2.2 MODELO DE REFERÊNCIA TCP/IP.....	18
2.3 PROTOCOLO IP.....	19
2.3.1 Formato do Datagrama IP	19
2.3.2 Endereçamento IP	21
2.3.3 Sub-redes	23
2.4 ALGORITMOS DE ROTEAMENTO.....	23
2.4.1 Vetor de Distâncias	25
2.4.2 Estado de Enlace	26
2.5 CONTROLE DE CONGESTIONAMENTO.....	27
2.6 NAT.....	29
2.7 ROTEAMENTO.....	29
2.7.1 RIP	31
2.7.2 OSPF	32
2.7.3 IGRP	33
2.7.4 EIGRP	33
2.7.5 BGP	35
2.8 BALANCEAMENTO DE CARGA ATRAVÉS DE ROTEAMENTO EM REDES IP ..	36
3 ESTUDO DE CASOS	39
3.1 MODELO 1.....	39
3.2 MODELO 2.....	40
3.3 MODELO 3.....	41
4 IMPLEMENTANDO O ROTEAMENTO	43

4.1 IMPLEMENTANDO O MODELO 1	44
4.1.1 MODELO 1 COM RIPv2	46
4.1.1.1 Configuração do RIPv2	46
4.1.2 MODELO 1 COM OSPF	50
4.1.2.1 Configuração do OSPF	50
4.1.3 MODELO 1 COM EIGRP	55
4.1.3.1 Configuração do EIGRP	55
4.1.4 Conclusão Modelo 1	61
4.2 IMPLEMENTANDO O MODELO 2	61
4.2.1 Balanceamento de Carga – Modelo 2 com RIPv2, OSPF e EIGRP	62
4.2.2 Conclusão Modelo 2	64
4.3 IMPLEMENTANDO O MODELO 3	65
4.3.1 Análise de caso usando o modelo 3.	65
4.3.2 Controlando o tráfego com BGP-4.....	67
4.3.2.1 Modelos de Conexão <i>Multihoming</i>	68
4.3.3 Configuração do BGP-4	69
4.3.3.1 Resumo das Configurações	74
4.3.4 Anúncio das Rotas	74
4.3.5 Balanceamento de Carga – Modelo 3 com BGP-4.....	77
4.3.6 Conclusão Modelo 3	80
5 CONCLUSÃO.....	81
REFERÊNCIAS	82

LISTA DE FIGURAS

Figura 2.1 – Modelo de Referência TCP/IP	18
Figura 2.2 – Formato do Datagrama IP	20
Figura 2.3 - Formato das Classes de Endereços IP	22
Figura 2.4 – Exemplo de topologia utilizando sub-redes.	23
Figura 2.5 – Modelo Abstrato de Grafo de uma Rede de Computadores	26
Figura 3.1 – Topologia de Rede do Modelo 1	39
Figura 3.2 – Topologia de Rede do Modelo 2	40
Figura 3.3 – Topologia de Rede do Modelo 3	41
Figura 4.1 – Topologia de rede do modelo 1 aplicada aos testes práticos	44
Figura 4.2 – Tela do <i>Software Free Download Manager</i>	49
Figura 4.3 – Tela do <i>Software Free Download Manager</i>	60
Figura 4.4 – Topologia de rede do modelo 2 aplicada aos testes práticos.	61
Figura 4.5 – Topologia de rede do modelo 3 aplicada aos testes práticos.	66
Figura 4.6 – Modelo de Conexão <i>Multihoming</i> - Apenas Rotas <i>Default</i>	68
Figura 4.7 – Modelo de Conexão <i>Multihoming</i> –Tabelas Parciais	68
Figura 4.8 – Modelo de Conexão <i>Multihoming</i> - <i>Full Route</i>	69
Figura 4.9 – Mapa parcial da rede conectada ao AS 28670.	77
Figura 4.10 – Uso do <i>link</i> entre REDEL_BGP e OP1 no período de 24 horas	79
Figura 4.11 – Uso do <i>link</i> entre REDEL_BGP e OP2 no período de 24 horas	79
Figura 4.12 – Uso total dos <i>link</i> entre REDEL_BGP e OP1 e REDEL_BGP e OP2 (eth0) no período de 24 horas	79

LISTA DE TABELAS

Tabela 2.1 – Informações de Referência Sobre as Classes de Endereços IP.....	22
Tabela 2.2 – Execução do Algoritmo de Estado de Enlace na Rede da Fig. 2.5	27
Tabela 4.1 – Configuração das Interfaces <i>Gigabit Ethernet</i> do Roteador REDEL_BGP	66
Tabela 4.2 – Distribuição dos Endereços e Sub-blocos IPs do Bloco 189.8.104.0/21 (OP2) ..	78
Tabela 4.2 – Distribuição dos Endereços e Sub-blocos IPs do Bloco 189.8.96.0/21 (OP1)	78

LISTA DE ABREVIACÕES

AS - Autonomous System
ASN - Autonomous System Number
BGP-4 - Border Gateway Protocol version 4
CIDR - Classless Inter-Domain Routing
DCE - Data Communications Equipment
DNS - Domain Name Server
DR - Designed Router
DTE - Data Terminal Equipment
EGP - Exterior Gateways Protocol
EIGRP - Enhanced Interior Gateway Routing Protocol
FTP - File Transfer Protocol
FWA - Fixed Wireless Access
HTTP - Hypertext Transfer Protocol
IGP - Interior Gateway Protocols
IGRP - Interior Gateway Routing Protocol
IHL - IP Header Length
IOS - Internetwork Operating System
IP - Internet Protocol
ISP - Internet Service Provider
IS-IS - Intermediate System To Intermediate System
LAN - Local Area Network
LSA - Link State Advertisement
MTU - Maximum Transmit Unit
NAT - Network Address Translation
OSPF - Open Shortest Path First
RED - Random Early Detection
RFC - Request for Comments
RIPv2 - Routing Information Protocol version 2
SMTP - Simple Mail Transfer Protocol
SPF - Short Path First
SVA - Serviços de Valor Adicionado
TCP - Transmission Control Protocol

TOS - Type-of-Service

TTL - Time-to-live

UDP - User Datagram Protocol

VLSM - Variable Length Subnet Masking

WAN - Wide Area Network

1 INTRODUÇÃO

Com o aumento da demanda de serviços de Internet, criada pelos usuários e sua crescente necessidade de trocar informações entre si, é cada vez mais importante a implementação de redes estáveis com maior qualidade e velocidade.

Atualmente, além dos fornecedores dos serviços de comunicação, as empresas e instituições usuárias destes serviços têm a necessidade de fazer uso da Internet e principalmente dos recursos de suas redes de computadores, 24 horas por dia, sem interrupções, tornando prática comum a implementação de redundância nestes recursos e nos *links* da rede. Nos modelos mais simples, o *link* redundante é utilizado apenas quando o principal falhar, deixando o *link* secundário ocioso na maior parte do tempo.

A implementação de técnicas de balanceamento de carga vem justamente ao encontro destes problemas, já que a utilização de múltiplos *links* e caminhos simultaneamente podem aumentar significativamente a disponibilidade e a qualidade da comunicação.

Além disso, o bom funcionamento das redes de telecomunicações depende de uma série de fatores e condições de operação, como padrão de tráfego, nível de carga e a topologia da rede. Da mesma forma que os serviços evoluem e as redes crescem, alteram-se as necessidades e as condições de operação, criando um ambiente dinâmico e complexo. De todos os processos responsáveis pela transmissão das informações, os sistemas de roteamento têm grande importância e merecem especial destaque na busca por melhor desempenho e qualidade.

Nas redes WAN os dados são enviados de um ponto (computador) ao outro, através de diversos caminhos e roteadores, sendo divididos em pacotes que podem ser roteados de forma independente, pelo melhor caminho, através de algoritmos de roteamento (TANENBAUM, 2003, p. 22). É importante destacar que há diferentes maneiras de direcionar e balancear o tráfego de uma rede, sendo que nem sempre o termo balanceamento significa fazer uso dos recursos de forma igualitária e sim de forma otimizada, mais rápida e dinâmica. A escolha do melhor caminho baseia-se neste princípio, fazendo uso de técnicas combinadas como algoritmos adaptativos, marcação de pacotes, mapeamento da rede, entre outras. Tudo isso, é claro, deve preceder à escolha adequada de um protocolo de roteamento e até mesmo das interfaces e meios de transmissão.

1.1 JUSTIFICATIVA

Com significativos avanços, diversas técnicas aplicadas ao roteamento de pacotes se consolidaram e vêm sendo usadas em larga escala. A implementação de balanceamento de carga através de roteamento em redes IP necessita de um estudo, visando adequar estas técnicas às diferentes necessidades de cada rede.

Da mesma forma que uma rede bem balanceada pode trazer benefícios, um balanceamento de cargas mal dimensionado pode trazer mais prejuízos do que vantagens. Toda vez que uma interface ou *link* trabalha sobrecarregada é inevitável a perda de dados e pacotes causados pelas colisões, derrubando o desempenho da transmissão significativamente. Além disso, de nada adianta uma rede com ótimos recursos se eles não puderem ser utilizados ao máximo.

1.2 OBJETIVO GERAL

Identificar os problemas e respectivas soluções na implementação de balanceamento de carga em redes, através de um estudo sobre roteamento em redes IP e de uma avaliação comparativa entre os casos de uso.

1.3 OBJETIVOS ESPECÍFICOS

- Estudo da arquitetura de rede, dos protocolos TCP e IP e de sub-redes;
- Estudo das técnicas e algoritmos de roteamento;
- Especificar os algoritmos de controle de congestionamento;
- Avaliar as técnicas e formas de balanceamento de carga através de roteamento em redes IP;
- Verificação das técnicas de balanceamento de carga na implementação de estudos de caso.

1.4 ESTRUTURA DO TRABALHO

O trabalho está estruturado em cinco capítulos.

O capítulo 1 inicia o trabalho com uma breve introdução e os objetivos do mesmo.

O capítulo 2 apresenta toda a fundamentação do trabalho, com o estudo do modelo de referência TCP/IP, do protocolo IP e de sub-redes, dos algoritmos de roteamento e controle de congestionamento, além de informações sobre o processo de roteamento em redes IP, protocolos de roteamento e das formas e técnicas de balanceamento de carga através de roteamento em redes IP.

O capítulo 3 mostra um estudo de caso da implementação, indicando os modelos ou topologias que serão implementados e testados.

O capítulo 4 apresenta o processo de implementação, com as especificações e configurações dos modelos indicados, bem como as medições realizadas.

O capítulo 5 traz a conclusão do trabalho.

2 REDES DE COMPUTADORES E INTERNET

Desde o surgimento dos primeiros computadores, as pessoas têm se maravilhado com a velocidade dos avanços tecnológicos da informática. De todos os avanços, aquele que mais representa as inovações nos hábitos de uso e consumo proporcionados pelas tecnologias é a Internet. Em poucos anos, a Internet gerou fortunas, sustentou industriais, criou novas formas de consumo e revolucionou a comunicação em todo o mundo.

Segundo a TeleGeography (2010, p. 2), empresa especializada em pesquisa em diversas áreas de telecomunicação, a expansão da Banda de Internet a nível mundial tem acontecido de forma acelerada e estima-se que o crescimento do tráfego na Internet deve, aproximadamente, dobrar a cada dois anos, projetando uma capacidade de banda internacional de quase 40 Tbps em 2010. Isto só é possível graças aos avanços tecnológicos do setor, que devido à constante pesquisa, trouxeram um aumento considerável das redes, refletindo maiores capacidades, melhor qualidade e menor custo.

2.1 INTERNET

Sendo a rede mais conhecida, a Internet é o modelo ideal para o estudo das redes de computadores. A Internet, na verdade, é um conjunto de redes diferentes que utilizam diversos meios e tecnologias, conectadas entre si através de protocolos em comum (TANENBAUM, 2003, p. 53).

Há duas maneiras de descrever a Internet: a primeira é descrevê-la como os componentes básicos que a formam, *softwares* e *hardwares*; a outra maneira é descrevê-la como uma infra-estrutura de rede que provê serviços para aplicações distribuídas (KUROSE, 2005, p. 2). Para este trabalho, será estudada a sua infra-estrutura, sua forma de comunicação, seus protocolos e seu gerenciamento.

Segundo Kurose (2005, p.3), as redes de computadores e os equipamentos (sistemas finais) ligados à Internet não são conectados diretamente através de enlaces de comunicação, e sim indiretamente através de comutadores de pacotes, que são os equipamentos (roteadores e *switches*) que fazem a interligação entre os enlaces e definem as rotas que os pacotes irão percorrer entre os sistemas de origem e destino. Esta técnica é conhecida como comutação de pacotes.

A forma com que as redes de computadores trocam informações é definida por diversos protocolos. A comunicação na Internet é definida, principalmente, pelos protocolos

TCP (*Transmission Control Protocol*) e pelo IP (*Internet Protocol*), que formam o modelo de referência TCP/IP. Tanenbaum (2003, p. 60), define que “...a máquina está na Internet quando executa a pilha de protocolos TCP/IP, tem um endereço IP e pode enviar pacotes IP a todas as outras máquinas da Internet.”

2.2 MODELO DE REFERÊNCIA TCP/IP

TCP/IP refere-se tanto ao modelo de referência quanto aos dois protocolos que estão diretamente ligados formando o modelo: o protocolo de transporte TCP que fornece o serviço orientado a conexão do modelo e o protocolo de rede IP que fornece o serviço não orientado a conexão. Elaborado pela Internet Activity Board (IAB) é o modelo de referência usado na Internet, sendo formado por 4 camadas (SOARES, 1995, p. 142) de acordo com a Fig. 2.1.

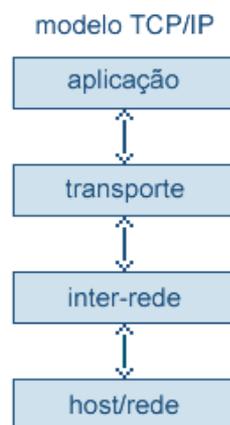


Figura 2.1 – Modelo de Referência TCP/IP
(Fonte: Tanenbaum, 2003)

Segundo Soares (1995, p. 144), a camada de aplicação contém os protocolos utilizados pelos programas de aplicação, tais como o SMTP (*Simple Mail Transfer Protocol*), o FTP (*File Transfer Protocol*) e o protocolo de terminal virtual (TELNET). Estes protocolos, por sua vez, fazem uso dos serviços fornecidos pela camada de transporte, sendo serviços orientados à conexão e fornecidos pelo protocolo TCP ou serviços não orientados à conexão fornecidos pelo UDP (*User Datagram Protocol*).

A camada de inter-rede é a principal camada da arquitetura TCP/IP, sendo a responsável pela interconexão de redes remotas sem conexão, injetando pacotes e entregando-os ao destinatário independentemente de sua ordem. A camada de inter-rede define oficialmente a implementação de um formato de pacote e do protocolo da Internet, IP, sendo

que seu papel é de grande importância no processo de roteamento dos pacotes. (TANENMAUM, 2003, p. 45).

A definição de endereços de rede feita pelo protocolo IP é feita de forma que as rotas possam ser definidas sistematicamente, fazendo-se uma comparação entre o endereço de origem e destino e aplicando máscaras de sub-rede.

Por fim, tem-se a camada de *host/rede*, que apesar de não ser muito bem especificada, é responsável pelo envio dos pacotes IP na rede fazendo a compatibilização das tecnologias de rede com o protocolo IP.

2.3 PROTOCOLO IP

O protocolo IP é o componente que mantém a Internet unida. Além de ser o mais importante componente da Internet, é o mais popular devido a sua capacidade e efetua a comunicação entre sistemas, independente de eles estarem na mesma rede ou em redes interconectadas, sendo que é igualmente funcional para redes LAN e WAN.

Sua tarefa é transferir datagramas do sistema de origem para o sistema de destino através da análise do endereço IP dos sistemas. O tamanho destes datagramas pode variar de acordo com a tecnologia de rede empregada, sendo que o protocolo IP deve fazer a fragmentação e remontagem dos datagramas (SOARES, 1995, p. 316).

Segundo Soares (1995, p. 316-317), algumas das principais características desse protocolo são:

- Serviço sem conexão/não confiável;
- Endereçamento hierárquico;
- Fragmentação e remontagem de datagramas;
- Identificação da importância e confiabilidade exigida do datagrama;
- Roteamento adaptativo;
- Checagem da integridade dos dados do cabeçalho (*checksum*);
- Controle do tempo de vida (*time to live*) dos pacotes;

2.3.1 Formato do Datagrama IP

O datagrama ou pacote é a unidade básica de dados no protocolo IP. Um datagrama está dividido em duas partes, uma parte de cabeçalho e outra de dados. O cabeçalho contém toda a informação necessária que identifica o conteúdo do datagrama, seu destino e origem.

Na parte de dados está encapsulado o pacote do nível superior (TANENMBAUM, 2003, p. 461).

De acordo com Kurose (2005, p. 256), o formato do datagrama IP é o seguinte:

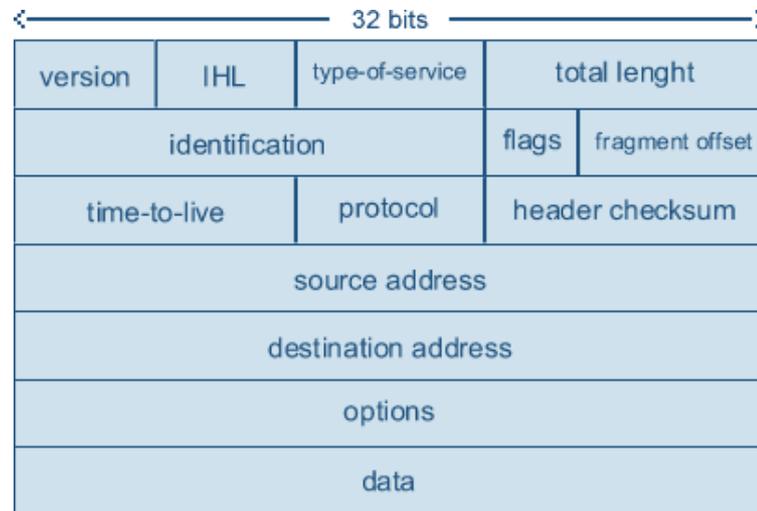


Figura 2.2 – Formato do Datagrama IP

(Fonte: Tanenbaum, 2003)

- **VERS (Version):** 4 *bits* que definem a versão do quadro, sendo usualmente IPv4 ou ainda IPv6.
- **IHL (IP Header Length):** indica o comprimento do cabeçalho.
- **TOS (Type-of-Service):** especifica o tipo de serviço, como o datagrama deve ser tratado e sua importância.
- **Total length:** define o comprimento total do datagrama IP em *bytes*, incluindo o cabeçalho e os dados.
- **Identification:** define a qual datagrama pertence um determinado fragmento. Todos os fragmentos de um mesmo datagrama possuem o mesmo valor de *Identification*.
- **Flags:** dividido em 3 *bits*. O primeiro não é utilizado e os dois seguintes controlam a fragmentação.
- **Fragment offset:** indica a localização do fragmento no datagrama.
- **TTL (Time-to-live):** mantém um contador que decresce a cada *hop* (salto), definindo o tempo de vida do pacote, eliminando-o ao atingir zero. Isso evita que o pacote trafegue na rede indefinidamente. Um aviso é enviado ao destinatário em caso de descarte.

- *Protocol*: indica qual protocolo da camada superior irá receber os pacotes após a sua chegada ao destinatário. Normalmente TCP e UDP.
- *Header checksum*: efetua uma soma de verificação do cabeçalho, identificando erros de *bits* em um datagrama IP.
- *Source address e Destination address*: indicam o endereço de origem e destino dos dados.
- *Options*: possibilita ao IP suportar diversas opções, tais como segurança, roteamento, etc.
- *Data*: contém os dados, a carga útil.

2.3.2 Endereçamento IP

O endereçamento IP é utilizado tanto para identificar computadores quanto redes e possui componentes específicos e um formato básico. Cada interface, seja ela de um computador ou de um roteador, ligada a uma rede TCP/IP, possui um endereço de IP único. No caso da Internet, os números são atribuídos pela ICANN (*Internet Corporation for Assigned Names and Numbers*) a ISPs (*Internet Service Providers*) e outras empresas (TANENBAUM, 2003, p. 464-465).

Utilizados nos campos *Source Address e Destination Address*, cada endereço possui 32 *bits*, separados em 4 octetos, sendo que inicia com o número de rede seguido pelo número de identificação do computador nesta rede (RFC 791, 1981, p. 6). Segundo a RFC 791, existem 5 formatos básicos para endereços na Internet, mas apenas 3 são utilizados para endereços *host* (computadores), ou “comercialmente”: classe A, onde o primeiro *bit* é zero seguido por 7 *bits* de rede e 24 *bits* de *host*; na classe B temos os dois primeiros *bits* com 1 e 0 respectivamente, seguido por 14 *bits* de rede e 16 *bits* de identificação do *host*; na classe C os três primeiros *bits* são 1, 1 e 0 respectivamente, seguidos por 21 *bits* de rede e 8 *bits* de *host*. As duas classes extras são: classe D utilizada para endereços de multidifusão e a classe E com endereços reservados para uso futuro. A Fig. 2.3, ilustra a divisão dos *bits* entre as classes.

n° de bits	1		7				24									
classe A	0		rede x				host x									
n° de bits	1		1		14				16							
classe B	1		0		rede x				host x							
n° de bits	1		1		1			21						8		
classe C	1		1		0			rede x						host x		

Figura 2.3 - Formato das Classes de Endereços IP

(Fonte: Cisco Systems, 1999)

O protocolo IP ainda especifica outras convenções de endereçamento. Um endereço IP com 0's no número de rede faz referência à própria rede e com 1's permite a difusão na rede local. Um endereço IP que possui 0's binários em todos os *bits* de *host* é reservado para a rede inteira e é chamado de endereço de rede. Como exemplo, em uma rede de classe A, temos o endereço 113.0.0.0 como endereço de rede. Os roteadores utilizam os endereços IP de uma rede ao encaminhar dados na Internet. Já para enviar dados para todos os *hosts* de uma rede é utilizado um endereço de *broadcast*, onde todos os *bits* de *host* do endereço são marcados com 1's binários. Por fim, há os endereços que iniciam com 127 (decimal) que são reservados para testes de *loopback*. Estes pacotes não são transmitidos e permitem que pacotes sejam enviados para o *host* local sem que o transmissor saiba seu número (TANENBAUM, 2003, p. 465-466). Mais detalhes na Tabela 2.1.

Tabela 2.1 – Informações de Referência Sobre as Classes de Endereços IP

(Fonte: Cisco Systems, 1999)

Classe	Propósito	Endereços	Endereços reservados	Número de bits Rede/Host	Máscara de sub-rede padrão
A	Grandes organizações	1.0.0.0 a 126.0.0.0	10.0.0.0 a 10.255.255.255	7/24	255.0.0.0
B	Organizações de médio porte	128.1.0.0 a 191.254.0.0	172.16.0.0 a 172.31.255.255	14/16	255.255.0.0
C	Organizações relativamente pequenas	192.0.1.0 a 223.255.254.0	192.168.0.0 a 192.168.255.255	21/8	255.255.255.0
D	Multicast	224.0.0.0 a 239.255.255.255	-	-	-
E	Experimental / Uso futuro	240.0.0.0 a 254.255.255.255	-	-	-

Cada classe ainda possui alguns endereços reservados para esquemas de endereçamento IP privado. Desta forma estes endereços (conforme coluna Endereços reservados da Tabela 2.1) não são atribuídos na Internet. Estes endereços são utilizados por computadores que não precisam de conectividade com a Internet ou os fazem através do NAT (*Network Address Translation*), que é uma técnica que reescreve o endereço IP de origem (*source address*) de um pacote, de forma que este tenha acesso a rede exterior.

2.3.3 Sub-redes

As redes IP podem ser divididas internamente em redes menores, chamadas de sub-redes, de forma que externamente funciona como uma única rede. As sub-redes foram criadas com o objetivo de trazer mais flexibilidade, eficiência no uso dos endereços e de diminuir o tráfego de *broadcast* (TANENBAUM, 2003, p. 466). Como exemplo, um endereço IP classe B, 176.1.0.0, pode ser dividido em diversas sub-redes, conforme ilustrado na Fig. 2.4.

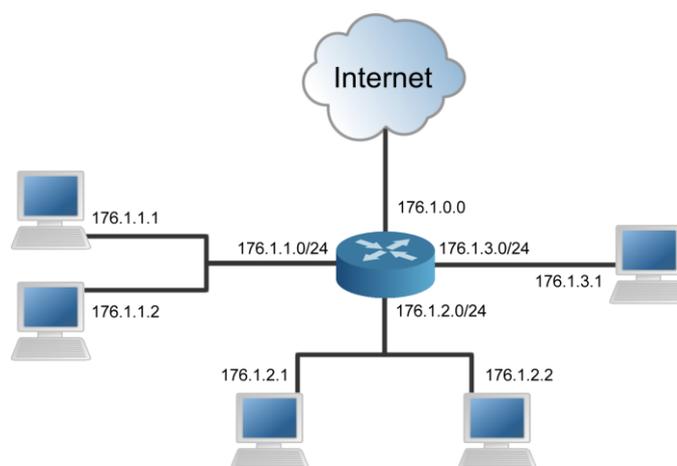


Figura 2.4 – Exemplo de topologia utilizando sub-redes.

(Fonte: Google Imagens, 2009)

A designação da sub-rede é feita através de uma máscara de sub-rede, que indica quais *bits* irão designar a parte de rede, sub-rede e *host* em um endereço. A máscara de sub-rede pode ser escrita de duas maneiras, como no exemplo da Fig. 2.4, 176.1.1.0/24, onde /24 indica que os 24 *bits* mais à esquerda definem endereços da sub-rede ou em notação decimal com pontos da mesma forma que o endereço IP (255.255.255.0 para o caso do exemplo).

2.4 ALGORITMOS DE ROTEAMENTO

Os algoritmos de roteamento são os responsáveis pelas decisões dos caminhos ou rotas que cada pacote deve seguir em uma transmissão. Existe um grande número de algoritmos de roteamento desenvolvidos, com propósitos diferenciados, causando diferentes impactos nos recursos de rede e de roteadores. De qualquer forma os algoritmos de roteamento geralmente possuem as seguintes propriedades: otimização, robustez, simplicidade, estabilidade e convergência. Duato (1997, p.115), define algumas das propriedades das redes de interconexão que são fornecidas pelos algoritmos de roteamento, conforme segue:

- Conectividade: habilidade de roteamento de pacotes de qualquer destino para qualquer origem;
- Adaptatividade: habilidade de rotear pacotes através de diferentes rotas, conforme relação entre a demanda e os recursos disponíveis ou falhas na rede;
- Controle de *deadlocks* e *livelocks*: habilidade de garantir que os pacotes não bloqueiem uns aos outros ou vaguem indefinidamente na rede;
- Tolerância a falhas: habilidade de rotear pacotes na presença de falhas de componentes. Além da adaptatividade, o roteamento dos pacotes pode ser feito em fases, armazenando-os em nós intermediários.

Os algoritmos de roteamento podem ser classificados de acordo com diversos critérios, tais como o destino dos pacotes, como são tomadas as decisões de roteamento, a forma com que são implementados e a forma mais utilizada pela sua adaptabilidade: estáticos ou dinâmicos.

Os algoritmos estáticos ou não adaptativos têm as rotas calculadas na sua inicialização, sendo que as mesmas quase não mudam ao longo do tempo. Como estes algoritmos não se adaptam às constantes alterações da rede, sendo que a intervenção nas tabelas muitas vezes é feita manualmente, são muito pouco utilizados atualmente.

Os algoritmos dinâmicos ou adaptativos utilizam informações de tráfego, topologia, além das condições e características dos enlaces para determinar o melhor caminho, evitando congestionamentos ou falhas em certas partes da rede.

De um ponto de vista de engenharia, a análise do algoritmo mais indicado deve levar em consideração o impacto que o mesmo causará no projeto, de forma que um algoritmo de roteamento mais complexo possa aumentar a utilização de canal em detrimento de um maior

atraso e de uma menor frequência de relógio (*clock*). É importante destacar que o objetivo é maximizar o desempenho, e de certa forma a relação desempenho/benefício (DUATO, 1997, p. 116-167).

A seguir fez-se um estudo dos algoritmos de roteamento mais utilizados. Para tal, uma de medida de “custo” para os enlaces ou rotas da rede será amplamente utilizada. Os custos são medidas métricas utilizadas, tais como: comprimento do enlace, velocidade de transmissão, largura de banda (capacidade de transmissão), retardo de tempo, número de saltos, segurança do enlace, entre outros. A escolha do parâmetro a ser utilizado vai depender da aplicação do algoritmo. Por exemplo, enlaces com elevado tempo de propagação terão um custo elevado para tráfegos que requerem baixo tempo de resposta.

2.4.1 Vetor de Distâncias

No roteamento baseado em vetor de distância (*distance vector*), cada roteador mantém uma tabela atualizada pelos seus roteadores vizinhos (*neighbors*), com as informações sobre o *status* da rede e modificações na topologia. É um algoritmo distribuído, já que um nó (roteador) recebe e envia o resultado de seus cálculos aos nós ligados diretamente a ele.

Também conhecido por algoritmo de Bellman-Ford, este algoritmo é baseado na equação de mesmo nome, como segue:

$$D_x(y) = \min_v \{ c(x,y) + D_v(y) \}$$

onde, segundo Kurose (2005, p. 281-282), $D_x(y)$ é o vetor de distância do nó x até o nó y , ou seja o custo do caminho de menor custo de x até y , \min_v é calculado para todos os vizinhos de x , onde $c(x,y)$ é o custo de x em relação ao nó vizinho v e $D_v(y)$ o vetor de distância do vizinho v ao nó y (informação repassada pelos roteadores vizinhos ao nó x). Por fim, o cálculo $D_x(y)$ é repassado por x aos seus vizinhos.

Desta forma o roteador mantém uma tabela, com a saída preferencial para o destino e uma estimativa de tempo ou da distância até o destino (custo). A unidade métrica utilizada no vetor de distância pode ser o número de *hops* (saltos), o retardo de tempo, a fila de pacotes no caminho, entre outros.

O algoritmo de Bellman-Ford é utilizado nos roteamentos RIP e BGP, sendo que tem sido amplamente substituído devido a problemas como a contagem até o infinito, lenta

convergência, entre outros. Além disso, os algoritmos de vetor de distância possuem informações não específicas sobre redes distantes e nenhum conhecimento sobre roteadores distantes.

2.4.2 Estado de Enlace

Nos algoritmos de roteamento baseados em estado de enlace (*link estate*), a topologia e os custos de toda a rede são conhecidos e armazenados em um banco de dados, ou seja, “... estão disponíveis como dados para o algoritmo...” (KUROSE, 2005, p. 278).

Desta forma, cada roteador da rede tem um conhecimento completo sobre roteadores distantes e não somente sobre seus vizinhos e a forma como estão interconectados. Para isso, cada roteador que utiliza o algoritmo de estado de enlace, inicia conhecendo o caminho de menor custo até seus vizinhos. Em seguida o roteador envia estas informações para todos os roteadores da rede (*broadcast*), através dos chamados pacotes LSA (*Link State Advertisement*). Com estas informações o algoritmo constrói uma árvore SPF (*Short Path First*), com todos os caminhos da rede, suas saídas e custos, podendo, então, escolher o caminho mais “curto”.

Kurose (2005, p. 279) exemplifica bem a forma com que o algoritmo monta esta tabela de roteamento e determina o caminho de menor custo. Considerando a Fig. 2.5, calcula-se o caminho de menor custo de *u* até todos os destinos da rede. Podem-se descrever os estágios deste processo da seguinte forma:

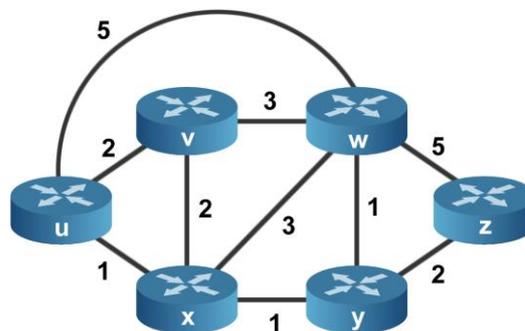


Figura 2.5 – Modelo Abstrato de Grafo de uma Rede de Computadores

(Fonte: Kurose, 2005)

- Considerando $D(v)$ o custo do caminho de menor custo entre a fonte e o nó v , $p(v)$ o nó anterior ao longo deste caminho e N' o subconjunto de nós.

- Na primeira etapa são conhecidos os caminhos de menor custo de **u** até seus vizinhos. Note que os custos dos nós **y** e **z** são estabelecidos como infinito, já que **u** não está diretamente ligado a eles e ainda não sabe o melhor caminho para alcançá-los.
- A seguir são examinados os nós que ainda não foram adicionados a **N'**. Neste caso, analisando **x**, observa-se que o custo até **y** até **x** é de 2. Além disso, descobre-se que o custo de **w** é menor pelo caminho do nó **x**. Este caminho é selecionado e substitui o caminho escolhido anteriormente.
- Este processo é efetuado para todos os nós até que o conjunto **N'** esteja completo, ou seja, que **u** conheça o melhor caminho até todos os destinos possíveis.

Tabela 2.2 – Execução do Algoritmo de Estado de Enlace na Rede da Fig. 2.5

(Fonte: Kurose, 2005)

Etapa	N	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(z)
0	u	2,u	5,u	1,u	∞	∞
1	ux	2,u	4,x		2,x	∞
2	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					

Os algoritmos baseados em estado de enlace são amplamente utilizados nas redes atuais. Um dos protocolos mais importantes que faz uso destes algoritmos é o protocolo OSPF.

2.5 CONTROLE DE CONGESTIONAMENTO

Quando a quantidade de pacotes depositados em uma sub-rede é superior a sua capacidade de transporte tem-se um congestionamento, causando uma degradação no desempenho da rede. Em casos extremos, a rede pode entrar em colapso total, causando um *deadlock* (impasse) e a perda de muitos pacotes.

Existem várias técnicas possíveis para se conseguir um controle do congestionamento e equidade do tráfego, traduzidas em algoritmos de controle de congestionamento. De certa forma o controle de congestionamento é feito em diversos níveis, desde o controle de fluxo da camada de enlace até a principal responsável pelo controle de congestionamento, a camada de transporte. O controle de congestionamento que se irá

estudar, baseia-se no tráfego da sub-rede, seus computadores e roteadores (TANENBAUM, 2003, p. 409-413).

Uma forma de identificar congestionamentos é analisar cada linha de saída em um roteador, monitorando o seu uso e o uso de seus recursos, exemplificado com a seguinte equação:

$$u_{nova} = au_{antiga} + (1 - a)f$$

Onde cada linha de saída recebe uma variável u que reflete a utilização recente desta linha (com valor entre 0,0 e 1,0) e é atualizada periodicamente, f é uma amostra da utilização instantânea da linha e a determina com que velocidade o roteador se esquece dos acontecimentos mais recentes. Com isso, o roteador pode colocar uma linha em sinal de advertência, sempre que u ultrapassar um valor pré-determinado, ou seja, que a linha estiver congestionada. Quando isto ocorrer, o roteador deverá adotar alguma ação específica para solucionar o problema. Algumas alternativas são:

- *Bit* de advertência: um *bit* especial é assinalado pelo destino e enviado de volta à origem indicando a advertência. A origem regula a velocidade de transmissão de acordo com o intervalo entre os *bits*;
- Pacotes reguladores: o destino encaminha à origem um pacote regulador avisando sobre o congestionamento. O destino reduzirá o tráfego durante um período determinado de tempo. Somente após o término deste tempo, o computador de origem irá aceitar outros pacotes reguladores, que caso continuem a chegar, obrigarão o computador de origem a reduzir ainda mais o tráfego;
- Pacotes reguladores *hop a hop*: em enlaces de alta velocidade ou de longa distância, o pacote regulador pode demorar a chegar à origem. A solução é identificar os pacotes reguladores e reduzir o tráfego a cada *hop* (salto);
- Escoamento de carga: quando muitos pacotes chegam a um roteador e ele não pode armazená-los, estes podem simplesmente ser descartados. Contudo, este processo deve ser feito com cuidado, já que nem todos os pacotes podem ser descartados e para que o algoritmo descarte primeiramente os menos importantes;
- Detecção aleatória prematura: esta técnica consiste em detectar o congestionamento antes que ele ocorra. Baseando-se em probabilidades, o algoritmo pode descartar pacotes sem que a origem precise reduzir a transmissão

imediatamente. Um algoritmo muito utilizado que utiliza esta técnica é RED (*Random Early Detection*);

- Controle de *jitter*: em transmissões em que o mais importante é a variação do tempo de chegada dos pacotes (flutuação) e não o tempo por si só (desde que o tempo seja viável), algoritmos de controle de *jitter* podem ser implementados, ajustando o tempo de armazenamento em *buffer* dos pacotes e a prioridade de transmissão de pacotes atrasados (TANENBAUM, 2003, p. 409 – 422).

2.6 NAT

Com a rápida evolução das redes de computadores e da Internet, um problema, inicialmente não identificado, apareceu: a escassez de endereços IPv4. Segundo Tanenbaum (2003, p.474), a solução a longo prazo é migrar a Internet inteira para endereços IPv6. Mas como o problema já está ocorrendo, uma solução temporária foi criada, a tradução de endereços de rede (*Network Address Translation – NAT*). O processo NAT possibilita que uma LAN com diversos computadores, utilize apenas um endereço de IP válido na Internet, atribuindo aos computadores da LAN endereços de IP reservados.

De forma resumida, podemos explicar o processo conforme segue: quando um determinado pacote é enviado de uma rede local para a Internet, seu endereço de IP de origem é alterado pelo roteador para um endereço de IP válido ao qual o roteador está conectado com a Internet. Neste processo o roteador armazena o IP reservado de origem, o IP de destino e as portas de origem e destino (TCP ou UDP) de cada pacote, sendo que através desta informação pode identificar o computador quando ocorre a resposta.

2.7 ROTEAMENTO

O processo de roteamento consiste basicamente em transportar informações de um ponto a outro através de uma rede. Filipetti (2006, p. 169) descreve o processo de modo geral como “Um conjunto de regras que definem como dados originados em uma determinada sub-rede devem alcançar uma outra”. Este processo é efetuado pelos roteadores, também chamados de *gateways*, que fazem a leitura dos cabeçalhos dos pacotes IP e utilizando as tabelas e os algoritmos de roteamento determinam as rotas ou caminhos que os pacotes irão percorrer. De modo mais detalhado o processo de roteamento IP ocorre da seguinte forma:

- O computador que deseja enviar um pacote verifica se o destino está na mesma sub-rede ou se está em outra sub-rede. Se estiver na mesma rede a comunicação ocorre diretamente entre os computadores, sem a utilização do roteador. Caso contrário, o pacote é encaminhado ao roteador que irá determinar o destino do pacote.
- O roteador armazena o pacote em seu *buffer* e verifica em sua tabela de roteamento se a rede do computador de destino está ligada diretamente a ele ou se deverá encaminhar o pacote a outro roteador. A tabela de roteamento ainda fornece qual a interface (canal) que está conectada à rede ou o melhor caminho até a rede de destino, neste caso o próximo salto.
- O roteamento IP especifica que o pacote deve viajar na rede, um salto por vez, sendo que sua rota é calculada em cada parada, ou seja, em cada roteador.
- Para determinar o endereço de rede de destino o roteador executa uma operação lógica *AND* utilizando a máscara de sub-rede, removendo a parte de *host* do endereço IP de destino, ficando apenas com o endereço de rede.
- Ao final, o roteador que está ligado à sub-rede do computador de destino irá entregar o pacote diretamente ao computador.
- É importante destacar que durante todo este processo, os endereços IP de destino e origem não são alterados, bem como todo o pacote de dados. Apenas os frames e os endereços físicos (*MAC address*) sofrem alterações.

O encaminhamento entre os roteadores é talvez o processo mais importante no roteamento de pacotes. Existem diferentes maneiras de se manter as tabelas de roteamento e definir a melhor rota que um pacote irá percorrer. Estas definições ocorrem através do modo de roteamento escolhido que pode ser estático, dinâmico ou *default*. As definições de roteamento estático e dinâmico são praticamente as mesmas dos algoritmos de roteamento classificados desta mesma maneira, conforme se viu anteriormente.

O processo de roteamento dinâmico utiliza protocolos de roteamento que definem as regras que um roteador irá utilizar ao se comunicar com seus vizinhos, fazendo a montagem e atualização das tabelas de roteamento. Estes protocolos são baseados e utilizam os algoritmos de roteamento dinâmicos.

De qualquer forma, é muito praticado o uso de rotas estáticas, mesmo em redes que utilizam protocolos dinâmicos. Isto ocorre por questões de segurança, já que parte do controle do processo de roteamento fica nas mãos do administrador da rede, tendo em vista que este

pode desejar ocultar partes de uma *internetwork*. Além disso, quando uma rede está disponível apenas por um caminho, uma rota estática será suficiente. Conhecidas como redes *stub*, a configuração destas rotas evita sobrecargas de roteamento dinâmico.

Além da definição de roteamento estático e roteamento dinâmico, o roteador ainda possui a opção de roteamento *default*. No roteamento *default*, os pacotes com destino desconhecido serão sempre encaminhados para uma rota padrão, ao invés de serem simplesmente descartados. Essa rota pode ser uma interface de saída conectada a um roteador que conhece as rotas para aquele destino (FILIPETTI, 2006, p. 169-194).

A Internet é um conjunto de redes interconectadas. Os responsáveis por estas interconexões são justamente os roteadores que, no caso da Internet, são organizados de forma hierárquica. No topo desta hierarquia estão os Sistemas Autônomos (AS – *Autonomous System*), que podem ser descritos como um grupo de roteadores (domínios) sobre uma mesma administração técnica, utilizando protocolos de roteamento interno (IGPs – *Interior Gateway Protocols*), enquanto ASs distintos utilizam protocolos de roteamento externos (EGPs – *Exterior Gateways Protocol*) para rotear pacotes entre si (RFC 1930, 1996, p.3).

Dentre os diversos protocolos de roteamento disponíveis, serão estudados o BGP, como protocolo de roteamento externo e os seguintes protocolos roteamento interno: RIP, OSPF, EIGRP e IGRP

2.7.1 RIP

O RIP (*Routing Information Protocol* – Protocolo de Informação de Roteamento) foi um dos primeiros protocolos, sendo ainda muito utilizado, principalmente em pequenas redes, devido ao fato de possuir baixo *overhead* em termos de largura de banda utilizada, além de ser de fácil implementação.

O RIP é um protocolo vetor distância e utiliza a contagem de saltos como métrica, sendo 15 o número máximo de saltos. Para determinar o melhor caminho até uma determinada sub-rede, o RIP avalia o número de sub-redes percorridas até a sub-rede de destino.

Um roteador utilizando o RIP mantém as suas tabelas de roteamento trocando informações com os seus vizinhos. Cada roteador envia ao seu vizinho, a cada 30 segundos, uma relação com até 25 sub-redes de destino, juntamente com a distância entre ele e estes destinos. (KUROSE, 2005, p. 292).

O problema mais comum ao RIP é o *loop* de roteamento. Se um roteador não ouvir nada de seu vizinho ao menos uma vez a cada 180 segundos, este roteador e as sub-redes conectadas a ele serão considerados inalcançáveis e o roteador enviará esta informação aos seus vizinhos. O problema ocorre quando um roteador que ainda não convergiu, ou seja, ainda não ficou sabendo que aquela devida rede não está operando, envia uma informação para outro que já havia recebido a informação, informando que ele sabe o caminho para aquela rede, atualizando assim a tabela deste roteador com uma informação errada, gerando um *loop*. Para solucionar este problema o protocolo utiliza uma técnica chamada de *split-horizon*, que basicamente faz com que um roteador não envie novamente uma informação sobre uma rota, através de uma interface que tenha aprendido uma informação sobre aquela rota.

Devido a algumas deficiências do RIP, em 1998 foi padronizada a versão 2 do protocolo RIP, chamada de RIPv2. A principal diferença da nova versão com relação a anterior é incluir o suporte à troca de informações de sub-redes, sendo este um protocolo *classless*. O RIPv2 é compatível com o RIP.

2.7.2 OSPF

O OSPF (*Open Shortest Path First*), a exemplo do RIP, é um protocolo aberto, disponível ao público, sendo que pode ser desenvolvido e aperfeiçoado por vários fabricantes. É um dos mais importantes protocolos que utilizam algoritmos de roteamento estado de enlace, também conhecidos como algoritmos SPF e é comumente utilizado (juntamente com o seu “primo”, o IS-IS) em ISPs de níveis mais altos, já que sua implementação é mais complexa.

Os roteadores de uma rede que utilizam o OSPF identificam os roteadores vizinhos e, então, se comunicam com eles. As informações recebidas dos roteadores vizinhos possibilitam a criação de um banco de dados *link-state*. O roteador então constrói um mapa topológico completo de toda a rede, utilizando um algoritmo de Dijkstra para determinar os caminhos mais curtos para todas as sub-redes, utilizando uma métrica baseada na largura de banda do caminho (KUROSE, 2005, p. 294).

Outra característica do OSPF é a capacidade de designação de um roteador, em um determinado segmento, que fará a comunicação, ou seja, o envio das informações de *link-state* para os demais roteadores. Em um segmento de rede multiacesso com *broadcast*, a troca de informações entre todos os roteadores vizinhos pode gerar uma grande sobrecarga. Com o roteador designado, chamado de DR (*Designed Router*), todos os roteadores de um segmento

enviam suas informações de *link-state* para este que age como um porta-voz do segmento, repassando estas informações para os demais roteadores do segmento.

2.7.3 IGRP

Quando se fala em roteamento IP não se pode deixar de falar da Cisco Systems. A Cisco é a empresa "líder mundial em infra-estrutura para a Internet e em soluções para conectividade ponta a ponta" (FILIPETTI, 2006, p. 1). Desde sua fundação, em 1984, a empresa investiu no desenvolvimento dos mais diversos equipamentos e segmentos de redes, tais como: *switches ethernet*, telefonia IP, *firewalls*, equipamentos *wireless* e, principalmente, roteadores, dos mais diversos tipos, entre outros.

Avaliando as necessidades do mercado a Cisco desenvolve também tecnologias, sendo a criadora dos protocolos de roteamento IGRP e EIGRP, desenvolvidos para atender às limitações do RIP. Não se pode deixar de incluir nos estudos, ambos os protocolos, sendo que são amplamente utilizados devido aos seus potenciais e por serem de fácil implementação, sendo usados em muitos casos para soluções de balanceamento de carga em redes IP. A desvantagem destes protocolos é que por serem proprietários, só são encontrados em roteadores da marca Cisco.

O IGRP (*Interior Gateway Routing Protocol*) é um protocolo que utiliza vetor de distância e diferentemente do RIP, pode ser usado em grandes segmentos de rede. Suporta múltiplas métricas para cada rota, incluindo, largura de banda (*bandwidth*), atraso da linha (*delay*), confiabilidade, carga e MTU (*Maximum Transmit Unit*). Para determinar o melhor caminho, estas métricas são combinadas em uma única métrica. Cada roteador envia regularmente, a cada 90 segundos, toda ou parte de sua tabela de roteamento em uma mensagem de atualização de roteamento a cada roteador vizinho. A contagem máxima de saltos para os pacotes é de 255, sendo que por padrão é configurada a contagem máxima de 100 saltos. É um protocolo de roteamento *classful* (não permite a prática do VLSM).

2.7.4 EIGRP

O EIGRP (*Enhanced Interior Gateway Routing Protocol*) é um protocolo híbrido, que utiliza técnicas de algoritmos vetor de distância e estado de enlace, combinadas de forma a reduzir as dificuldades, principalmente em termos de convergência para o caso de mudanças na rede.

Como o próprio nome já diz (*enhanced* = reforçado), é uma versão melhorada do IGRP, sendo um protocolo *classless*, possibilitando a prática do VLSM, CIDR e sumarização de rotas além de melhorar a segurança da rede, através de autenticação (FILIPETTI, 2006, p. 218). Por ser muito semelhante ao IGRP, o EIGRP é compatível como o seu “irmão”, proporcionando interoperabilidade entre os protocolos, fazendo um compartilhamento de rotas automático, sem a necessidade de configurações avançadas.

Roteadores que utilizam EIGRP mantêm as informações sobre rotas e topologia em três tabelas:

- Tabela de vizinhos;
- Tabela de topologia;
- Tabela de roteamento;

A tabela de vizinhos é a mais importante do EIGRP. Ela contém a lista de roteadores adjacentes. Para trocar informações entre os vizinhos os roteadores utilizam pacotes do tipo *hello*. Pacotes *hello* são utilizados para descobrir e manter adjacências e para a eleição de um DR em redes multiacesso. Quando um roteador envia ao seu vizinho um pacote *hello*, este marca o seu endereço e a interface a qual está conectado. O pacote anuncia um *hold time* (tempo de retenção), que é o período de tempo no qual um roteador considera um vizinho como alcançável e operacional. Após este período se o roteador não receber outro pacote *hello* do vizinho, este será considerado inalcançável e a rota deverá ser recalculada (Cisco Systems, 2003, p. 52).

A tabela de topologia contém todos os destinos informados pelos roteadores vizinhos. Para cada destino é associado um endereço IP e uma lista de vizinhos que anunciaram este destino, juntamente com a métrica anunciada. É importante destacar que os roteadores irão anunciar apenas rotas que estejam utilizando, sendo esta uma regra básica de protocolos que utilizam vetor distância.

Com estas informações é possível montar a tabela de roteamento, que irá conter apenas as melhores rotas para cada destino. Para este estudo é importante entender mais a fundo quais as métricas utilizadas pelo EIGRP e como elas são combinadas para a determinação das melhores rotas.

Para cada rota o EIGRP associa 5 (cinco) diferentes métricas, que são praticamente as mesmas utilizadas pela IGRP. São elas:

- *Delay* – atraso (10s de micro-segundos)
- *Bandwidth* – largura de banda ou taxa de transferência (*kilobits* por segundo)
- *Reliability* – confiabilidade (número entre 1 e 255; sendo 255 o mais confiável)
- *Load* – carga (número entre 1 e 255; sendo 255 o mais saturado)
- MTU (*Minimum path Maximum Transmission Unit*)

Por padrão, apenas *Delay* e *Bandwidth* são habilitados em um roteador usando EIGRP, sendo que as métricas podem ser habilitadas ou desabilitadas pelo administrador conforme necessário.

Para compor a métrica “final”, as métricas acima são combinadas na seguinte fórmula:

$$\left[\left(K_1 \cdot \text{Bandwidth} + \frac{K_2 \cdot \text{Bandwidth}}{256 - \text{Load}} + K_3 \cdot \text{Delay} \right) \cdot \frac{K_5}{K_4 + \text{Reliability}} \right] \cdot 256$$

Onde as constantes (K1 a K5) podem ser configuradas para variar o comportamento das decisões, desde que sejam configuradas da mesma forma em todos os roteadores em sistema EIGRP, evitando *loops* de roteamento. Como por padrão as constantes K1 e K3 são definidas como 1 e as demais como 0, a fórmula simplificada é $(\text{Bandwidth} + \text{Delay}) * 256$.

2.7.5 BGP

O BGP é um protocolo de roteamento projetado para ser utilizado em grandes redes. É um protocolo bastante complexo, sendo usado na Internet e em empresas multinacionais. Sua funcionalidade básica é manter a tabela de redes IP ou destinos, designando o alcance a estes destinos entre os Sistemas Autônomos (Gough, 2002, p. 339).

A grande característica do BGP é a atualização do roteamento, sendo que o torna apropriado a grandes ambientes. A atualização do roteamento transporta uma lista de números de Sistemas Autônomos e prefixos agregados, assim como algumas informações do roteamento baseado na política.

A atual versão do protocolo é a versão 4 (BGP-4), adotada em 1995 e definida na RFC 4271. Os principais recursos do BGP-4 incluem:

- Protocolo vetor de distância;

- São enviadas informações completas do roteamento no início da sessão;
- A conexão é mantida por pacotes do tipo *keepalives* periódicos;
- Qualquer alteração na rede resulta no acionamento de uma atualização;
- A métrica, chamada de atributo, é intrínseca e permite ótima granularidade;
- O uso de endereçamento hierárquico e a capacidade de manipular o fluxo de tráfego resultam em uma rede projetada para crescer;
- Possui sua própria tabela de roteamento, sendo capaz de compartilhar e pesquisar a tabela de roteamento IP interna;
- É possível manipular o fluxo de tráfego usando-se a métrica complexa, mas o roteamento baseado na política utiliza o paradigma salto a salto. Isto significa que nenhum roteador pode enviar tráfego em uma rota que o roteador do próximo salto não escolheria para si próprio.

O BGP-4 é orientado à conexão, sendo que após estabelecer uma conexão com um vizinho o BGP-4 envia testes periódicos (*keepalives*) para manter o enlace e a sessão.

2.8 BALANCEAMENTO DE CARGA ATRAVÉS DE ROTEAMENTO EM REDES IP

O balanceamento de carga através de roteamento é uma técnica muito praticada e está disponível em grande parte dos roteadores das mais diversas plataformas. Sempre que uma tabela de roteamento possui múltiplas rotas para alcançar um determinado destino, é possível configurar opções de balanceamento de carga baseadas nos protocolos de roteamento, tais como o RIP, RIPv2, EIGRP, IGRP, OSPF e BGP.

Dependendo da topologia de uma determinada rede, alguns roteadores podem aprender mais de uma rota para um mesmo destino, sendo que neste caso, o roteador irá escolher a rota de menor custo. Como vimos anteriormente, cada processo de roteamento calcula os custos de forma diferente, pois muitas vezes é necessário se fazer a manipulação das métricas para se conseguir um balanceamento de carga ideal. Desta forma, a maneira de implementar o balanceamento de carga varia conforme o protocolo de roteamento a ser utilizado, sendo que a escolha do protocolo é importante até mesmo devido às características de balanceamento de cada protocolo, como por exemplo, o número de caminhos máximos a serem balanceados para um determinado destino (6 caminhos com o EIGRP).

As formas de se fazer balanceamento de carga através de roteamento devem variar conforme a necessidade e a realidade de cada rede e sua topologia. É possível se determinar

formas de roteamento dinâmicas, em que cada pacote é encaminhado para um determinado destino de acordo com escolhas dinâmicas do roteador ou estáticas e o administrador da rede deve pré-determinar os caminhos para cada destino. Mesmo no balanceamento de carga dinâmico, é importante que o administrador da rede determine ou manipule estas escolhas, a fim de se obter o resultado esperado.

Em resumo, o balanceamento de carga (através de roteamento) é a capacidade que um roteador possui para distribuir tráfego sobre uma rede através das interfaces que estão a uma mesma distância de um endereço de destino, aumentando a utilização dos segmentos de rede e assim sendo da largura de banda efetiva. A Cisco classifica as formas de balanceamento de carga de acordo com os custos das rotas (Cisco Systems, 2009, p.1):

- Rotas com custos iguais: aplicado quando diferentes caminhos para um determinado destino possuem métricas de roteamento de igual valor;
- Rotas sem custos iguais: aplicado quando diferentes caminhos para um determinado destino possuem métricas de valores diferentes.

É possível configurar um roteador para realizar balanceamento de carga de forma dinâmica de duas maneiras, por destino (*per-destination*) ou por pacote (*per-packet*).

No balanceamento por destino, o roteador distribui os pacotes baseado no endereço IP de destino do mesmo. Se existem dois ou mais destinos para uma determinada rede 1, o roteador irá escolher o primeiro caminho disponível e, da mesma forma, se existem dois ou mais destinos para uma determinada rede 2, o roteador irá escolher o segundo caminho disponível e assim sucessivamente, alternando entre os caminhos ou rotas. Esta forma de balanceamento de tráfego garante uma maior qualidade ao que se refere à preservação dos pacotes, já que tendem a chegar de forma ordenada ao destino. O ponto negativo desta técnica é que através dela irá ocorrer um potencial uso desigual entre as rotas disponíveis. Quanto maior o número de endereços de destino utilizados pela rede, maior a igualdade entre o uso dos *links* disponíveis.

Já no balanceamento por pacote, o roteador envia um pacote destinado à rede 1 por uma rota e o segundo pacote destinado para a mesma rede 1 pela outra rota disponível e assim sucessivamente, alternando os pacotes entre as rotas disponíveis, garantindo assim um equilíbrio de carga entre os *links* utilizados. No entanto, com esta técnica, existe a possibilidade de que os pacotes cheguem fora de ordem no destino, devido a possíveis

diferenças no atraso da rede e, dependendo da topologia da rede, pode ser inviável a sua aplicação.

3 ESTUDO DE CASOS

O objetivo da implementação é realizar uma comparação entre as topologias de roteamento em redes IP mais utilizadas em soluções para médias e grandes empresas de modo geral e pequenos provedores de Internet (ISPs), apresentando seus pontos positivos e negativos e quais formas de roteamento e balanceamento de carga podem ser aplicados. Serão considerados três casos ou MODELOS mais comumente implantados.

3.1 MODELO 1

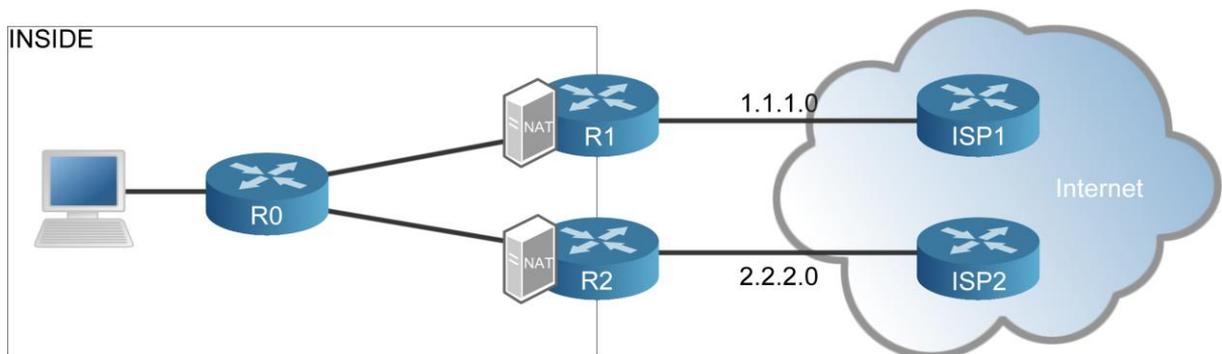


Figura 3.1 – Topologia de Rede do Modelo 1

O modelo de rede conforme Fig. 3.1, apresenta duas “saídas”, ou seja, dois *links* com a Internet fornecidos por dois ISPs diferentes (ISP1 e ISP2). Com isso os IPs de saída e classes de IPs fornecidos pelos ISPs serão apenas utilizáveis no *link* do respectivo ISP.

Para a maior disponibilidade devem-se utilizar endereços de IP privados para computadores, servidores e roteadores internos. Com isso, caso ocorra uma falha em um dos *links*, o outro irá assumir normalmente as requisições, podendo variar conforme o protocolo de roteamento e o tipo de tráfego. O problema deste modelo é que caso ocorra uma falha em um *link*, as conexões TCP estabelecidas através deste *link* não serão automaticamente assumidas pelo outro *link*. De forma mais prática, se um usuário está navegando na Internet (HTTP), é provável que não perceba uma parada nos serviços, já que deve constantemente encaminhar novas requisições. Porém, se este usuário estiver estabelecido uma conexão permanente, como um *streaming* de áudio e vídeo ou mesmo um *download*, o usuário deverá reiniciar a requisição.

Neste modelo seria possível aplicar técnicas de balanceamento de carga para minimizar o problema de indisponibilidade dos serviços caso ocorra uma parada e,

principalmente, para melhor utilizar os recursos de ambos os *links*. Neste caso não deve ser utilizado o balanceamento de carga por pacotes, já que as conexões TCP são estabelecidas utilizando o endereço IP de origem. Se aplicarmos o balanceamento de carga por pacotes, não ocorrerá a troca de informações com sucesso, uma vez que cada requisição terá um IP de origem diferente.

Os protocolos indicados para este modelo são: RIPv2, OSPF, EIGRP. A escolha destes protocolos pode variar conforme os equipamentos utilizados, a topologia da rede e o tipo de balanceamento de carga a ser utilizado.

3.2 MODELO 2

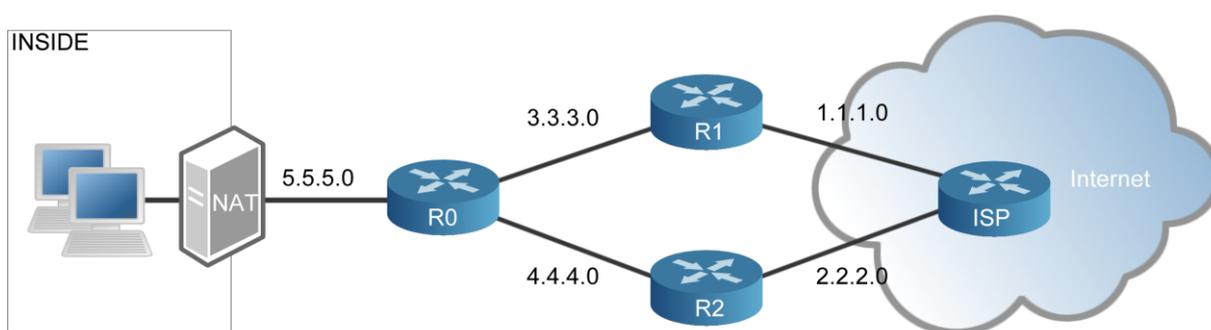


Figura 3.2 – Topologia de Rede do Modelo 2

O modelo da Fig. 3.2 apresenta uma topologia em que a rede do cliente possui dois *links*, ou seja, duas saídas para a Internet, com um mesmo fornecedor. Neste caso a empresa poderá ter implantado duas ou mais rotas internas até a saída para a Internet no intuito de estabelecer uma melhor disponibilidade dos serviços para o caso de falhas nos *links* internos, ou mesmo para um melhor desempenho da rede.

Para esta topologia a disponibilidade dos serviços pode ser configurada de forma mais eficiente internamente, podendo-se distribuir classes de IPs públicos, fornecidos pelo ISP, aos demais roteadores, servidores e até mesmo computadores da rede “interna”. Estas configurações dependerão de que o fornecedor ISP concorde e configure a sua rede para atender a tal.

Com isso a disponibilidade dos serviços é maior, uma vez que mesmo com a parada de um dos *links* com o ISP, o usuário que está fazendo uma requisição não terá seu IP de saída alterado e, dependendo do tamanho da rede e do protocolo de roteamento utilizado, nem mesmo serviços orientados à conexão serão interrompidos. O exemplo do modelo 1 apresenta

como solução para tal, em que ocorre a implantação de um servidor de NAT entre o R0 e a rede de computadores.

Da mesma forma que o modelo 1, o segundo modelo apresenta condições para a aplicação de técnicas de balanceamento de carga com o objetivo de minimizar o problema da indisponibilidade dos serviços caso ocorra uma parada em um dos *links* – neste caso com a mesma operadora – e para melhor utilizar os recursos de ambos os *links*. É possível até mesmo utilizar técnicas de balanceamento de carga por pacotes, além das técnicas de balanceamento de carga por destino. Em certos casos, dependendo do protocolo de roteamento, das configurações e da capacidade dos equipamentos e *softwares* implementados, os *links* podem ter uma operação semelhante a de um único *link*, com capacidade somada, quando do ponto de vista da capacidade de tráfego (até seis, utilizando o EIGRP).

Os protocolos indicados para este modelo são os mesmos do primeiro modelo, devendo seguir os mesmos cuidados apresentados no modelo anterior: RIPv2, OSPF, EIGRP.

3.3 MODELO 3

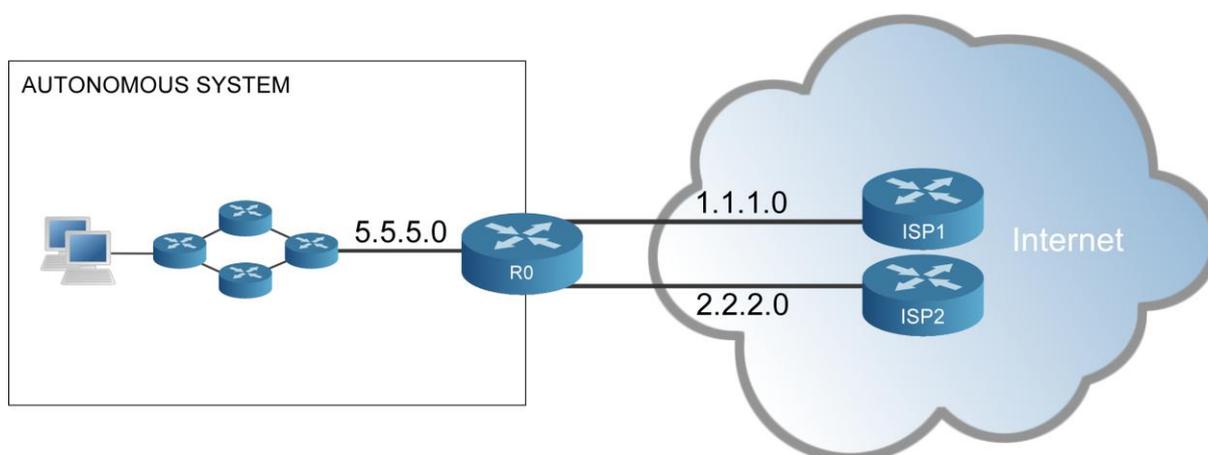


Figura 3.3 – Topologia de Rede do Modelo 3

Para grandes empresas e principalmente provedores de serviços de Internet, mesmo de pequeno porte, recomenda-se a implantação de um Sistema Autônomo (AS). A designação de ASNs (*Autonomous System Numbers*) é controlada pela IANA (*Internet Assigned Numbers Authority*), sendo que no Brasil é representada pelo Núcleo de Informação e Coordenação do Ponto BR (NIC.br) juntamente com a LACNIC (*Latin American and Caribbean Internet Addresses Registry*), desde que uma empresa ou instituição justifique a estes órgãos a

necessidade para tal e preencha os pré-requisitos para ser um Sistema Autônomo. A Fig 3.3 ilustra este modelo.

Juntamente com o ASN a empresa ou instituição recebe um ou mais blocos de IP “próprios” por assim dizer, podendo utilizar e divulgar estes blocos independentemente de suas conexões junto a outros ISPs (AS), ou seja, com isso pode designar estas classes de IPs internamente, fazendo com que os dispositivos que as recebem possam ser acessados por dois ou mais caminhos (acessos) de diferentes fornecedores de serviços.

Desta forma a disponibilidade da rede de computadores e roteadores dentro de um mesmo AS, em caso de falha em um dos *links*, pode ser melhor dimensionada e ocorrer de forma mais prática e segura, podendo uma empresa optar por possuir acessos à Internet com dois ou mais ISPs e distribuir IPs públicos aos componentes da rede.

O balanceamento de carga entre os *links* em um AS também ocorre de forma diferenciada. Neste caso é possível apenas designar de forma direta o caminho a ser seguido pelo tráfego que sai da rede (*upload*) sendo que o tráfego que entra na rede (*download*) pode apenas ser influenciado. Isto ocorre, pois mesmo quando uma solicitação é feita através de um *link*, ela não necessariamente deverá voltar por este *link*, uma vez que cada AS na Internet visualiza as rotas do AS local de forma diferenciada, podendo variar conforme os custos designados a cada rota de acordo com as métricas aplicadas pelo protocolo BGP, que é o protocolo de roteamento usado entre ASs. A melhor forma de se balancear o tráfego entre os *links* é segmentar os blocos de endereços IP e dividir a rede de computadores e roteadores entre estes blocos menores, fazendo a divulgação destes blocos de forma diferenciada entre os ISPs.

4 IMPLEMENTANDO O ROTEAMENTO

A implementação e os testes que serão realizados podem ser subdivididos em duas partes, sendo a primeira englobando os dois primeiros modelos, aplicando a eles protocolos de roteamento e técnicas de implantação e balanceamento de carga de protocolos IGP e a segunda parte aplicando técnicas de protocolos EGP no terceiro modelo.

Para a implementação e testes dos primeiros dois modelos serão utilizados 4 roteadores Cisco, com as seguintes características técnicas e especificações:

- **Router0 (R0):**

```
Cisco 2811 (revision 53.50) with 251904K/10240K bytes of memory.
Processor board ID FTX1050A4SS
2 FastEthernet interfaces
4 Serial(sync/async) interfaces
DRAM configuration is 64 bits wide with parity enabled.
239K bytes of non-volatile configuration memory.
62720K bytes of ATA CompactFlash (Read/Write)

Cisco IOS Software, 2800 Software (C2800NM-SPSERVICESK9-M), Version
12.4(10a), RELEASE SOFTWARE (fc2)
```

- **Router1 (R1):**

```
Cisco 1721 (MPC860P) processor (revision 0x100) with 54388K/11148K bytes of
memory.
Processor board ID FOC06460BW7 (2069882113), with hardware revision 0000
MPC860P processor: part number 5, mask 2
Bridging software.
X.25 software, Version 3.0.0.
1 FastEthernet/IEEE 802.3 interface(s)
4 Serial(sync/async) network interface(s)
32K bytes of non-volatile configuration memory.
32768K bytes of processor board System flash (Read/Write)

Cisco Internetwork Operating System Software
IOS (tm) C1700 Software (C1700-Y-M), Version 12.3(10), RELEASE SOFTWARE
(fc3)
```

- **Router2 (R2):**

```
Cisco 1721 (MPC860P) processor (revision 0x100) with 54388K/11148K bytes of
memory.
Processor board ID FOC07230C0K (290377455), with hardware revision 0000
MPC860P processor: part number 5, mask 2
Bridging software.
X.25 software, Version 3.0.0.
1 FastEthernet/IEEE 802.3 interface(s)
2 Serial(sync/async) network interface(s)
2 Low-speed serial(sync/async) network interface(s)
32K bytes of non-volatile configuration memory.
32768K bytes of processor board System flash (Read/Write)

Cisco Internetwork Operating System Software
IOS (tm) C1700 Software (C1700-Y-M), Version 12.3(10), RELEASE SOFTWARE
(fc3)
```

- Router3 (R3):

```

Cisco MC3810 (MPC860) processor (revision 10.01) with 28672K/4096K bytes of
memory.
Processor board ID 17034114
PPC860 PowerQUICC, partnum 0x0000, version C01(0x0031)
Channelized E1, Version 1.0.
Bridging software.
X.25 software, Version 3.0.0.
MC3810 SCB board (v16.B0)
1 Six-Slot Analog Voice Module (v07.B0)
1   Analog E&M voice interface (v05.B0) port 1/1
1 3-DSP(slot2) Voice Compression Module(v01.C0)
1 Ethernet/IEEE 802.3 interface(s)
2 Serial(sync/async) network interface(s)
256K bytes of non-volatile configuration memory.
8192K bytes of processor board System flash (AMD29F016)

Cisco Internetwork Operating System Software
IOS (tm) MC3810 Software (MC3810-IS-M), Version 12.2(32), RELEASE SOFTWARE
(fc1)

```

Os roteadores serão utilizados e interligados para simular as condições dos modelos propostos a fim de se obter condições para os testes, comparações entre as técnicas e protocolos utilizados e comparações entre os modelos propostos.

Para os dois primeiros modelos serão implementados e testados os protocolos de roteamento mais utilizados, sendo eles o RIPv2, o EIGRP e o OSPF e suas respectivas técnicas de roteamento.

4.1 IMPLEMENTANDO O MODELO 1

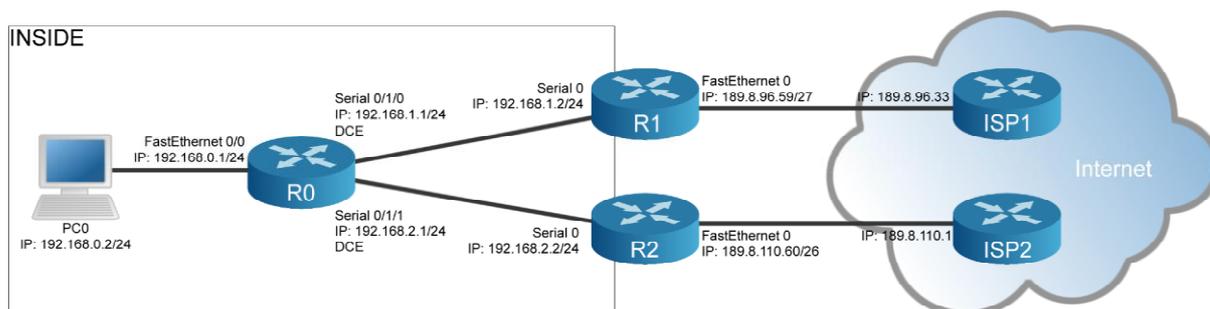


Figura 4.1 – Topologia de rede do modelo 1 aplicada aos testes práticos

A Fig. 4.1 ilustra a topologia da rede do modelo 1 aplicada aos equipamentos e suas interfaces, bem como os endereços IP utilizados.

As interfaces seriais foram interligadas através de cabos DCE/DTE V.35. O roteador R0 será o gerador do *clock* (DCE). A taxa de transmissão máxima destas interfaces é de 8 Mbps (Cisco Systems, 2009, p. 4). Através de medições de vazão foi possível confirmar a

taxa de transmissão dos enlaces entre o roteador R0 e os roteadores R1 e R2. O programa utilizado para tais medições foi o Iperf, um programa muito utilizado em testes de taxa de transmissão em redes IP. Para tais testes, os roteadores ISP1 e ISP2 foram substituídos por um computador compatível com o sistema Iperf. Os testes apresentaram os seguintes resultados:

Entre 192.168.0.2 e 189.8.110.1:

```
PC0>iperf -c 189.8.110.1 -t 10 -i 1 -r
-----
Server listening on TCP port 5001
TCP window size: 8.00 KByte (default)
-----
Client connecting to 189.8.110.1, TCP port 5001
TCP window size: 8.00 KByte (default)
-----
[1876] local 192.168.0.2 port 2011 connected with 189.8.110.1 port 5001
[ ID] Interval      Transfer    Bandwidth
...
[1876] 1.0- 2.0 sec   984 KBytes  8.06 Mbits/sec
```

Entre 192.168.0.2 e 189.8.96.33:

```
PC0>iperf -c 189.8.96.33 -t 10 -i 1 -r
-----
Server listening on TCP port 5001
TCP window size: 8.00 KByte (default)
-----
Client connecting to 189.8.96.33, TCP port 5001
TCP window size: 8.00 KByte (default)
-----
[1868] local 192.168.0.2 port 2008 connected with 189.8.96.33 port 5001
[ ID] Interval      Transfer    Bandwidth
[1868] 1.0- 2.0 sec   984 KBytes  8.06 Mbits/sec
```

Além das configurações básicas (endereços IP, máscara de rede/sub-rede, *clock rate*, etc.), para o modelo 1 será necessário configurar os roteadores R1 e R2 com as rotas *default* para a Internet e os serviços NAT, já que internamente (*INSIDE*) serão utilizados endereços IPs reservados. Os seguintes comandos habilitam o serviço nos roteadores:

```
Router1(config)#ip route 0.0.0.0 0.0.0.0 189.8.96.33
Router1(config)#ip nat pool ovrl 189.8.96.59 189.8.96.59 prefix-length 24
Router1(config)#ip nat inside source list 7 pool ovrl overload
Router1(config)#access-list 7 permit 192.168.0.0 0.0.255.255
Router1(config)#interface serial 0
Router1(config-if)#ip nat inside
Router1(config)#interface fastEthernet 0
Router1(config-if)#ip nat outside

Router2(config)#ip route 0.0.0.0 0.0.0.0 189.8.110.1
Router2(config)#ip nat pool ovrl 189.8.110.60 189.8.110.60 prefix-length 24
Router2(config)#ip nat inside source list 7 pool ovrl overload
```

```
Router2(config)#access-list 7 permit 192.168.0.0 0.0.255.255
Router2(config)#interface serial 0
Router2(config-if)#ip nat inside
Router2(config)#interface fastEthernet 0
Router2(config-if)#ip nat outside
```

A seguir estão detalhados os comandos e procedimentos para a configuração dos protocolos de roteamento RIPv2, EIGRP e OSPF.

4.1.1 MODELO 1 COM RIPv2

4.1.1.1 Configuração do RIPv2

```
Router0(config)#router rip
Router0(config-router)#version 2
Router0(config-router)#network 192.168.0.0
Router0(config-router)#network 192.168.1.0
Router0(config-router)#network 192.168.2.0
Router0(config-router)#no auto-summary

Router1(config)#router rip
Router1(config-router)#version 2
Router1(config-router)#network 192.168.1.0
Router1(config-router)#network 189.8.96.32
Router1(config-router)#no auto-summary
Router1(config-router)#redistribute static

Router2(config)#router rip
Router2(config-router)#version 2
Router2(config-router)#network 192.168.2.0
Router2(config-router)#network 189.8.110.0
Router2(config-router)#passive-interface fastEthernet 0
Router2(config-router)#no auto-summary
Router2(config-router)#redistribute static
```

O comando **#router rip** especifica o protocolo de roteamento a ser utilizado, seguido do comando **#version 2** que configura a utilização do RIPv2. O comando **#network [endereço ip de rede]**, identifica as redes conectadas diretamente ao roteador que serão divulgadas pelo protocolo dinâmico. No nosso exemplo, devemos ainda habilitar o comando **#no auto-summary** para evitar problemas na divulgação da rede *classfull* (189.8.0.0) que está subdividida. Através do comando **#show ip route** é possível visualizar a tabela de roteamento. Executando em R0, temos:

```
Router0#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
```

```

        ia - IS-IS inter area, * - candidate default, U - per-user static
route
        o - ODR, P - periodic downloaded static route

Gateway of last resort is 192.168.2.2 to network 0.0.0.0

    189.8.0.0/16 is variably subnetted, 2 subnets, 2 masks
R       189.8.110.0/26 [120/1] via 192.168.2.2, 00:00:16, Serial0/1/1
R       189.8.96.32/27 [120/1] via 192.168.1.2, 00:00:23, Serial0/1/0
C       192.168.0.0/24 is directly connected, FastEthernet0/0
C       192.168.1.0/24 is directly connected, Serial0/1/0
C       192.168.2.0/24 is directly connected, Serial0/1/1
R*     0.0.0.0/0 [120/1] via 192.168.2.2, 00:00:16, Serial0/1/1
        [120/1] via 192.168.1.2, 00:00:23, Serial0/1/0

```

É possível observar (no destaque) que o roteador R0 adicionou à sua tabela de roteamento duas rotas com o mesmo custo para a rede 0.0.0.0 (*default network*), sendo que o campo **[120/1]** apresenta a distância administrativa da fonte de informação e o custo da rota, respectivamente. Neste caso (RIPv2) o custo é 1, que representa o número de saltos até a rede padrão.

4.1.1.2 Balanceamento de Carga - Modelo 1 com RIPv2

Nos roteadores Cisco, ao utilizarmos o protocolo RIPv2, quando existirem duas rotas ou mais instaladas na tabela de roteamento, ou seja, com o mesmo custo para um determinado destino, o balanceamento de carga ocorre automaticamente. Nesta caso, a configuração padrão estabelece o balanceamento de carga por destino. Para verificarmos tal técnica utiliza-se o comando **#ping** para dois destinos diferentes e o comando **#debug ip packet** que fará o *log* dos pacotes enviados e recebidos.

```

Router0#debug ip packet
IP packet debugging is on
Router0#ping 201.54.192.6

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 201.54.192.6, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/20/24 ms
Router0#
May  5 22:22:26.146: IP: tableid=0, s=192.168.2.1 (local), d=201.54.192.6
(Serial0/1/0), routed via FIB
May  5 22:22:26.146: IP: s=192.168.2.1 (local), d=201.54.192.6
(Serial0/1/0), len 100, sending
May  5 22:22:26.166: IP: tableid=0, s=201.54.192.6 (Serial0/1/0),
d=192.168.2.1(Serial0/1/1), routed via RIB
May  5 22:22:26.166: IP: s=201.54.192.6 (Serial0/1/0), d=192.168.2.1, len
100, rcvd 4
May  5 22:22:26.166: IP: tableid=0, s=192.168.2.1 (local), d=201.54.192.6
(Serial0/1/0), routed via FIB
May  5 22:22:26.166: IP: s=192.168.2.1 (local), d=201.54.192.6
(Serial0/1/0), len 100, sending
May  5 22:22:26.186: IP: tableid=0, s=201.54.192.6 (Serial0/1/0),

```

```

d=192.168.2.1(Serial0/1/1), routed via RIB
May 5 22:22:26.186: IP: s=201.54.192.6 (Serial0/1/0), d=192.168.2.1, len
100, rcvd 4
May 5 22:22:26.186: IP: tableid=0, s=192.168.2.1 (local), d=201.54.192.6
(Serial0/1/0), routed via FIB
May 5 22:22:26.186: IP: s=192.168.2.1 (local), d=201.54.192.6
(Serial0/1/0), len 100, sending
May 5 22:22:26.206: IP: tableid=0, s=201.54.192.6 (Serial0/1/0),
d=192.168.2.1(Serial0/1/1), routed via RIB
May 5 22:22:26.206: IP: s=201.54.192.6 (Serial0/1/0), d=192.168.2.1, len
100, rcvd 4
May 5 22:22:26.206: IP: tableid=0, s=192.168.2.1 (local), d=201.54.192.6
(Serial0/1/0), routed via FIB
May 5 22:22:26.206: IP: s=192.168.2.1 (local), d=201.54.192.6
(Serial0/1/0), len 100, sending
May 5 22:22:26.226: IP: tableid=0, s=201.54.192.6 (Serial0/1/0),
d=192.168.2.1(Serial0/1/1), routed via RIB
May 5 22:22:26.226: IP: s=201.54.192.6 (Serial0/1/0), d=192.168.2.1, len
100, rcvd 4
May 5 22:22:26.226: IP: tableid=0, s=192.168.2.1 (local), d=201.54.192.6
(Serial0/1/0), routed via FIB
May 5 22:22:26.226: IP: s=192.168.2.1 (local), d=201.54.192.6
(Serial0/1/0), len 100, sending
May 5 22:22:26.246: IP: tableid=0, s=201.54.192.6 (Serial0/1/0),
d=192.168.2.1(Serial0/1/1), routed via RIB
May 5 22:22:26.246: IP: s=201.54.192.6 (Serial0/1/0), d=192.168.2.1, len
100, rcvd 4

```

Os pacotes *ping* enviados para o endereço 201.54.192.6 (furb.br) saíram e entraram pela interface Serial 0/1/0, ou seja, pelo caminho do Router1. Na sequência fez-se o mesmo teste com o comando **#ping** para o endereço 200.144.145.14 (pucsp.br):

```

Router0#ping 200.144.145.14

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 200.144.145.14, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/16/20 ms
Router0#
May 5 22:23:49.289: IP: tableid=0, s=192.168.2.1 (local), d=200.144.145.14
(Serial0/1/1), routed via FIB
May 5 22:23:49.289: IP: s=192.168.2.1 (local), d=200.144.145.14
(Serial0/1/1), len 100, sending
May 5 22:23:49.309: IP: tableid=0, s=200.144.145.14 (Serial0/1/1),
d=192.168.2.1 (Serial0/1/1), routed via RIB
May 5 22:23:49.309: IP: s=200.144.145.14 (Serial0/1/1), d=192.168.2.1
(Serial0/1/1), len 100, rcvd 3
May 5 22:23:49.309: IP: tableid=0, s=192.168.2.1 (local), d=200.144.145.14
(Serial0/1/1), routed via FIB
May 5 22:23:49.309: IP: s=192.168.2.1 (local), d=200.144.145.14
(Serial0/1/1), len 100, sending
May 5 22:23:49.325: IP: tableid=0, s=200.144.145.14 (Serial0/1/1),
d=192.168.2.1 (Serial0/1/1), routed via RIB
May 5 22:23:49.325: IP: s=200.144.145.14 (Serial0/1/1), d=192.168.2.1
(Serial0/1/1), len 100, rcvd 3
May 5 22:23:49.325: IP: tableid=0, s=192.168.2.1 (local), d=200.144.145.14
(Serial0/1/1), routed via FIB
May 5 22:23:49.325: IP: s=192.168.2.1 (local), d=200.144.145.14
(Serial0/1/1), len 100, sending
May 5 22:23:49.341: IP: tableid=0, s=200.144.145.14 (Serial0/1/1),
d=192.168.2.1 (Serial0/1/1), routed via RIB
May 5 22:23:49.341: IP: s=200.144.145.14 (Serial0/1/1), d=192.168.2.1
(Serial0/1/1), len 100, rcvd 3

```

May 5 22:23:49.341: IP: tableid=0, s=192.168.2.1 (local), d=200.144.145.14 (Serial0/1/1), routed via FIB
 May 5 22:23:49.341: IP: s=192.168.2.1 (local), d=200.144.145.14 (Serial0/1/1), len 100, sending
 May 5 22:23:49.353: IP: tableid=0, s=200.144.145.14 (Serial0/1/1), d=192.168.2.1 (Serial0/1/1), routed via RIB
 May 5 22:23:49.353: IP: s=200.144.145.14 (Serial0/1/1), d=192.168.2.1 (Serial0/1/1), len 100, rcvd 3
 May 5 22:23:49.357: IP: tableid=0, s=192.168.2.1 (local), d=200.144.145.14 (Serial0/1/1), routed via FIB
 May 5 22:23:49.357: IP: s=192.168.2.1 (local), d=200.144.145.14 (Serial0/1/1), len 100, sending
 May 5 22:23:49.381: IP: tableid=0, s=200.144.145.14 (Serial0/1/1), d=192.168.2.1 (Serial0/1/1), routed via RIB
 May 5 22:23:49.381: IP: s=200.144.145.14 (Serial0/1/1), d=192.168.2.1 (Serial0/1/1), len 100, rcvd 3

Neste caso o caminho escolhido foi a rota através do Router2 (Serial0/1/1). Este processo continuará alternando para todos os destinos requisitados pela rede, balanceando a carga a ser utilizada (Cisco Systems, 2003, p. 91). Na Fig. 4.2 é possível verificar tal distribuição através de diversos downloads realizados com o uso do *software Free Download Manager* (gerenciador de downloads) instalado no computador PC0.

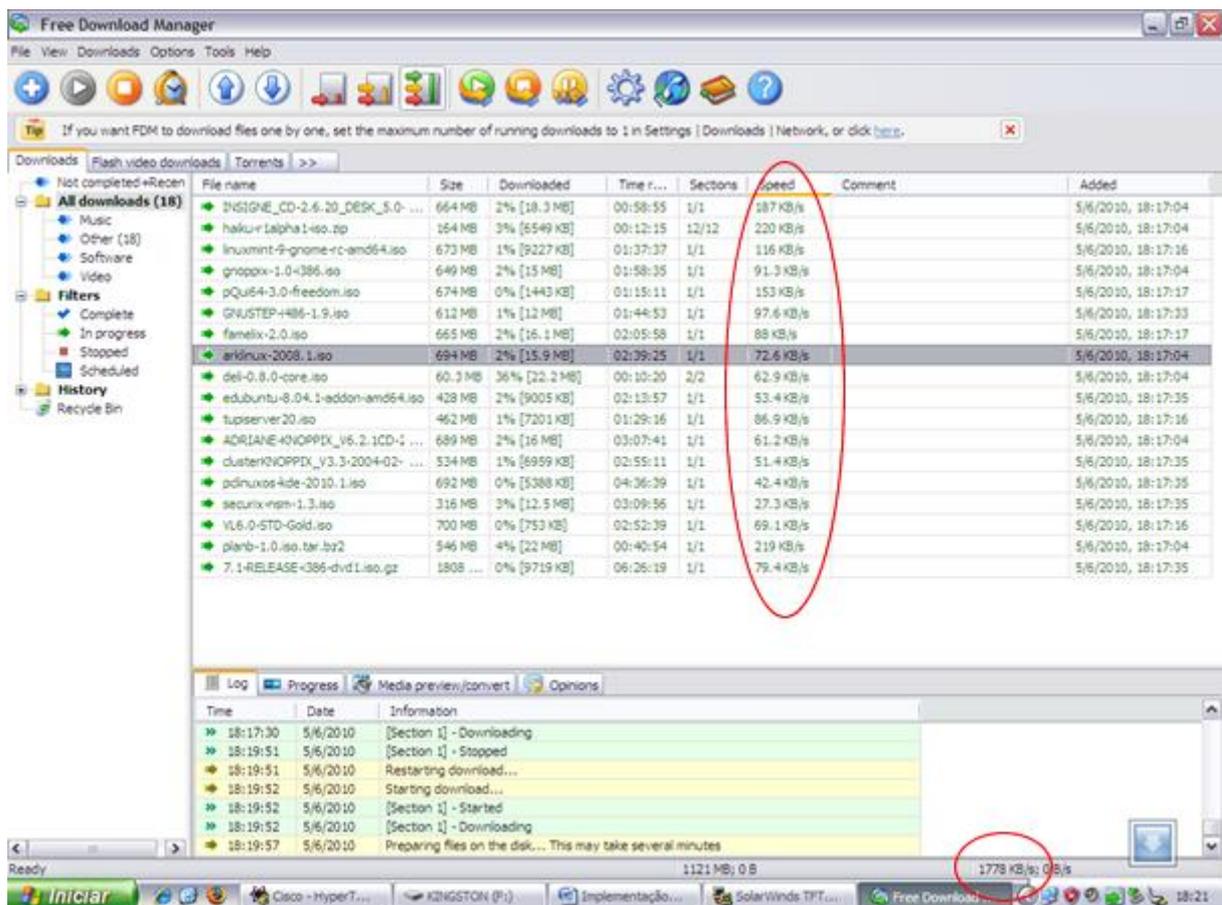


Figura 4.2 – Tela do Software Free Download Manager

A velocidade total dos *downloads* em um mesmo momento chegou a 1778 KB/s, conforme destacado na Fig. 4.2, ou seja, 14,2 Mbps, sendo que a velocidade máxima de cada enlace entre R0-R1 e R0-R2 é de 8Mbps.

Através do comando **#show interfaces stats** é possível observar que, de aproximadamente 470 mil pacotes enviados e recebidos durante um certo período de tempo, aproximadamente 243 mil foram enviados e recebidos através de R1 e 227 mil através de R2, conforme segue:

```
Router0#show interfaces stats
...
Serial0/1/0
  Switching path  Pkts In   Chars In   Pkts Out   Chars Out
  Processor        77         5176      71          5241
  Route cache     243190    354354685 176842     8678337
  Total           243267    354359861 176913     8683578
Serial0/1/1
  Switching path  Pkts In   Chars In   Pkts Out   Chars Out
  Processor        70         4406      67          4931
  Route cache     227061    327143449 151804     7485358
  Total           227131    327147855 151871     7490289
```

É possível notar uma diferença de 3,43% entre o total de pacotes que trafegou entre as rotas. Essa diferença tende a diminuir quanto maior for o número de destinos requisitados pela rede.

O RIP é capaz de fazer balanceamento de carga em até 6 caminhos com custo iguais. Caso deseje-se restringir o número máximo de caminhos a serem utilizados pelo RIP, deve-se utilizar o comando **#maximum-paths**. Caso não seja do interesse trabalhar com balanceamento de cargas, basta efetuar o seguinte comando (Cisco Systems, 2003, p. 93):

```
Router0(config-router)#maximum-paths 1
```

4.1.2 MODELO 1 COM OSPF

4.1.2.1 Configuração do OSPF

A ativação do OSPF em um roteador inicia com o comando **#router ospf [número do processo]**. O número do processo nada tem a ver com o AS ao qual o roteador pertence, sendo que poderá existir mais de um processo em execução em um mesmo roteador, apesar de

não ser algo comum. Para o experimento usa-se um número de processo comum para todos os roteadores e áreas. O comando **#network** irá especificar quais as redes ligadas ao roteador que irão participar do processo OSPF. Este comando desempenha um papel semelhante ao que desempenha nos protocolos RIPv2 e EIGRP, sendo que nestes dois o comando especifica a rede a nível de classe e no OSPF é possível identificar o endereço específico de uma interface (Gough, 2002, p. 195). Os parâmetros do comando **#network** são o endereço da rede, seguido de uma máscara curinga (*wildcard mask*) e a área a qual a rede pertence. Para a configuração da rede foi utilizado apenas uma área para todas as redes. É recomendado especificar a largura de banda real das interfaces para que o OSPF possa fazer o cálculo correto dos custos de cada rota. A largura de banda das interfaces é configurada através do comando **#bandwidth** (em kbps).

```
Router0(config)#router ospf 1
Router0(config-router)#network 192.168.0.0 0.0.0.255 area 0
Router0(config-router)#network 192.168.1.0 0.0.0.255 area 0
Router0(config-router)#network 192.168.2.0 0.0.0.255 area 0
Router0(config-router)#passive-interface fastEthernet 0/0
Router0(config)#interface serial 0/1/0
Router0(config-if)#bandwidth 8192
Router0(config)#interface serial 0/1/1
Router0(config-if)#bandwidth 8192

Router1(config)#router ospf 1
Router1(config-router)#network 192.168.1.0 0.0.0.255 area 0
Router1(config-router)#network 189.8.96.32 0.0.0.31 area 0
Router1(config-router)#passive-interface fastEthernet 0
Router1(config-router)#default-information originate
Router1(config)#interface serial 0
Router1(config-if)#bandwidth 8192

Router2(config)#router ospf 1
Router2(config-router)#network 192.168.2.0 0.0.0.255 area 0
Router2(config-router)#network 189.8.110.0 0.0.0.63 area 0
Router2(config-router)#passive-interface fastEthernet 0
Router2(config-router)#default-information originate
Router2(config)#interface serial 0
Router2(config-if)#bandwidth 8192
```

Configuram-se todas as interfaces seriais com largura de banda de 8 Mbps (8192 kbps). Não é necessário configurar a largura de banda das interfaces *Fast Ethernet*, pois o roteador sabe por padrão o seu valor (100 Mbps). Por padrão, as interfaces seriais em roteadores Cisco possuem a especificação de largura de banda de 1,544 Mbps. É importante destacar que esta configuração ou mesmo a configuração padrão de largura de banda não influencia no tráfego real da rede, elas servem apenas para que o OSPF e outros processos efetuem os cálculos necessários corretamente.

Verificando as rotas instaladas em R0, tem-se:

```

Router0#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 192.168.2.2 to network 0.0.0.0

    192.168.31.0/32 is subnetted, 1 subnets
C       192.168.31.31 is directly connected, Loopback0
    189.8.0.0/16 is variably subnetted, 2 subnets, 2 masks
O       189.8.110.0/26 [110/13] via 192.168.2.2, 00:01:27, Serial0/1/1
O       189.8.96.32/27 [110/13] via 192.168.1.2, 00:01:27, Serial0/1/0
C       192.168.0.0/24 is directly connected, FastEthernet0/0
C       192.168.1.0/24 is directly connected, Serial0/1/0
C       192.168.2.0/24 is directly connected, Serial0/1/1
O*E2 0.0.0.0/0 [110/1] via 192.168.2.2, 00:01:28, Serial0/1/1
        [110/1] via 192.168.1.2, 00:01:28, Serial0/1/0

```

Para o cálculo de custo de uma rota, o roteador realiza o cálculo do custo de suas interfaces ligadas a esta rede mais o valor do custo total informado pelo roteador que será o próximo salto para a rede. É possível observar que as rotas para 189.8.0.0 possuem custo 13. Para chegar a este valor o roteador efetuou os seguintes cálculos:

(10^8) / largura de banda da interface em bps

Como exemplo, tem-se o seguinte custo para a rota 189.8.96.32/27 que utiliza a interface Serial 0/1/0:

Custo interface Serial 0/1/0: $((10^8) / (8192 * 1024)) = 11,92 = 12$

Custo interface *FastEthernet* 0 em R1: $((10^8) / (100000 * 1024)) = 0,98 = 1$

Custo total = 12 + 1 = 13

Por padrão, o OSPF informa rotas *default* com o custo igual a 1:

```

Router0#show ip route 0.0.0.0
Routing entry for 0.0.0.0/0, supernet
  Known via "ospf 1", distance 110, metric 1, candidate default path
  Tag 1, type extern 2, forward metric 12
  Last update from 192.168.2.2 on Serial0/1/1, 00:05:12 ago
Routing Descriptor Blocks:
  192.168.2.2, from 192.168.2.2, 00:05:12 ago, via Serial0/1/1
    Route metric is 1, traffic share count is 1

```

```

Route tag 1
* 192.168.1.2, from 192.168.1.2, 00:05:12 ago, via Serial0/1/0
Route metric is 1, traffic share count is 1
Route tag 1

```

Caso seja desejado um maior controle sobre a rota *default* propagada pelos roteadores R1 e R2, deve-se utilizar o comando **#default-information originate metric [valor]**, especificando no campo **valor** o custo correto para a rota *default*. Para este experimento não há necessidade de fazer tal configuração, pois ambas as rotas têm o mesmo custo, tendo a mesma largura de banda. Além disso, não seria possível balancear carga entre *links* com custos desiguais com o OSPF, conforme veremos a seguir.

4.1.2.2 Balanceamento de Carga - Modelo 1 com OSPF

A exemplo do RIPv2, o OSPF faz o equilíbrio de carga automaticamente entre enlaces de mesmo custo e apenas para *links* de mesmo custo. A configuração padrão dos roteadores Cisco estabelece o balanceamento de carga por destino quando dois ou mais *links* do mesmo custo forem instalados na tabela de roteamento em um determinado roteador. No caso do OSPF é possível alterar manualmente o custo das interfaces que fazem parte do processo. Desta forma, mesmo que se dinamicamente o OSPF não instalar uma rota para um determinado destino em sua tabela por ter outra rota com menor custo instalada, é possível forçar o roteador a instalar tal rota ao configurá-la com o mesmo custo da menor rota instalada. Para testar tal condição, “enganou-se” o Router0 a respeito da largura de banda de uma de suas interfaces:

```

Router0(config)#interface serial 0/1/1
Router0(config-if)#bandwidth 3072

Router2(config)#interface serial 0
Router2(config-if)#bandwidth 3072

```

Após é possível verificar que a rota para 0.0.0.0 entre R0 e R2 foi removida da tabela de roteamento em R0, conforme segue:

```

Router0#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route

```

o - ODR, P - periodic downloaded static route

Gateway of last resort is 192.168.1.2 to network 0.0.0.0

```

192.168.31.0/32 is subnetted, 1 subnets
C    192.168.31.31 is directly connected, Loopback0
189.8.0.0/16 is variably subnetted, 2 subnets, 2 masks
O    189.8.110.0/26 [110/33] via 192.168.2.2, 00:00:12, Serial0/1/1
O    189.8.96.32/27 [110/13] via 192.168.1.2, 00:00:12, Serial0/1/0
C    192.168.0.0/24 is directly connected, FastEthernet0/0
C    192.168.1.0/24 is directly connected, Serial0/1/0
C    192.168.2.0/24 is directly connected, Serial0/1/1
O*E2 0.0.0.0/0 [110/1] via 192.168.1.2, 00:00:13, Serial0/1/0

```

A rota para 189.8.110.0 passou a ter o custo no valor de 33. Para forçar R0 a instalar novamente a rota através de R2, especificou-se manualmente os custos de ambos os enlaces com o mesmo valor, através do comando **#ip ospf cost [valor]** nas interfaces seriais:

```

Router0(config)#interface serial 0/1/0
Router0(config-if)#ip ospf cost 3

Router0(config)#interface serial 0/1/1
Router0(config-if)#ip ospf cost 3

```

Novamente verificando a tabela de roteamento:

```

Router0#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 192.168.2.2 to network 0.0.0.0

192.168.31.0/32 is subnetted, 1 subnets
C    192.168.31.31 is directly connected, Loopback0
189.8.0.0/16 is variably subnetted, 2 subnets, 2 masks
O    189.8.110.0/26 [110/4] via 192.168.2.2, 00:00:00, Serial0/1/1
O    189.8.96.32/27 [110/4] via 192.168.1.2, 00:00:00, Serial0/1/0
C    192.168.0.0/24 is directly connected, FastEthernet0/0
C    192.168.1.0/24 is directly connected, Serial0/1/0
C    192.168.2.0/24 is directly connected, Serial0/1/1
O*E2 0.0.0.0/0 [110/1] via 192.168.2.2, 00:00:01, Serial0/1/1
      [110/1] via 192.168.1.2, 00:00:01, Serial0/1/0

```

Ambas as rotas para 0.0.0.0 possuem o mesmo custo e estão instaladas. Além disso, é possível notar que o custo para as rotas 189.8.0.0 voltaram a ter o mesmo custo, agora igual a 4. Após gerar tráfego entre PC0 e diversos destinos tem-se o seguinte relatório de pacotes enviados e recebidos nas interfaces seriais:

```

Router0#show interfaces stat
...
Serial0/1/0
  Switching path  Pkts In   Chars In   Pkts Out   Chars Out
  Processor        51         3856       48          3788
  Route cache     71635     102747202  48844       2210189
  Total           71686     102751058  48892       2213977
Serial0/1/1
  Switching path  Pkts In   Chars In   Pkts Out   Chars Out
  Processor        47         3276       43          3141
  Route cache     63981     94941650   43729       2249077
  Total           64028     94944926   43772       2252218

```

De 228.378 pacotes enviados e recebidos nas interfaces seriais, 47,20% foram enviados e recebidos através de R2 e 52,80% através de R1.

4.1.3 MODELO 1 COM EIGRP

4.1.3.1 Configuração do EIGRP

A configuração do EIGRP é muito semelhante à do RIPv2. A única diferença neste caso é que deve se especificar o número do sistema autônomo EIGRP, ao qual o roteador participa, através do comando **#router eigrp [número do sistema autônomo]**. Não há necessidade de especificar a versão, já que o EIGRP possui apenas uma versão e é recomendado especificar a largura de banda real das interfaces (a exemplo do OSPF) para que o EIGRP possa fazer o cálculo correto das custos de cada rota. A largura de banda das interfaces é configurada através do comando **#bandwidth** (em kbps).

```

Router0(config)#router eigrp 1
Router0(config-router)#network 192.168.0.0
Router0(config-router)#network 192.168.1.0
Router0(config-router)#network 192.168.2.0
Router0(config-router)#no auto-summary
Router0(config)#interface serial 0/1/0
Router0(config-if)#bandwidth 8192
Router0(config)#interface serial 0/1/1
Router0(config-if)#bandwidth 8192

Router1(config)#router eigrp 1
Router1(config-router)#network 192.168.1.0
Router1(config-router)#network 189.8.96.32
Router1(config-router)#no auto-summary
Router1(config-router)#redistribute static
Router1(config)#interface serial 0
Router1(config-if)#bandwidth 8192

Router2(config)#router eigrp 1
Router2(config-router)#network 192.168.2.0
Router2(config-router)#network 189.8.110.0
Router2(config-router)#no auto-summary

```

```
Router2(config-router)#redistribute static
Router2(config)#interface serial 0
Router2(config-if)#bandwidth 8192
```

A configuração da largura de banda no EIGRP tem o mesmo papel que no OSPF. O que difere neste caso é a forma com a qual o roteador irá utilizar esta informação para calcular os custos dos enlaces e os recursos da rede utilizados para os pacotes EIGRP. O EIGRP não usará mais do que 50% da largura de banda definida em um enlace para transmissão de seus pacotes.

Verificando as rotas instaladas em R0:

```
Router0#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 192.168.2.2 to network 0.0.0.0

   189.8.0.0/16 is variably subnetted, 2 subnets, 2 masks
D       189.8.110.0/26 [90/826880] via 192.168.2.2, 00:02:39, Serial0/1/1
D       189.8.96.32/27 [90/826880] via 192.168.1.2, 00:02:48, Serial0/1/0
C       192.168.0.0/24 is directly connected, FastEthernet0/0
C       192.168.1.0/24 is directly connected, Serial0/1/0
C       192.168.2.0/24 is directly connected, Serial0/1/1
D*EX 0.0.0.0/0 [170/826880] via 192.168.2.2, 00:02:39, Serial0/1/1
        [170/826880] via 192.168.1.2, 00:02:39, Serial0/1/0
```

É possível visualizar a rota instalada para 0.0.0.0 com mais detalhes através do comando:

```
Router0#show ip route 0.0.0.0
Routing entry for 0.0.0.0/0, supernet
  Known via "eigrp 1", distance 170, metric 826880, candidate default path,
  type
  external
  Redistributing via eigrp 1
  Last update from 192.168.2.2 on Serial0/1/1, 00:04:38 ago
  Routing Descriptor Blocks:
    192.168.2.2, from 192.168.2.2, 00:04:38 ago, via Serial0/1/1
    Route metric is 826880, traffic share count is 1
    Total delay is 20100 microseconds, minimum bandwidth is 8196 Kbit
    Reliability 255/255, minimum MTU 1500 bytes
    Loading 1/255, Hops 1
  * 192.168.1.2, from 192.168.1.2, 00:04:38 ago, via Serial0/1/0
    Route metric is 826880, traffic share count is 1
    Total delay is 20100 microseconds, minimum bandwidth is 8196 Kbit
    Reliability 255/255, minimum MTU 1500 bytes
    Loading 1/255, Hops 1
```

Por se tratarem de interfaces iguais em ambas as rotas, o EIGRP calculou o mesmo custo para os dois destinos padrão (826880). A equação (métrica) padrão usada para calcular o custo é: $[(10000000 / \text{menor kbps para largura de banda}) + \text{atraso em 10 ms}] * 256$ (Gough, 2002, p. 293). Aplicando a esta equação os valores apresentados através do comando **#show ip route 0.0.0.0** em R0, tem-se o seguinte custo para o destino padrão:

$$[(10000000 / 8196) + (20100 / 10)] * 256 = 826907$$

A diferença com relação ao valor real do custo apresentado (826880) deve-se ao arredondamento dos valores de *delay* e *bandwidth* apresentados.

4.1.3.2 Balanceamento de Carga - Modelo 1 com EIGRP

O EIGRP faz o equilíbrio de carga automaticamente entre enlaces de mesmo custo, da mesma forma que o RIP e o OSPF (Gough, 2002, p. 307). A grande diferença do EIGRP é a possibilidade de balancear a carga entre caminhos desiguais, ou seja, com custos diferentes, sem a necessidade de alterar os custos diretamente. Após se gerar tráfego entre o PC0 e diversos destinos diferenciados (na Internet), durante um determinado período de tempo, foi obtido o seguinte relatório das interfaces:

```
Router0#show interfaces stat
FastEthernet0/0
  Switching path  Pkts In  Chars In  Pkts Out  Chars Out
  Processor        5         300       21         1848
  Route cache     65579     4067426  91695     138131413
  Total           65584     4067726  91716     138133261
Interface FastEthernet0/1 is disabled

Serial0/1/0
  Switching path  Pkts In  Chars In  Pkts Out  Chars Out
  Processor        51        3428      48         3070
  Route cache     46704     69639381 35022     1695011
  Total           46755     69642809 35070     1698081

Serial0/1/1
  Switching path  Pkts In  Chars In  Pkts Out  Chars Out
  Processor        47        2904      46         3173
  Route cache     44992     67576586 30558     1427085
  Total           45039     67579490 30604     1430258
```

Dos 157.468 pacotes recebidos nas interfaces seriais de R0 tem-se uma diferença de apenas 3,93 % de pacotes entre os enlaces, ou seja, é possível verificar que existe uma divisão igualitária (entre destinos) entre as rotas, lembrando que ambas as rotas possuem custos

iguais.

Para fazer uma comparação com o funcionamento do balanceamento de carga entre enlaces desiguais, será necessário “enganar” o cálculo da métrica em um dos enlaces, a fim de simular custos desiguais. Para tal, deve-se alterar o valor de *bandwidth* de 8192 kbps para 3072 kbps do enlace R0-R2, conforme segue:

```
Router0(config)#interface serial 0/1/1
Router0(config-if)#bandwidth 3072

Router2(config)#interface serial 0
Router2(config-if)#bandwidth 3072
```

Após, é possível constatar que a rota não está mais instalada em R0:

```
Router0#show ip route 0.0.0.0
Routing entry for 0.0.0.0/0, supernet
  Known via "eigrp 1", distance 170, metric 826880, candidate default path,
  type
    external
    Redistributing via eigrp 1
    Last update from 192.168.1.2 on Serial0/1/0, 00:02:43 ago
    Routing Descriptor Blocks:
    * 192.168.1.2, from 192.168.1.2, 00:02:43 ago, via Serial0/1/0
      Route metric is 826880, traffic share count is 1
      Total delay is 20100 microseconds, minimum bandwidth is 8196 Kbit
      Reliability 238/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1
```

No entanto, é possível visualizar a rota desinstalada através do comando **#show ip eigrp topology**. Observe que seu custo é de 1347840, ou seja, 1,63 vezes maior que a rota com R1:

```
Router0#show ip eigrp topology
IP-EIGRP Topology Table for AS(1)/ID(192.168.2.1)

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 0.0.0.0/0, 1 successors, FD is 826880
   via 192.168.1.2 (826880/28160), Serial0/1/0
   via 192.168.2.2 (1347840/28160), Serial0/1/1
P 192.168.0.0/24, 1 successors, FD is 28160
   via Connected, FastEthernet0/0
P 192.168.1.0/24, 1 successors, FD is 824320
   via Connected, Serial0/1/0
P 192.168.2.0/24, 1 successors, FD is 1345280
   via Connected, Serial0/1/1
P 189.8.110.0/26, 1 successors, FD is 1347840
   via 192.168.2.2 (1347840/28160), Serial0/1/1
P 189.8.96.32/27, 1 successors, FD is 826880
   via 192.168.1.2 (826880/28160), Serial0/1/0
```

Com o protocolo EIGRP é possível instruir o roteador a instalar tal rota na tabela de roteamento sem, no entanto, alterar o seu custo. Para tal, usa-se o comando **#variance**. Este comando é um multiplicador e permite ao administrador informar ao roteador que o mesmo deverá instalar em sua tabela de roteamento qualquer rota que possua um custo menor que o custo do melhor caminho instalado, multiplicado pelo valor especificado no comando **#variance**. Sendo assim deve-se configurar R0 com os seguintes comandos:

```
Router0(config)#router eigrp 1
Router0(config-router)#variance 2
Router0(config-router)#traffic-share balanced
```

Novamente verificando as rotas instaladas para 0.0.0.0 em R0:

```
Router0#show ip route 0.0.0.0
Routing entry for 0.0.0.0/0, supernet
  Known via "eigrp 1", distance 170, metric 826880, candidate default path,
  type
    external
    Redistributing via eigrp 1
    Last update from 192.168.2.2 on Serial0/1/1, 00:03:36 ago
    Routing Descriptor Blocks:
      192.168.2.2, from 192.168.2.2, 00:03:36 ago, via Serial0/1/1
        Route metric is 1347840, traffic share count is 49
        Total delay is 20100 microseconds, minimum bandwidth is 3072 Kbit
        Reliability 253/255, minimum MTU 1500 bytes
        Loading 1/255, Hops 1
      * 192.168.1.2, from 192.168.1.2, 00:03:36 ago, via Serial0/1/0
        Route metric is 826880, traffic share count is 80
        Total delay is 20100 microseconds, minimum bandwidth is 8196 Kbit
        Reliability 238/255, minimum MTU 1500 bytes
        Loading 1/255, Hops 1
```

O comando **#traffic-share balanced** informa ao roteador para balancear a carga de forma inversamente proporcional às métricas dos enlaces. Para o exemplo R0, deverá balancear a carga de forma que aproximadamente 62% do tráfego seja realizado entre R0-R1 e 38% entre R0-R2. No caso deste teste, estes valores de distribuição percentual não se aplicam a pacotes, mas sim a destinos. Após gerar tráfego entre PC0 e diversos destinos tem-se as seguintes estatísticas em R0:

```
Router0#show interfaces stats
...
Serial0/1/0
  Switching path  Pkts In   Chars In   Pkts Out   Chars Out
  Processor        91      5832      86         5620
  Route cache     137488  204838889  84221     3997019
  Total           137579  204844721  84307     4002639
Serial0/1/1
  Switching path  Pkts In   Chars In   Pkts Out   Chars Out
  Processor        87      5512      84         5691
  Route cache     80584  120378122  46980     2136455
```

Total 80670 120383634 47064 2142146

De um total de 283.923 pacotes recebidos e enviados nas interfaces seriais, 60,81% foram recebidos e enviados através de R1 e 39,19% através de R2, ou seja, valores muito próximos ao projetado. Novamente, quanto maior o número de destinos, mais próximo dos valores projetados se chega. Na Fig. 4.3 é possível verificar também que a velocidade total alcançado entre PC0 e diversos destinos na Internet é de 1625 KB/s (13 Mbps).

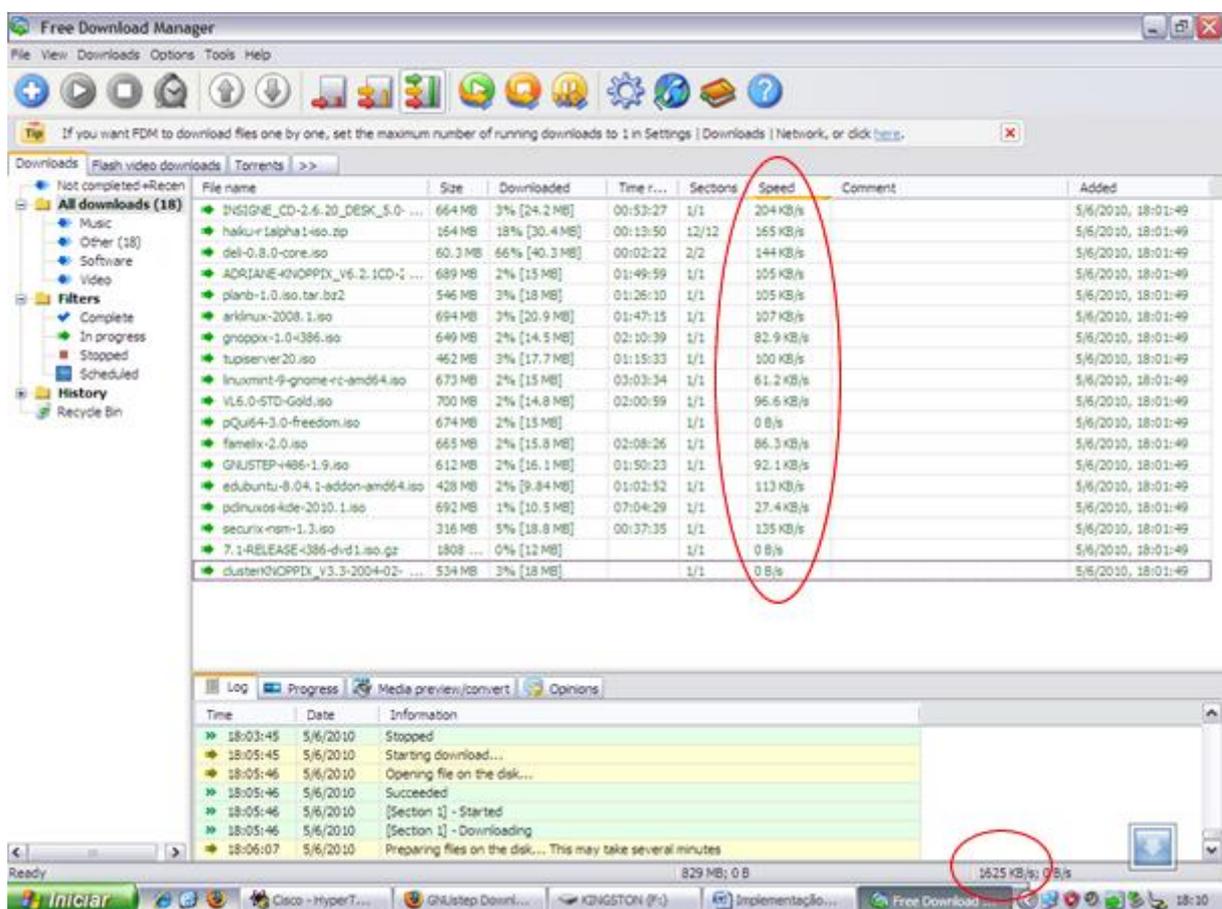


Figura 4.3 – Tela do *Software Free Download Manager*

Outra forma de balancear carga entre enlaces desiguais no EIGRP, sem a utilização do comando **#variance**, é alterar manualmente as especificações de *bandwidth* e *delay* das interfaces. Neste caso o resultado será semelhante ao alcançado no balanceamento para *links* com custos desiguais com o OSPF. Os seguintes comandos devem ser aplicados:

```
R0(config)#interface serial [interface number]
R0(config-if)#bandwidth [bandwidth in kbps]
```

```
R0(config-if)#delay [delay in ms]
```

O EIGRP fará o balanceamento de carga em até 6 caminhos. Também é possível limitar o número máximo de caminhos a serem utilizados pelo EIGRP através do comando **#maximum-paths**.

4.1.4 Conclusão Modelo 1

Conforme destacado anteriormente para o modelo 1, não é possível utilizar a técnica de balanceamento de carga por pacotes. As configurações e testes desta técnica serão apresentadas no modelo 2. Outro ponto importante a respeito deste primeiro modelo é que, caso ocorra a parada de um dos *links*, “metade” dos destinos ficarão indisponíveis, até que ocorra a atualização da tabela de roteamento, sendo esta a vantagem de se usar o balanceamento no modelo 1, quando do ponto de vista da disponibilidade dos serviços, ou seja, em um modelo semelhante, sem balanceamento de carga, todos os destinos ficariam indisponíveis caso o enlace “principal” fique indisponível. Para redes onde os usuários utilizam diversos destinos na Internet, como exemplo, onde os principais tipos de serviços utilizados são através do protocolo HTTP o uso do balanceamento de carga por destino é a melhor opção, pois além de oferecer um balanceamento de carga igualitário entre os *links*, consome menos recursos dos equipamentos.

4.2 IMPLEMENTANDO O MODELO 2

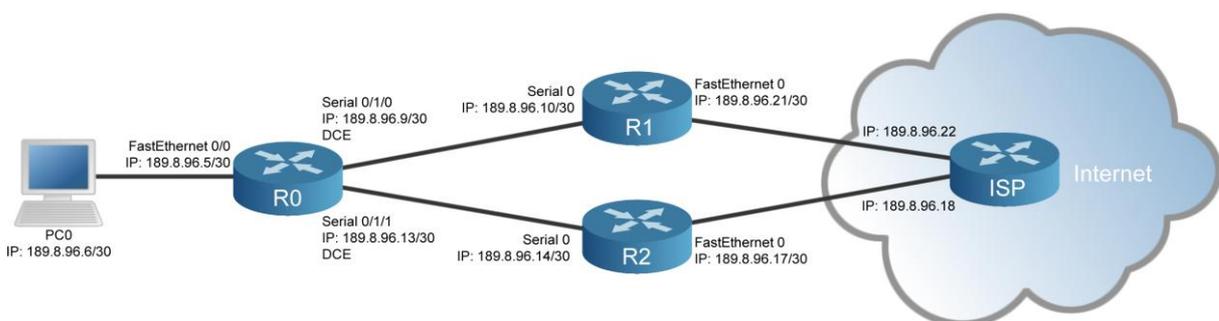


Figura 4.4 – Topologia de rede do modelo 2 aplicada aos testes práticos.

A Fig. 4.4 ilustra a topologia da rede do modelo 2 aplicada aos equipamentos e suas interfaces, bem como os endereços IP utilizados.

Após aplicarmos as configurações básicas (endereços IP, máscara de rede/sub-rede, *clock rate*, etc.) para o modelo 2 não será necessário configurar os roteadores R1 e R2 com as rotas *default*. Esta função deverá ocorrer no roteador ISP, que também deverá estar configurado com o protocolo de roteamento pré-definido entre o ISP e o cliente. O serviço NAT também não precisará ser configurado na rede, já que o ISP poderá fornecer uma ou mais classes de endereçamento IP ao cliente e este poderá utilizar em toda a rede estes endereços. No exemplo de nossa implementação o ISP forneceu uma classe de IP 189.8.96.0/24 que foi dividida em sub-redes menores /30 e estas aplicadas a todas as interfaces e dispositivos da rede. A configuração dos protocolos de roteamento RIPv2, OSPF e EIGRP se assemelham às do modelo 1.

4.2.1 Balanceamento de Carga – Modelo 2 com RIPv2, OSPF e EIGRP

Da mesma forma que as configurações básicas são semelhantes no primeiro e segundo modelos deste trabalho, o mesmo ocorre com o processo de configuração do balanceamento de carga, sendo que neste caso não será necessário detalhar tais processos para a comparação entre os modelos. A diferença básica para este modelo ocorre pelo fato de tais configurações, tanto básicas como de roteamento, devem ser aplicadas também no roteador do ISP de acordo o pré-definido entre as partes, conforme citado anteriormente.

É importante lembrar que os protocolos RIPv2, OSPF e EIGRP fazem o equilíbrio de carga automaticamente entre enlaces de mesmo custo, sendo que, por padrão, os roteadores Cisco utilizam o balanceamento de carga por destino quando dois ou mais *links* de mesmo custos, forem instalados na tabela de roteamento em um determinado roteador. Neste etapa, os resultados apresentados pelos testes realizados no modelo 2, são os mesmos do primeiro modelo.

A grande diferença ao se utilizar o segundo modelo ocorre no fato da possibilidade de utilizar a técnica de balanceamento de carga por pacotes. Novamente, para aplicar tal técnica, é necessário que os *links* possuam custos iguais para o caso da utilização dos protocolos RIPv2 e OSPF. A configuração para ativar tal técnica é a mesma para todos os protocolos de roteamento, conforme segue:

Em Router0:

```
Router0(config)#interface serial 0/1/0
Router0(config-if)#no ip route-cache
```

```
Router0(config)#interface serial 0/1/1
Router0(config-if)#no ip route-cache

Router0(config)#interface fastEthernet 0/0
Router0(config-if)#no ip route-cache

Router0(config)#no ip cef
```

Para a verificação desta técnica dispara-se um *ping* de PC0 (189.8.96.6) para um mesmo destino, neste caso 8.8.8.8 e o comando **#debug ip packet** que fará o *log* dos pacotes enviados e recebidos, conforme segue:

```
PC0>ping 8.8.8.8
```

```
Disparando contra 8.8.8.8 com 32 bytes de dados:
```

```
Resposta de 8.8.8.8: bytes=32 tempo=19ms TTL=52
Resposta de 8.8.8.8: bytes=32 tempo=20ms TTL=52
Resposta de 8.8.8.8: bytes=32 tempo=20ms TTL=52
Resposta de 8.8.8.8: bytes=32 tempo=23ms TTL=52
```

```
Estatísticas do Ping para 8.8.8.8:
```

```
Pacotes: Enviados = 4, Recebidos = 4, Perdidos = 0 (0% de perda),
Aproximar um número redondo de vezes em milissegundos:
Mínimo = 19ms, Máximo = 23ms, Média = 20ms
```

```
Router0#debug ip packet
IP packet debugging is on
Router0#
Jun 23 21:40:22.856: IP: tableid=0, s=189.8.97.6 (FastEthernet0/0),
d=8.8.8.8 (Serial0/1/1), routed via RIB
Jun 23 21:40:22.856: IP: s=189.8.97.6 (FastEthernet0/0), d=8.8.8.8
(Serial0/1/1), g=189.8.97.14, len 60, forward
Jun 23 21:40:22.872: IP: tableid=0, s=8.8.8.8 (Serial0/1/0), d=189.8.97.6
(FastEthernet0/0), routed via RIB
Jun 23 21:40:22.872: IP: s=8.8.8.8 (Serial0/1/0), d=189.8.97.6
(FastEthernet0/0), g=189.8.97.6, len 60, forward
Jun 23 21:40:23.844: IP: tableid=0, s=189.8.97.6 (FastEthernet0/0),
d=8.8.8.8 (Serial0/1/0), routed via RIB
Jun 23 21:40:23.844: IP: s=189.8.97.6 (FastEthernet0/0), d=8.8.8.8
(Serial0/1/0), g=189.8.97.10, len 60, forward
Jun 23 21:40:23.864: IP: tableid=0, s=8.8.8.8 (Serial0/1/0), d=189.8.97.6
(FastEthernet0/0), routed via RIB
Jun 23 21:40:23.864: IP: s=8.8.8.8 (Serial0/1/0), d=189.8.97.6
(FastEthernet0/0), g=189.8.97.6, len 60, forward
Jun 23 21:40:24.844: IP: tableid=0, s=189.8.97.6 (FastEthernet0/0),
d=8.8.8.8 (Serial0/1/1), routed via RIB
Jun 23 21:40:24.844: IP: s=189.8.97.6 (FastEthernet0/0), d=8.8.8.8
(Serial0/1/1), g=189.8.97.14, len 60, forward
Jun 23 21:40:24.860: IP: tableid=0, s=8.8.8.8 (Serial0/1/0), d=189.8.97.6
(FastEthernet0/0), routed via RIB
Jun 23 21:40:24.860: IP: s=8.8.8.8 (Serial0/1/0), d=189.8.97.6
(FastEthernet0/0), g=189.8.97.6, len 60, forward
Jun 23 21:40:25.844: IP: tableid=0, s=189.8.97.6 (FastEthernet0/0),
d=8.8.8.8 (Serial0/1/0), routed via RIB
Jun 23 21:40:25.844: IP: s=189.8.97.6 (FastEthernet0/0), d=8.8.8.8
(Serial0/1/0), g=189.8.97.10, len 60, forward
Jun 23 21:40:25.864: IP: tableid=0, s=8.8.8.8 (Serial0/1/0), d=189.8.97.6
(FastEthernet0/0), routed via RIB
Jun 23 21:40:25.864: IP: s=8.8.8.8 (Serial0/1/0), d=189.8.97.6
(FastEthernet0/0), g=189.8.97.6, len 60, forwardundeall
```

Observa-se que, do total de quatro pacotes enviados, dois foram enviados via R1 e dois via R2, alternadamente. No resultado obtido também é possível verificar que todos os pacotes de resposta foram recebidos em R0 através do enlace com R1. Isto ocorre, pois a configuração de balanceamento de carga não foi aplicada no roteador ISP. Sendo assim, a configuração a ser aplicada em ISP deve ser conforme segue:

```
ISP(config)#interface serial 0
ISP(config-if)#no ip route-cache

ISP(config)#interface serial 1
ISP(config-if)#no ip route-cache

ISP(config)#interface ethernet 0
ISP(config-if)#no ip route-cache

ISP(config)#no ip cef
```

Após aplicar as configurações em ISP e gerar tráfego entre PC0 e diversos destinos, tem-se as seguintes estatísticas em R0:

```
Router0#show interfaces stat
FastEthernet0/0
  Switching path  Pkts In  Chars In  Pkts Out  Chars Out
    Processor      42329   2826357   49730     74219937
    Route cache    0        0         0         0
    Total          42329   2826357   49730     74219937
Interface FastEthernet0/1 is disabled

Serial0/1/0
  Switching path  Pkts In  Chars In  Pkts Out  Chars Out
    Processor      24835   36857483  21192     1155205
    Route cache    0        0         0         0
    Total          24835   36857483  21192     1155205

Serial0/1/1
  Switching path  Pkts In  Chars In  Pkts Out  Chars Out
    Processor      24928   36864174  21193     1152636
    Route cache    0        0         0         0
    Total          24928   36864174  21193     1152636
```

Observe que o total de pacotes enviados pelas duas interfaces de R0 é praticamente o mesmo (21192 e 21193). O total de pacotes recebido sofre uma variação de aproximadamente 1% entre os enlaces, graças à perda de pacotes que ocorre na Internet. Este mesmo padrão ocorre para todos os protocolos testados, OSPF e EIGRP.

4.2.2 Conclusão Modelo 2

O balanceamento de carga por pacotes pode ser uma boa alternativa para aqueles que desejam obter um balanceamento total entre os *links*, desde que estes sejam contratados com

um mesmo ISP e que os roteadores utilizados possuam especificações técnicas e capacidade de suportar o processamento dos dados de acordo com o tráfego que se planeja utilizar. Do ponto de vista da disponibilidade dos serviços em caso de falha em um ou mais *links*, todos os destinos ficarão “alcançáveis” pelos demais *links*, mesmo antes da convergência da rede. O problema ocorre, pois metade dos pacotes (em caso de dois *links*, ocorrendo a falha de um deles) serão perdidos até que a convergência ocorra e que R0 remova a rota inoperante de sua tabela, gerando lentidão nos serviços. Para algumas aplicações, esta perda de pacotes pode até mesmo comprometer o serviço. É importante destacar que, para aqueles que procuram uma boa redundância, a utilização de apenas uma operadora (ISP) não é uma boa escolha, sendo que falhas que possam comprometer todos os *links* com o ISP não ocorrem de forma tão rara.

Quanto aos protocolos de roteamento testados e sua respectivas técnicas de balanceamento de carga, o EIGRP mostrou melhores resultados, sendo o mais flexível e praticamente englobando as técnicas dos demais protocolos. É o mais indicado caso o(s) roteador(es) utilizado(s) seja(m) da marca Cisco, independentemente da topologia e da complexidade da rede. Para os demais casos o protocolo RIPv2 pode ser mais indicado para o caso de uma rede que não ofereça muita complexidade e devido a sua simplicidade de implantação. O OSPF é mais indicado em caso de uma rede mais complexa ou que necessite de maior escalabilidade.

4.3 IMPLEMENTANDO O MODELO 3

4.3.1 Análise de caso usando o modelo 3.

Para os testes e análises de balanceamento de carga através do protocolo BGP, deverá ser analisado um caso real. A empresa Redel Internet (ISP) presta serviços de acesso à Internet “via rádio” (WISP), também conhecido como *Fixed Wireless Access* (FWA), além de Serviços de Valor Adicionado (SVA), correio eletrônico e hospedagem de domínios. A empresa é detentora do ASN 28670, um bloco de endereços IP classe B (máscara 20 com 4094 *hosts*) e possui uma saída BGP, com *links* de duas operadoras (OP1 e OP2), ambos com 50 Mbps cada.

A implantação de tais serviços necessita de especial atenção já que envolve diversos serviços, prestados para cerca de 20.000 clientes. Os principais objetivos da empresa são, primeiramente, possuir acesso redundante a Internet (*multihoming*) sendo que em caso de falha em um dos *links*, todo o tráfego seja transferido para o outro *link*, no menor tempo

possível e em segundo lugar distribuir o tráfego entre os *links* fazendo um balanceamento de forma que nenhum dos *links* fique congestionado em determinados momentos, havendo ociosidade no outro. O primeiro passo é mais simples, sendo que é pré-requisito para a obtenção do recurso de AS. Para o balanceamento de carga será utilizada uma técnica de distribuição do bloco de IPs públicos em posse do ISP, conforme será visto a seguir.

Para tal implementação será utilizado um computador, instalado com um sistema operacional Linux Debian e utilizando o *software* de roteamento Zebra. Mais informações sobre o equipamento a ser utilizado:

Processador: Intel(R) Xeon(R) CPU X3323 2.50GHz
 Memória RAM: 2070900 kB
 Hard Disk: 234612,6 MB
 Interfaces: 4 Interfaces Gigabit Ethernet (eth0, eth1, eth2 e eth3)
 Sistema Operacional: Linux version 2.6.26-2-686 (Debian 2.6.26-21lenny4)
 Zebra version: Quagga 0.99.10
 Hostname: REDEL_BGP

O Zebra é um *software* livre, desenvolvido para plataformas Unix, que gerencia protocolos de roteamento TCP/IP. É parte do Projeto GNU, sendo que suporta os protocolos BGP-4, RIPv1, RIPv2 e OSPF. A interface do usuário e os comando utilizados no Zebra são praticamente os mesmo dos IOS Cisco. A configuração das interfaces *Gigabit Ethernet* é detalhada na Tabela 4.1.

Tabela 4.1 – Configuração das Interfaces *Gigabit Ethernet* do Roteador REDEL_BGP

Interface	Endereço IP	Máscara	Gateway
eth0	189.8.96.1	255.255.0.0	
eth1	187.59.81.134	255.255.255.252	187.59.81.133
eth2	201.70.217.86	255.255.255.252	201.70.217.87
eth3	Não utilizada		

A Fig. 4.5 ilustra a topologia da rede aplicada ao equipamento e suas interfaces, bem como o endereçamento IP utilizado.

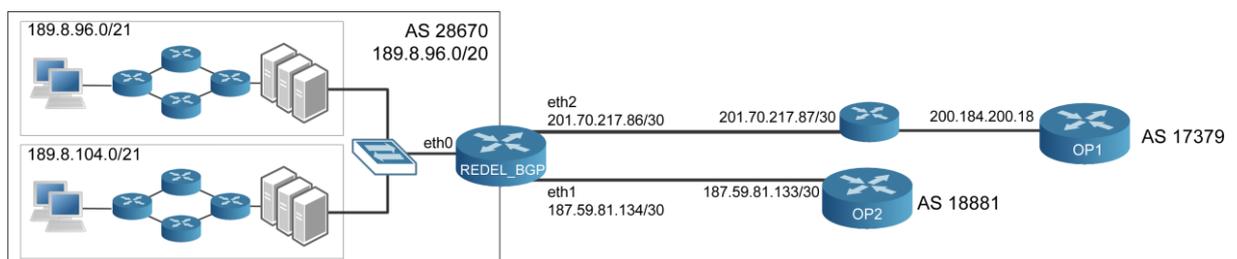


Figura 4.5 – Topologia de rede do modelo 3 aplicada aos testes práticos.

4.3.2 Controlando o tráfego com BGP-4.

A melhor maneira de controlar o tráfego de um determinado roteador BGP é limitar o modo como são propagadas as atualizações de roteamento. Projetar como as informações de roteamento devem ser encaminhadas através da rede forma um nível básico de segurança e dinamiza o fluxo de dados, além de simplificar a sua manutenção (Gough, 2002, p. 398). Para a nossa implementação serão aplicados dois tipos de filtragem:

- Lista de prefixos (*prefix-list*) – Listas de prefixos filtram atualizações de roteamento com base no prefixo do endereço, especialmente na redistribuição. Estas regras filtram números de rede, e não caminhos de sistemas autônomos.
- Mapas de rotas (*route-map*) – um mapa de rota é uma lista de acesso sofisticada que define critérios de acordo com os quais um roteador atua quando é encontrada uma correspondência para os critérios estabelecidos. É usado no BGP-4 para definir os atributos que determinam a base para a seleção de melhor caminho a um destino.

O termo *multihoming* designa que a rede possui mais de uma conexão com a Internet. Vários enlaces não só fornecem redundância como também possibilitam o balanceamento de carga, apresentando assim uma melhora no desempenho. Para tal, é preciso observar algumas questões a respeito da conexão com mais de um ISP:

- Os provedores podem não estar propagando as mesmas rotas na Internet. Se os provedores estiverem enviando subconjuntos das rotas necessárias, poderá ocorrer um problema maior com a conectividade se o enlace com um dos provedores falhar.
- É possível que, se a rede estiver conectada a dois diferentes provedores, seu sistema autônomo se torne um ponto de trânsito entre os ISP. Isto pode ocorrer se um dos provedores visualizar o seu sistema autônomo como a melhor rota para um ou mais destinos do outro provedor.

A configuração no nível do ISP é a solução para estas questões e deve ser tratado ao se estabelecer o serviço com cada ISP.

4.3.2.1 Modelos de Conexão *Multihoming*

Segundo Gough (2002, p. 402), existem três formas de fazer a conexão com a Internet utilizando *multihoming* BGP. A primeira é configurar o roteador para aceitar apenas rotas *default* dos provedores conforme demonstra a Fig. 4.6.

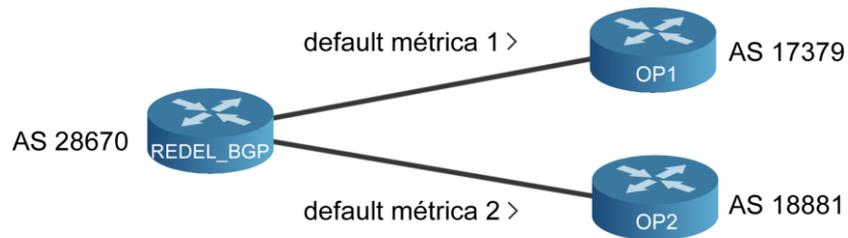


Figura 4.6 – Modelo de Conexão *Multihoming* - Apenas Rotas *Default*

A segunda forma de fazer essa conexão é receber uma tabela de roteamento parcial dos provedores conforme demonstra a Fig. 4.7.

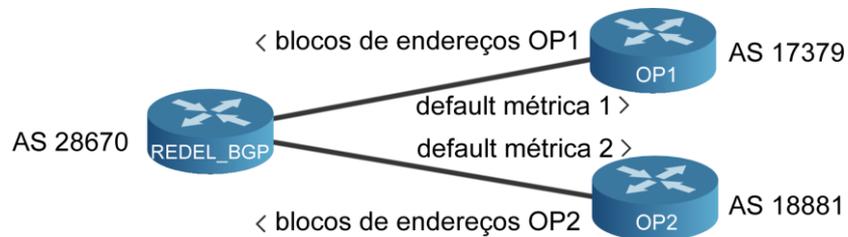


Figura 4.7 – Modelo de Conexão *Multihoming* –Tabelas Parciais

Neste ambiente, o roteador do AS 28670 recebe via BGP apenas os blocos de endereços que cada provedor é responsável, ou seja, está dentro de seus domínios. Através desse mecanismo o roteamento para o bloco de endereços de cada provedor é otimizado, contudo, para o restante dos endereços é necessário definir rotas *default*. Esse mecanismo também é simples, mas exige um pouco mais de recurso computacional se comparado à solução anterior.

A terceira opção é a mais trabalhosa e que exige mais dos roteadores de borda. Trata-se do modelo denominado *full route*. A Fig. 4.8 ilustra esse modelo.

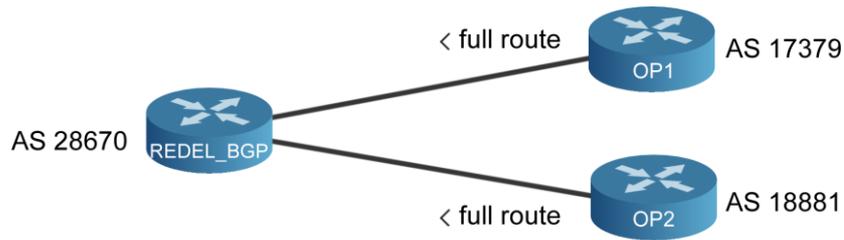


Figura 4.8 – Modelo de Conexão *Multihoming - Full Route*

Nessa arquitetura os roteadores de borda recebem toda a tabela de roteamento dos provedores podendo tomar a decisão de roteamento através de qualquer provedor. O ponto de atenção nesse modelo é que o AS privado pode se tornar um ponto de trânsito entre os dois provedores. Esse comportamento deve ser inibido sob pena de super utilização dos recursos da organização sem benefício próprio.

O protocolo BGP é definido como sendo de roteamento “*path vector*”, ou seja, ele anuncia caminhos e as redes que são alcançáveis no final desse caminho. Esse anúncio contém os atributos utilizados no caminho que são similares à métrica. Esse protocolo permite aos administradores definir políticas de como os dados fluirão pelo AS.

4.3.3 Configuração do BGP-4

A configuração do roteador REDEL_BGP em relação ao protocolo BGP e as técnicas acima descritas são:

CONFIGURAÇÃO LOCAL:

<code>router bgp 28670</code>	Define o AS local.
<code>no synchronization</code>	Não realiza o sincronismo entre os protocolos IGP e BGP.
<code>bgp router-id 189.8.96.48</code>	Identifica o roteador local e seu respectivo endereço. Esse é o endereço que deve ser definido como vizinho nos roteadores remotos.
<code>bgp log-neighbor-changes</code>	Registra para efeito de gerência de eventos as mudanças nas vizinhanças.
<code>redistribute connected</code>	Redistribui todas as rotas localmente conectadas e ativas no roteador. Substitui o uso do comando #network .
<code>redistribute static</code>	Redistribui as rotas estáticas configuradas no roteador.
<code>no auto-summary</code>	Não realiza a sumarização automática dos endereços.

CONFIGURAÇÃO PARA OP1:

<code>neighbor 200.184.200.18 remote-as 17379</code>	Define o endereço IP do AS 17379 (vizinho). Esse endereço deve estar diretamente conectado ou o roteador deve possuir uma rota para ele. Neste caso, como não está diretamente conectado, os seguintes comandos devem ser configurados: #neighbor 200.184.200.18 interface eth2 #neighbor 200.184.200.18 ebgp-multihop 5
<code>neighbor 200.184.200.18 description BGP OP1</code>	Apenas descreve o vizinho.
<code>neighbor 200.184.200.18 interface eth2</code>	Define qual a interface ou rota até o roteador vizinho.
<code>neighbor 200.184.200.18 ebgp-multihop 5</code>	Define que o endereço do vizinho (<i>neighbor</i>) pode estar a até 5 saltos (<i>hops</i>) de distância. É necessário quando o endereço declarado do vizinho não está diretamente conectado.
<code>neighbor 200.184.200.18 remove-private-AS</code>	Remove ASN privados das atualizações de roteamento.
<code>neighbor 200.184.200.18 send-community</code>	Envia o atributo da comunidade ao vizinho. O atributo de comunidade é especificado através dos mapas de rotas.
<code>neighbor 200.184.200.18 soft-reconfiguration inbound</code>	Roteador armazena as atualizações.
<code>neighbor 200.184.200.18 route-map AS17379_IN in</code>	Define a política de entrada de rotas de acordo com o mapa de rotas AS17379_IN.
<code>neighbor 200.184.200.18 route-map AS17379_OUT out</code>	Política para filtragem de rotas de saída de acordo com o mapa de rotas AS17379_OUT.

CONFIGURAÇÃO PARA OP2:

<code>neighbor 187.59.81.133 remote-as 18881</code>	Define o endereço IP do AS 18881 (vizinho). Esse endereço deve estar diretamente conectado ou o roteador deve possuir uma rota para ele. Neste caso está diretamente conectado.
<code>neighbor 187.59.81.133 description BGP OP2</code>	Apenas descreve o vizinho.
<code>neighbor 187.59.81.133 remove-private-AS</code>	Remove ASN privados das atualizações de roteamento.
<code>neighbor 187.59.81.133 send-community</code>	Envia o atributo da comunidade ao vizinho. O atributo de comunidade é especificado através dos mapas de rotas.
<code>neighbor 187.59.81.133 soft-reconfiguration inbound</code>	Roteador armazena as atualizações.
<code>neighbor 187.59.81.133 route-map AS18881_IN in</code>	Define a política de entrada de rotas de acordo com o mapa de rotas AS18881_IN.
<code>neighbor 187.59.81.133 route-map AS18881_OUT out</code>	Política para filtragem de rotas de saída de acordo com o mapa de rotas AS18881_OUT.

CONFIGURAÇÃO DAS LISTAS DE PREFIXOS:

ip prefix-list ANY seq 10 permit 0.0.0.0/0 le 32	Define que todos os prefixos são permitidos.
ip prefix-list DEFAULT-ONLY seq 10 permit 0.0.0.0/0	Define que apenas rotas <i>default</i> serão permitidas. Será utilizado para configurar a primeira forma de <i>mulihoming</i> apresentado no item 5.3.2.1.
ip prefix-list DENY-IN seq 100 permit 0.0.0.0/0 le 32	Define que rotas <i>default</i> serão permitidas.
ip prefix-list OP2 seq 10 permit 189.8.104.0/21	Define que os prefixos 189.8.104.0/21 são permitidos.
ip prefix-list OP1 seq 10 permit 189.8.96.0/21	Define que os prefixos 189.8.96.0/21 são permitidos.
ip prefix-list LOCAL-AS seq 5 permit 189.8.96.0/20 le 32	Define que os prefixos 189.8.0.0/20 com comprimento até /32 são permitidos.

CONFIGURAÇÃO DOS MAPAS DE ROTAS (469):

Mapas de rotas servem para o BGP controlar e modificar informações de roteamento e também definir as condições da propagação de rotas entre domínios de roteamento. A sintaxe de um mapa de rota é:

#route-map nome [[permit | deny] | [número sequencial]]

O nome identifica o mapa. O número sequencial indica a posição que a instância do mapa de rota deve ter em relação a outras instâncias do mesmo nome, sendo as mesmas ordenadas sequencialmente, ou seja, a ordem com que será verificado.

Os comandos **#match** e **#set** são usados para definir as condições. O comando **#match** define a condição a ser satisfeita e o comando **#set** especifica a ação a ser tomada caso o *update* corresponda à condição.

Para o diagrama da rede apresentada, o AS 28670 deseja atingir a seguinte política de roteamento:

- O tráfego para a rede 189.8.96.0/21 deve ser recebido através do *link* com a OP1. Em caso de falha deste *link*, todo o tráfego deve ser recebido através do *link* com a OP2.
- O tráfego para a rede 189.8.104.0/21 deve ser recebido através do *link* com a OP2. Em caso de falha deste *link*, todo o tráfego deve ser recebido através do *link* com a

OP1.

Para tal, o roteador REDEL_BGP deve anunciar e negar os prefixos, na ordem, conforme segue:

Para OP1:

- Divulgar o prefixo 189.8.96.0/21.
- Negar a divulgação do prefixo 189.8.104.0/21.
- Divulgar o prefixo 189.8.96.0/20.

Para OP2:

- Divulgar o prefixo 189.8.104.0/21.
- Negar a divulgação do prefixo 189.8.96.0/21.
- Divulgar o prefixo 189.8.96.0/20.

MAPAS DE ROTAS PARA POLÍTICAS DE ENTRADA DE ROTAS ATRAVÉS DO AS 17379 (OP1):

<pre>route-map AS17379_IN permit 10 match ip address prefix-list DEFAULT-ONLY set community 28670:1 28670:17379 additive</pre>	<p>Define que as rotas que combinam com a lista de prefixos DEFAULT-ONLY são aditivadas com os atributos de comunidade 28670:1 e 28670:17379.</p>
<pre>route-map AS17379_IN permit 1000 set community 28670:17379 additive</pre>	<p>Define a adição dos atributos de comunidade 28670:1 e 28670:18881. É utilizada para o caso de não haver combinação com o <i>match</i> do mapa acima. Se um <i>match</i> não é configurado o <i>set</i> é aplicado e a próxima cláusula é verificada.</p>

MAPAS DE ROTAS PARA POLÍTICAS DE ENTRADA DE ROTAS ATRAVÉS DO AS 18881 (OP2):

<pre>route-map AS18881_IN permit 10 match ip address prefix-list DEFAULT-ONLY set community 28670:1 28670:18881 additive</pre>	<p>Define que as rotas que combinam com a lista de prefixos DEFAULT-ONLY são aditivadas com os atributos de comunidade 28670:1 e 28670:18881.</p>
<pre>route-map AS18881_IN permit 1000 set community 28670:18881 additive</pre>	<p>Define a adição dos atributos de comunidade 28670:1 e 28670:18881. É utilizada para o caso de não haver combinação com o <i>match</i> do mapa acima. Se um <i>match</i> não é configurado, o <i>set</i> é aplicado e a próxima cláusula é verificada.</p>

Os mapas de rotas para políticas de entrada determinam que o roteador irá aprender apenas rotas *default*. Os atributos de comunidade adicionados servem para identificação com relação ao processo que estão sendo aplicados e o caminho a que se referem.

MAPAS DE ROTAS PARA POLÍTICAS DE SAÍDA DE ROTAS ATRAVÉS DO AS 17379 (OP1):

<pre>route-map AS17379_OUT permit 10 match ip address prefix-list OP1 set origin igp set community 17379:23014 17379:23034 17379:23044 17379:23054 28670:20 additive</pre>	<p>Rotas que combinam com a lista de prefixos OP1 são definidas como rotas de origem interna com relação ao AS (#set origin igp) e possuem os seguintes atributos de comunidade 17379:23014 17379:23034 17379:23044 17379:23054 28670:20. Os primeiro quatro atributos são informados pela OP1 afim de que se possa manipular o tráfego através da rede da própria OP1. O último atributo (28670:20) adiciona em 20 os atributo.</p>
<pre>route-map AS17379_OUT deny 20 match ip address prefix-list OP2 set origin igp set community 28670:20 additive set as-path prepend 28670 28670 28670 28670</pre>	<p>Rotas que combinam com a lista de prefixos OP2, não serão distribuídas. Além disso, são definidas como rotas de origem interna com relação ao AS (#set origin igp) e possuem o atributo de comunidade adicionado em 20 (28670:20). O comando #set as-path define a sequência de ASs que uma rota segue para atingir um determinado destino. Adicionando #prepend 28670 28670 28670 28670, estamos “piorando” esta rota, afim de influenciar o tráfego. Ao aumentar a rota estamos desestimulando o tráfego através deste caminho, já que rotas com caminhos mais curtos tendem a ser escolhidas.</p>
<pre>route-map AS17379_OUT permit 100 match ip address prefix-list LOCAL-AS set origin igp set community 28670:20 additive</pre>	<p>Rotas que combinam com a lista de prefixos LOCAL-AS são marcadas como rotas de origem interna e possuem o atributo de comunidade adicionado em 20 (28670:20).</p>

MAPAS DE ROTAS PARA POLÍTICAS DE SAÍDA DE ROTAS ATRAVÉS DO AS 18881 (OP2):

<pre>route-map AS18881_OUT permit 10 match ip address prefix-list OP2 set origin igp set community 28670:10 additive</pre>	<p>Rotas que combinam com a lista de prefixos OP2 são definidas como rotas de origem interna com relação ao AS (#set origin igp) e possuem o atributo de comunidade adicionado em 10 (28670:10).</p>
<pre>route-map AS18881_OUT deny 20 match ip address prefix-list OP1</pre>	<p>Rotas que combinam com a lista de prefixos</p>

<pre>set origin igp set community 28670:10 additive set as-path prepend 28670 28670</pre>	<p>OP1, não serão distribuídas. Além disso, são definidas como rotas de origem interna com relação ao AS (#set origin igp) e possuem o atributo de comunidade adicionado em 10 (28670:10). O comando #set as-path define a sequência de ASs que uma rota segue para atingir um determinado destino. Adicionando o #prepend 28670 28670, estamos “piorando” esta rota afim de influenciar o tráfego. Ao aumentar a rota estamos desestimulando o tráfego através deste caminho, já que rotas com caminhos mais curtos tendem a ser escolhidas.</p>
<pre>route-map AS18881_OUT permit 100 match ip address prefix-list LOCAL-AS set origin igp set community 28670:10 additive</pre>	<p>Rotas que combinam com a lista de prefixos LOCAL-AS são marcadas como rotas de origem interna e possuem o atributo de comunidade adicionado em 10 (28670:10).</p>

4.3.3.1 Resumo das Configurações

As listas de prefixos OP1 e OP2 definem quais sub-blocos do bloco 189.8.96.0/20 serão anunciados para cada um dos provedores.

Assim o comando **#route-map AS18881_OUT deny 20** simplesmente nega o anúncio para a OP2 do sub-bloco /21 enviado para a OP1. Os demais comandos não têm efeito e foram aplicados em algum momento durante o “ajuste fino” da rede para testes. O mesmo ocorre (de maneira inversa) no comando **#route-map AS17379_OUT deny 20**.

Ao anunciar o sub-bloco /21 separadamente, os roteadores da Internet irão entender que devem acessar as redes através de caminhos separados. Ao anunciar sequencialmente o bloco completo /20 através de ambos os provedores, os demais roteadores da Internet também irão entender que podem acessar a todos os destinos através de ambos os caminhos. Esta segunda configuração servirá para o caso de um dos *links* falhar.

Neste exemplo, as comunidades são simplesmente termos associados às rotas que são usados para tomar alguma ação, dentro do roteador usando um mapa de rotas. Por exemplo, no caso da comunidade 28670:10, sua função é simplesmente identificar de forma rápida que uma rota está sendo anunciada para OP2. A comunidade 28670:20 identifica as rotas enviadas para OP1.

4.3.4 Anúncio das Rotas

O primeiro teste ocorre internamente que, através do comando **#show ip bgp neighbors [neighbor address] advertised-routes**, pode-se verificar quais rotas estão sendo anunciadas aos roteadores vizinhos (OP1 e OP2).

```
REDEL_BGP# show ip bgp neighbors 200.184.200.18 advertised-routes
BGP table version is 0, local router ID is 189.8.96.48
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
                r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 189.8.96.0/20	189.8.111.254	0		32768	i
*> 189.8.96.0/21	189.8.111.254	0		32768	i

```
REDEL_BGP# show ip bgp neighbors 187.59.81.133 advertised-routes
BGP table version is 0, local router ID is 189.8.96.48
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
                r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 189.8.96.0/20	187.59.81.134	0		32768	i
*> 189.8.104.0/21	187.59.81.134	0		32768	i

Cada roteador vizinho recebe o anúncio do sub-bloco específico (/21), conforme configuração e do bloco completo /20. Para testar as rotas escolhidas pelos roteadores da Internet utilizou-se um dos roteadores públicos disponíveis para verificação de rotas e testes. O roteador utilizado será o roteador da Tiscali (AS 3257), empresa italiana de telecomunicações. Para tal deve-se efetuar um acesso telnet no endereço `telnet://route-server.ip.tiscali.net`.

Para verificar a rota até ambos os sub-blocos da Redel Internet escolhida por este roteador:

```
route-server.as3257.net>show ip bgp 189.8.96.0
BGP routing table entry for 189.8.96.0/20, version 262839
Paths: (1 available, best #1, table Default-IP-Routing-Table)
Flag: 0x820
Not advertised to any peer
3257 6762 17379 28670
213.200.64.93 from 213.200.64.93 (213.200.87.12)
Origin IGP, metric 5, localpref 100, valid, external, best
Community: 3257:8102 3257:30023 3257:54900 3257:54901
```

```
route-server.as3257.net>show ip bgp 189.8.104.0
BGP routing table entry for 189.8.104.0/21, version 614173
Paths: (1 available, best #1, table Default-IP-Routing-Table)
Flag: 0x820
Not advertised to any peer
3257 12956 18881 28670
213.200.64.93 from 213.200.64.93 (213.200.87.12)
Origin IGP, metric 5, localpref 100, valid, external, best
Community: 3257:8102 3257:30060 3257:54900 3257:54901
```

Observe que para o sub-bloco 189.8.104.0/21 o roteador escolheu uma rota que passa pelo AS 18881 (OP2). Para o sub-bloco 189.8.96.0/21 o roteador escolheu uma rota que passa pelo ASN 17379 (OP1). Observe que para o caso de OP1, como o roteador recebe também o bloco completo, ele insere apenas o bloco completo /20. De qualquer forma esta opção de englobar todo o bloco através da rota via OP1 não irá sobrescrever a opção da rota via OP2 para o sub-bloco 189.8.96.104/21. É possível confirmar este fato ao se verificar as rotas adicionadas pelo roteador da OP2. Através da ferramenta *Looking Glass* fornecida pela operadora, uma ferramenta fornecida pela operadora para verificação de configurações e testes por parte de seus clientes e parceiros.

Ao submeter o comando `#show ip bgp` para o endereço/prefixo 189.8.96.0, tem-se o seguinte resultado:

```

flags: * = Valid, > = Selected, I = via IBGP, A = Announced
origin: i = IGP, e = EGP, ? = Incomplete

flags destination          gateway          lpref    med aspath origin
I  189.8.96.0/21          189.59.255.11    200      0 17379 28670 i
I  189.8.96.0/21          189.59.255.11    200      0 17379 28670 i

success.

```

O resultado demonstra que o roteador da OP2 “entende” o prefixo 189.8.96.0 com máscara /21 e como melhor caminho via AS 17379 (OP1). Quando solicitamos o prefixo 189.8.104.0, tem-se o seguinte resultado:

```

flags: * = Valid, > = Selected, I = via IBGP, A = Announced
origin: i = IGP, e = EGP, ? = Incomplete

flags destination          gateway          lpref    med aspath origin
I  189.8.104.0/21         189.59.255.75    600      0 28670 i
I  189.8.104.0/21         189.59.255.75    600      0 28670 i

success.

```

Outra ferramenta interessante para entender melhor a rede ao qual seu AS está conectado é fornecida pela *Hurricane Electric Internet Services*, através do endereço eletrônico <http://bgp.he.net>. A Fig. 4.9 apresenta a topologia de parte do mapa da rede de ASs ao qual o AS 28670 está conectado.

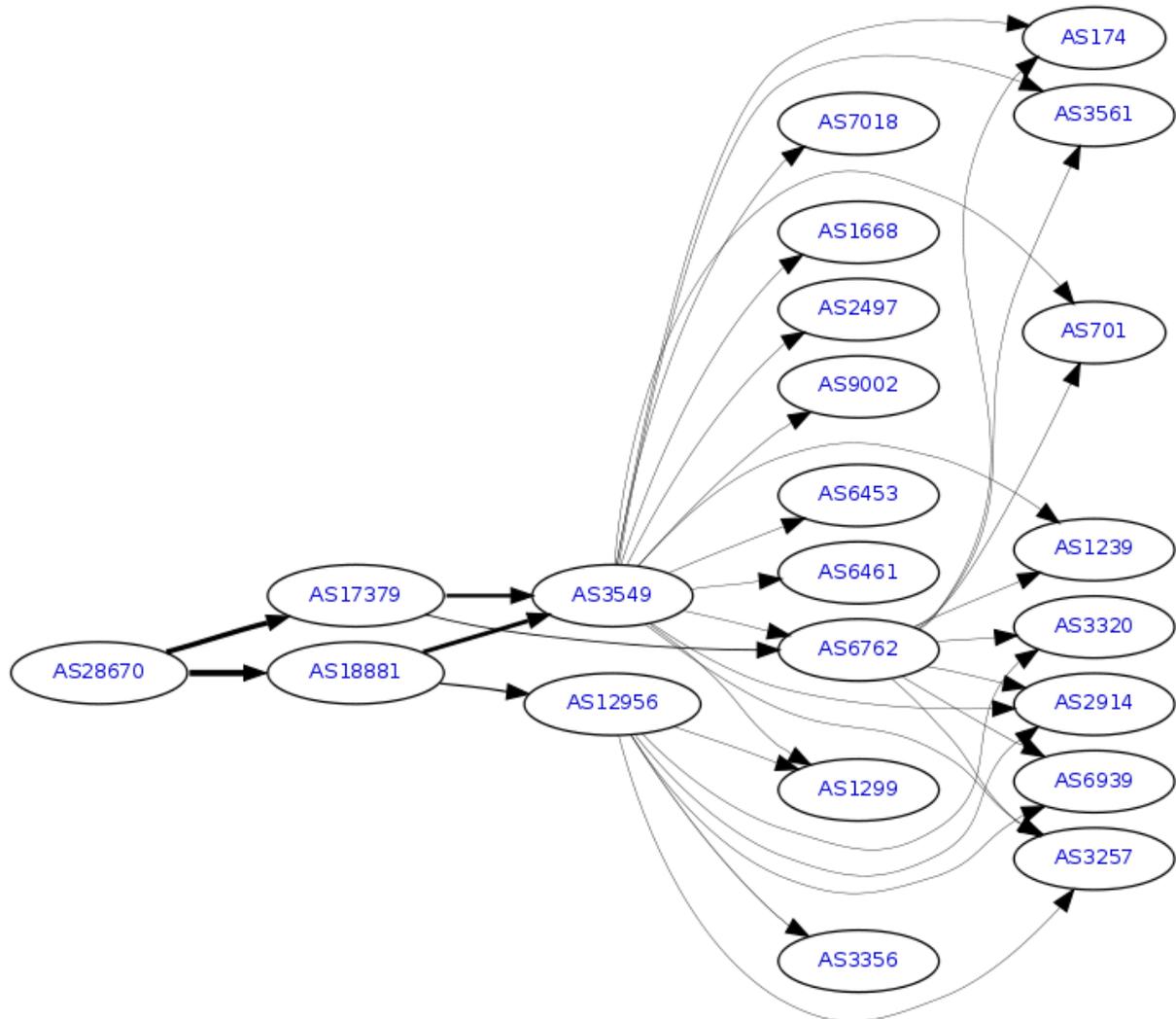


Figura 4.9 – Mapa parcial da rede conectada ao AS 28670.

4.3.5 Balanceamento de Carga – Modelo 3 com BGP-4

Com a separação do bloco de IPs /20 em dois sub-blocos /21, o balanceamento de carga ocorrerá automaticamente entre os endereços de IP de cada bloco. Um endereço IP pertencente ao sub-bloco 189.8.104.0/21, enviará e receberá dados através do *link* com a OP2. Para os endereços do sub-bloco 189.8.96.0/21, o tráfego ocorrerá através do *link* com a OP1.

Neste caso, o uso de cada *link* dependerá de como o administrador de rede distribuir os computadores, redes e servidores e de acordo com o uso dos recursos necessários, entre ambos os sub-blocos. Um exemplo simplificado seria distribuir os clientes da empresa atribuindo IPs do sub-bloco 189.8.96.0/21 para uma metade destes clientes e IPs do sub-bloco 189.8.104.0/21 para a outra metade.

Como a empresa possui diversos serviços como hospedagem de *sites*, acesso e provimento de banda de Internet, correio eletrônico, entre outros, a distribuição ocorre de acordo com as Tabelas 4.2 e 4.3:

Tabela 4.2 – Distribuição dos Endereços e Sub-blocos IPs do Bloco 189.8.96.0/21 (OP1)

IP/SUB-BLOCO	HOST(S)	DETALHES
189.8.98.0/24	254 <i>hosts</i>	Clientes empresariais de acesso e provimento de banda com IP fixo.
189.8.96.35	Servidor MySQL	Banco de dados MySQL.
189.8.96.38	<i>Gateway</i> rede interna	Servidor <i>gateway</i> para a rede privada da empresa (com aproximadamente 30 computadores)
189.8.96.40	DNS Primário	Servidor primário utilizado para hospedagem de sites (DNS primário).
189.8.96.45	<i>Mail Server</i> Redel	Servidor de <i>e-mail</i>
189.8.96.50	<i>Mail Server</i> Domains	Servidor de <i>e-mail</i> e DNS secundário para hospedagem de sites.
189.8.96.73	Servidor <i>Proxy</i> (RGS)	Servidor <i>proxy</i> e NAT com 534 clientes
189.8.96.81	Servidor <i>Proxy</i> (CCE)	Servidor <i>proxy</i> e NAT com 1.169 clientes
189.8.96.84	Servidor <i>Proxy</i> (WVI)	Servidor <i>proxy</i> e NAT com 669 clientes

Tabela 4.2 – Distribuição dos Endereços e Sub-blocos IPs do Bloco 189.8.104.0/21 (OP2)

IP/SUB-BLOCO	HOST	DETALHES
189.8.108.0/24	254 <i>hosts</i>	Clientes empresariais de acesso e provimento de banda com IP Fixo. Dividido em 64 sub-blocos /30.
189.8.109.0/24	254 <i>hosts</i>	Clientes residenciais de acesso e provimento de banda com IP variável. Dividido em 4 sub-blocos /26.
189.8.110.5	Servidor <i>Proxy</i> (QP3)	Servidor <i>proxy</i> e NAT com 1.188 clientes
189.8.110.20	Servidor <i>Proxy</i> (CRI)	Servidor <i>proxy</i> e NAT com 1.602 clientes

Todos os clientes citados no detalhamento dos Servidores *Proxy* referem-se a clientes residenciais com acesso e provimento de banda de Internet.

Para esta implementação, o balanceamento de carga é observável através de uma ferramenta que recolhe e exibe informações a respeito do uso dos *links* através de gráficos, instalada em um dos servidores, chamada Cacti, as Fig. 4.10, 4.11 e 4.12 apresentam estes gráficos.

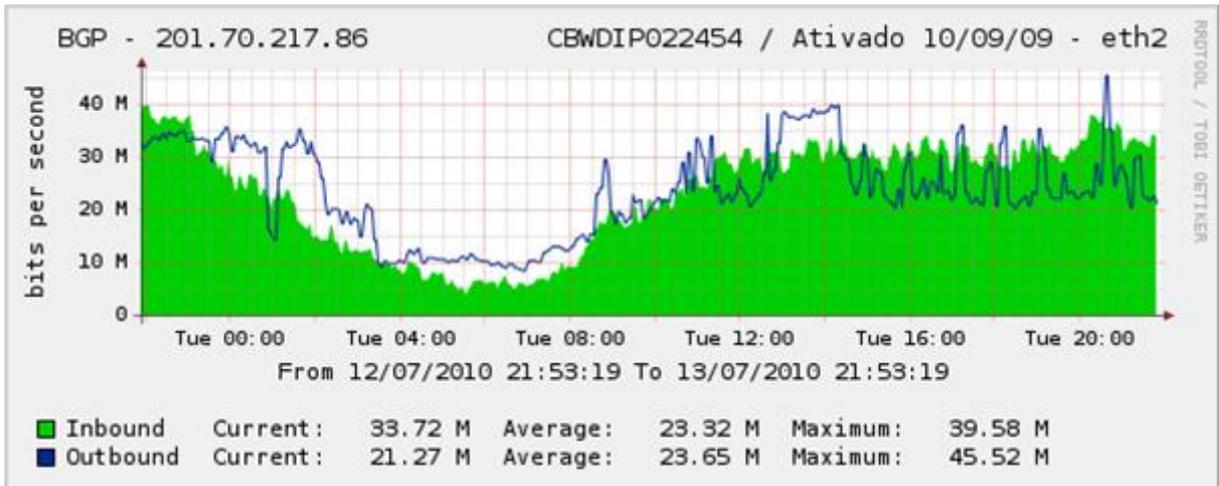


Figura 4.10 – Uso do *link* entre REDEL_BGP e OP1 no período de 24 horas

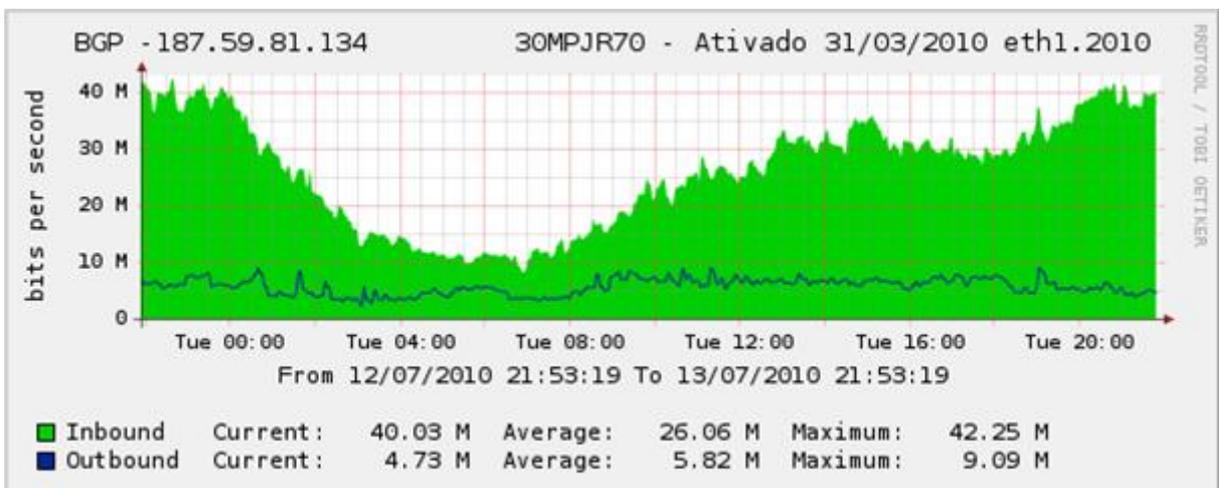


Figura 4.11 – Uso do *link* entre REDEL_BGP e OP2 no período de 24 horas

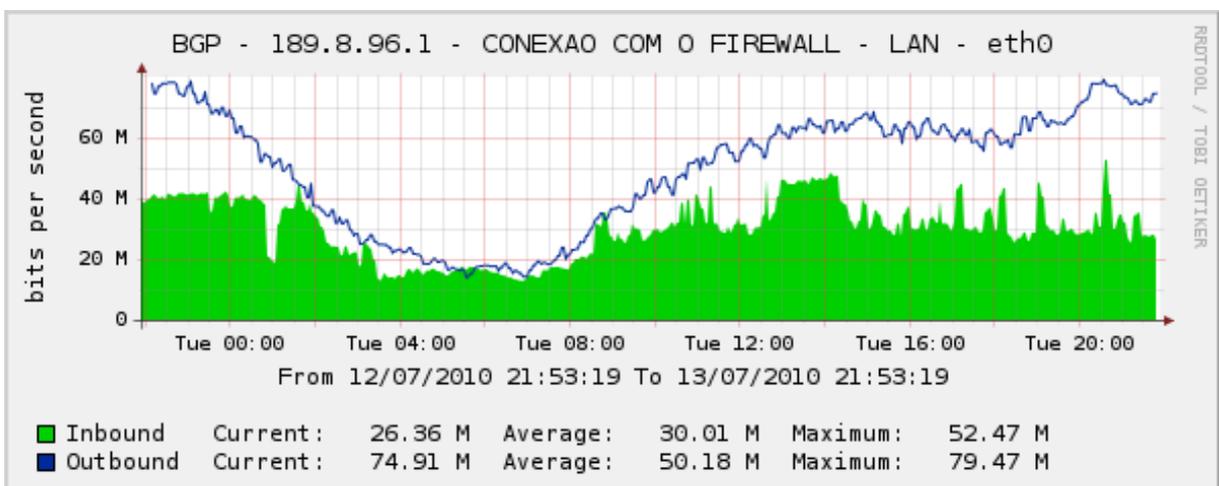


Figura 4.12 – Uso total dos *link* entre REDEL_BGP e OP1 e REDEL_BGP e OP2 (eth0) no período de 24 horas

Através dos gráficos das Fig. 4.10, 4.11 e 4.12 é possível observar que, com relação ao tráfego de entrada (*Inbound*), em média, 47,23% ocorre através do *link* com a OP1 e 52,77% ocorre através do *link* com a OP2. Este balanceamento ocorre desta forma, graças à separação dos clientes de acesso e provimento de banda de Internet, que utilizam principalmente aplicações que necessitam de mais tráfego de *download* do que de *upload*. Do total de 5.924 *hosts* e clientes destes serviços, 2.626 (44,33%) estão localizados no sub-bloco de IPs divulgados através da OP1, e 3.298 (55,67%) no sub-bloco de IPs divulgados através da OP2.

Também pode-se notar uma grande diferença com relação ao tráfego de saída (*Outbound*) entre os *links*, sendo que, em média, 80,25% deste tráfego ocorre através do *link* com a OP1. Esta diferença ocorre, devido as aplicações que mais utilizam a parte de *upload* dos *links* (e-mail, hospedagem de sites e banco de dados) estarem localizadas no sub-bloco de IPs divulgados para a OP1.

4.3.6 Conclusão Modelo 3

A abordagem mais comum para se balancear carga utilizando-se o protocolo de roteamento BGP é a segmentação dos blocos de endereços IPs conforme foi abordado neste trabalho. Este modelo utilizando esta abordagem evidentemente é mais complexo e pode parecer pouco flexível com relação aos demais, mas é importante lembrar que as vantagens são consideráveis, dependendo de sua necessidade. Através desta implementação, é possível distribuir endereços de IPs públicos aos seus clientes e/ou dispositivos de rede, com a possibilidade de utilizar diversos fornecedores (ISPs), melhorando assim a disponibilidade dos seus serviços. A própria detenção de endereços de IP próprios, conquistados ao se adquirir o título de *Autonomous System*, incrementa o leque de possibilidades e soluções para os problemas de uma determinada rede.

Quanto à flexibilidade da rede, pode-se sempre subdividir seu bloco de IPs e até diversos blocos /24, ou até mesmo solicitar novos blocos ao NIC.br. É claro que esta abordagem requer uma constante observação da rede e eventuais manutenções, alterando configurações e migrando serviços entre os blocos, afim de se obter o melhor dos *links* e equipamentos, algo que também deve ocorrer nos demais modelos.

5 CONCLUSÃO

O trabalho apresentou as técnicas mais utilizadas de balanceamento de carga através de roteamento em redes IP, bem como seus respectivos protocolos de roteamento e aspectos técnicos. A implementação e os testes aplicados aos modelos propostos apresentaram as informações necessárias para as devidas comparações inicialmente planejadas, bem como para entender a importância do balanceamento de carga.

Verificou-se, através dos testes, que a implementação de balanceamento de carga em redes IP permite que se utilize os recursos da rede ao máximo, trazendo consigo, além da melhoria no desempenho, maior economia e uma maior disponibilidade dos serviços, com os recursos de redundância vinculados às técnicas, as quais são necessidades primordiais em praticamente todas as redes de empresas e instituições que utilizam recursos da Internet com grande frequência.

Dentre os protocolos de roteamento IGP e suas respectivas técnicas, aquele que apresentou melhores resultados foi o EIGRP, sendo que o mesmo praticamente engloba as funcionalidades dos demais (RIPv2 e OSPF), e é o mais indicado em caso de uso de roteadores Cisco.

Como resultado final da análise dos modelos propostos, concluiu-se ainda que para grandes redes ou redes que necessitem de uma maior escalabilidade, a implementação do protocolo de roteamento BGP é a mais indicada. É importante lembrar que para tal, é necessário a obtenção do AS e de um maior planejamento.

Por questões de segurança em redes ativas, em caso de manutenção ou de novas implementações, em qualquer dos casos e modelos verificados, indica-se manter um “plano de recuperação” prevenindo a rede de possíveis casos de falhas nas configurações aplicadas, bem como realizá-las em horários em que há pouco uso dos serviços.

Os resultados práticos e teóricos obtidos neste Trabalho de Conclusão de Curso demonstraram que os objetivos inicialmente traçados foram alcançados com sucesso. Afim de se aprofundar o estudo de balanceamento de carga em redes, sugere-se como tema para trabalhos futuros o estudo de técnicas de balanceamento de carga em níveis superiores ao da camada de inter-rede do Modelo TCP/IP, como por exemplo, o estudo de técnicas de priorização de tráfego de acordo com os protocolos de transporte e dos tipos de aplicações mais utilizados nas redes.

REFERÊNCIAS

- 1 Cisco Systems. **CCNA 2: Conceitos Básicos de Roteadores**, v. 3.1.1, 2003.
- 2 Cisco Systems. **CCNA 3: Conceitos Básicos de Switching e Roteamento Intermediária**, v. 3.1.1, 2003.
- 3 Cisco Systems. **How Does Unequal Cost Path Load Balancing (Variance) Work in IGRP and EIGRP.** Disponível em: <http://www.cisco.com/application/pdf/paws/13677/19.pdf>, Acesso em 29 de junho de 2010.
- 4 Cisco Systems. **Internetworking Technology Handbook – Internet Protocols (IP)**, Junho de 1999.
- 5 Cisco Systems. **One- and 2-Port Serial and Asynchronous High-Speed WAN Interface Cards for Cisco 1800, 1900, 2800, 2900, 3800, and 3900 Series Integrated Services Routers Datasheet**, 2009.
- 6 DUATO, José. **Interconnection Networks: an engineering approach**. Piscataway: IEEE, 1997.
- 7 FILIPPETTI, Marco Aurélio. **CCNA 4.0 – Guia Completo de Estudo**. Florianópolis: Visual Books, 2006.
- 8 GOUGH, Clare. **Guia de Certificação do exame CCNP Routing**. Tradução de Ana Beatriz Tavares. Alta Books: Rio de Janeiro, 2002. Título original: Cisco CCNP Routing.
- 9 KUROSE, James F. **Redes de Computadores e a Internet: uma abordagem top-down**. Tradução de Arlete Simille Marques. 3. ed. São Paulo: Pearson Addison Wesley, 2006. Título original: Computer networking third edition a top-down approach featuring the Internet.

- 10 RFC 791; University of Southern California; **Internet Protocol**; Setembro 1981.
- 11 RFC 1930; J. Hawkinson, BBN Planet, T. Bates, MCI; **Guidelines for creation, selection, and registration of an Autonomous System (AS)**; Março 1996.
- 12 RFC 4271; Y. Rekhter, T. Li, S. Hares; **A Border Gateway Protocol 4 (BGP-4)**; Janeiro 2006
- 13 SOARES, Luiz Fernando G. **Redes de Computadores: das LANs, MANs e WANs às redes ATM**. Rio de Janeiro: Campus, 1995.
- 14 TANENBAUM, Andrew S. **Redes de Computadores**. Tradução de Vandenberg D. De Souza. 4. ed. Rio de Janeiro: Elsevier, 2003. Título original: Computer Networks.
- 15 Tele Geography, **Global Internet Geography Executive Summary**, 2010.

OUTROS TRABALHOS EM:

www.projeteredes.com.br