UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA

Dissertação de Mestrado

Gerenciamento de Sites Multi-Homed

por

Ioram Schechtman Sette

UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA

Ioram Schechtman Sette

Gerenciamento de Sites Multi-Homed

Este trabalho foi apresentado à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

ORIENTADOR: Dr. André Santos
COORIENTADOR: Dr. Geber Ramalho

Recife, 02 de dezembro de 2002

A meus avós Iosif (em memória) e Zelda e meus pais Sergio e Sonia.

Agradecimentos

Agradeço, sinceramente, a André pela orientação e acompanhamento deste trabalho, me fornecendo todas as condições técnicas necessárias para a realização do mesmo, e a Geber pela co-orientação e pelas dicas de como fundamentar e produzir um trabalho científico.

Gostaria de agradecer também aos professores Fábio e José Neuman por participarem da banca e pelos comentários e sugestões que aprimoraram esta dissertação.

Agradecimentos especiais a Rossam, que fez a instalação, pré-configuração e conexão aos *sites* RNP e Embratel da máquina utilizada na pesquisa, a João Paulo pela ajuda nas configurações dos softwares de captura de pacotes da rede e a Ricardo pela ajuda no estudo sobre análise de performance de sistemas.

Agradeço aos amigos da turma “Mestrandos 2000” pelo coleguismo, amizade e colaboração, criando um ambiente tão agradável para estudar e pesquisar.

Ao meu avô, Iosif, pelo carinho e incentivo aos estudos, desde pequeno. À minha avó, Zelda, aos meus pais, Sergio e Sonia, e aos meus irmãos Iara e Sergio, que me deram amor, carinho e apoio em todas as passagens de minha vida.

À minha namorada, Gabriela, que sempre acreditou e mim, me dando força e servindo de inspiração para a conclusão deste trabalho.

E aos demais que direta e indiretamente contribuíram para o alcance de meu objetivo.

Resumo

A confiabilidade em suas conexões à Internet tem sido uma necessidade cada dia mais presente nas empresas. Uma solução para este problema, considerando-se a queda do preço das conexões, é a contratação de conexões de provedores diferentes, tornando a empresa em um *site multi-homed*.

Nesse contexto, surge a dificuldade de se administrar estas conexões da melhor forma para a empresa. São considerados então os conceitos de sistemas autônomos (AS), e não autônomos, abordados nesta dissertação, utilizando como estudo de caso o Centro de Informática da UFPE, um *site multi-homed* não autônomo. Baseado em análises de tráfego deste site, sugerimos uma solução geral para gerenciamento de *sites multi-homed* não autônomos.

A solução proposta concentra-se na automatização do processo de gerenciamento de *sites multi-homed*, normalmente realizado pelos administradores de rede, manualmente. Este processo baseia-se na configuração de mecanismos como as rotas estáticas, serviço de nomes (DNS), tradução de endereços (NAT) e roteamento por políticas, que influenciam nos caminhos percorridos pelos pacotes na rede.

Abstract

Reliability in Internet connections has become a need in many enterprises lives. A solution to this problem, considering the lower prices of Internet links, is the acquisition of connections from distinct providers (ISPs), making the enterprise a multi-homed site.

In this context, managing the site at the best way for the enterprise becomes a difficult task. We consider the concepts of autonomous (AS) and non-autonomous systems, explained in this dissertation, using the "Centro de Informática" of UFPE as case study of a multi-homed non-autonomous site. Based in traffic analysis from this site, we propose a general solution for managing multi-homed non-autonomous sites.

The proposed solution concentrates in the multi-homed site's management process automation, normally performed by the network administrators, manually. This process is based on the configuration of mechanisms like static routes, domain name service (DNS), network address translation (NAT) and policy routing, that influence in the ways the packets follow in the network.

Conteúdo

Lista de Figuras

Lista de Tabelas

Capítulo1

Introdução

A maioria das empresas possui apenas um ponto de acesso (conexão) à Internet, através de um provedor de acesso (ISP – Internet Service Provider). Esta estrutura é bastante simples de ser configurada, e normalmente atende às necessidades de qualidade de serviço das empresas.

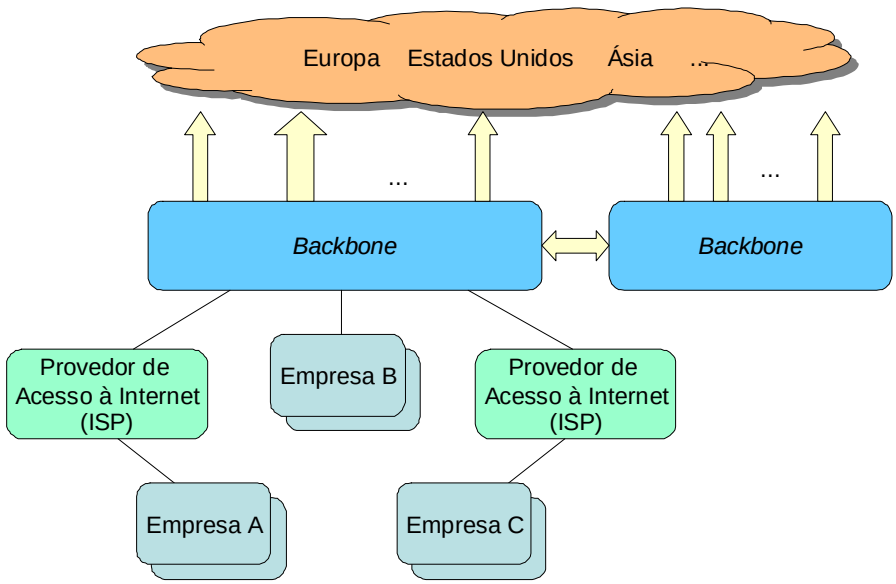


Figura 1.1: Hierarquia entre os provedores de acesso à Internet

O diagrama da figura 1.1 ilustra a hierarquia entre os provedores de acesso à Internet. Os *backbones* são grandes provedores com pontos de acesso disseminados nas grandes cidades do país, interligados com conexões de alta velocidade, formando uma estrutura similar a uma "espinha dorsal" (*backbone*, em inglês). Os *backbones* possuem várias conexões, também de alta velocidade, com outros *backbones* nacionais e internacionais. Neles também se conectam algumas empresas que necessitam de conexões de alta velocidade (como a empresa B acima) e provedores de acesso locais, que têm a finalidade de conectar usuários finais e empresas menores (como as empresas A e C do diagrama).

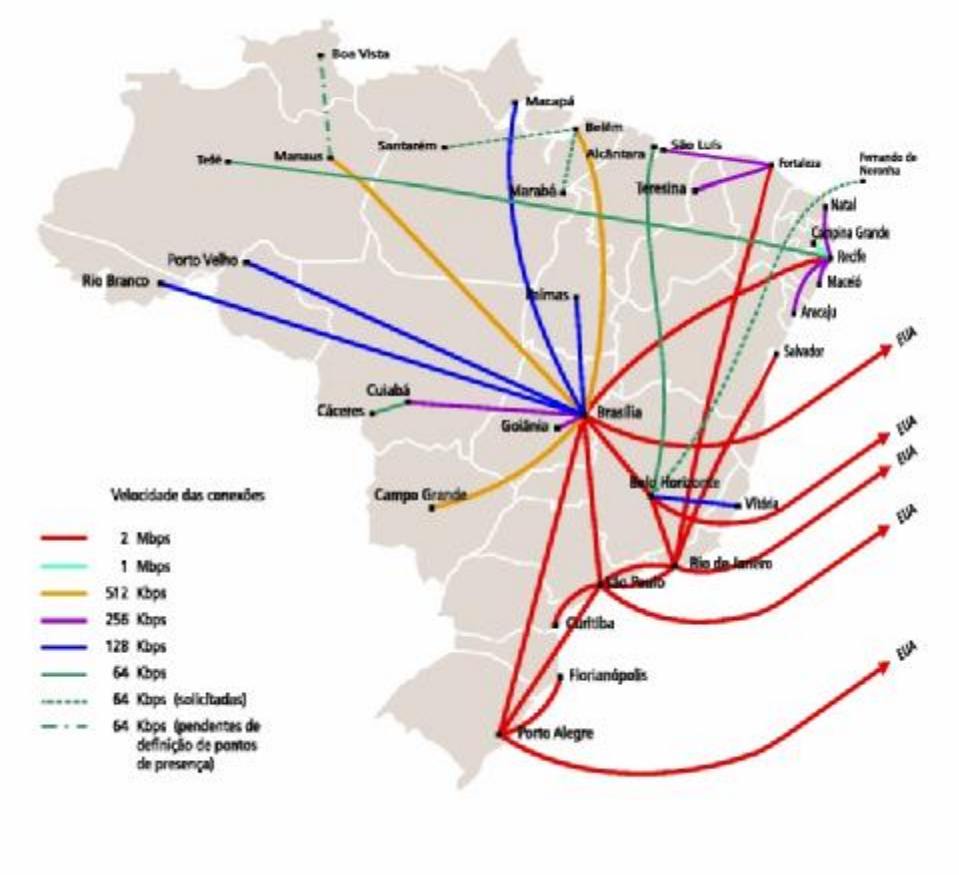


Figura 1.2: Mapa de um *backbone* brasileiro: a RNP (Rede Nacional de Pesquisa), em 1998

Entretanto, ao terem apenas um provedor de acesso, estas empresas estão sujeitas a um maior risco de perda de conectividade à Internet por várias razões:

- Queda da conexão da empresa até o provedor;
- Queda da conexão do provedor até o *backbone* ao qual ele se conecta à Internet, se o provedor não possuir conexões e/ou *backbones* redundantes;
- Sobrecarga de tráfego na conexão do provedor ao *backbone* ou nas interconexões de troca de tráfego entre os *backbones*.

Organizações que precisam minimizar estes riscos de desconexão ou perda de performance têm como alternativa a instalação de conexões múltiplas a um ou mais provedores. A estes *sites* com múltiplas conexões, dá-se o nome de *sites multi-homed*.

Outras razões que podem levar ao uso de múltiplas conexões podem ser a economia de custo (contratando-se provedores mais baratos para utilizar serviços que não necessitem de tanta qualidade), ou a melhora na velocidade de acesso a alguns *sites*, que fiquem mais próximos aos novos provedores contratados.

1.1 Dificuldades no Gerenciamento de Múltiplas Conexões

Ao optar por usar múltiplas conexões com provedores diferentes, uma empresa tem dificuldades em gerenciar o uso dos mesmos, pois terá que trabalhar com faixas de endereços IP diferentes para cada um das conexões. Transformar o *site* da empresa em um Sistema Autônomo (*Autonomous System* - AS) é a solução oferecida pelos órgãos da Internet para esse problema. Entretanto, é extremamente difícil se tornar um AS.

Faixas de endereços IP são atribuídas aos provedores pelos organismos que gerenciam a alocação desses números. A IANA (*International Assigned Numbers Authority*), órgão responsável por coordenar as funções centrais da Internet, delega a alocação de endereços a órgãos responsáveis nas regiões geográficas:

- APNIC (*Asia Pacific Network Information Centre*) (http://www.apnic.net/) – Asia/Região do Pacífico;
- ARIN (*American Registry for Internet Numbers*) (http://www.arin.net/) – Américas e África (abaixo do Saara);
- RIPE NCC (*Réseaux IP Européens*) (http://www.ripe.net/) – Europa e arredores.

Por sua vez, estas instituições dividem a tarefa de alocação de suas faixas com outras entidades chamadas de NIC (*Network Information Center*). No Brasil, o NIC responsável

pelo controle de faixas IP é o Registro.br (http://www.registro.br/), órgão controlado pela RNP/Fapesp.

Os NIC também são os órgãos responsáveis por decidir quando os *sites* podem se tornar AS. No Brasil, é necessário que a empresa seja bastante grande, justificando o uso de pelo menos 4096 números IP (faixa /20). Com a escassez de endereços IPv4, pouquíssimas empresas (apenas as grandes provedoras de acesso) conseguem justificar esta necessidade.

Sistemas Autônomos recebem um número único de identificação chamado ASN (*Autonomous System Number)*. Os roteadores dos ASs, através de protocolos como o BGP (*Border Gateway Protocol*) [17], trocam informações sobre as faixas de endereços IP que são acessíveis através dos ASs vizinhos e os respectivos custos (distâncias em *hops*) para se chegar até elas. Desta forma, o roteador decide através de que provedor enviar cada pacote, baseado na proximidade do endereço IP destino.

A dificuldade para um *site* se tornar um AS é necessária para evitar um tráfego muito grande de pacotes BGP entre os roteadores da Internet, além de evitar o aumento das tabelas dinâmicas de roteamento dos mesmos, comprometendo sua performance. Outro motivo para a dificuldade é a escassez de números ASN, pois eles não foram criados para suportar um número tão grande de *sites* autônomos.

Uma solução para *sites* que não conseguem tornar-se autônomos seria possuir conexões redundantes com apenas um provedor. Assim, o *site* pode utilizar um ASN privado (a faixa de 64512 a 65535 é reservada para esses casos) e se comunicar através de BGP com o provedor. Existem, entretanto, desvantagens em se ter conexões com um mesmo provedor. Se a banda do provedor se esgotar, por exemplo, ambas as conexões ficarão lentas. Se houver ainda um problema mais sério com o provedor, todas as conexões serão comprometidas, deixando o *site* sem conectividade.

As soluções existentes para gerenciamento de *sites multi-homed* (de provedores distintos) não-autônomos baseiam-se em geral na intuição dos administradores do *sites*, que utilizam técnicas ou ferramentas para desviar o tráfego entre as conexões, estaticamente.

Sistemas Autônomos, apesar de garantirem a redundância na conexão do *site* à Internet e de acessar os *hosts* remotos pelo provedor mais próximo, não resolvem problemas de balanceamento de carga entre múltiplas conexões.

1.2 Objetivos

Neste trabalho, analisamos os aspectos determinantes de tráfego em *sites multi-homed* não-autônomos, apresentamos uma metodologia e arquitetura para a melhoria do problema de gerenciamento destes *sites* e apresentamos um estudo de caso onde analisamos o tráfego real de um *site multi-homed* e simulamos a adoção de nossa estratégia neste *site*.

Atualmente, a administração desses *sites* é difícil e realizada de forma manual pelos administradores de redes e sistemas. Neste trabalho propomos um sistema que faça tal gerenciamento de forma transparente ao usuário/administrador.

Para isso, definimos uma metodologia para facilitar o trabalho de gerenciamento e melhorar o aproveitamento das conexões em *sites multi-homed*. Esta metodologia consiste na análise de tráfego da rede, com o propósito de adquirir conhecimento sobre a mesma. Este conhecimento deve ser utilizado como entrada em mecanismos que distribuem o tráfego da rede entre as conexões de forma inteligente, dinâmica e automatizada, em contrapartida à intuição do administrador de sistemas que é utilizada, na maioria dos casos, de forma estática e manual.

No Capítulo 2, apresentamos o estado da arte em gerenciamento de *sites multi-homed*. Fazemos um estudo dos mecanismos de roteamento, DNS (*Domain Name Service*) e NAT (*Network Address Translation*) que influem diretamente no fluxo do tráfego da rede e abordamos as soluções existentes atualmente para gerenciamento de Sistemas Autônomos e Não-autônomos, mostrando suas deficiências.

No Capítulo 3, descrevemos métodos de análise de tráfego de uma rede IP a fim de distribuí-lo entre conexões com a Internet. Utilizamos como estudo de caso para esta análise a rede do Centro de Informática da UFPE.

No Capítulo 4, propomos uma solução para o balanceamento do tráfego da rede entre as conexões, baseada na análise feita e em políticas pré-estabelecidas pelos administradores. Em seguida, simulamos a solução proposta e analisamos seu desempenho.

No Capítulo 6, apresentamos as conclusões desta dissertação e sugestões para trabalhos futuros.

Capítulo 2

Sites Multi-Homed

Com o rápido crescimento da Internet, as empresas e usuários dependem cada vez mais dessa rede para suas atividades diárias. Nos casos onde a rede passa a ser essencial e até mesmo vital, é necessário que as empresas tenham a garantia de que suas conexões com a Internet nunca sejam interrompidas e que proporcionem sempre uma boa velocidade de acesso a seus serviços, para os usuários.

Para garantir conectividade na rede, a solução é manter mais de um ponto de acesso à Internet, preferencialmente por ***backbones***[1] distintos. Desta forma, a empresa ficaria segura no caso de uma conexão ficar fora do ar por algum período, pois haveria(m) outra(s) que garantiria(m) o acesso.

Com duas ou mais conexões, a empresa, além de garantir seu acesso à Internet, a não ser que todas as conexões fiquem inacessíveis, poderá usufruir de outra vantagem. Se

[1] *"Backbone"* textualmente significa espinha dorsal. Assim são chamados os fornecedores dos grandes *links* de acesso à Internet com conectividade nacional e internacional. Geralmente são as grandes empresas de telecomunicação que ocupam este papel. No Brasil, podemos citar a Embratel, Telemar, etc. O *backbone* utilizado no Brasil por instituições de ensino e pesquisa é a RNP – Rede Nacional de Pesquisa.

as conexões forem a provedores que utilizam *backbones* distintos, a empresa pode utilizar sempre a conexão mais rápida em relação aos *hosts* que acessa e aos *hosts* (possivelmente seus clientes) que acessam seus serviços. Na figura a seguir, ilustramos um modelo onde o *site multi-homed* possui conexões (linhas pretas) aos *backbones* 1 e 2 através dos quais pode acessar os clientes 1 e 2 (linhas pretas tracejadas). Os caminhos indicados em verde utilizam a menor rota ao *site* destino. Já os indicados em vermelho utilizam rotas mais longas, aumentando a chance de passar por conexões saturadas. O critério de proximidade é uma possibilidade quando desejamos determinar a conexão para se chegar a um *site*.

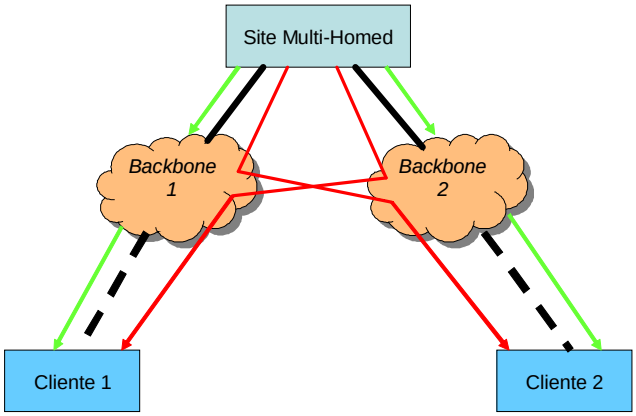


Figura 2.1: Rotas entre um *site multi-homed* e um *site* cliente. Menor (verde) e maior (vermelho) distância.

2.1 Aspectos Determinantes do Tráfego

A fim de utilizarmos a conexão desejada para trafegar os pacotes de nossa rede em cada ocasião, precisamos em primeiro lugar saber como direcionar o tráfego através das conexões existentes. Portanto, estudaremos nesta seção mecanismos que influenciam de alguma forma o caminho que os pacotes de uma rede IP seguem para chegar ao seu destino.

Os principais mecanismos que exercem influência no tráfego IP são:

- **Roteamento estático**, que indica o caminho por onde pacotes IP devem seguir baseado no endereço de destino. Este caminho pode ser uma interface de rede ou um endereço IP, que servirá de ponte (*gateway*) para os pacotes chegarem ao *host* destino.
- **Roteamento por políticas (*Policy Routing*)**, que indica o caminho por onde pacotes IP devem seguir, baseado em políticas como, por exemplo, o endereço de origem ou o serviço utilizado.
- **Serviço de nomes (DNS)**, que indica aos *hosts* externos qual o endereço IP dos *hosts* internos através de consultas por nomes.
- **Tradução de Endereços (NAT)**, que traduz os endereços privados da rede interna (não roteáveis na Internet) para endereços reais da Internet. É para este endereço real que o *host* extemo enviará o pacote seguinte do fluxo IP.

A seguir, discutiremos como utilizar esses mecanismos, como fazê-los exercer influência na escolha da conexão a ser utilizada pelo tráfego, e como combiná-los para obter um determinado resultado.

2.1.1 Roteamento Estático

Roteamento é o mecanismo que, através de rotas, indica o caminho na rede por onde os pacotes devem seguir para chegarem a seus destinos. Este mecanismo é ativado quando pacotes IP estão sendo enviados para algum *host*, portanto, o roteamento está relacionado ao tráfego de saída.

O roteamento pode ser estático ou dinâmico. No primeiro caso, a tabela de rotas não sofre alteração automática. Elas são modificadas através de comandos dados manualmente pelo administrador da rede. No segundo, a tabela de roteamento sofre alterações automaticamente, normalmente através de protocolos de roteamento.

Na Internet, existem dois tipos de protocolos de roteamento dinâmico: o intra-domínio e o extra-domínio. Protocolos de roteamento intra-domínio são utilizados para escolha dinâmica da melhor rota, internamente nos *sites* entre um *site* e um AS. Uma das

aplicações deste protocolo seria num *site* com múltiplas conexões a um mesmo provedor. Neste caso, o protocolo pode inclusive gerenciar o balanço da carga entre tais conexões. Exemplos de tais protocolos são o RIP (*Routing Internet Protocol*), implementando um algoritmo baseado em "vetor distância" e usado no início da rede, e o OSPF (*Open Short Path First*), baseado no algoritmo do "caminho mais curto" e atualmente recomendado. O protocolo de roteamento extra-domínio é usado para roteamento entre os ASs e tem como diferença o suporte a regras de política. Na Internet, utiliza-se o BGP (*Border Gateway Protocol*).

Normalmente, os protocolos de roteamento dinâmico exigem a troca de informações dos roteadores com seus vizinhos. Em nossos estudos, sugerimos um mecanismo de roteamento dinâmico baseado na alteração automática da tabela de rotas estáticas. Este mecanismo poderia ser usado sem a necessidade de troca de informações entre os roteadores e, portanto, sem a necessidade do *site* tornar-se um sistema autônomo (AS).

A rota estática é composta de:

- endereço de rede
- máscara de rede
- endereço do *Gateway* ou Interface de rede
- outros parâmetros (opcionais)

Quando o pacote está sendo enviado para algum *host*, o roteamento estático percorre as rotas existentes em sua tabela de rotas até que alguma seja encontrada, da seguinte forma: para cada rota, a "máscara de rede" é aplicada (fazendo-se um "e" binário) ao endereço IP de destino. O resultado é comparado com o "endereço de rede": se forem iguais, o pacote é encaminhado para o endereço do *gateway* ou em *broadcast* pela interface de rede definida; se forem diferentes, a rota seguinte da tabela é testada.

A última linha da tabela de rotas representa uma rota especial: a *default*. Essa rota tem como endereço de rede o IP 0.0.0.0 e máscara 0.0.0.0, forçando os pacotes, que não se enquadraram nas rotas anteriores, a seguirem um caminho especificado.

A figura 2.2 ilustra um exemplo típico de roteamento estático em um roteador com duas portas seriais e uma porta *ethernet*. Tal roteador possui uma tabela de rota, como mostra a figura. Um *host* interno à rede se comunica com o mesmo através da interface Ethernet0, utilizando a rota para a faixa de IP 10.0.0.0. Este host envia pacotes para quatro endereços IP utilizando o roteador como seu *gateway*. O roteador então, encaminha o pacote para o endereço ou interface de acordo com suas regras internas:

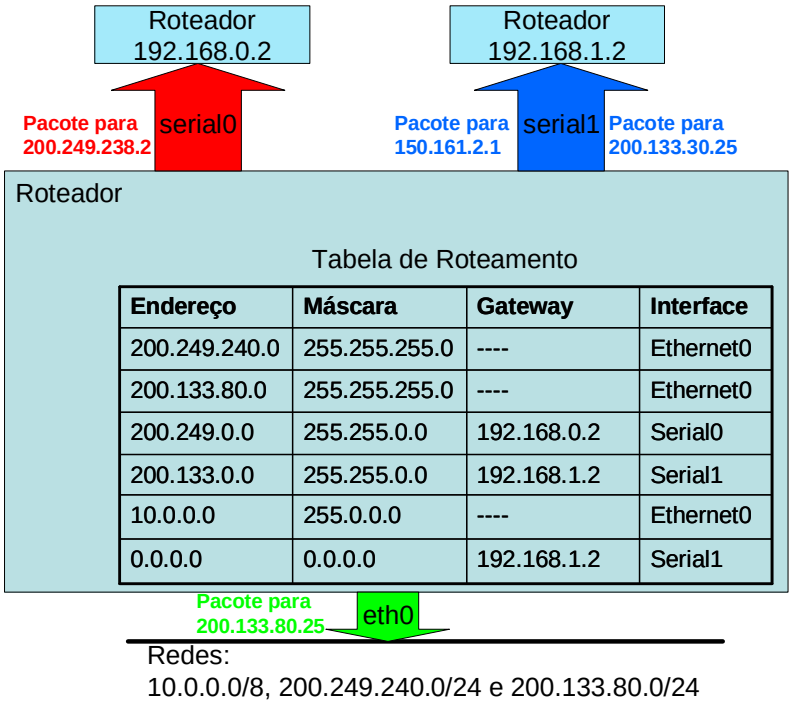


Figura 2.2: Roteamento Estático

a) O pacote destinado ao endereço 200.249.238.2, com máscara 255.255.255.0, retorna 200.249.238.0, portanto não casa nem com o padrão de endereço da primeira nem com o da segunda regra. O pacote é então mascarado com 255.255.0.0, retornando 200.249.0.0. Desta vez, o resultado é igual ao endereço da terceira regra e assim o pacote é encaminhado para o *gateway* de endereço 192.168.0.2, através da interface Serial0.

b) O pacote destinado ao endereço 200.133.80.25 é mascarado com 255.255.255.0 e retorna 200.133.80.0. Tal endereço é idêntico ao da segunda regra, o que leva o pacote a ser lançado em *broadcast* na interface Ethernet0.

c) O pacote destinado ao endereço 200.133.30.25 encontrará seu destino quando mascarado com 255.255.0.0, resultando no endereço 200.133.0.0. Desta forma, o pacote segue através do *gateway* 192.168.1.2 pela interface Serial1, segundo a quarta regra.

d) Finalmente o pacote endereçado a 150.161.2.1 não casa padrão com as cinco primeiras regras, chegando ao caso da rota *default*, que casa padrão com qualquer IP que apareça. Assim, o pacote será encaminhado através do *gateway* 192.168.1.2 pela interface Serial1.

Concluímos que o roteamento estático possui influência direta sobre o tráfego de saída da rede. Assim, ele pode ser usado para direcionar os pacotes que seguem para destinos, situados fora do nosso *site*, por uma das conexões. Para isso, basta criarmos uma rota para cada faixa de endereços IP (endereço de rede e máscara associada) que desejamos redirecionar, utilizando o endereço de *gateway* e a interface de acordo com a conexão desejada.

Se alguma das conexões ficar inoperante por algum motivo, é possível criar um mecanismo que, percebendo isto, altere de forma automática as rotas relacionadas ao mesmo. Desta forma, o tráfego que sairia por esta conexão seria distribuído entre as outras.

2.1.2 Roteamento por políticas (*Policy Routing*)

O roteamento normalmente baseia-se em métodos que utilizam o endereço de destino contido no pacote para escolher o caminho para onde enviá-lo. Já o Roteamento Baseado em Políticas (***Policy-Based Routing - PBR***) é um recurso existente em vários roteadores (por exemplo, certos modelos da CISCO com IOS a partir da versão 11, e roteadores baseados em Linux) que possibilita aos administradores criar políticas de roteamento para permitir ou negar caminhos baseando-se em fatores como:

- endereço do *host* de origem
- serviço (porta) utilizado
- protocolo (tcp/udp/icmp/etc)

Com esse tipo de roteamento, conseguimos alguns benefícios como:

- **Seleção da conexão de saída baseado no *host* de origem:** Permite que grupos de usuários/*hosts* do *site* acessem a Internet através de conexões distintas baseando-se no endereço de origem dos pacotes enviados.
- **Qualidade de Serviço (QoS):** Permite que o tráfego seja roteado por conexões diferentes baseando-se em seu tipo, ou seja, nos campos existentes no cabeçalho IP (tamanho do pacote, etc.).
- **Economia de custo:** Tráfego interativo e programado é distribuído entre conexões permanentes de baixo custo e baixa banda e conexões temporárias (permutáveis) de alta velocidade e alto custo. Conexões ISDN podem ser criadas quando necessárias e desfeitas quando o tráfego não for necessário.
- **Balanceamento de carga:** Políticas são implementadas para distribuir o tráfego entre múltiplas conexões à Internet baseando-se em características deste tráfego, por exemplo, o serviço utilizado.

Portanto, com *Policy-Based Routing*, podemos direcionar o tráfego de saída através das conexões de forma semelhante ao roteamento estático, mas utilizando outros critérios além do endereço IP de destino.

2.1.3 Serviço de Nomes (DNS)

O Serviço de Nomes, ou DNS (*Domain Name Service*), traduz nomes em endereços IP (numéricos) e vice-versa (números IP em nomes) e foi criado para facilitar o acesso aos *hosts* pelo usuário, pois nomes são mais intuitivos e fáceis de memorizar que endereços IP.

Quando *hosts* remotos tentam acessar um *host* interno pelo nome, a consulta é repassada para um servidor DNS com autoridade sobre o domínio. Domínio é a parte de um nome, que é gerenciada pelo *site*, por exemplo *cin.ufpe.br*. Quando o servidor de DNS tem autoridade, ele procura o nome do *host*, por exemplo *recife.cin.ufpe.br*, em uma tabela interna que relaciona nomes em endereços IP e responde com o endereço associado ao nome solicitado. Enfim, o *host* remoto acessa o *host* local utilizando tal IP.

Além dessa facilidade, já suficiente para justificar sua existência, o DNS ainda nos fornece outras vantagens. Uma delas é a de possibilitar a troca do endereço IP da máquina que executa algum serviço, alterando-se apenas o IP associado ao nome da máquina na tabela de DNS. Como exemplo podemos ilustrar o nome *www.cin.ufpe.br* associado ao IP 150.161.2.40. Se quisermos trocar o serviço para a máquina 150.161.2.41, basta trocar a entrada *www* na tabela de DNS para o novo IP. Outro recurso oferecido por alguns servidores de DNS, como o *djbdns* (http://cr.yp.to/djbdns.html), é a possibilidade de balancearmos o tráfego entre várias máquinas que respondem por um mesmo nome. Nesse caso, cabe ao servidor DNS a tarefa de distribuir o tráfego entre as máquinas. Um exemplo seria associarmos ao serviço *www* os dois endereços IP, por exemplo: 150.161.2.40 e 150.161.2.50, e o DNS escolhe qual endereço responder com o percentual desejado para cada um.

A figura 2.3 mostra uma situação onde o *host* que atende pelo nome *www.cin.ufpe.br* , pode ter seu tráfego com o *host externo* passando pela conexão ao ISP1 (2a) ou pela conexão ao ISP2 (2b), dependendo da resposta à consulta DNS feita anteriormente (1).

Como a maioria das aplicações utiliza nomes para acessar serviços em outros *hosts*, devido às vantagens listadas, podemos concluir que o serviço de DNS é responsável por definir a conexão por onde o tráfego de entrada, em conexões iniciadas remotamente, chegará, pois, dependendo do endereço IP respondido na consulta de nomes, o *host* estabelece a conexão utilizando o ISP relacionado a este endereço.

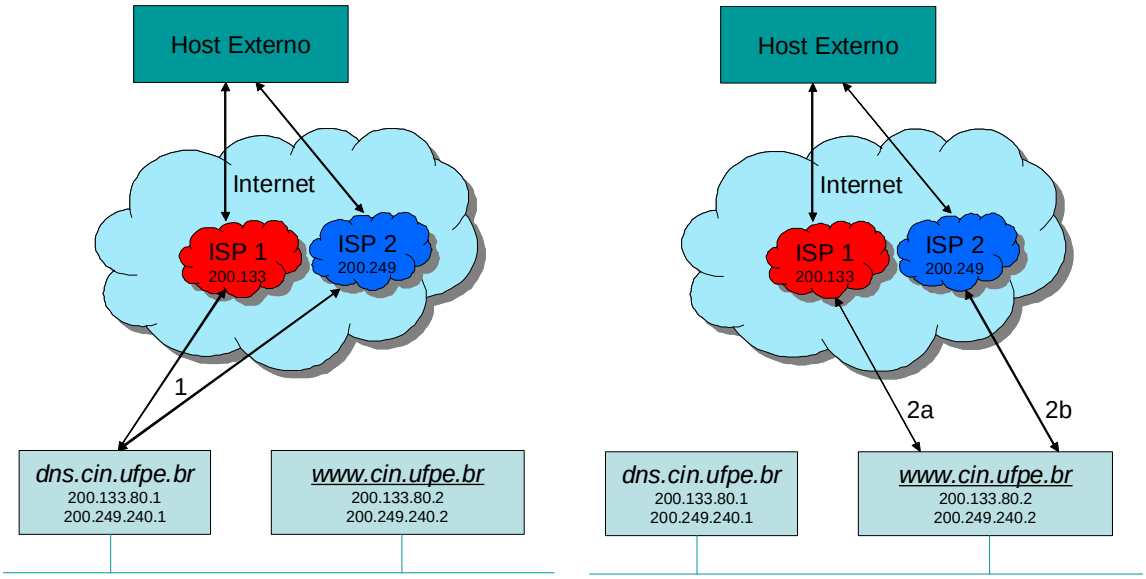


Figura 2.3: Serviço de Nomes (DNS)

Se alguma das conexões for desfeita por algum motivo, podemos redirecionar o tráfego de saída iniciado externamente para outras conexões de forma automática. Para isso, alteramos a tabela de DNS a fim de responder às consultas com IPs relativos aos provedores que estão conectados.

Outro recurso existente nos servidores *djbdns* e *bind* (http://www.isc.org/products/BIND/) a partir da versão 9, é a resolução baseada no endereço de origem. Isso permite que o servidor DNS responda às chamadas de forma diferente, baseada no endereço IP do *host* que fez a consulta. Assim, pode-se escolher a partir do endereço do *host*, qual conexão desejamos utilizar.

Para cada conexão à Internet que o *site* possui, podemos instalar um servidor DNS. Desta forma, é possível responder de forma diferente, também baseada na conexão pela qual chegou a requisição. Se isso não for necessário, temos a opção de instalar apenas um servidor DNS e configurá-lo de forma a escutar as consultas que chegam a partir de qualquer uma das conexões existentes.

2.1.4 Tradução de Endereços (NAT)

Tradução de endereços (*Network Address Translation* – NAT) é um serviço fornecido por roteadores que traduz endereços IP privados (reservados) em endereços reais (válidos na Internet), quando os pacotes saem do *site*, e inversamente, quando os pacotes chegam aos *hosts* do *site*. Esse recurso, além de influenciar o caminho pelo qual o tráfego flui, facilita bastante a administração dos *sites*, principalmente quando se trata de *sites multi-homed*, conforme veremos a seguir.

A IANA, que provê as faixas de IP para cada provedor, reserva as seguintes faixas de endereços IP para redes privadas [18]: 10.0.0.0/8, 172.16.0.0/12 e 192.168.0.0/16. Cabe ao administrador escolher quais faixas, ou sub-faixas destas, ele deseja utilizar em sua rede interna.

Porém, para os *hosts* locais acessarem e serem acessados por *hosts* na Internet, eles precisam de um IP válido na Internet. O roteador, utilizando o serviço de NAT, faz a tradução do endereço privado do *host* para um verdadeiro na Internet, utilizando uma tabela interna que associa os dois endereços. Quando os *hosts* externos acessam os internos, o roteador também faz a tradução para o endereço da rede privada do *host*, fazendo com que os pacotes cheguem aos seus destinos dentro da rede.

Na figura 2.4, o *host* externo se comunica com o interno (*www.cin.ufpe.br*), utilizando o IP real 200.133.82.2. O roteador, através do serviço de NAT, substitui os endereços destino dos pacotes recebidos pelo endereço privado 192.168.0.2 e faz o inverso para os pacotes enviados pelo *host* interno ao externo.

Desta forma, o NAT facilita a administração das redes, pois, havendo qualquer mudança do *backbone*, ou apenas da faixa IP utilizada, basta alterar-se a tabela de tradução no roteador, ao invés de reconfigurar cada *host* alterando-se seus endereços.

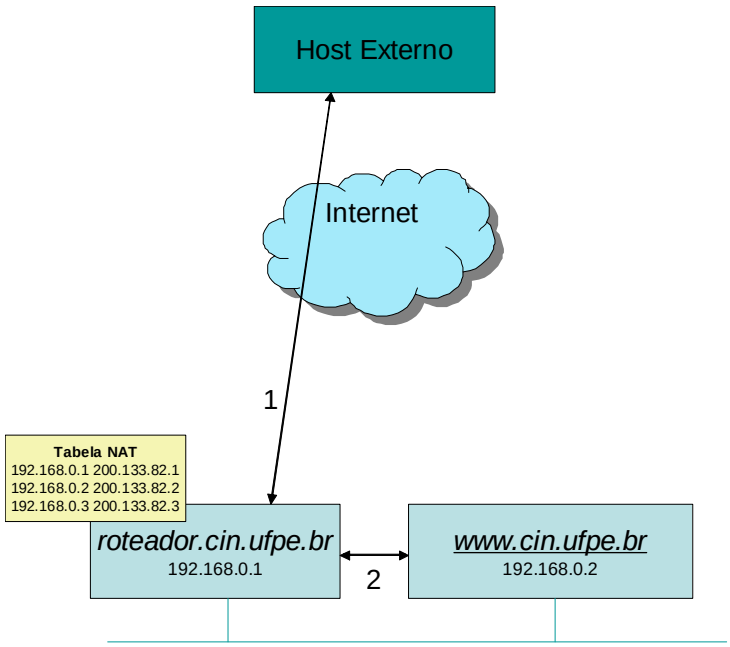


Figura 2.4: Tradução de Endereços (NAT)

No caso do *site* possuir mais de uma conexão, o NAT evita que cada *host* da rede tenha que ser configurado com vários endereços IP, um para cada conexão que se deseje utilizar. Neste caso, o *host* é configurado apenas com um endereço privado, ficando a cargo do roteador substituir tal endereço por um válido, adequado na ocasião, determinando a conexão a ser utilizada.

O NAT, quando utilizado, influi diretamente no tráfego de entrada do *site*. Isto ocorre porque os *hosts* externos respondem aos pacotes recebidos, utilizando os endereços de origem contidos nestes pacotes. Assim, dependendo do endereço traduzido pela regra de NAT, o fluxo estabelecido utilizará a conexão associada a tal endereço de resposta.

Na figura 2.5, o tráfego é estabelecido utilizando a conexão com o ISP 1 (a) ou a conexão com o ISP 2 (b), dependendo do endereço traduzido pela tabela de NAT do roteador.

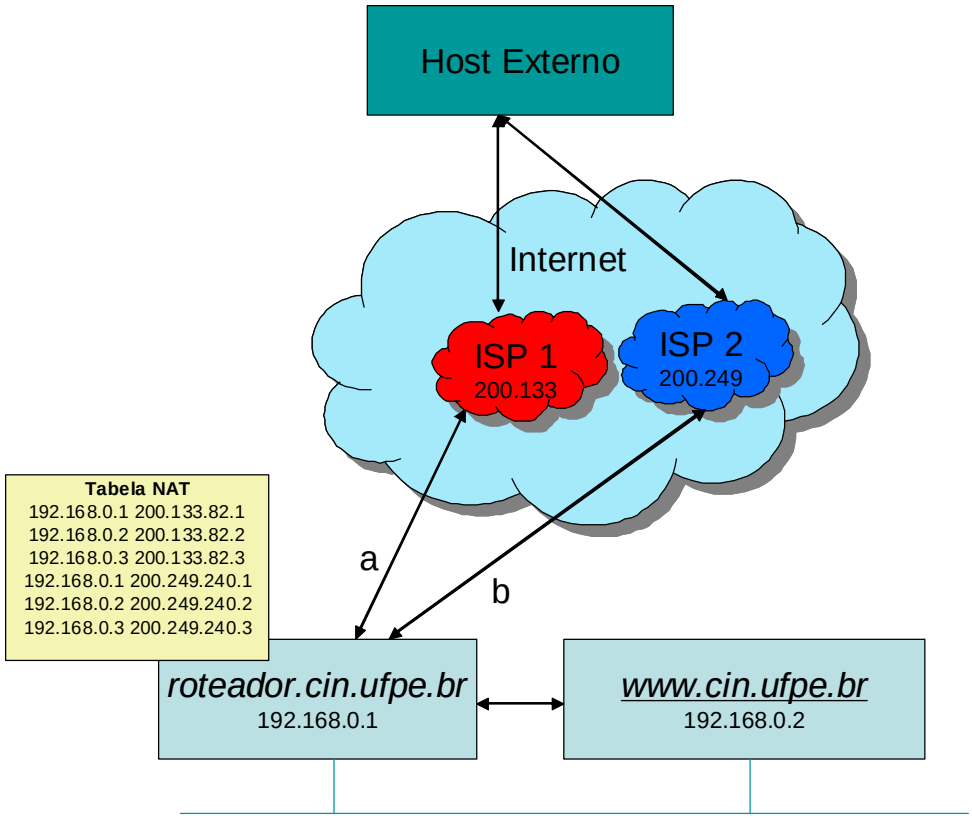


Figura 2.5: Tradução de Endereços (NAT) em *sites multi-homed*

2.1.5 Combinação de Mecanismos

Como vimos nas seções anteriores, existem diversos mecanismos que influenciam diretamente a rota dos pacotes IP que chegam e saem de uma rede. Mostraremos a seguir como combinar esses mecanismos de forma a se obter determinada divisão do tráfego entre as conexões.

A tabela 2.1 resume os principais mecanismos, mostrando suas influências no tráfego da rede, ressaltando dois aspectos: o sentido do tráfego influenciado, e a influência exercida no início do tráfego, quando uma das conexões é escolhida.

Mecanismo	**Sentido do tráfego influenciado diretamente pelo mecanismo**	**Influência exercida na escolha do *link* a ser utilizado**
Roteamento	Tráfego de saída	Tráfego iniciado por *hosts* internos utiliza tal mecanismo para escolher a conexão de saída.
DNS	Tráfego de entrada	Tráfego iniciado por *hosts* remotos utiliza tal mecanismo para escolher a conexão de entrada.
NAT	Tráfego de entrada	Tráfego iniciado por *hosts* internos e remotos utiliza tal mecanismo para escolher a conexão de entrada.

Tabela 2.1: Mecanismos que influenciam o tráfego numa rede TCP/IP

Roteamento + NAT

Para garantir que o tráfego iniciado por *hosts* internos tenha um comportamento regular, devemos garantir que os pacotes saiam e cheguem através do mesmo canal. Se isto não acontece, comportamentos indesejados podem ocorrer conforme mostraremos.

Quando o tráfego é iniciado por *hosts* internos, o roteamento é o fator de escolha da conexão de saída, conforme estudamos anteriormente. Se a rede utiliza NAT, o roteador utiliza determinada regra de tradução para definir o endereço de origem. É para este endereço que os pacotes serão enviados de volta, induzindo então o tráfego de entrada por tal conexão.

Quando os *hosts* não utilizam de forma sincronizada o roteamento e o NAT, os pacotes podem sair por uma conexão e chegar por outra. Isso não é desejado por vários fatores. Em primeiro lugar, isso exige que as duas conexões aos provedores estejam funcionando. Se ocorrer algum problema com uma delas, o caminho dos pacotes é interrompido. Em segundo lugar, existem *backbones*, como é o caso da Embratel, que

filtram os pacotes, descartando os que utilizam um endereço de origem diferente dos fornecidos por eles. Desta forma, a conexão não seria estabelecida, pois os pacotes seriam perdidos.

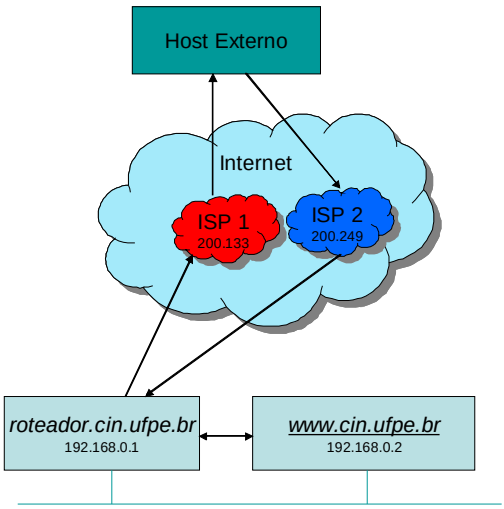


Figura 2.6: Rotas assimétricas na combinação da Tradução de Endereços (NAT) com o Roteamento Estático

Na figura 2.6, o roteador utiliza o ISP 1 para enviar os pacotes, de acordo com sua tabela de roteamento, e o ISP 2 para receber os pacotes, de acordo com sua tabela de NAT. Tal comportamento é indesejado pelos motivos explicados.

DNS + Roteamento

Quando o tráfego é iniciado por *hosts* externos, geralmente um novo elemento aparece neste novo cenário: o DNS. Isto ocorre, porque nomes são bastante utilizados para acessar os *hosts* em lugar do endereço IP.

Assim, quando uma consulta DNS chega ao *host*, é baseado no endereço IP de resposta que a conexão por onde o tráfego de entrada será estabelecido é escolhido. Os pacotes são respondidos e seguem ao *host* externo pela conexão escolhida pelo roteamento.

Pelos mesmos motivos indesejáveis estudados na seção anterior, que ocorrem

quando o tráfego de entrada e saída seguem por conexões distintas, devemos sincronizar o roteamento com o DNS, para que os IPs utilizados sejam sempre os mesmos para as mesmas faixas de endereços IP externos. Assim garantimos que os tráfegos de entrada e saída utilizarão sempre a mesma conexão.

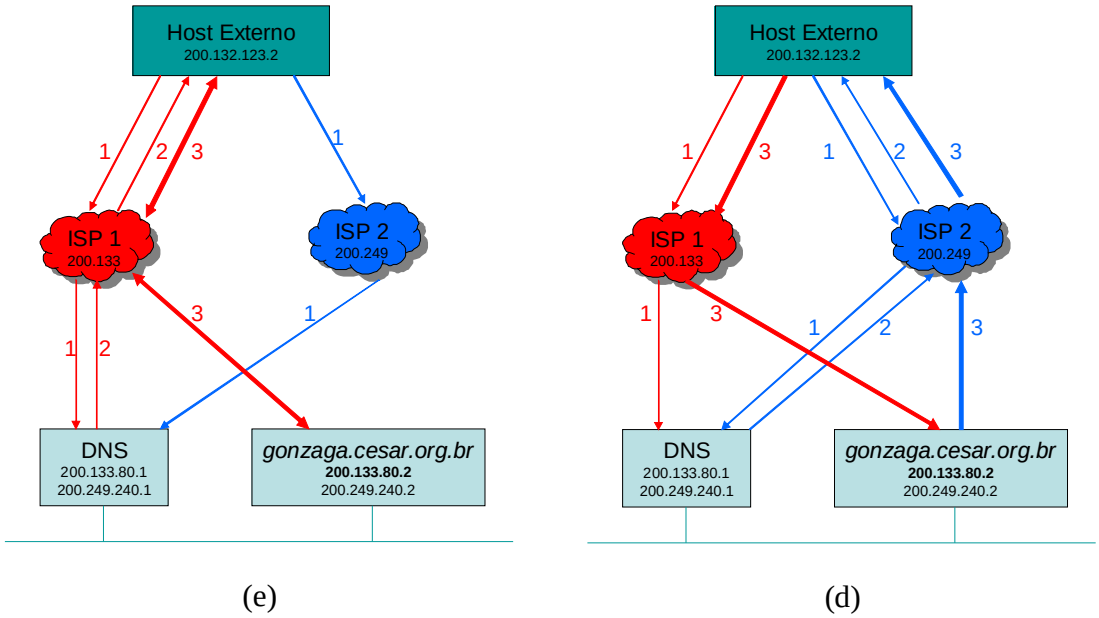


Figura 2.7: Combinação de Roteamento com Serviço de Nomes (DNS)

Na figura 2.7, ilustramos dois casos de uma mesma consulta de nome. Na figura à esquerda (*e*), a rota para o *site* externo é através do ISP 1 e na figura à direita (*d*), através do ISP 2. Em ambos os cenários, o IP respondido é o 200.133.80.2 e as rotas no *site* externo são através do ISP 1 para endereços IP 200.133.80.x e do ISP 2 para endereços 200.249.240.x.

Ocorre então, no passo 1, a requisição do endereço para o nome *gonzaga.cesar.org.br*. Esta requisição pode chegar por qualquer uma das conexões e será respondida com o IP 200.133.80.2 no passo 2. Na figura *e*, o pacote é roteado pelo ISP1 e na figura *d*, o caminho via o ISP 2 é escolhido. No passo 3, o tráfego é estabelecido entre os *hosts*. Na figura *e*, os pacotes chegam pelo ISP 1 devido ao roteamento externo para o IP

respondido e também saem pela mesma conexão, devido ao roteamento interno. Já na figura *d*, os pacotes chegam pelo ISP 1, mas saem pelo ISP2.

Podemos gerenciar as conexões por onde o tráfego de entrada chegará através de duas maneiras: baseando-nos na conexão em que a consulta de DNS chegou, ou baseando-nos no endereço do *host* que fez tal consulta. Podemos implementar a primeira solução utilizando qualquer servidor de DNS, porém a segunda só é possível utilizando-se o servidor BIND a partir de sua versão 9, ou o servidor djbdns.

Quando nos baseamos na conexão por onde chega a consulta, a idéia é colocar vários servidores de DNS, cada um respondendo na porta 53/udp do endereço IP associado ao ISP correspondente. Assim, quando a consulta chega, o respectivo servidor de DNS responde com o IP desejado e a partir daí a conexão é estabelecida pela conexão relacionado a este IP. A figura 2.8 ilustra um exemplo deste caso onde cada servidor DNS responde com o IP relacionado ao ISP ao qual ele responde as consultas, conduzindo o tráfego através dele.

Quando *hosts* remotos perguntam qual é o IP de *ioram.cesar.org.br*, a consulta chega a um dos servidores de DNS do *site*, que tem autoridade sobre o domínio *cesar.org.br*. Dependendo por qual servidor esta consulta chegue, o endereço IP é devolvido de forma diferente. Isso garante que a conexão seja estabelecida por este *site*.

Essa arquitetura garante que, se uma das conexões tiver problema, nada precisa ser feito, já que os pacotes deixarão de chegar apenas ao servidor de DNS relativo a esta conexão.

Porém, para termos um controle maior sobre por onde será conduzido o tráfego, de forma a controlarmos o balanceamento de carga entre as conexões, podemos utilizar um recurso disponível na versão 9 do BIND e no djbdns. Esse recurso permite a um mesmo servidor de DNS responder as consultas diferentemente dependendo de regras, como, por exemplo, baseado no endereço do *host* que originou a consulta. Desta forma, podemos direcionar faixas de IP pelas conexões da forma que desejarmos, não ficando mais dependentes do balanceamento proveniente da Internet.

A figura 2.9 mostra um exemplo de arquitetura onde um único servidor de DNS está conectado através das 4 conexões à Internet, recebendo conexões pelos seus 4 endereços IP

(um relativo a cada provedor) nas respectivas portas 53. Independente da conexão por onde chegue a requisição, o servidor verificará qual endereço IP e responderá baseado no endereço do *host* que fez a consulta. Podem existir regras *default* para os endereços IP que não casem com regra alguma.

Dessa forma, conseguimos balancear como desejarmos o tráfego de entrada da nossa rede. Combinando este método com o anterior (roteamento estático), temos como balancear os tráfegos de entrada e de saída do nosso *site*.

Na figura 2.8, o host a realiza uma consulta DNS que chega ao site cesar.org.br pelo ISP3. Da mesma forma, os hosts b, c e d realizam as consultas, mas desta vez as consultas chegam pelos ISP 1, 4 e 2 respectivamente. Assim, os servidores de DNS 3, 1, 4 e 2 recebem as perguntas e respondem com os endereços IP que lhes forem convenientes. No nosso exemplo, cada servidor responde com o endereço relativo ao próprio ISP. Desta forma, o tráfego gerado para cada host é conduzido pelo ISP por onde chegou a consulta, conforme vemos na figura 2.9.

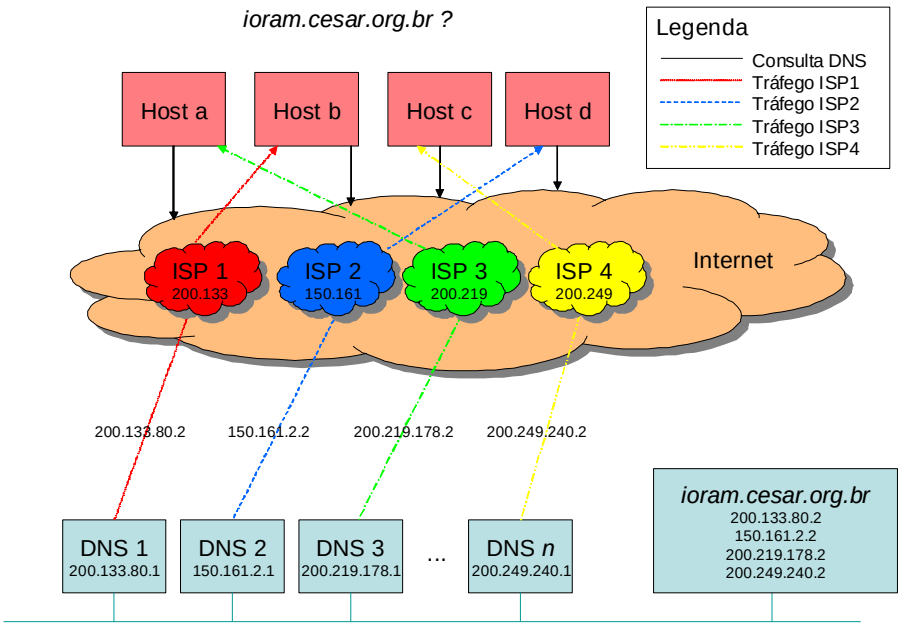


Figura 2.8: Serviço de Nomes e Roteamento com vários servidores de DNS: consulta.

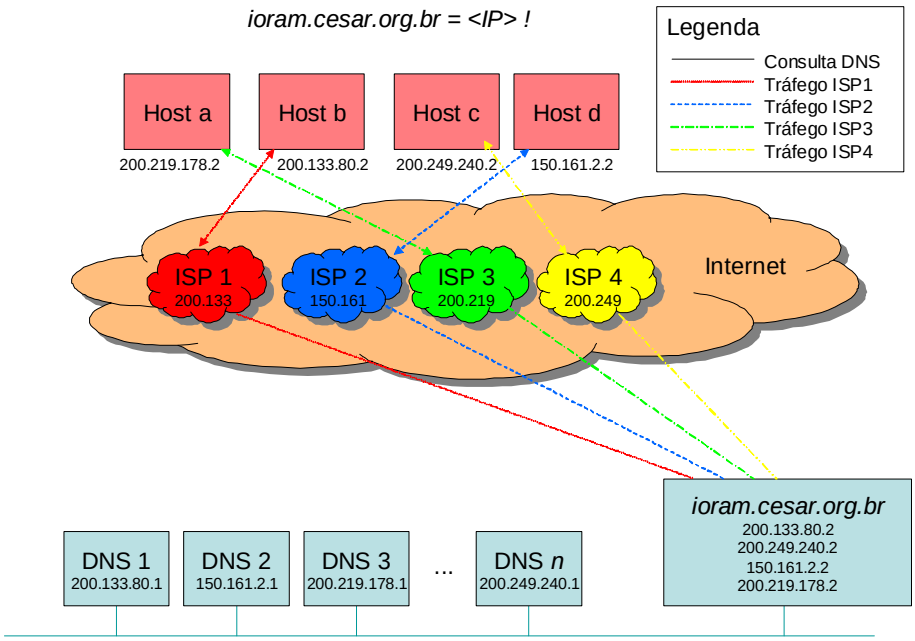


Figura 2.9: Serviço de Nomes e Roteamento com vários servidores de DNS: resposta.

Já na Figura 2.10, todas as requisições de consulta DNS chegam a um mesmo servidor DNS, que atende às chamadas de todos os ISPs. Tal servidor tem uma tabela de regras, e casa padrão do endereço IP do host que origina a consulta. Dependendo da regra utilizada, um IP relativo a um determinado ISP é devolvido. O tráfego então é conduzido pelo ISP escolhido pelo servidor de DNS como mostra a figura 2.11.

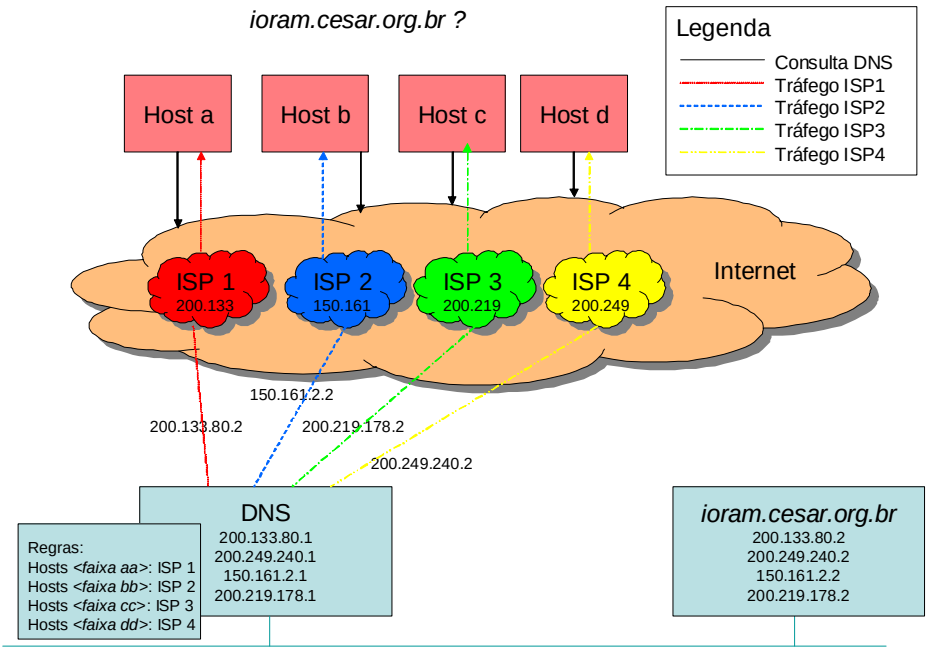


Figura 2.10: Serviço de Nomes e Roteamento com único servidor de DNS: consulta.

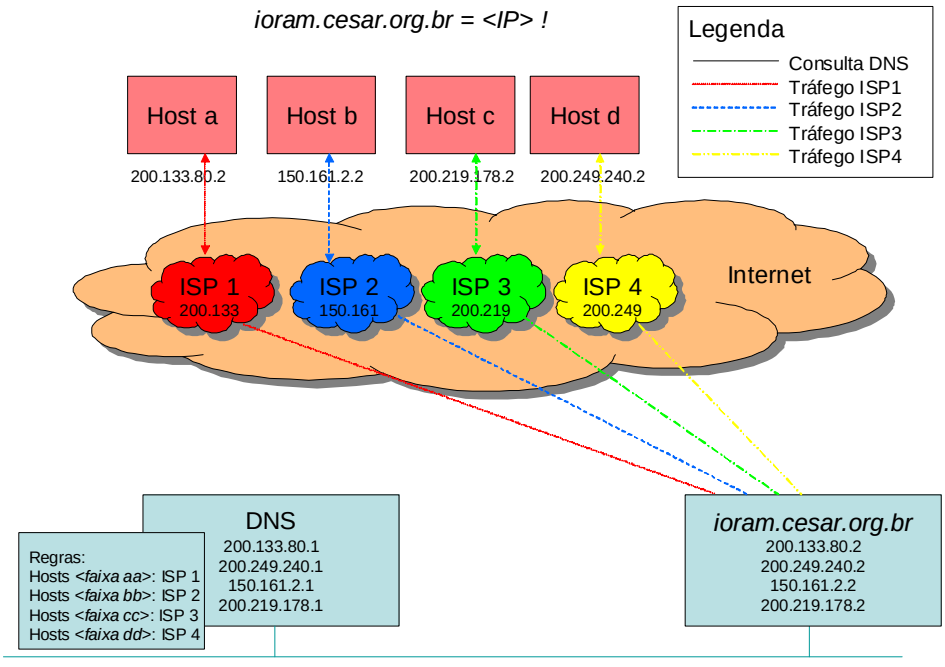


Figura 2.11: Serviço de Nomes e Roteamento com único servidor de DNS: resposta.

2.2 Sistemas Autônomos x Sistemas Não-Autônomos

Para utilizar sempre a melhor conexão ao cliente, a empresa precisa gerenciar dinamicamente suas rotas, criando sempre a mais adequada em cada situação. Para isso, criou-se na Internet o conceito de Sistemas Autônomos (*Autonomous System* - AS). Os Sistemas Autônomos utilizam um protocolo chamado BGP (*Border Gateway Protocol*) nos roteadores que conectam o *site* aos seus provedores de acesso. Esse protocolo permite que os roteadores troquem informações sobre a proximidade com os demais *sites* e gerem sempre a melhor rota aos mesmos.

Porém, quando se criou o conceito de Sistemas Autônomos, não se dimensionou a quantidade de empresas que se tornariam AS. Assim, com um grande número de *sites* que necessitam deste recurso, o tráfego gerado entre os roteadores, utilizando-se o protocolo BGP torna-se de tal ordem, que chega a ocupar parte significativa da banda, tornando a conexão mais lenta. Por esta razão, os comitês responsáveis por cadastrar os Sistemas Autônomos dificultam tal cadastramento, fazendo-o apenas no caso de grandes *sites*. No Brasil, a RNP (Rede Nacional de Pesquisa) é responsável pelo cadastro de Sistemas Autônomos, através da FAPESP (Fundação de Amparo a Pesquisa do Estado de São Paulo). Para efetuar o cadastro de AS, o *site* precisa justificar a necessidade de mais de 4000 endereços IP distintos. Os únicos *sites* que atingem este patamar são os grandes provedores de acesso.

Assim, pequenos e médios *sites* conseguem conectar-se a mais de um provedor, mas têm dificuldades em gerenciar tais conexões da melhor maneira. Podemos classificar os Sites Multi-Homed em relação ao gerenciamento de suas conexões em: *Sistemas Autônomos* e *Sistemas Não-Autônomos*.

2.2.1 Sistemas Autônomos

Sistemas Autônomos, ou AS (*Autonomous System*), existem para gerenciar o roteamento na Internet. Cada AS possui um número inteiro de 16 bits, único globalmente, chamado ASN (*Autonomous System Number*). O ASN é designado pela InterNIC (no Brasil, pela Fapesp – Registro.br). De posse desse número, os roteadores das empresas passam a ter capacidade de trocar informações sobre a localização dos números IP na

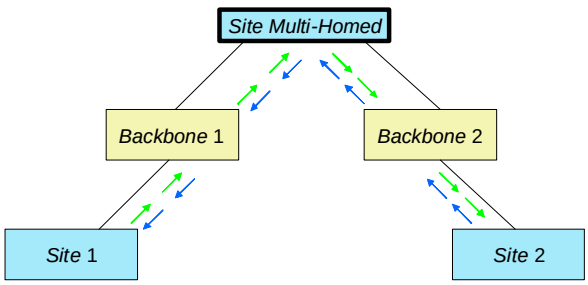
Internet, isto é, a distância de cada roteador a grupos de números IP. Essa troca de informações entre os roteadores é feita no protocolo BGP (*Border Gateway Protocol*) e a partir dela são definidas as melhores rotas.

Os seguintes tipos de AS existem na Internet:

- **AS *Stub*** é conectado apenas a um outro AS. Para fins de roteamento, ele poderia ser considerado como uma simples extensão do outro AS. De fato, a maioria das redes com uma única conexão à Internet não tem números AS associados e seus endereços de rede são tratados como parte do AS pai.
- **AS de Trânsito** tem conexões a mais de um AS, e permite ser usado como conduite de tráfego (em trânsito) entre outros AS. Muitos ISP grandes são AS de Trânsito.
- **AS *Multi-Homed*** tem conexões a mais de um AS, mas não permite a passagem de tráfego em trânsito, pois seus *hosts* interiores podem rotear o tráfego através de múltiplos AS. Essa é a configuração típica para uma grande rede corporativa, com múltiplas conexões redundantes, mas que não desejam passar tráfego de outras redes.

A diferença entre ASs de Trânsito e *Multi-Homed* é sutil. Nos de trânsito, a conexão do *site* a mais de um *backbone* pode ser utilizada para beneficiar outros *sites*, e em contrapartida, degradar a performance das conexões. Já nos *Multi-Homed*, isso não é possível. A figura a seguir ilustra os dois cenários. No primeiro, o *site* 1 se comunica com o *site* 2 utilizando as conexões do *site multi-homed*. No segundo, o *site* 1 *e o site* 2 conseguem apenas se comunicar com o *site multi-homed* utilizando tais conexões.

AS de Trânsito



AS Multi-Homed

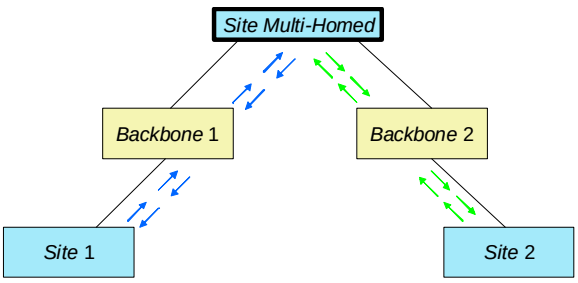


Figura 2.12: Sistemas Autônomos de Trânsito e *Multi-Homed*

Os ASs podem também, caso uma conexão até um provedor seja interrompida, utilizar os endereços IP relativos a este provedor pela conexão de algum outro provedor que conectado ao *site*. A RFC 2260 [04] explica com detalhes como funciona este mecanismo.

Devido às restrições impostas pela tecnologia atual de roteamento (aumento das tabelas de roteamento na chamada "*default-free zone*" da Internet), somente são alocados números AS no Brasil a entidades que comprovam uso mínimo de prefixo /20, isto é, possuam cerca de 4096 *hosts*.

Além da dificuldade de um *site* se tornar um AS, Sistemas Autônomos não satisfazem todas as necessidades de gerenciamento de *sites multi-homed*. Eles são bons para se escolher a rota mais curta até um determinado *site*. Entretanto deixa a desejar quando desejamos estabelecer políticas para o uso das conexões, por exemplo, dividir a carga entre as conexões, obedecendo critérios como:

- Estabelecer um percentual de uso para cada conexão (ex.: 70% da carga através do ISP A e 30% através do ISP B)
- Definir serviços aos quais deseja-se usar por determinada conexão (ex.: O uso dos protocolos http, smtp e ftp seguem pelo ISP A, e os demais pelo ISP B)

2.2.2 Sistemas Não-Autônomos

Sem possuir um endereço AS, os roteadores do *site Multi-Homed* não tem informações de qual *backbone* mais adequado para enviar os pacotes de dados a um determinado destino. Aos administradores desses sistemas resta então criar rotas estáticas para alguns grupos de endereços já conhecidos por eles e criar uma rota *default* através de um dos ISP conectados. Outra solução paliativa é criar rotas baseando-se no tipo de serviço utilizado na rede, direcionando o tráfego de pacotes http, por exemplo, sempre por determinado ISP, o de e-mail por outro, etc. Esse estilo de roteamento baseado em políticas é chamado "*policy routing*".

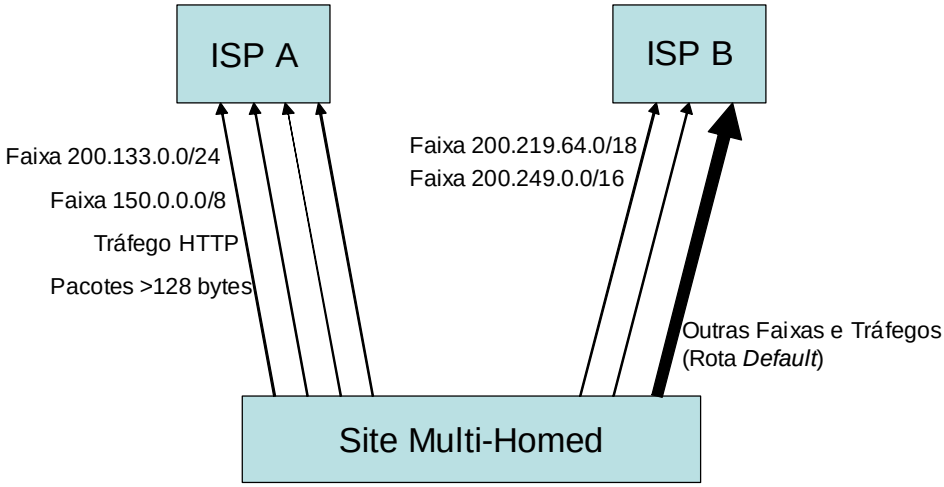


Figura 2.13: Sistemas Não Autônomos

Existe ainda uma forma de balancear o tráfego entre várias conexões a um mesmo provedor utilizando várias rotas *default*. Esse tipo básico de balanceamento é suportado por alguns roteadores. Em particular, nos roteadores Cisco, é possível criar várias rotas *default* e ele distribui os pacotes por igual entre essas rotas.

Porém desta forma, ficamos sem poder controlar por qual conexão sairiam os pacotes e, além disso, a saída é dividida por igual entre as *n* conexões, fato que nem sempre é desejado. Esta solução exige também a configuração nos equipamentos do provedor.

2.3 Soluções Existentes

Algumas soluções foram definidas para configuração de roteadores utilizando BGP para balanceamento de tráfego em *Sites Multi-Homed*. Porém, como vimos nas sessões anteriores, o uso do protocolo BGP implica no *site* possuir um ASN, que possibilite com que ele converse com os roteadores vizinhos.

2.3.1 Solução proposta pela RFC2260

A RFC2260 [04] descreve estratégias de endereçamento e roteamento para empresas conectadas a múltiplos ISP, com a intenção de reduzir o excesso de tráfego de roteamento para essas empresas no roteamento global da Internet.

O diagrama a seguir nos mostra a configuração típica de um *site multi-homed*. Neste caso, considera-se que os roteadores trocam informações no protocolo BGP, tratando-se, então, de um Sistema Autônomo.

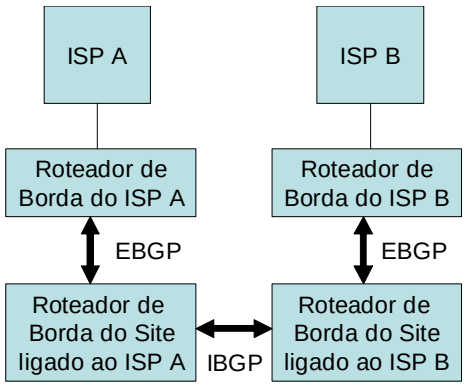


Figura 2.14: Configuração de *Site Multi-Homed* proposta pela RFC 2260.

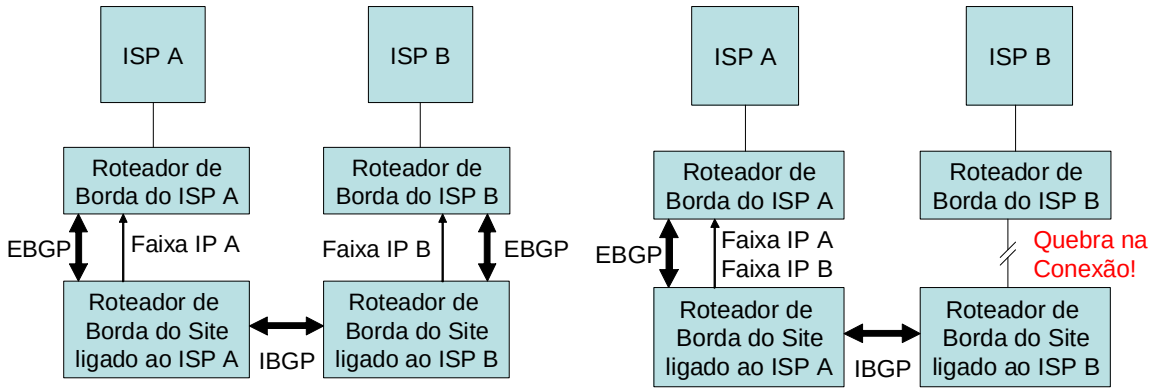
A solução descrita pela RFC 2260 se baseia nas informações enviadas pelo roteador de borda da empresa ao roteador de borda do provedor sobre o como alcançar as faixas de endereço IP da empresa.

Quando todas as conexões da empresa estão estabelecidas, os roteadores de borda do *site* avisam aos roteadores do ISP que o *site* é acessível apenas pela faixa de endereço IP alocada pelo ISP em questão. Estas regras já estão agregadas ao roteamento dos ISPs e não são enviadas à "*default-free zone*" da Internet.

Quando ocorre um problema em uma das conexões da empresa, os roteadores de borda da empresa que estão conectados em conexões que estão funcionando informam aos roteadores do ISP que conseguem alcançar a faixa de endereços associada aos ISPs com problema. Isto acarretará na inclusão de novas rotas na "*default-free zone*" da Internet.

A solução acima se baseia que os roteadores são inteligentes o suficiente para determinar quando a conexão do outro roteador caiu e qual a faixa de endereços a ser divulgada caso isso aconteça. Essas duas premissas são resolvidas utilizando mecanismos fornecidos pelo BGP. Os roteadores da empresa recebem informações dos roteadores do

ISP diretamente conectados a eles, através de EBGP (External BGP), sobre quais as faixas de endereços IP acessíveis através do ISP. Os roteadores da empresa também trocam informações entre si, através de IBGP (Internal BGP), sobre quais faixas de endereços IP são acessíveis a partir deles. Se a interseção dos dois conjuntos de faixas recebidos por EBGP e IBGP for vazia, o roteador de borda da empresa começa a avisar ao roteador do ISP diretamente conectado que as faixas designadas pelos outros ISPs são acessíveis a partir dele. Este roteador recebe as faixas dos outros ISPs via IBGP pelos outros roteadores de borda da empresa. Esta abordagem é chamada de "injeção automática de rotas".



(a) Interseção não vazia **(b)** Interseção vazia

Figura 2.15: Primeira solução proposta pela RFC 2260

A soluções fornecidas por essa RFC utiliza o protocolo BGP: IBGP (Internal BGP) entre os roteadores de borda do *site* internamente e EBGP (External BGP) entre os roteadores de borda do *site* e os roteadores de borda do ISP. O protocolo BGP é usado para troca de informações entre os roteadores, pressupondo que a empresa possui um número AS.

Uma alternativa para melhorar o mecanismo descrito acima é também proposto na RFC 2260. Para isso, cada roteador de borda da empresa conversaria através de EBGP, não apenas com o roteador do ISP diretamente ligado a ele, mas também com os outros roteadores de borda dos outros ISPs. Assim, eles dariam sempre preferência às rotas recebidas pelo seu próprio ISP.

Enquanto as conexões estiverem estabelecidas, a empresa é acessada pelas conexões utilizando suas respectivas faixas. Porém, quando um das conexões cai, o roteador do ISP ao qual o *link* caiu envia os pacotes encapsulados ao roteador de borda que ainda tem conexão à Internet. A figura a seguir ilustra este novo mecanismo.

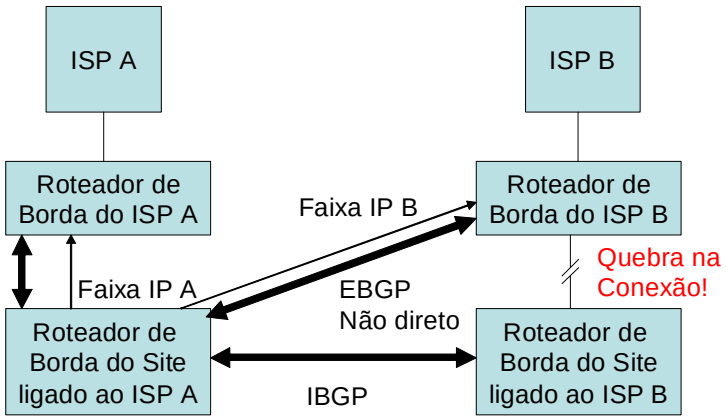


Figura 2.16: Segunda solução proposta pela RFC 2260.

Este novo mecanismo evita a inserção de rotas na "*default-free zone*" da Internet. Essas duas abordagens ainda podem ser associadas para uma melhor detecção do momento em que uma das conexões tem problema.

2.3.2 Multihoming com NAT - Cisco

As soluções propostas no white paper "Enabling Enterprise Multihoming with CISCO IOS Network Address Translation (NAT)" [02] lançado pela CISCO são baseadas nas soluções propostas pela RFC2260. Porém, a grande novidade é a utilização de NAT para evitar que a rede interna da empresa sofra impactos quando se deseja trocar o provedor (ISP), ou até quando se for instalar a solução pela primeira vez.

A figura a seguir ilustra o mesmo cenário da RFC 2260, mas enfatizando o uso de NAT. Os ISP A e B possuem faixas de endereço roteáveis globalmente na Internet (OG). Os roteadores de borda, do ISP e da empresa, utilizam faixas de endereço reservadas para se comunicarem entre si (OL). Nos roteadores de borda da empresa, onde está programado

o NAT, estão configuradas as faixas da empresa que são roteáveis na Internet (IG). Finalmente, dentro da empresa, utiliza-se uma faixa reservada (IL), que não precisará ser alterada na troca do pool de endereços fornecidos pelos provedores.

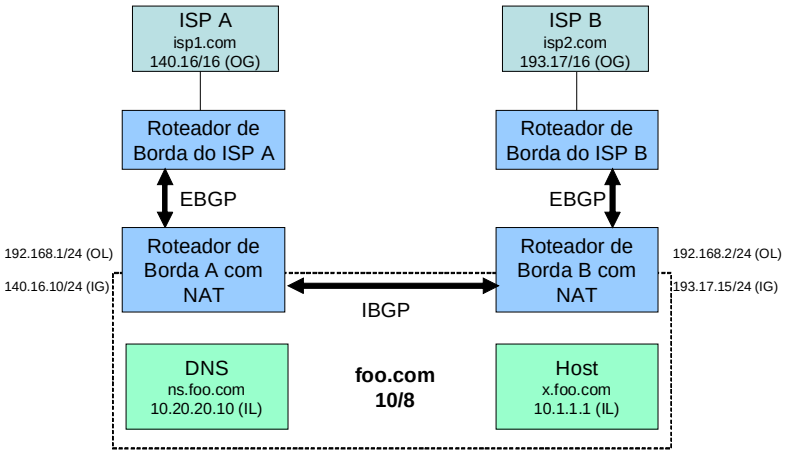


Figura 2.17: Solução proposta pela CISCO, baseada na solução da RFC 2260.

Os roteadores de borda da empresa fazem a troca dos endereços internos privados (IL) da empresa por endereços globalmente roteáveis na Internet (IG) e vice-versa quando as máquinas da rede interna acessam *hosts* na Internet.

2.3.3 Soluções Ad-hoc

Não existe solução que otimize o uso de múltiplas conexões à Internet sem o uso do protocolo BGP. As empresas que não justificam ser AS e possuem mais de uma conexão partem para soluções *Ad-Hoc* para conseguirem gerenciar de alguma forma essas conexões.

Como exemplo, explicaremos a solução usada pelo CIn/UFPE e CESAR, que possuem dois provedores de acesso: a RNP (Rede Nacional de Pesquisa) e a Embratel (Empresa Brasileira de Telecomunicações), e baseia-se em rotas estáticas, utilizando *policy routing* para direcionar os pacotes por um ou outro provedor e na resposta diferenciada do DNS.

A solução utiliza os mecanismos que estudamos na seção 2.1 para alterar o fluxo de

pacotes trocados com *sites* pertencentes a faixas de endereços IP que se conhece previamente que estão mais próximos ao *backbone* da RNP, utilizando para os demais *sites* o *backbone* da Embratel.

Esta solução está longe de ser a desejada já que é necessário um conhecimento prévio dessas faixas. Assim, se uma nova faixa passar a ser usada pela RNP, deverá ser acrescentada manualmente nas tabelas de DNS, roteamento e NAT pelo administrador de sistemas. Também não conseguimos dividir o tráfego entre as conexões de maneira proporcional com esta solução.

Capítulo 3

Análise de Tráfego de uma rede IP: Um Estudo de Caso

No capítulo anterior, discutimos os mecanismos que podem ser utilizados para dividir o tráfego de uma rede IP entre *n* conexões. Agora, precisamos alimentar estes mecanismos com dados que informem como efetuar tal divisão. Para isso, buscaremos informações históricas sobre o tráfego na rede para efetuar o balanceamento de carga.

3.1 Elementos da Análise de Tráfego de uma rede IP

Para analisarmos o comportamento do tráfego de uma rede IP, observamos os elementos existentes nos pacotes que compõem este tráfego. Tais elementos podem ser: o endereço IP de origem, o endereço IP de destino, o protocolo utilizado (TCP ou UDP), o tamanho do pacote, o serviço utilizado (observando-se a porta TCP ou UDP), entre outros.

Com o tráfego classificado por esses elementos, o passo seguinte é ordená-lo de acordo com critérios que possibilitem a divisão do tráfego entre as conexões. Estes critérios podem ser: a quantidade de bytes trafegada, a quantidade de pacotes trafegada ou o tempo de resposta e a distância (em *hops*) para cada conexão, quando o elemento utilizado é o endereço IP do *host* externo ao *site* analisado.

Para utilizar a quantidade de bytes ou pacotes como critério, é importante isolar o tráfego de entrada do tráfego de saída. O motivo é simples: as conexões utilizadas numa rede IP são geralmente *full-duplex*, ou seja, suportam simultaneamente tráfego em ambos os sentidos, sem que um interfira no outro. Então, temos que considerar o tráfego de entrada e de saída isoladamente, já que os mesmos não concorrem pela mesma banda.

Para nossas análises, devemos escolher qual dos dois tipos de tráfego é o mais relevante. Num *site* que tem como objetivo disponibilizar serviços, como servidores de *web* ou outros serviços, o tráfego de saída deve ser a mais relevante. Num *site* cujo objetivo é fornecer acesso à Internet a usuários internos, como em Universidades, escolas, etc., o tráfego de entrada é que seria o principal fator de decisão.

Os mecanismos de controle de tráfego estudados utilizam entradas que podem ser alimentadas com dados adquiridos em análises, segundo mostra a tabela a seguir:

Mecanismo	Entrada	Dados Históricos Relevantes
Roteamento Estático	Tabela de Roteamento (faixa de IP destino e Interface/Gateway)	Faixas de endereços IP dos *hosts* remotos que interagiram com o *site*
Roteamento por Políticas	Tabela de Roteamento (serviço/faixa de IP origem/tamanho do pacote/etc. e Interface/Gateway)	Faixas de endereços IP dos *hosts* locais, serviços utilizados, etc.
Serviço de Nomes (DNS)	Faixa de endereços IP dos *hosts* remotos e tabela de nomes e endereços IP locais que se deseja utilizar	Faixas de endereços IP dos *hosts* remotos que interagiram com o *site*
Tradução de Endereços	Faixa de endereços IP dos *hosts* remotos e Faixa roteável de IP (do *site*) que se deseja utilizar	Faixas de endereços IP dos *hosts* remotos que interagiram com o *site*

Tabela 3.1: Mecanismos que influenciam o tráfego.

Apenas o roteamento por políticas não se beneficia diretamente das faixas de endereços IP dos *hosts* que se comunicam com o nosso *site*. Por isso, primeiramente, estudaremos a análise do tráfego baseando-nos nestas faixas. Em seguida, analisaremos o tráfego baseado nos serviços utilizados, que serviria como auxílio no roteamento por políticas.

Essas duas abordagens podem ser utilizadas conjuntamente, dando-se preferência ao tráfego de alguns serviços e/ou ao tráfego de alguns *sites* seguirem por determinadas conexões.

3.1.1 Análise do Tráfego Baseado nas Faixas de Endereços IP dos *hosts* externos

Nosso objetivo é detectar as faixas de endereços IP com maior quantidade de tráfego associada, para dividirmos seus tráfegos entre as conexões, utilizando os mecanismos de roteamento estático, serviço de nomes (DNS) e tradução de endereços (NAT).

Faixas de endereços IP são formadas por um endereço de rede associado a uma máscara de rede. Elas servem para agrupar endereços que provavelmente estão próximos uns dos outros. Assim, evitamos tratar cada *host* individualmente, já que o número de *hosts* que interage com uma rede é normalmente bastante elevado.

Em nossa análise temos como medir o tráfego para *hosts* individuais, mas não possuímos informação sobre suas redes. Desta forma, necessitamos de um mecanismo que agrupe tais *hosts* em faixas mais abrangentes. A máscara de rede é a responsável por separar os bits dos endereços IP que representam a rede dos que representam os *hosts* desta rede. Então, escolhendo-se a máscara apropriada, estaremos associando os endereços IP de mesma rede numa única faixa de tamanho definido por ela.

As vantagens de usarmos faixas mais abrangentes (de maior quantidade de *hosts*) são a menor necessidade de uso de espaço na memória do roteador para armazenar as rotas e o menor processamento do roteador para efetuar o roteamento. Já a vantagem de faixas menos abrangentes é o maior grau de certeza que a rota será a melhor para acessar determinado endereço externo. Quando somos abrangentes demais, deixamos de garantir que o tráfego para um *host* está indo pelo melhor caminho, já que não conhecemos os detalhes da rede do mesmo. Devemos, então, escolher a máscara mais adequada à nossa análise.

Com a máscara definida, medimos o tráfego relativo a cada uma das faixas e fazemos a divisão do tráfego entre as conexões. Utilizaremos apenas as faixas que obtiverem juntas uma quantidade de tráfego significativo. As faixas com baixa quantidade de tráfego devem ser evitadas para não gerar *overhead* desnecessário nos mecanismos de divisão de tráfego. Assim, podemos escolher um limite para a quantidade de faixas que

desejamos considerar, e/ou um percentual de tráfego que desejamos considerar em nossa análise.

3.1.2 Análise do Tráfego Baseado no Serviço Utilizado

Nossa segunda abordagem analisa o tráfego de acordo com o serviço de transporte utilizado. Esta lista de serviços será utilizada como entrada para o roteamento por políticas.

Numa rede IP trafegam pacotes de dois protocolos de transporte: o TCP e o UDP. Além deles, pacotes ICMP (como o ping), IGMP, OSPF também trafegam sobre o IP, porém não costumam representar parte significativa do tráfego, pois são usados para gerenciamento da própria rede.

Nos protocolos TCP e UDP, as aplicações se comunicam utilizando endereços de transporte. Esses endereços são formados pelo endereço IP associados a uma porta (número de 0 a 65535). As aplicações que rodam como serviços ficam escutando portas conhecidas, associadas ao endereço IP do *host*. Então, os clientes utilizam seu endereço IP associado a uma porta livre para estabelecer conexões a estes serviços.

Os protocolos TCP e UDP possuem seus conjuntos de 65536 portas. A porta 80/TCP, por exemplo, é utilizada para atender a conexões utilizando o protocolo HTTP. Já a 53/UDP é utilizada pelo serviço de nomes (DNS).

As portas listadas na RFC 1700 [17] são reservadas e devem ser utilizadas por protocolos devidamente cadastrados no órgão responsável: a IANA [24]. As demais podem ser utilizadas para serviços definidos pelo usuário. Baseando-se nessas portas, podemos classificar o tráfego de acordo com o serviço utilizado.

Como podem existir 65536 portas em cada um dos protocolos, e muitas delas são usadas apenas para acessar um serviço, vale a pena utilizar algum critério de filtragem para considerarmos apenas as portas que representam algum serviço. Este critério poderia ser, por exemplo, considerar apenas as portas listadas na RFC 1700 para análise do tráfego. Porém, este critério pode deixar de fora alguma porta de serviço não cadastrado responsável por parte significante do tráfego. Estas devem ser analisadas com cuidado uma a uma.

Utilizamos então, para gerar as tabelas necessárias para o roteamento por políticas, listas com as portas tcp e udp relacionadas aos serviços mais utilizados e o tráfego

associado às mesmas. Dessa forma, podemos balancear o tráfego entre as conexões pelo serviço utilizado.

Como na análise por faixas IP, devemos utilizar somente as portas que representem uma quantidade significativa de tráfego na rede.

Podemos utilizar também como critério de divisão entre as conexões a relevância do serviço para a rede. Por exemplo, pode-se querer priorizar o tráfego dos serviços http, smtp e pop3, alocando-os para conexões mais velozes e limitar o tráfego de serviços de transferências de arquivos multimedia (como o kazaa e o gnutella), alocando conexões de menor capacidade para os mesmos.

3.2 Estudo de Caso: Rede do CIn/CESAR

Em nosso estudo de caso, fazemos a análise no tráfego da rede do CIn com o objetivo de adquirir informações relevantes para o balanceamento de tráfego.

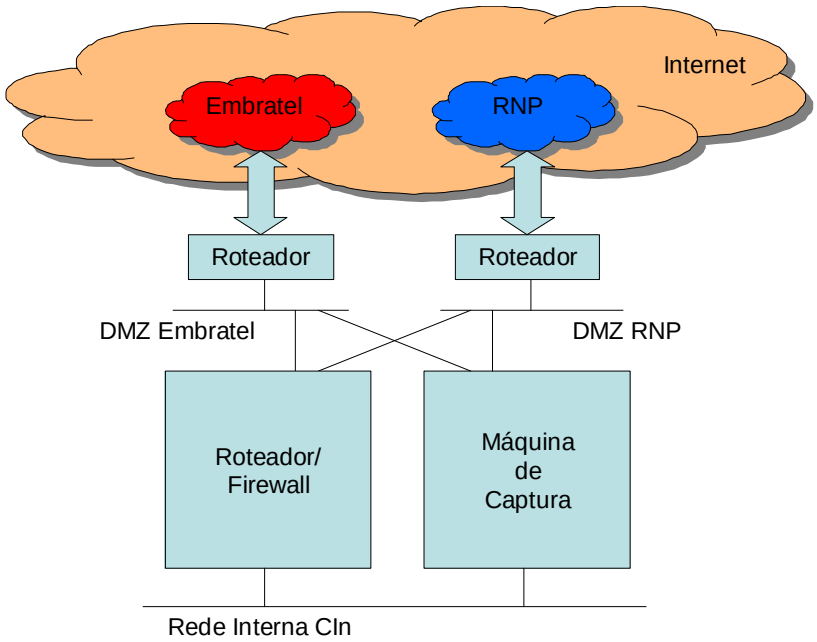


Figura 3.1: Solução de hardware proposta para captura do tráfego de uma rede TCP/IP.

Para coletar informações sobre o tipo de tráfego de uma rede interligada por mais de um canal à Internet, são necessários alguns cuidados:

1. O computador responsável pela leitura dos pacotes deve possuir interfaces nas LANs onde se encontram ligados os roteadores que se ligam à Internet. Essas interfaces devem operar em modo promíscuo para capturar todos os pacotes que passam pela rede.
2. Para que essas interfaces não interfiram no tráfego da rede, não é recomendável associar endereços IP a elas. Desta forma, as interfaces capturam os pacotes para análise, mas não interferem no tráfego da rede, evitando também ataques à estação.
3. Estando a máquina com as interfaces configuradas, necessitamos de um software que capture os pacotes que passam por elas, leia seus cabeçalhos e salve as informações necessárias para a análise em arquivos de *log*.

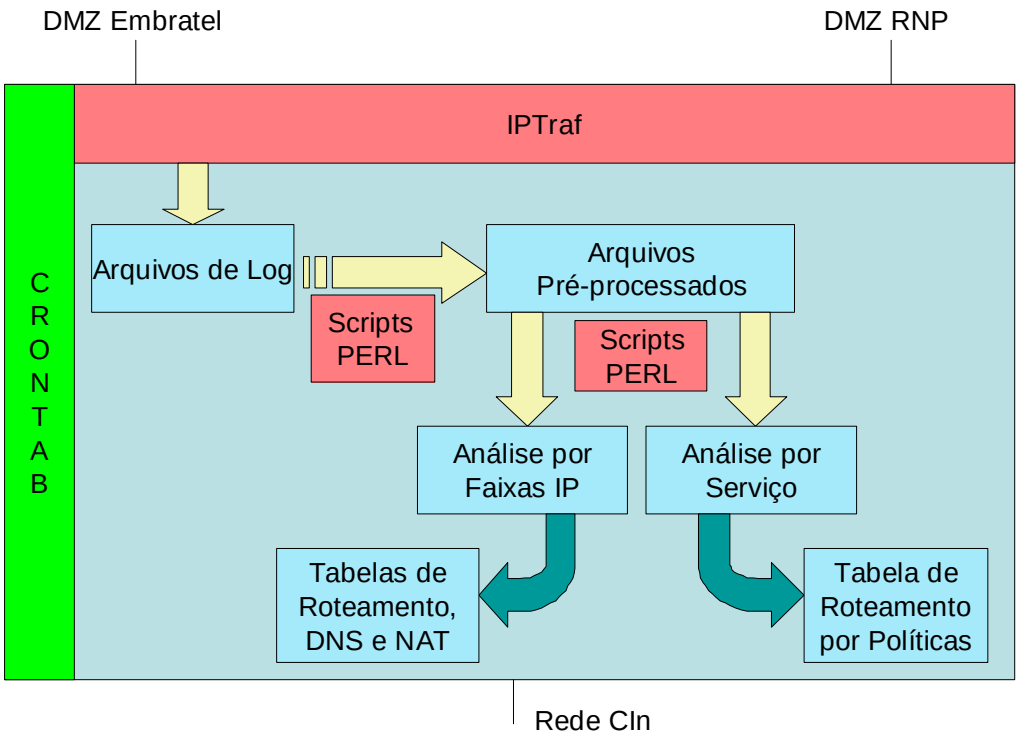


Figura 3.2: Solução de software proposta para captura do tráfego de uma rede TCP/IP.

Foram analisados dois softwares de monitoração de redes IP: o nTOP [07] e IPTraf [12]. Ambos funcionam no sistema operacional *linux*.

O nTOP, no período da análise, apesar de estar em sua versão 2.0, não possuía versões oficiais (*releases*) e mostrou-se bastante instável. A vantagem deste software é

possuir uma interface gráfica via *web* bastante amigável, além de possuir interface com alguns bancos de dados como o *mySQL* para gravar as informações.

O IPTraf possui interface em modo texto, porém gera arquivos de *log* bastante completos e já possuía versão estável no momento da análise.

Como estávamos interessados em estabilidade e completude das informações, escolhemos o IPTraf como ferramenta de auxílio na análise de tráfego da rede.

O arquivo de LOG do IPTRAF contém os seguintes campos, separados por ponto e vírgula (;) como mostraremos a seguir:

```
Campo 1: Data e Hora
Campo 2: Protocolo utilizado (TCP, UDP, ICMP, OSPF, ARP, ou Não IP)
Campo 3: Interface (eth0, eth1, eth2, ...)
Campo 4: Tamanho do Pacote (em bytes)
Campo 5: Endereços de Origem e Destino do Pacote (IP e Porta, nos
         casos de TCP/UDP)
Campo 6: Informações Extra (opcional)
```

Apenas os pacotes iniciais e finais das sessões TCP são armazenados nos arquivos de *log*. A informação sobre o final da sessão contém o a quantidade de pacotes e bytes transferidos no campo de “Informações Extras” dos pacotes finais.

Exemplo:

```
Tue Jan 22 11:00:01 2002; TCP; eth1; 552 bytes; from
        150.161.2.34:9191 to 200.252.46.183:1318; first packet

Tue Jan 22 11:00:01 2002; UDP; eth0; 229 bytes; from
        172.17.108.5:138 to 172.17.255.255:138

Tue Jan 22 11:00:01 2002; TCP; eth1; 1500 bytes; from
        216.35.123.119:80 to 150.161.2.4:3608; first packet

Tue Jan 22 11:00:01 2002; TCP; eth1; 60 bytes; from 150.161.2.34:80
        to 200.199.11.206:1033; first packet

Tue Jan 22 11:00:02 2002; TCP; eth2; 52 bytes; from 200.185.63.5:25
        to 200.249.235.15:2388; FIN sent; 2 packets, 160 bytes
```

3.2.1 Classificação do tráfego em Protocolos IP

Na primeira análise, classificou-se o tráfego separando-o em protocolos. Os protocolos que apareceram no *log* gerado pelo IPTraf foram TCP, UDP, ICMP, OSPF e

ARP. Também foram detectados pacotes de rede que não utilizam IP. Chamamos esta categoria de “Não IP”. O tráfego foi medido no intervalo de 25 de Outubro de 2001 a 31 de Dezembro de 2001 e o resultado foi o seguinte:

	Mês	% de Entrada	% de Saída	% Total
TCP	10	98.5	94.8	97.6
	11	97.3	93.1	95.9
	12	95.5	94.9	95.0
UDP	10	0.67	2.67	2.46
	11	1.99	4.56	2.30
	12	3.80	4.33	0.72
ICMP	10	0.73	2.46	1.09
	11	0.69	2.30	1.15
	12	0.61	0.72	0.73
Não IP	10	-	-	0.12
	11	-	-	0.14
	12	-	-	0.11
ARP	10	-	-	0.00
	11	-	-	0.01
	12	-	-	0.01
OSPF	10	-	-	0.04
	11	-	-	0.05
	12	-	-	0.04

Tabela 3.2: Percentual de tráfego por protocolo de transporte utilizado.

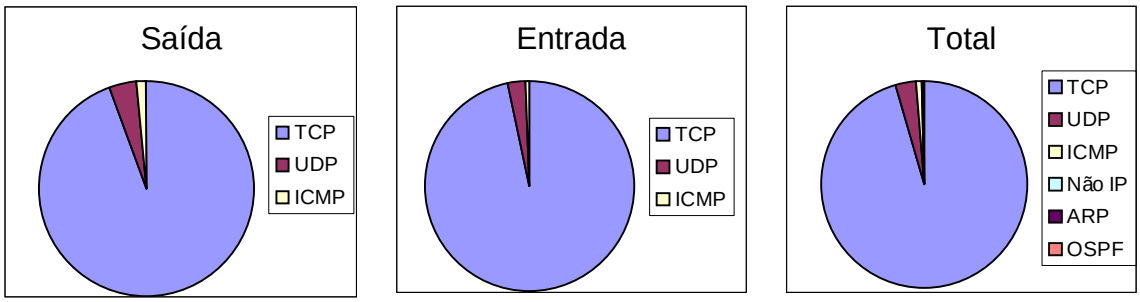


Figura 3.3: Gráficos de saída (a) entrada (b) e total (c) com os percentuais de tráfego por protocolo de transporte utilizado.

Desta forma, concluímos que o protocolo TCP representa mais de 95% do tráfego da rede IP estudada, tanto para o tráfego de entrada como para o de saída. Além disso, o protocolo TCP é responsável por menos de 10% das linhas dos arquivos de *log*. Por este motivo, consideramos apenas este protocolo, reduzindo bastante o nosso processamento.

Descartamos então, em nossas análises futuras, o tráfego dos protocolos UDP, ICMP, OSPF, ARP e pacotes não IP. Como a maioria deles são protocolos de gerenciamento, já se esperava uma baixa relevância dos mesmos. No caso de redes onde o UDP é bastante usado, podemos passar a considerá-lo em nossas análises. Esta escolha fica a critério do administrador de sistemas, que deve se basear na performance da solução.

3.2.2 Classificação do tráfego TCP em Serviços

Nossa segunda análise se relaciona ao tráfego TCP. Resolvemos classificá-lo em serviços, baseando-nos nas portas TCP utilizadas. Observamos que as principais portas utilizadas em nossa análise foram as seguintes:

21 (ftp)	22 (ssh)	23 (telnet)	25 (smtp)
53 (dns)	80 (http)	84 (interno)	110 (pop3)
113 (auth)	119 (nntp)	143 (imap)	443 (https)
993(imap-ssl)	995(pop3-ssl)	1214 (kazaa)	1755 (ms streaming)
3128 (squid)	6346 (gnutella svc)	6347 (gnutella rtr)	8080 (http proxy)

Tabela 3.3: Serviços TCP mais utilizados na rede em estudo.
Os nomes dos serviço foram obtidos nos registros da IANA.

As portas não listadas acima foram classificadas num grupo a parte, denominado “outras”. Essa classe acabou possuindo um tráfego significativo em alguns momentos, o que nos indicou que é freqüente o uso de serviços utilizando portas inusitadas que significam grande parte do tráfego geral.

Desta forma, no mesmo período anterior (de 25 Outubro de 2001 a 31 de Dezembro de 2001), obtivemos o seguinte resultado:

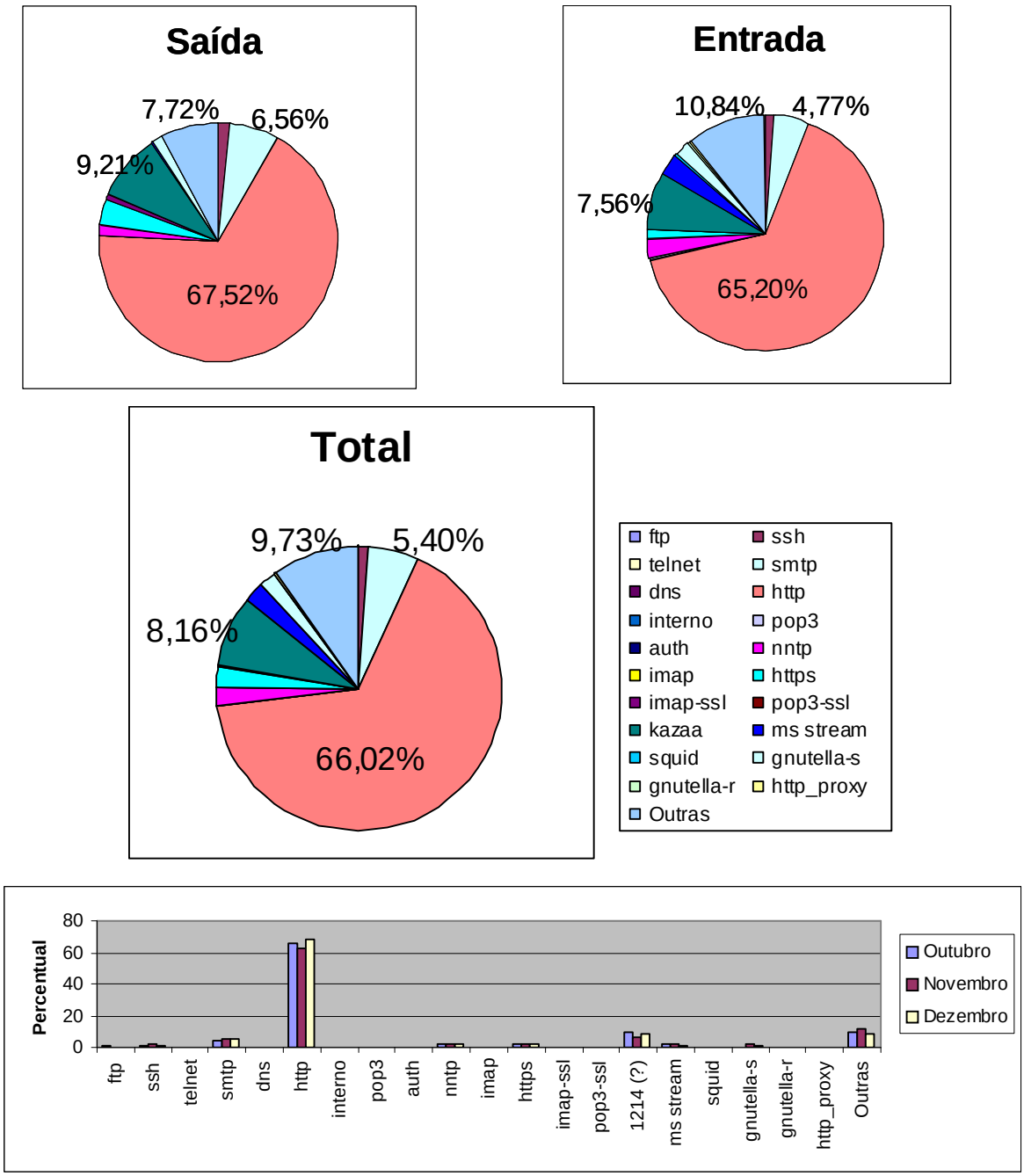


Figura 3.4: Gráficos de saída (a) entrada (b) e total (c) com os percentuais de tráfego, por serviço TCP utilizado. O gráfico (d) mostra os totais de 3 meses, indicando que os percentuais quase se mantêm.

Apesar do protocolo http (porta 80) ser o mais utilizado, isto não ocorre em todos os momentos, e o percentual de seu tráfego não é tão significativo de forma a se justificar considerarmos apenas seu tráfego, como fizemos com o protocolo TCP. Portanto,

levaremos em conta todos os serviços TCP nas análises seguintes.

Podemos também utilizar os percentuais destas portas para, através do roteamento por políticas, direcionar estes tráfegos através das conexões.

3.2.3 Classificação do tráfego TCP por Sites acessados

Nossa próxima análise leva em consideração apenas o tráfego TCP, porém com uma abordagem diferente. Analisamos os *hosts* externos que acessaram o *site* do CIn e os *hosts* externos que foram acessados pelos usuários do CIn.

Como o número de *hosts* que interagem diariamente com a rede do CIn é bastante grande, resolvemos agrupá-los em faixas de endereços IP de máscaras /8, /16 e /24 para testar qual delas é a mais adequada ao nosso caso.

Agrupar endereços IP em faixas de máscara /8, /16 e /24 significa aplicar as máscaras 255.0.0.0 (8 bits), 255.255.0.0 (16 bits) e 255.255.255.0 (24 bits), respectivamente, aos endereços IP que possuímos. Por exemplo, o IP 200.249.243.1 seria agrupado nas faixas 200.0.0.0 (/8), 200.249.0.0 (/16) e 200.249.243.0 (/24). De posse destas faixas, as ordenamos de acordo com os totais de seus tráfegos de entrada e de saída.

Para análise do tráfego por faixas, medimos o tráfego no período de 25 de outubro a 31 de dezembro de 2001. Utilizaremos, porém, a título de exemplo, o período de uma semana entre 23 e 29 de dezembro. Consideramos que o tráfego em outros períodos comporta-se de maneira semelhante.

Em nossos exemplos, também, consideramos apenas o tráfego de entrada, porém o comportamento do tráfego de saída é semelhante.

Ordenamos as faixas de acordo com o tráfego de entrada, e calculamos os percentuais de cada faixa relativos ao tráfego total de entrada. Posteriormente, calculamos os percentuais acumulados das faixas e traçamos os gráficos abaixo para cada tipo de faixa.

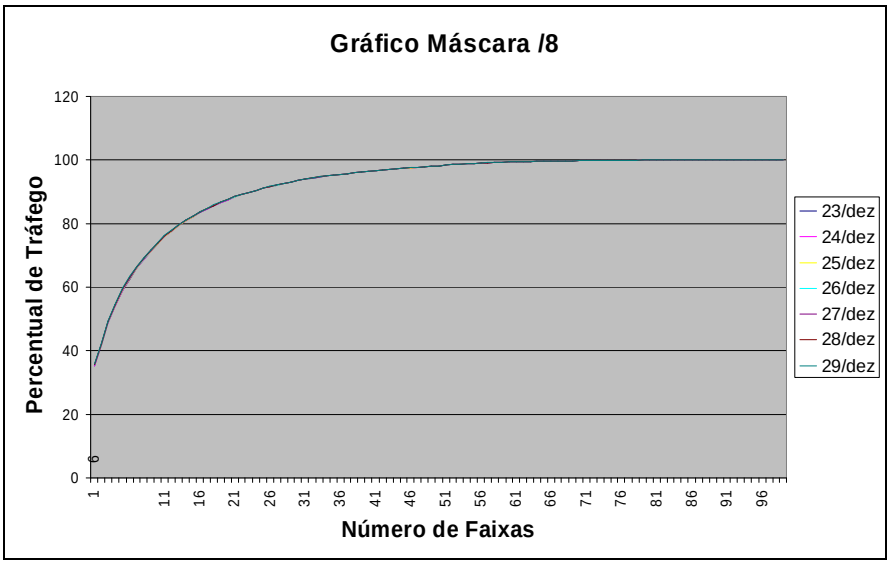


Figura 3.5: Percentual de tráfego do dia por quantidade de faixas de endereços IP de máscara /8, por dia

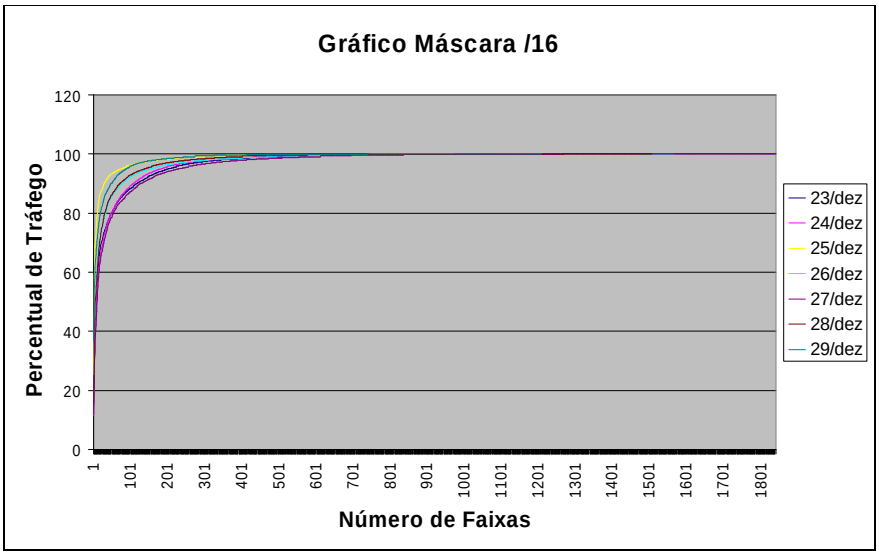


Figura 3.6: Percentual de tráfego do dia por quantidade de faixas de endereços IP de máscara /16, por dia

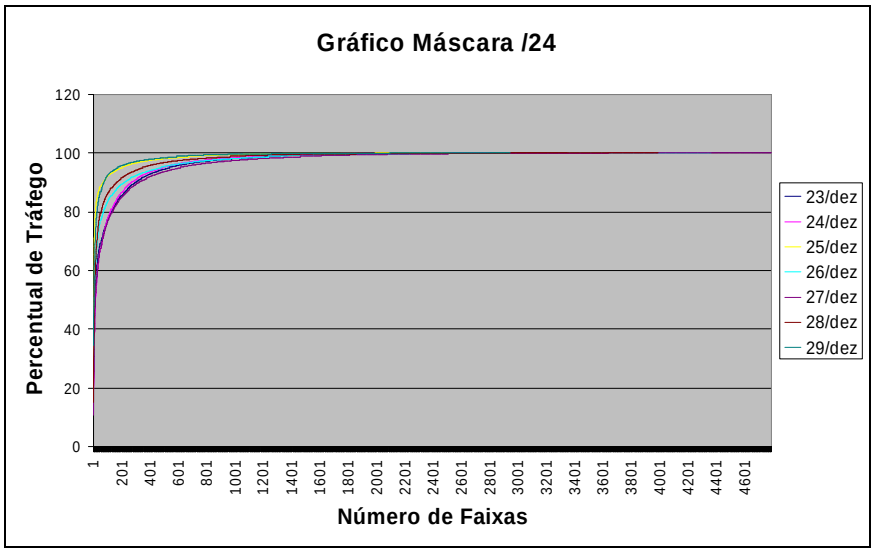


Figura 3.7: Percentual de tráfego do dia por quantidade de faixas de endereços IP de máscara /24, por dia

Número de faixas para atingirmos 90% do tráfego, total de faixas e percentual:

Dia	23/12	24/12	25/12	26/12	27/12	28/12	29/12
Faixas /8	**24**/99 24,24%	**24**/99 24,24%	**24**/99 24,24%	**24**/99 24,24%	**24**/99 24,24%	**24**/99 24,24%	**24**/99 24,24%
Faixas /16	**118**/1168 10,1%	**108**/1165 9,27%	**30**/1046 2,86%	**78**/1568 4,97%	**129**/1842 7%	**73**/1507 4,84%	**49**/1210 4,05%
Faixas /24	**296**/2814 10,52%	**270**/2801 9,63%	**68**/2522 2,7%	**218**/4367 4,99%	**322**/4795 6,71%	**166**/3990 4,16%	**74**/2946 2,51%

Tabela 3.4: Número absoluto e percentual de faixas necessárias para obter 90% do tráfego, por dia.

Em nossas análises, aparecem apenas 99 faixas de máscara /8 com apenas 24 delas representando 90% do tráfego. Cada faixa dessas é responsável por um percentual de tráfego bastante alto. Por isso, temos pouca flexibilidade para direcionarmos o tráfego utilizando faixas com máscara /8, considerada abrangente demais.

Aparecem também cerca de 1500 faixas de máscara /16, com cerca de 100 delas representando 90% do tráfego e cerca de 4000 faixas de máscara /24, com aproximadamente 300 faixas representando 90% do tráfego. Com faixas de máscara /16 abrangemos uma quantidade maior de endereços IP e ainda tempos uma quantidade menor de faixas representando 90% do tráfego. Isso nos fez escolher a máscara /16 para usarmos no balanceamento de tráfego. Apesar de sugerirmos esta máscara, o administrador pode utilizar máscaras /14, /15, /17, /18 ou alguma outra, se achar mais conveniente.

3.3 Conclusões

Concluímos que a análise de tráfego de uma rede IP é de grande utilidade para obtermos um melhor desempenho em nossa solução. Isso acontece porque, com base no conhecimento do tráfego da rede, podemos customizar nosso sistema focando apenas no tráfego relevante.

Como exemplo, analisamos o tráfego da rede do CIn – UFPE e detectamos que:

- Utilizando apenas o tráfego TCP da rede teríamos um ganho de performance de mais de 90% (percentual de linhas de *log* relativas aos outros protocolos) com uma representatividade de mais de 95% do tráfego total da rede.
- A relevância dos serviços utilizados na rede muda bastante com o tempo. Porém, os diversos serviços trafegados na rede podem ser usados como critério para o balanço de carga entre as conexões.
- Escolhendo-se faixas de IP de tamanhos diferentes, ganha-se e perde-se em desempenho e abrangência. Faixas /16 pareceram ser de bom tamanho para balancearmos o tráfego entre as conexões. Este tamanho, porém, pode ser um parâmetro ajustável em nossa solução.

Capítulo 4

Análise de Previsibilidade de Tráfego

Neste capítulo, tentaremos prever o comportamento do tráfego da rede. O objetivo desta previsão é descobrir formas de dividir o tráfego entre as conexões existentes com a Internet.

Existem várias técnicas para previsibilidade de tráfego. Podemos simplesmente conhecer os gostos dos usuários que acessarão os *hosts* a partir do *site* e, baseado neles deduzirmos os principais *hosts* que estes acessarão. Esta técnica em particular é bastante difícil de ser usada, pois na maioria dos *sites* possuem inúmeros usuários bastante heterogêneos e a Internet possui uma quantidade maior ainda de *hosts*.

Técnicas que se baseiam no tráfego passado do *site* tendem a ter uma capacidade maior de acerto, já que normalmente as pessoas tendem a repetir o acesso aos *hosts* que já acessaram em algum momento anterior. Se tivermos este histórico do tráfego da rede, então, temos chances de acertar o tráfego em um período seguinte. Existem várias famílias de algoritmos e mecanismos que utilizam a análise de tráfego de um período e gera um conjunto dos *hosts* mais prováveis de serem acessados, e com que percentual de tráfego.

Métodos estatísticos e técnicas de inteligência artificial como CBR (*Constraint Based Reasoning*) e algoritmos de aprendizagem (redes neurais, algoritmos genéticos, aprendizado por reforço, etc.) podem ser usados eficientemente nesta previsão.

Optamos por utilizar fórmulas estatísticas para termos um sentimento maior de como se comporta o tráfego da rede. Em trabalhos futuros, podemos, de posse destes resultados, implementar as técnicas de I.A., já com uma idéia de como as coisas funcionam.

Decidimos também utilizar a banda das conexões existentes como critério para a divisão do tráfego. Assim, obtendo uma amostra do tráfego em um determinado período, escolhemos faixas que representem percentuais do tráfego total na rede e, a partir destas, tentaremos dividir percentualmente o tráfego entre as conexões. Portanto, não levamos em consideração a distância de *hops* entre os *sites*.

A primeira dúvida que surge então é: qual a medida de tempo que devemos utilizar para obtermos amostras em nossa análise de tráfego e de quanto em quanto tempo deveremos efetuar tal previsão? Intuitivamente, resolvemos utilizar as medidas: hora, dia, semana e mês em nossa análise. Também, pelo mesmo método, resolvemos prever o tráfego de uma hora e de um dia.

4.1 Estudo de Caso: Previsão da Rede do Centro de Informática

Tentaremos prever o tráfego do Centro de Informática. Para isso, temos que escolher alguns parâmetros que sejam mais adequados ao *site*. A primeira decisão que tomamos foi se iríamos utilizar o tráfego de entrada ou de saída nesta previsão. Em seguida, escolhemos a máscara (que define o tamanho das faixas) mais adequada, os protocolos relevantes no *site* e outros parâmetros. Para tais escolhas, utilizamos o conhecimento que temos sobre o *site* e os dados das análises anteriores.

Os dados de entrada que possuímos são os arquivos de *log* gerados pelo IPTraf. Estes arquivos são gerados a cada hora com todo o tráfego dos pacotes que transitaram na rede neste período. Temos que filtrar e trabalhar esses dados para obtermos informações relevantes com as quais poderemos utilizar em nossa previsão.

Para o caso do CIn, trabalhamos com o tráfego de entrada da rede, já que se trata de um *site* onde a maior parte do tráfego está associado ao acesso à Internet de seus usuários.

O gráfico a seguir, medido no período de 06 de outubro de 2001 a 16 de julho de 2002, nos comprova isto.

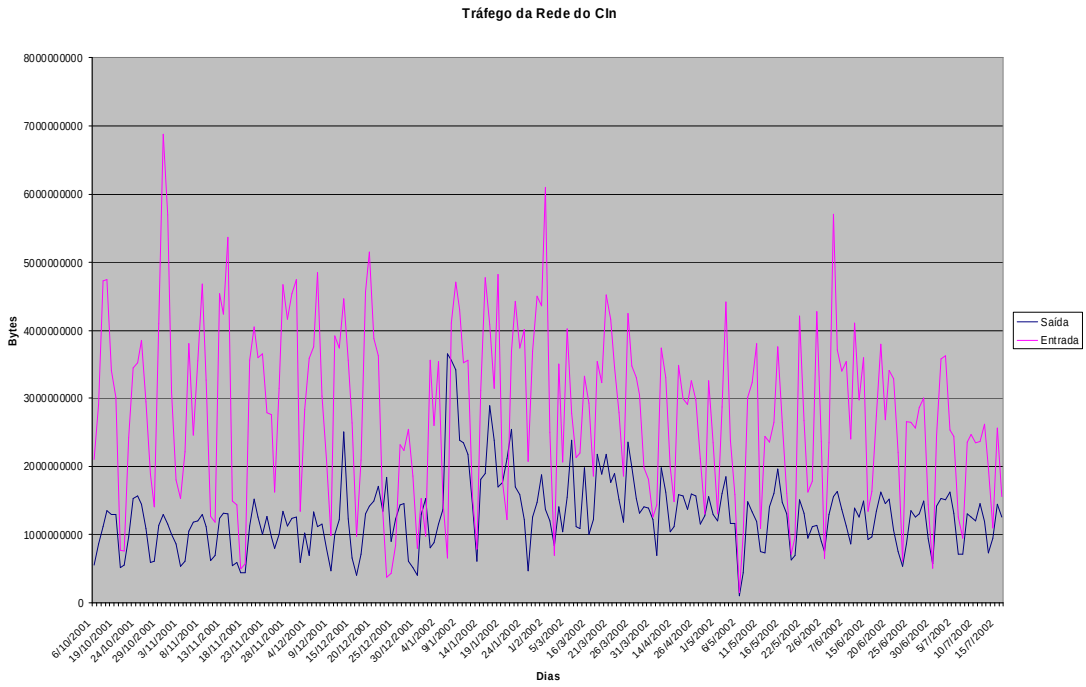


Figura 4.1: Gráfico do tráfego de entrada e saída da rede do Centro de Informática – UFPE

Desprezaremos também todo o tráfego não-TCP/IP, já que este representa um percentual de mais de 90% do tráfego, em bytes, e apenas cerca de 20% das linhas de *log*. Usaremos a máscara /16 (255.255.0.0) para agruparmos os endereços dos *hosts* que conversamos, pelo resultado de nossa análise feita no capítulo anterior.

4.1.1 Análise do tráfego de uma hora para prever o da hora seguinte

Nesta primeira análise, usaremos o tráfego coletado durante uma hora como base para previsão do tráfego da hora seguinte. Utilizaremos as informações sobre o tráfego, que foram coletadas pelo aplicativo IPTraf durante o mês de Dezembro de 2001. Tal aplicativo foi configurado para gerar arquivos de *log* a cada hora com informações sobre todos os pacotes que passam pela rede.

Efetuamos os seguintes passos em nossa previsão:

1. Isolamos o tráfego TCP/IP dos demais protocolos capturados pelo IPTraf. Como vimos no capítulo anterior, este protocolo é responsável por mais de 90% do tráfego total. Desta forma, então, diminuímos o tempo de processamento, já que as linhas do *log* referentes ao TCP representam apenas 20% do arquivo.

2. Separamos o tráfego por faixas (de máscara /16) de endereços que trafegaram com o *site*. Contabilizamos, separadamente, o tráfego de entrada e o tráfego de saída para cada uma delas. Isso é feito para cada arquivo de *log* com o tráfego TCP/IP de cada hora. Escolhemos trabalhar com o tráfego de entrada, já que o *site* do Centro de Informática é formado principalmente por usuários que acessam a Internet.

3. Geramos novos arquivos de *log* com as faixas ordenadas decrescentemente pelo tráfego de entrada. Desta forma sabemos quais os *sites* que mais enviaram dados à nossa rede na hora em questão.

4. Calculamos o total de tráfego de entrada de cada hora somando o tráfego de entrada de cada faixa, gravando num arquivo.

5. Com a quantidade de tráfego de cada faixa e com o total, calculamos o percentual de cada uma delas em cada hora.

6. Com as faixas ordenadas decrescentemente pelo tráfego de entrada em cada hora, obtemos o grupo de faixas que representam um percentual desejado do tráfego da hora. Em nossas análises, tentaremos prever 50%, 75% e 80% do tráfego.

7. Verificamos finalmente, o quanto o tráfego gerado por estas faixas na hora seguinte representa percentualmente em relação ao tráfego total da hora seguinte.

Todo este processo é feito a partir de *scripts* programados na linguagem *Perl*. Escolhemos tal linguagem pela praticidade de se trabalhar com arquivos texto (como nossos *logs*). Cada passo gerou arquivos de *log* que serviram como entrada para o passo seguinte. Com os dados percentuais gerados no passo 7, concluímos que a medida de uma hora não é adequada para previsibilidade do tráfego da hora seguinte. Atribuímos a esta conclusão ao fato do tráfego neste período é pequeno e, portanto, irrelevante para

especularmos o tráfego gerado na hora posterior à hora analisada.

4.1.2 Análise do tráfego de um dia para prever o do dia seguinte

Desta vez, tentaremos utilizar o tráfego diário para prever o tráfego do dia seguinte.

As informações sobre o tráfego foram as mesmas coletadas através do aplicativo IPTraf, mas num período maior: de 15 de outubro de 2001 a 16 de julho de 2002. Neste período de 270 dias, obtivemos sucesso na coleta de 201 destes dias (74,4%). Nos outros 25,5% houve perda de informações devido a problemas técnicos referentes ao espaço em disco destinado à captura dos dados. Consideramos o volume de dados satisfatório para tirarmos conclusões a partir de sua análise.

Como vimos anteriormente, os arquivos de *log* são gerados de hora em hora.

Efetuamos os seguintes passos para obtermos nossa nova previsão:

1. Isolamos o tráfego que utiliza o protocolo TCP/IP dos demais, como anteriormente. Podemos utilizar também o tráfego UDP caso ele seja significativo na rede em questão.
2. No início do dia seguinte, percorremos os arquivos de *log* gerados por hora e totalizamos os tráfegos de entrada e de saída dos endereços IP que trafegaram com o *site* durante o dia.
3. Em seguida, os endereços IP são agrupados em faixas de máscara /16. Apenas neste passo, utilizamos um programa em Java. A idéia inicial era calcular uma forma mais elaborada de gerar máscaras entre /16 e /24, as mais abrangentes possíveis. Como percebemos que a maioria das faixas acabava utilizando máscaras /16, resolvemos considerar apenas os dois primeiros bytes do endereço IP.
4. Ordenamos as faixas de endereço pelo tráfego de entrada (por ser o mais relevante ao *site*) em ordem decrescente e calculamos o total de tráfego de entrada de cada dia somando todos tráfegos de entrada.
5. Obtemos em seguida o percentual de cada uma dessas faixas no dia.
6. Com as faixas ordenadas pelo tráfego em cada dia, obtemos o grupo de faixas

que representam 50%, 75% ou 80% (percentual desejado) do tráfego do dia. Utilizamos, em seguida, estas faixas para calcular o quanto as mesmas geram de tráfego no dia seguinte. Esses dados foram gerados em termos percentuais.

Todo o processo (exceto o passo 3) é feito a partir de *scripts* programados na linguagem *Perl*. Cada passo gerou arquivos de *log* que serviram como entrada para o passo seguinte. Com os dados percentuais gerados no passo 6, produzimos os gráficos a seguir, utilizando o Microsoft Excel. Nestes gráficos, eliminamos os dados "faltosos" para uma melhor visualização.

O gráfico da Figura 4.2 mostra que nos primeiros meses, durante um período de greve da Universidade, acarretando um tráfego menor na rede, a previsão oscilou nos 35%. Porém, com a volta às aulas, a previsão oscila em torno de 45% para prever o tráfego de 50% do total.

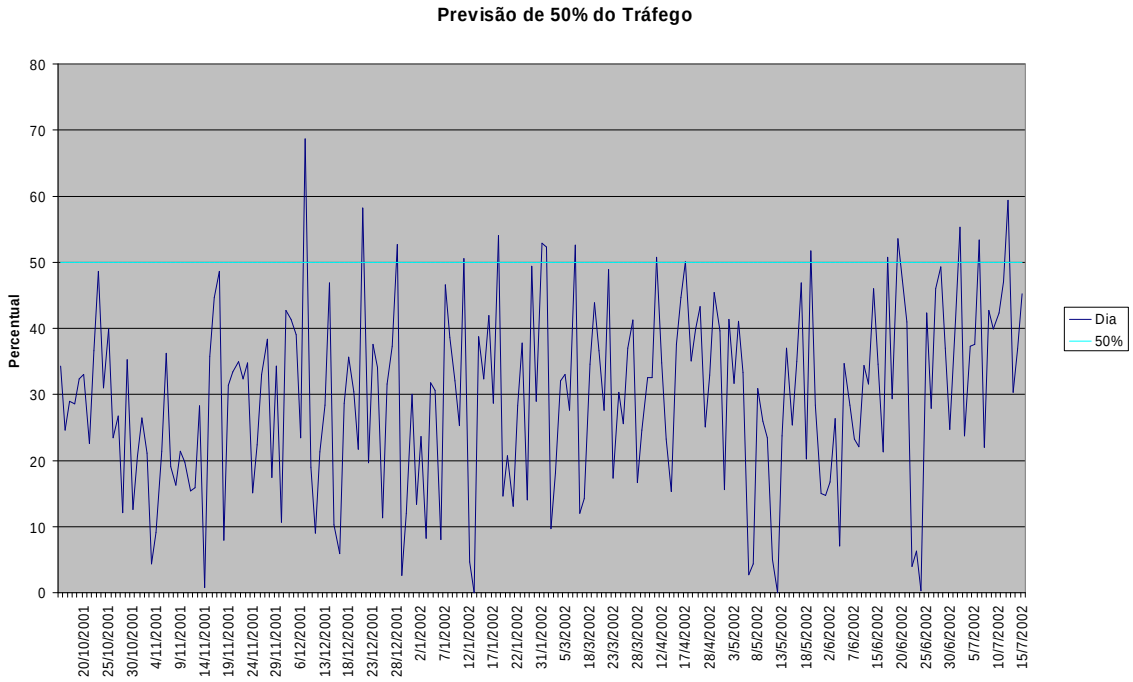


Figura 4.2: Previsão de 50% do tráfego de um dia baseado no tráfego do dia anterior.

Da mesma forma, na Figura 4.3, os primeiros meses oscilam na média em 50% para se prever 75% do tráfego total. Já no período final, a média é de 65% e na figura 4.4, para

previsão de tráfego de um dia utilizando o dia anterior como base, temos nos primeiros meses uma média de 60% quando queríamos chegar no 80%. Nos meses finais, a oscilação fica em torno dos 70%.

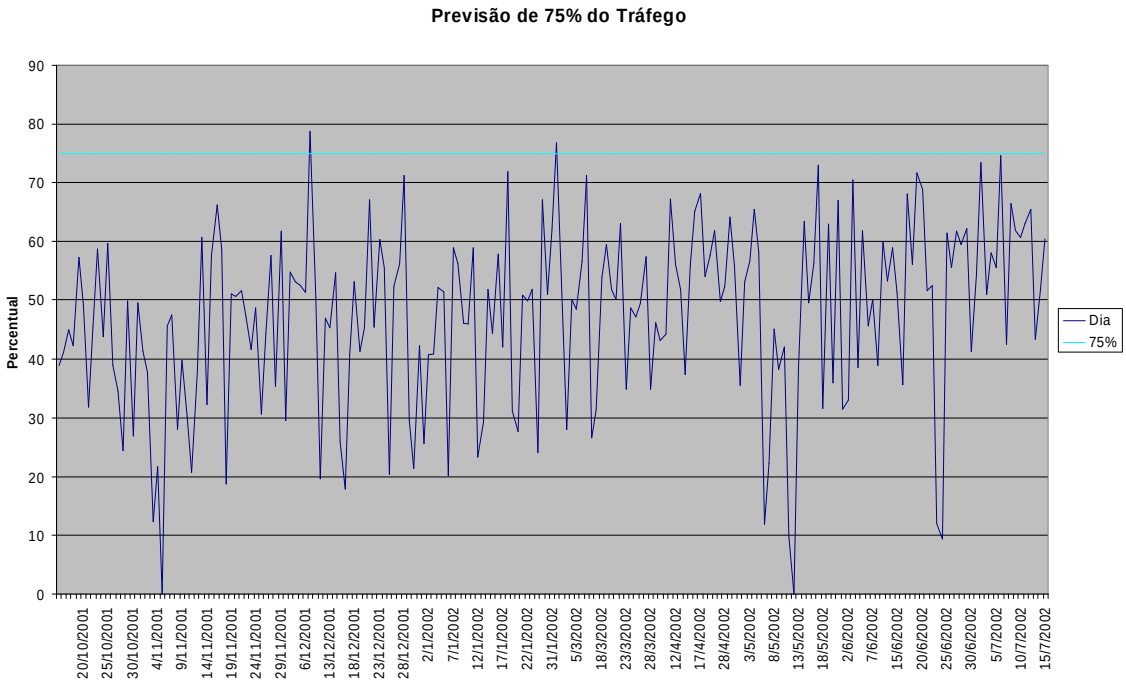


Figura 4.3: Previsão de 75% do tráfego de um dia baseado no tráfego do dia anterior.

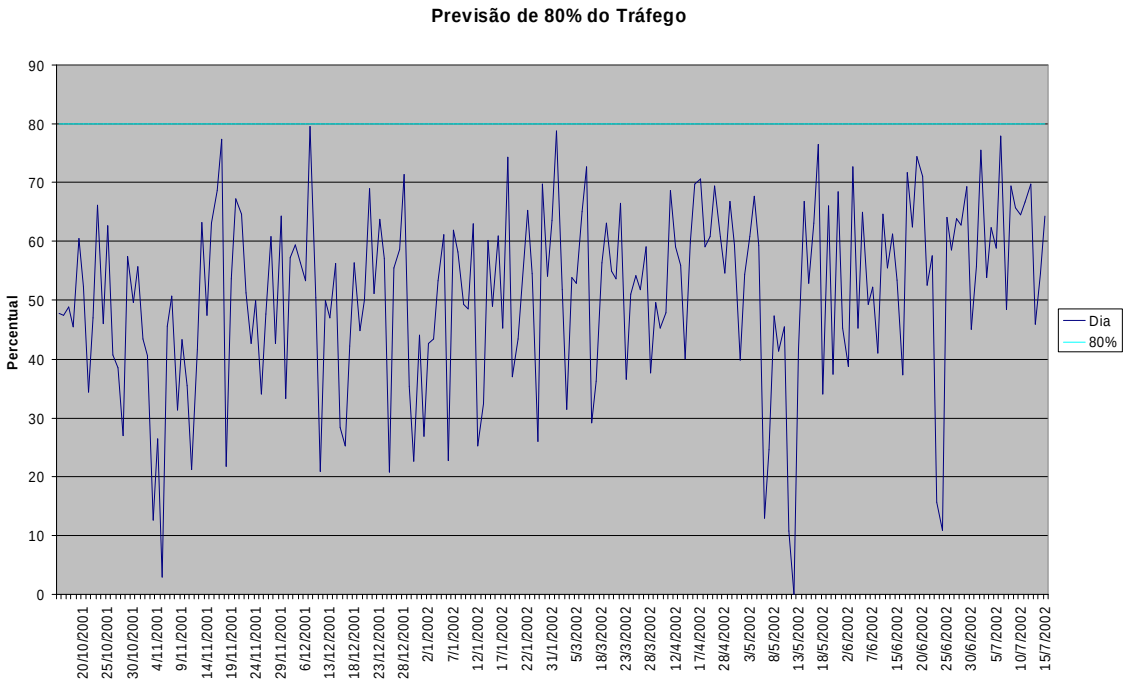


Figura 4.4: Previsão de 80% do tráfego de um dia baseado no tráfego do dia anterior.

4.1.3 Análise do tráfego de uma semana para prever o do dia seguinte

Neste passo, utilizaremos o tráfego acumulado de toda semana para prever o tráfego do dia seguinte a mesma.

As informações sobre o tráfego são coletadas da mesma forma que anteriormente, através do aplicativo IPTraf. O período de coleta foi o mesmo que na seção anterior: de 15 de outubro de 2001 a 16 de julho de 2002. Neste período de 270 dias, obtivemos sucesso na coleta de 201 destes dias (74,4%). Nos outros 25,5% houve perda de informações devido a problemas técnicos referentes ao espaço em disco destinado à captura dos dados. Consideramos o volume de dados satisfatório para tirarmos conclusões a partir de sua análise.

Efetuamos os seguintes passos para obtermos nossa nova previsão:

0. Na análise anterior, geramos um arquivo com o total de tráfego por dia. Utilizamos este arquivo para obter os mesmos totais por semana. Consideramos como semana “07 de dezembro de 2001”, o período que vai de 01 a 07 de

dezembro de 2001 (7 dias).

1. Isolamos o tráfego que utiliza o protocolo TCP/IP dos demais, como anteriormente. Podemos utilizar também o tráfego UDP caso ele seja significativo na rede em questão.
2. No início do dia seguinte, percorremos os arquivos de *log* gerados por hora e totalizamos os tráfegos de entrada e de saída dos endereços IP que trafegaram com o *site* durante o dia.
3. Em seguida, os endereços IP que trocaram pacotes com o *site* são agrupados em faixas de máscara /16 (programa em Java).
4. Carregamos as faixas e seus tráfegos diários em memória e somamos o tráfego de cada uma delas no período de uma semana (últimos 7 dias).
5. Ordenamos as faixas de endereço pelo tráfego de entrada (por ser o mais relevante ao *site*) em ordem decrescente e calculamos o total de tráfego de entrada, por faixa, de cada semana somando todos tráfegos de entrada.
6. Obtemos em seguida o percentual de cada uma dessas faixas na semana utilizando o arquivo de totais por semana, gerado no passo 0.
7. Com as faixas ordenadas pelo tráfego em cada semana, obtemos o grupo de faixas que representam 50%, 75% ou 80% (percentual desejado) do tráfego da semana. Utilizamos, em seguida, estas faixas para calcular o quanto as mesmas geram de tráfego no dia seguinte. Esses dados foram gerados em termos percentuais.

Todo o processo (exceto o passo 3) é feito a partir de *scripts* programados na linguagem *Perl*. Cada passo gerou arquivos de *log* que serviram como entrada para o passo seguinte. Com os dados percentuais gerados no passo 7, produzimos os gráficos a seguir, utilizando o Microsoft Excel. Nestes gráficos, eliminamos os dados "faltosos" para uma melhor visualização.

No gráfico da Figura 4.5, tentamos prever 50% do tráfego de um dia, nos baseando no tráfego da semana que o antecede. No primeiro período vemos que, devido ao baixo volume de tráfego, a média vai caindo de 50% até 30%. Com o retorno do movimento

normal na rede, a oscilação fica em torno dos 50% que gostaríamos.

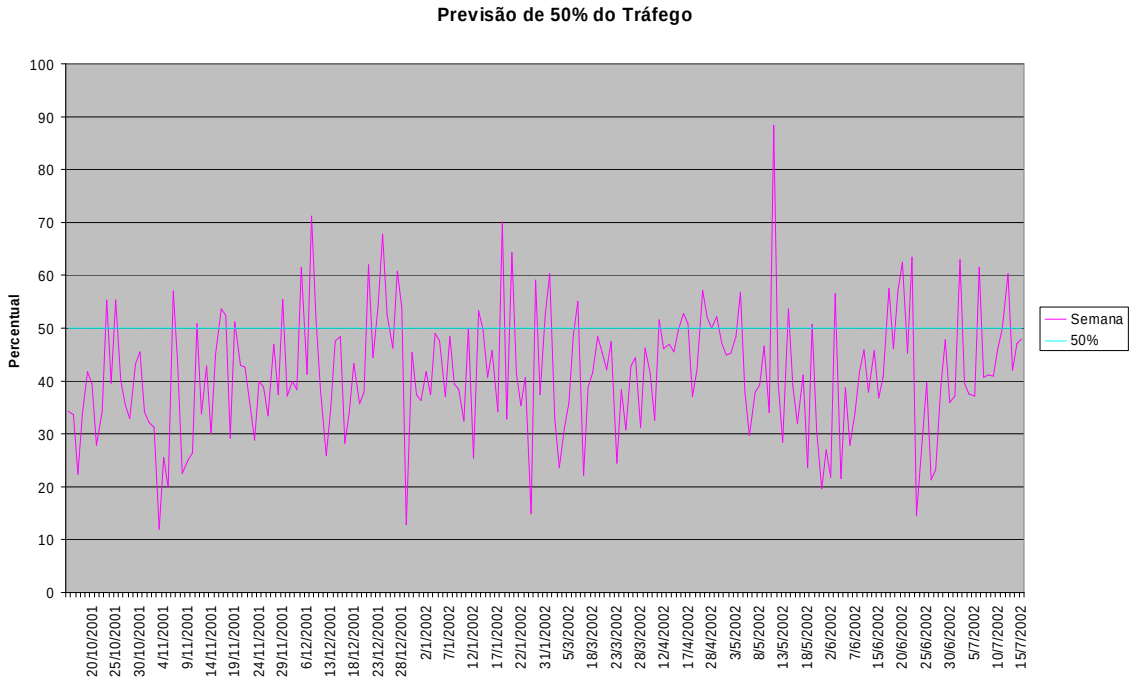


Figura 4.5: Previsão de 50% do tráfego de um dia baseado no tráfego da semana anterior.

Já na Figura 4.6, queremos prever 75% do tráfego. Nos primeiros meses, a média oscila de 70% a 40%. Com o movimento normal, o tráfego previsto oscila nos 75% desejados. Na Figura 4.7, tentamos prever 80% do tráfego de um dia com base no tráfego da semana anterior ao mesmo. Nos primeiros meses temos médias que vão dos 70% aos 45%. No período final, a oscilação fica em torno dos 80% como gostaríamos.

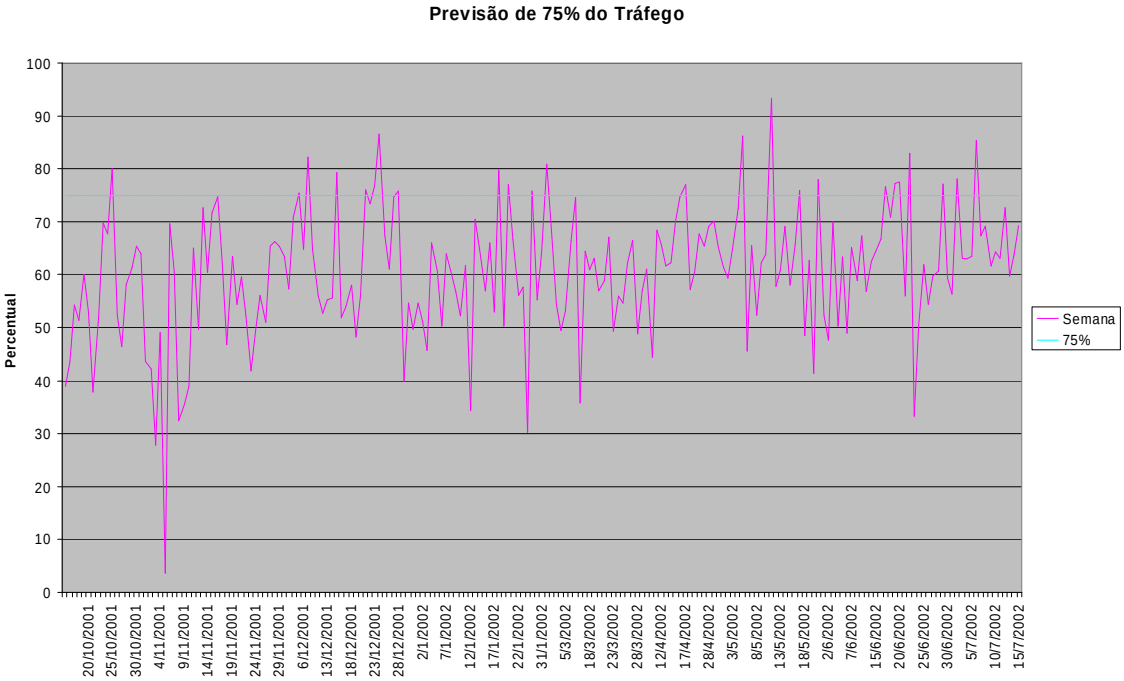


Figura 4.6: Previsão de 75% do tráfego de um dia baseado no tráfego da semana anterior.

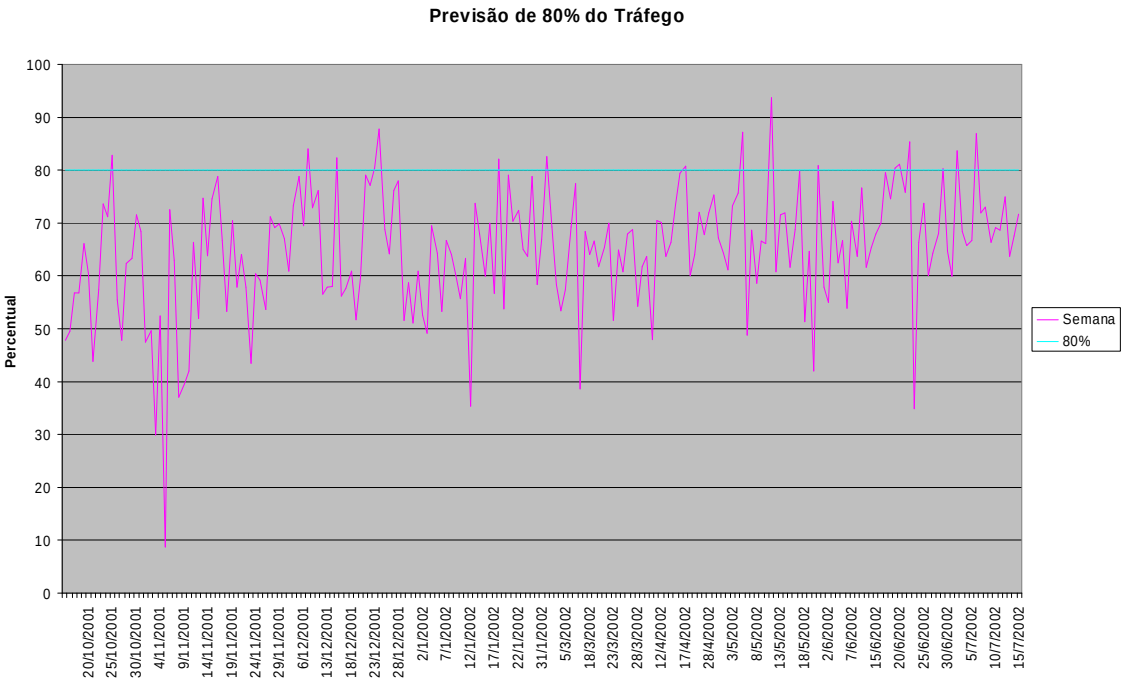


Figura 4.7: Previsão de 80% do tráfego de um dia baseado no tráfego da semana anterior.

4.1.4 Análise do tráfego de um mês para prever o do dia seguinte

Em nossa última análise, faremos um processo semelhante ao anterior para prever o tráfego de um dia. Só que no lugar de medirmos o tráfego de uma semana, desta vez mediremos o tráfego de um mês. Entendemos por mês, nesta análise, o período de 30 dias que antecedem ao dia que terá o tráfego previsto.

As informações sobre o tráfego são coletadas da mesma forma que anteriormente, através do aplicativo IPTraf. O período de coleta foi o mesmo que na seção anterior: de 15 de outubro de 2001 a 16 de julho de 2002. Neste período de 270 dias, obtivemos sucesso na coleta de 201 destes dias (74,4%). Nos outros 25,5% houve perda de informações devido a problemas técnicos referentes ao espaço em disco destinado à captura dos dados. Consideramos o volume de dados satisfatório para tirarmos conclusões a partir de sua análise.

Efetuamos os seguintes passos para obtermos nossa nova previsão:

0. Na análise anterior, geramos um arquivo com o total de tráfego por dia. Utilizamos este arquivo para obter os mesmos totais por mês. Consideramos como mês “31 de dezembro de 2001”, o período que vai de 01 a 30 de dezembro de 2001 (30 dias).
1. Isolamos o tráfego que utiliza o protocolo TCP/IP dos demais, como anteriormente. Podemos utilizar também o tráfego UDP caso ele seja significativo na rede em questão.
2. No início do dia seguinte, percorremos os arquivos de *log* gerados por hora e totalizamos os tráfegos de entrada e de saída dos endereços IP que trafegaram com o *site* durante o dia.
3. Em seguida, os endereços IP que trocaram pacotes com o *site* são agrupados em faixas de máscara /16 (programa em Java).
4. Carregamos as faixas e seus tráfegos diários em memória e somamos o tráfego de cada uma delas no período de um mês (últimos 30 dias).
5. Ordenamos as faixas de endereço pelo tráfego de entrada (por ser o mais relevante ao *site*) em ordem decrescente e calculamos o total de tráfego de

entrada, por faixa, de cada mês somando todos tráfegos de entrada.

6. Obtemos em seguida o percentual de cada uma dessas faixas no mês utilizando o arquivo de totais por mês, gerado no passo 0.

7. Com as faixas ordenadas pelo tráfego em cada mês, obtemos o grupo de faixas que representam 50%, 75% ou 80% (percentual desejado) do tráfego do mês. Utilizamos, em seguida, estas faixas para calcular o quanto as mesmas geram de tráfego no dia seguinte. Esses dados foram gerados em termos percentuais.

Todo o processo (exceto o passo 3) é feito a partir de *scripts* programados na linguagem *Perl*. Cada passo gerou arquivos de *log* que serviram como entrada para o passo seguinte. Com os dados percentuais gerados no passo 7, produzimos os gráficos a seguir, utilizando o Microsoft Excel. Nestes gráficos, eliminamos os dados "faltosos" para uma melhor visualização.

Nos gráficos das Figuras 4.8, 4.9 e 4.10 temos a análise da influência do tráfego de entrada de um mês na tentativa de prever 50%, 75% e 80% do tráfego de entrada de um dia após este mês, respectivamente. Percebemos que os dados oscilam nestas faixas como desejávamos.

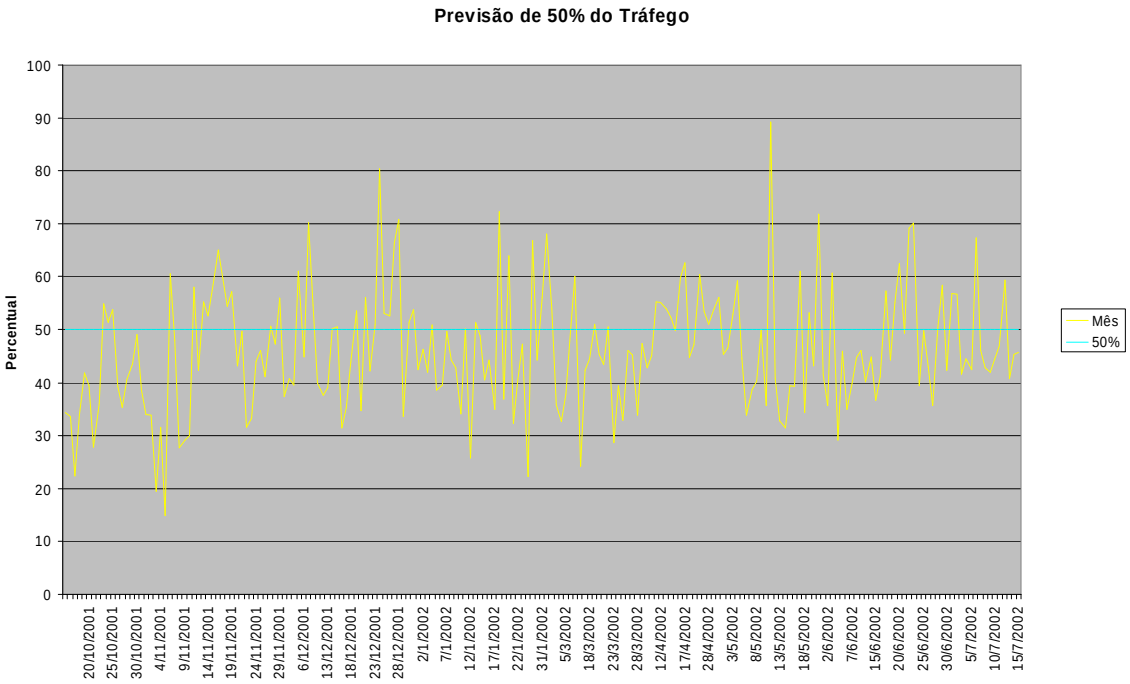


Figura 4.8: Previsão de 50% do tráfego de um dia baseado no tráfego do mês anterior.

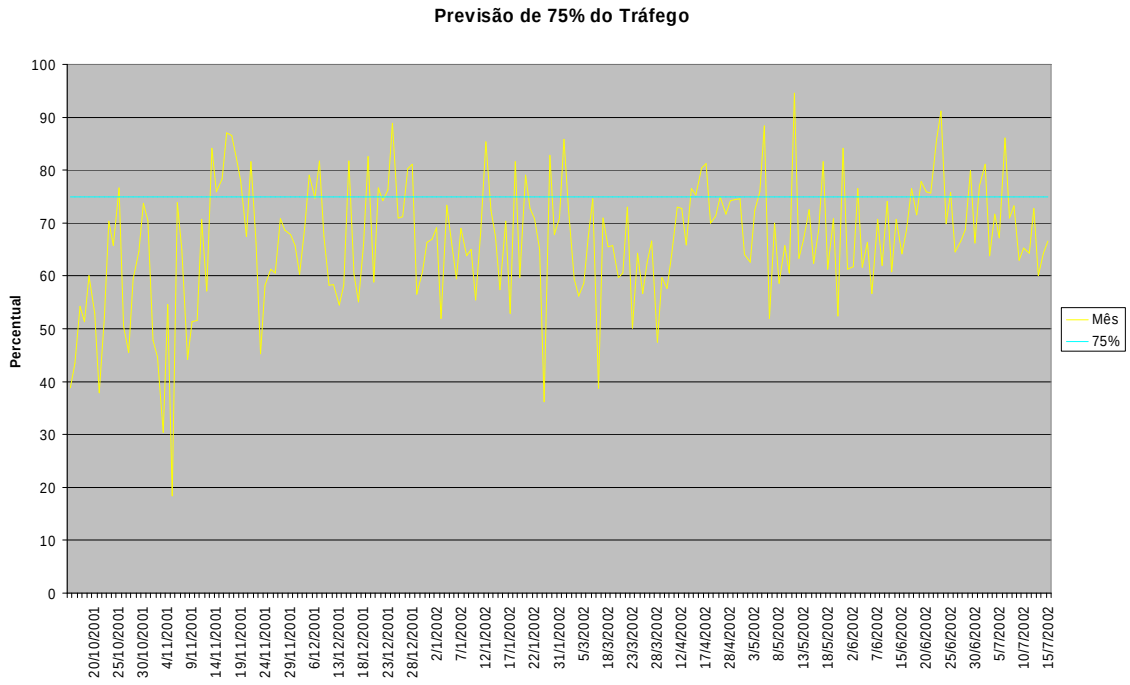


Figura 4.9: Previsão de 75% do tráfego de um dia baseado no tráfego do mês anterior.

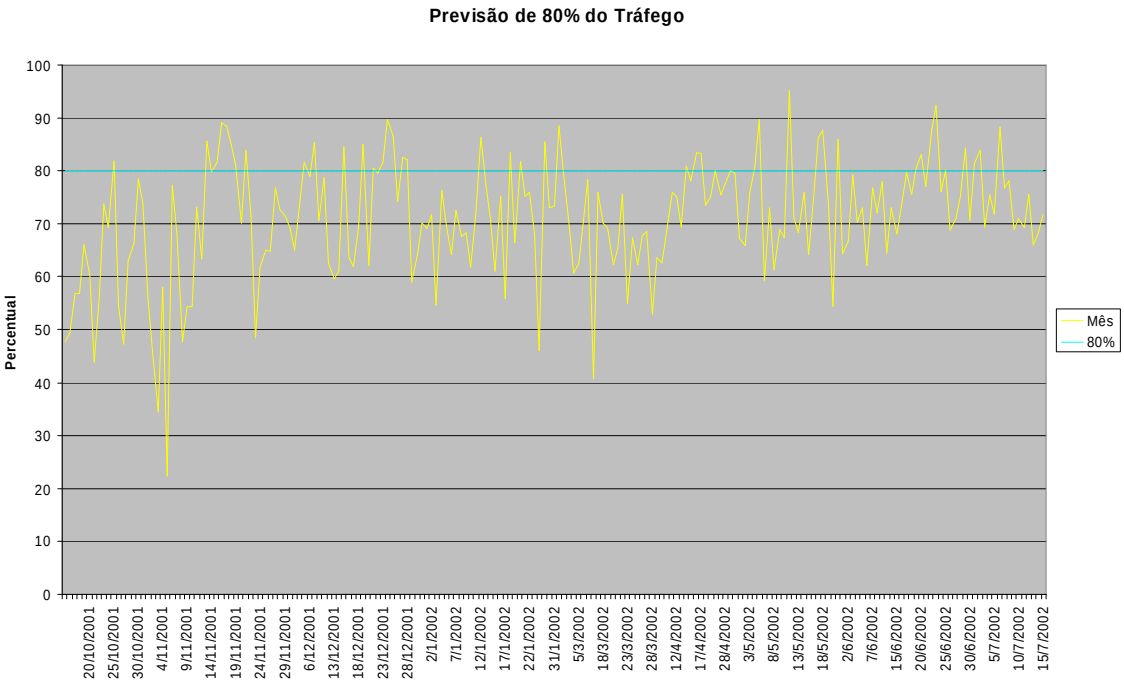


Figura 4.10: Previsão de 80% do tráfego de um dia baseado no tráfego do mês anterior.

4.2 Conclusões

Com este estudo, concluímos que a técnica utilizada foi de grande utilidade para compreendermos o quanto influi algumas unidades de tempo no tráfego futuro de uma rede. Porém, percebemos que podemos chegar mais próximos de nossos objetivos se utilizarmos técnicas mais sofisticadas como as de I.A.

Percebemos, com nossas análises, que:

- Utilizar uma hora para prever o tráfego da hora seguinte não é uma boa idéia para o nosso *site*. Atribuímos isto ao volume de tráfego, que consideramos pequeno para tirarmos uma amostragem e obtermos resultados para o período seguinte. Em *sites* com maior volume de tráfego esta medida pode apresentar melhores resultados.
- Utilizar um dia para prever o tráfego do dia seguinte já nos mostra um resultado mais próximo do que procuramos, mas ainda ruim. Vemos que tanto faz termos um bom grau de acerto num dia, como um péssimo no dia

seguinte, oscilando bastante. Porém concluímos que um dia nos pareceu ser uma boa unidade de tempo para repetirmos a realização da previsão.

- Utilizar uma semana (últimos 7 dias) para prever o tráfego do dia posterior nos mostrou um resultado melhor que quando utilizamos apenas a análise do tráfego de um dia. A sensação que tivemos é que os números começavam a convergir. Porém ainda não foi o ideal, já que as oscilações ainda foram grandes.
- Enfim, utilizar um mês (últimos 30 dias) para se prever o tráfego do dia em seguida nos mostrou um resultado ainda mais convergente que o da semana. Concluímos que já é uma boa aproximação do que desejamos, mas ainda podemos melhorar, ou aumentando o período, dando pesos aos dias da semana iguais, ou alguma outra heurística.

A fim de melhorar o índice de acerto, podemos tentar achar perfis para horas ou dias que possuam tráfegos de comportamentos semelhantes e utilizá-los na previsão do tráfego de uma nova hora ou um novo dia. Técnicas de Inteligência Artificial podem ser usadas para encontrar tais perfis ou padrões.

Como exemplos destes perfis que poderiam ser encontrados, podemos citar:

- Usar o tráfego de quintas-feiras com maior influência para a previsão do tráfego de uma quinta-feira.
- Usar os tráfegos dos dias 1 dos meses anteriores com maior influência para a previsão do tráfego do dia 1 de um determinado mês.
- O comportamento do tráfego do dia 25 de dezembro de um ano teria grande influencia na previsão do tráfego deste mesmo dia, no ano seguinte.
- Para previsão do tráfego de um dia da semana, não utilizar dias de finais de semana e vice-versa.
- No lugar de dias inteiros, poderíamos trabalhar com períodos de algumas horas e tentar achar semelhanças nestes períodos em dias distintos.

Capítulo 5

Arquitetura da Solução Proposta

Neste capítulo descreveremos a solução proposta para gerenciamento de sites multi-homed que não possuem status de AS. Esta solução foi baseada em nossas análises de tráfego da rede e em nosso estudo de caso, a rede do Centro de Informática da UFPE.

Dividiremos este capítulo em duas sessões. Na primeira, explicaremos como deve ser configurado o hardware para nossa solução funcionar. Na segunda, explicaremos quais softwares devem ser instalados e como estes se comunicam.

5.1 Arquitetura de Hardware utilizada

Para por em prática a nossa solução de gerenciamento, precisamos instalar uma máquina para captura do tráfego da rede. Para isso, esta máquina deve se conectar diretamente aos canais de saída à Internet da rede através de interfaces de rede. O sistema operacional que utilizamos é o Unix, necessário para executarmos o software IPTraf.

Se o *site* estiver conectado a *n* provedores através de *n* redes "DMZ" (zonas desmilitarizadas), devemos ter então (*n* + 1) placas de rede na máquina de captura - uma placa de rede para cada "DMZ" e uma extra para o acesso na rede privativa do *site*. As interfaces conectadas às redes "DMZ" devem ser configuradas da seguinte forma:

- **Modo promíscuo**, para capturar para si todo o tráfego existente na rede. Normalmente, a placa apenas captura os pacotes destinados para o IP configurado na placa. Para tal configuração, devemos utilizar o comando: "ifconfig -i *eth0* promisc", onde *eth0* é a interface em questão.
- **Sem endereços IP associados**, para que esta interface não se tornar ponto de invasão do sistema, já que esta máquina não está protegida por *firewall* para enxergar o mundo externo. Sem endereço a placa não troca pacotes IP com os provedores. Para tal configuração, basta não atribuirmos o endereço IP à Interface em suas configurações. Os arquivos de configuração variam de acordo com a versão do Unix em questão. No caso do *linux*, na distribuição *debian* devemos omitir a configuração da interface no arquivo "*/etc/network/interfaces*". Já na distribuição *red hat*, devemos omitir as linhas BROADCAST, IPADDR, NETMASK e NETWORK do arquivo "*/etc/sysconfig/network-scripts/ifcfg-eth0*" (onde eth0 deve ser substituído pela interface).

A placa de rede conectada à rede privativa deve possuir endereço IP pertencente a rede local, e não deve estar em modo promíscuo, já que não necessita capturar pacotes da rede local que não sejam referentes a si.

As figuras a seguir, mostram dois cenários possíveis para nosso sistema de gerenciamento. Nelas, indicamos onde deve se inserir a máquina para captura dos dados no contexto de um *site* multi-homed. A máquina não deve ser utilizada para conexão com a Internet. Para isso, devemos utilizar uma firewall ou um roteador como indicado.

A arquitetura exemplificada pela figura 5.1 utiliza um roteador para cada provedor de acesso. Neste caso, a *firewall* além de proteger a rede privativa da empresa das redes desmilitarizadas (DMZ) ainda tem o papel de rotear o tráfego entre as redes. A máquina de captura está conectada às redes DMZ através de interfaces sem endereços IP associados, para proteger a rede privativa. O servidor de DNS encontra-se na rede privativa e é acessado através da firewall pelos *hosts* externos.

A arquitetura exemplificada pela figura 5.2 utiliza um único roteador ligado por diferentes interfaces aos provedores de acesso. Desta vez, a *firewall* tem o único papel de proteger a rede privativa da rede "DMZ". O roteamento, entretanto, é feito pelo próprio

roteador. A máquina de captura está conectada à rede DMZ através de uma interface sem endereço IP associado, para proteger a rede privativa. O servidor de DNS encontra-se na rede privativa e é acessado através da firewall pelos *hosts* externos.

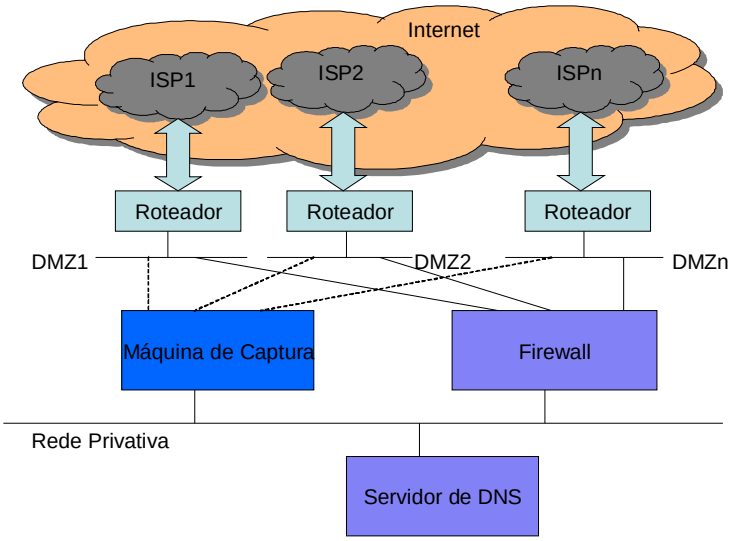


Figura 5.1: Solução de hardware proposta 1.

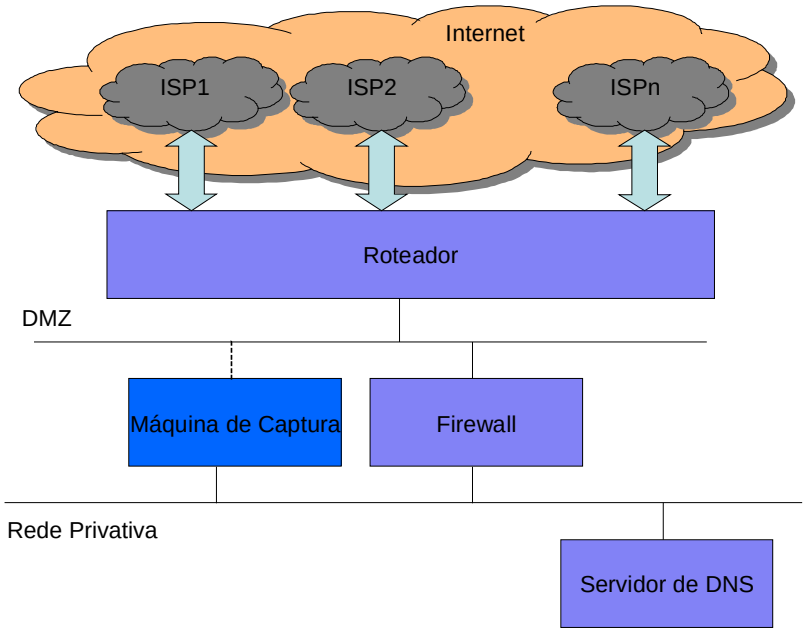


Figura 5.2: Solução de hardware proposta 2.

5.2 Software utilizado

Os softwares utilizados na máquina de captura, de hardware e sistema operacional configurados conforme a seção anterior, são:

- IPTraf, para captura do tráfego da rede.
- Perl, para rodar os scripts que trabalham com os arquivos de *log*.
- Java, para rodar programas que trabalham junto aos scripts em Perl.
- Crontab, para executar os scripts em tempos agendados.

O *crontab* é um serviço que executa programas em períodos programados. Os *scripts* escritos em Perl são disparados a partir do *crontab* de acordo com a arquitetura a seguir:

Na Figura 5.3, vemos que o nosso sistema divide-se em 3 partes: aquisição de dados da rede (fundo verde), processamento e análise dos dados (fundo azul) e atuação, que altera os mecanismos responsáveis pelo roteamento (fundo cinza).

Os scripts em Perl (em vermelho) são executados pelo crontab em horas determinadas, ou por outros scripts, após as suas execuções. Eles geram arquivos (em azul), geralmente de log, que são entradas para outros scripts conforme as setas. Alguns programas (em verde) são chamados por estes scripts. Explicaremos todo o processo com mais detalhes a seguir.

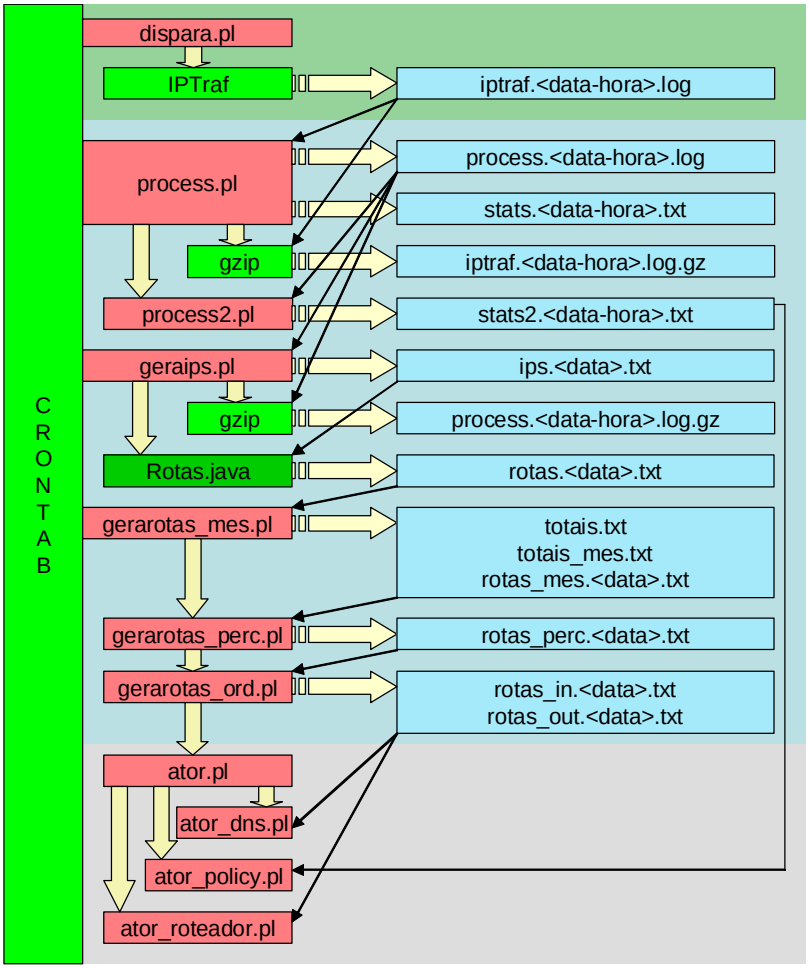


Figura 5.3: Solução de software poposta

5.2.1 Aquisição de dados sobre o tráfego na Rede

O seguinte *script* em Perl dispara o processo de aquisição de dados sobre o tráfego na rede, executando o IPTraf com parâmetros apropriados.

dispara.pl

É o *script* responsável por executar o IPTraf. É executado a cada hora, nas horas exatas, pelo crontab. Cada execução do IPTraf dá origem a um arquivo de *log*. Na chamada do IPTraf, o *script* passa os seguintes parâmetros:

- **-i all**, que indica ao IPTraf que ele capturar pacotes de todas as interfaces existentes.

- **-B**, que indica ao IPTraf que seu processo deve ser executado em *background*.
- **-t 59**, que indica ao IPTraf que ele deve ser executado por 59 minutos. Damos um minuto de folga, para não corrermos o risco do IPTraf estar rodando quando o crontab executasse o script novamente, o que ocasionaria erro.
- **-L *$arquivo***, que indica que a captura deve ser armazenada no arquivo de *log* indicado na variável $arquivo. Esta variável contém o nome do arquivo de *log* no seguinte formato: "**iptraf.*aaaa-mm-dd.hh*.log**", onde *aaaa-mm-dd.hh* são a data e hora corrente.

A linha do *crontab* que executa o script "*dispara.pl*" é a seguinte: "0 * * * * /*path*/dispara.pl", onde *path* é o caminho par o script. Esta linha indica que o script deve ser executado em horas exatas (quando o minuto for 0). Os arquivos de *log* gerados a cada hora pelo IPTraf variam de 4MB a 25MB.

5.2.2 Processamento e Análise de dados sobre o tráfego na Rede

Os seguintes *scripts* em Perl são responsáveis por pré-processar os dados brutos adquiridos no passo anterior, e analisá-los para transformá-los em informação de entrada para os mecanismos que controlam o fluxo do tráfego da rede TCP/IP.

process.pl

Este *script* executa uma série de processamentos sobre os dados adquiridos pelo IPTraf. Esses processamentos são iniciados 10 minutos após o arquivo de *log* gerado pelo IPTraf ser fechado. Os processamentos são de análise de dados, de geração novos *logs* eliminando dados desnecessários e de compactação dos arquivos de *log* "brutos".

A análise feita por este *script* nos informa a quantidade de pacotes e bytes por protocolo IP (TCP, UDP, ICMP, ...) trafegado. Arquivos de nome: "**stats.*aaaa-mm-dd.hh*.txt**" são gerados com estas informações.

Outra função deste *script* é remover informações desnecessárias existentes no *log* gerado pelo IPTraf. Tais informações podem ser os dados capturados pela interface local, dados de protocolos irrelevantes, entre outras. Arquivos de nome "**process.*aaaa-mm-dd.hh*.log**" são gerados com as informações filtradas.

Ao final do processamento, o arquivo de *log* gerado pelo IPTraf, que não será mais necessário, é compactado para economia de espaço em disco.

A linha do crontab que executa o script "*process.pl*" é a seguinte: "10 * * * * /*path*/process.pl", onde *path* é o caminho par o script. Esta linha indica que o script deve ser executado cada hora aos 10 minutos.

O arquivo de *log* gerado apenas com o tráfego TCP é cerca de 80% menor que o original gerado pelo IPTraf. A compactação utilizando o **gzip** gera arquivos 90% menores.

Ao final de sua execução, outro script é executado: o "*process2.pl*".

process2.pl

O *script* "*process2.pl*" classifica o tráfego pré-processado, dividindo o tráfego de entrada e o tráfego de saída, de acordo com os serviços TCP trafegados. Os serviços mais relevantes são agrupados, enquanto os demais são classificados em um grupo a parte (outros).

Arquivos de nome "**stats2.*aaaa-mm-dd.hh*.txt**" contendo as informações sobre os serviços TCP por hora são gerados.

geraips.pl

Este *script* é executado diariamente, ao contrário dos outros dois que rodam a cada hora. Ele dá uma folga de 20 minutos para o script "*process2.pl*", que é executado as 00:10, e as 00:30 executa as seguintes tarefas:

- ler todos os arquivos de tráfego TCP/IP do dia anterior (process.*.log, gerados a cada hora) e acumula o tráfego de entrada e de saída dos *hosts* externos que se comunicaram com o *site* em questão. Arquivos no formato "**ips.*aaaa-mm-dd*.txt**" são gerados listando os endereços IP e seus respectivos tráfegos de entrada e de saída.
- Compactar os arquivos "**process.*aaaa-mm-dd.hh*.log**" com os dados do tráfego TCP/IP a cada hora. Os arquivos são reduzidos em 90% com a compactação utilizando o software **gzip**.

- Executar um programa em Java (**Rotas.java**), que lê o arquivo "**ip.*aaaa-mm-dd*.txt**" e agrupa os endereços IP em faixas de máscara entre /16 e /24, somando o tráfego de todos os endereços pertencentes às faixas. Tal procedimento gera arquivos com o nome "**rotas.*aaaa-mm-dd*.txt**" contendo os endereços das faixas, a máscara utilizada e os respectivos tráfegos de entrada e de saída.
- Por fim, executar o *script* "**geraexcel.pl**" que organiza os dados gerados num arquivo facilmente importado pelo "Microsoft Excel" (opcional para análise).

gerarotas_mes.pl

Este *script*, executado uma vez ao dia, é responsável por, a partir do tráfego de cada faixa, contidas nos arquivos "**rotas.*aaaa-mm-dd*.txt**", gerar uma previsão do tráfego do dia seguinte. Poderíamos substituí-lo por um programa inteligente para adquirir melhores resultados.

Da forma que concebemos, ele trabalha utilizando o tráfego de um período para prever o de outro período no futuro. Utilizaremos o período de um mês para prever o tráfego de um dia, conforme ilustramos em nossos testes acima.

A primeira tarefa é calcular o total de tráfego de entrada e de saída por dia. Para obtermos esta informação, soma-se os tráfegos de todas as faixas existentes no arquivo "**rotas.*aaaa-mm-dd*.txt**". Para cada dia, uma linha é adicionada com os totais de entrada e de saída no arquivo "**totais.txt**" gerado.

Em seguida, precisaremos dos totais por mês. Criamos o arquivo "**totais_mes.txt**" acumulando os tráfegos totais dos últimos 30 dias.

A próxima tarefa, então, é calcular o tráfego de cada faixa no período utilizado, no nosso caso: 30 (trinta) dias. O *script* percorre os arquivos "**rotas.*aaaa-mm-dd*.txt**" dos últimos 30 dias e acumula os tráfegos de entrada e de saída de todas as faixas. No final, é gerado o arquivo "**rotas_mes.*aaaa-mm-dd*.txt**" com o tráfego de 30 dias até o dia indicado no nome do arquivo.

O *script* chama então outro *script*: o **gerarotas_perc.pl**.

gerarotas_perc.pl

Este *script* é responsável por calcular os percentuais de cada faixa no período utilizado (30 dias, no nosso caso). Ele utiliza os dados dos arquivos "**totais_mes.txt**" e "**rotas_mes.*aaaa-mm-dd*.txt**" para isso.

É gerado então o arquivo "**rotas_perc.*aaaa-mm-dd*.txt**" com os endereços IP das faixas, e seus percentuais de entrada e de saída associados ao mês em questão.

Em seguida, chamamos o *script* **gerarotas_ord.pl**.

gerarotas_ord.pl

O objetivo deste *script* é ordenar os dados do arquivo "**rotas_perc.*aaaa-mm-dd*.txt**" em ordem decrescente, ou seja, as primeiras linhas contém as faixas de maior tráfego da rede. É também calculado o tráfego percentual acumulado. Estes cálculos são efetuados separadamente para entrada e saída, gerando os arquivos "**rotas_in.*aaaa-mm-dd*.txt**" e "**rotas_out.*aaaa-mm-dd*.txt**" respectivamente.

A partir deste arquivo com as faixas ordenadas pelo tráfego do mês é que balancearemos a carga entre as conexões existentes. Para isso, chamamos o *script* **ator.pl**.

5.2.3 Atuação no Roteamento de Tráfego da Rede

ator.pl

Este *script* tem a utilidade de chamar os *scripts* que atuarão junto aos mecanismos de roteamento do tráfego da rede: o DNS, o Roteamento Estático e o Roteamento por Políticas. Ele é executado 1 vez ao dia, após a finalização do *script* **gerarotas_ord.pl.**

Estes *scripts* não chegaram a ser implementados, mas podem ser facilmente construídos utilizando *sockets* ou um modelo de sistema distribuído utilizando agentes nas máquinas que executam os serviços de roteamento e DNS.

ator_dns.pl

O *script* **ator_dns.pl** se comunica com um agente que se localiza na máquina onde

funciona o serviço de DNS principal, isto é, o que é consultado pelos *hosts* remotos ao *site*.

O *script*, baseado nas informações do arquivo "**rotas_in.*aaaa-mm-dd*.txt**" ou "**rotas_out.*aaaa-mm-dd*.txt**", dependendo de qual seja o principal tráfego do *site*, balanceia o tráfego entre as conexões e gera uma nova tabela de DNS com as novas informações de balanceamento. Finalmente, a tabela é enviada ao agente que substitui a tabela corrente e reinicia o serviço de nomes.

O servidor de DNS deve possuir mecanismo de responder de diferentes maneiras as consultas dependendo do endereço do *host* remoto. Verificamos que os servidores ***bind*** (a partir da versão 9) e ***djbdns*** possuem tal requisito.

ator_roteador.pl

O *script* **ator_roteador.pl** se comunica com a máquina que faz o roteamento entre os provedores de acesso à Internet. Esta máquina pode ser um roteador, uma firewall ou uma simples máquina que faça o roteamento.

A comunicação pode ser realizada através de um agente localizado na máquina, ou via protocolo *telnet* caso a máquina seja um roteador. Neste caso, o *script* simulará o acesso de uma pessoa.

Então, baseado nas informações do arquivo "**rotas_in.*aaaa-mm-dd*.txt**" ou "**rotas_out.*aaaa-mm-dd*.txt**", dependendo de qual seja o principal tráfego do *site*, o *script* balanceia o tráfego entre as conexões e gera uma nova tabela de roteamento. Então ele envia esta nova tabela para o agente, que a atualiza imediatamente, ou atualiza via *telnet* no caso de um roteador. A partir daí o novo roteamento já entra em vigor, funcionando com o novo balanceamento.

ator_policy.pl

O *script* **ator_policy.pl** é utilizado no caso do roteamento ser feito por um roteador que possua o recurso de "Roteamento por Políticas".

Nesse caso, o *script* analisa as informações dos arquivos "**stats2.*aaaa-mm-dd.hh*.log**" do dia e balanceia o tráfego entre as conexões existentes baseado nos protocolos TCP/IP. Em seguida o *script* se comunica (via *telnet*) com o roteador e altera a tabela de

roteamento por políticas de acordo com o balanceamento desejado, simulando uma pessoa.

Com a tabela alterada, o roteador automaticamente passará a rotear seguindo as modificações realizadas.

Capítulo 6

Conclusões e Trabalhos Futuros

Neste capítulo, concluímos nossos estudos e propomos novos trabalhos baseados em nossa pesquisa.

6.1 Conclusões

Nesta dissertação estudamos formas de gerenciar *sites multi-homed*, ou seja, que possuem conexões redundantes de acesso à Internet, e não têm a possibilidade de utilizar protocolos de roteamento dinâmico, que permite a utilização das conexões da maneira mais eficiente.

Estudamos o estado da arte sobre *sites multi-homed*, incluindo *sites* autônomos (AS) e não-autônomos, o foco de nosso problema. Verificamos neste estudo que não existe solução para otimização de roteamento. O sistema utilizado na maioria das empresas é, então, criar regras estáticas de roteamento para utilização das conexões.

Estudamos como otimizar o uso das conexões através da medição do tráfego da rede, e uso de mecanismos que fazem o roteamento. Para medir o tráfego, usamos programas que "monitoram" o tráfego da rede, nos fornecendo informações sobre o protocolo de rede utilizado, o serviço utilizado e as quantidades de pacotes e de bytes trafegados. Para alterar a rota do tráfego através das conexões propomos o uso de uma combinação de ferramentas do sistema: o roteamento estático, o serviço de nomes (DNS), roteamento por políticas e a tradução de endereços (NAT).

Na tabela abaixo comparamos a solução que utiliza BGP em Sistemas Autônomos (AS), com a solução proposta nesta dissertação para *sites multi-homed* em geral.

Otimizações Possíveis	**Solução utilizando BGP (apenas para AS)**	**Solução proposta para *sites multi-homed***
Redundância	SIM	SIM
Escolha da conexão de saída pela proximidade	SIM	SIM
Escolha da conexão de saída e entrada pelo tráfego / congestionamento, ou pela proximidade	NÃO	SIM

Tabela 6.1: Comparação da solução utilizando BGP/AS com a solução proposta

Em ambas as soluções, temos redundância de conectividade, de forma que se uma das conexões caírem, a contingência é feita de forma automática. Também temos nas duas soluções a possibilidade de utilizar a proximidade como fator de escolha da conexão de saída. Contudo, apenas na solução proposta nesta dissertação podemos utilizar outros critérios como: a largura de banda (reduzindo-se o congestionamento), os serviços trafegados e a própria proximidade, e, pode-se escolher tanto a conexão de entrada quanto a de saída.

Como estudo de caso, usamos o *site* do Centro de Informática (CIn) da Universidade Federal de Pernambuco (UFPE). Monitoramos o tráfego deste *site* durante vários meses e baseado neste tráfego, estudamos formas para dividi-lo entre as conexões de forma regular. Estudamos também uma forma de rotear o tráfego entre as conexões, baseando-se na banda consumida pelos serviços utilizados na rede.

Comparamos os resultados do uso do sistema com as atuais regras de roteamento estático e percebemos que haveria um uso bem mais equilibrado entre as conexões utilizando os dados de nossa medição. Também descobrimos com a medição, protocolos de transferência de arquivos como “filmes”, “músicas”, que podem ser roteados por conexões mais lentas como forma de desestimular seus usos.

6.2 Trabalhos Futuros

Propomos, como extensão a esta pesquisa, alguns temas para novas pesquisas sobre *sites multi-homed*.

- Estudo do uso de métodos estatísticos e inteligentes para maior acerto na previsão da divisão de tráfego entre as conexões.
- Finalização na implementação dos agentes “atores” que alimentam os mecanismos de desvio de tráfego, possibilitando a utilização do sistema num *site multi-homed* real.
- Avaliar como as técnicas apresentadas poderiam ser usadas dentro de um contexto de um *site* que seja um Sistema Autônomo, já que, neste caso, o *site* teria automaticamente resolvida a questão de tolerância a falhas, mas não as questões relativas a balanceamento de trafego de entrada e de saída em relação aos critérios descritos neste trabalho.
- Estudar formas de se efetuar o balanceamento de carga em *sites* não-autônomos baseado em proximidade para os *hosts* de forma semelhante ao comportamento do BGP.

Referências Bibliográficas

[01] Aamodt, A.; Plaza, E.

Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches, 1994

[02] Akkiraju, P.; Delgadillo, K.; Rekhter, Y.

White Paper – Enabling Enterprise Multihoming with Cisco IOS Network Address Translation (NAT), 2002

http://www.cisco.com/warp/public/cc/pd/iosw/ioft/ionetn/tech/emios_wp.htm

[03] Baccala, Brent

Connected: An Internet Encyclopedia, Third Edition, 1997

http://www.freesoft.org/CIE/

[04] Bates, T.; Rekhter, Y.

RFC2260 – "Scalable Support for Multi-homed Multi-provider Connectivity", 1998

http://www.faqs.org/rfcs/rfc2260.html

[05] Bernstein, D. J.

Site do djbdns

http://cr.yp.to/djbdns.html

[06] Braun, H-W.

RFC 1104 – "Models of Policy Based Routing", 1989

http://rfc.sunsite.dk/rfc/rfc1104.html

[07] Deri, L.

nTOP – network top

http://www.ntop.org/

[08] Egevang, K.; Francis, P.

RFC1631 - "The IP Network Address Translator (NAT)", 1994

http://www.faqs.org/rfcs/rfc1631.html

[09] Hawkinson, J.

RFC 1930 - ”Guidelines for creation, selection, and registration of an Autonomous System (AS)”, 1996

http://www.faqs.org/rfcs/rfc1930.html

[10] Internet Software Consortium

BIND 9 Administrator Reference Manual, 2001

http://www.isc.org/products/BIND/bind9.html

http://www.nominum.com/content/documents/bind9arm.pdf

[11] Jain, R.

The Art of Computer Systems Performance Analysis

Techniques for Experimental Design, Measurement, Simulation, and Modeling, Wiley, 1991

[12] Java, G. P.

IPTraf – IP Network Monitoring Software

http://iptraf.seul.org/

[13] Kaelbling, L. P.; Littman, M. L.

Reinforcement Learning: A Survey

Journal of Artificial Intelligence Research 4 (1996) 237-285

[14] Kewl Homepage

Policy Routing with Linux and Cisco IOS

http://kewl.phear.org/policy/index.html.en

[15] Leake, D. B.

CBR in Context: The Present and Future, 1996

[16] Marsh, M. G.

Policy Routing with Linux - Online Edition,

http://www.policyrouting.org/PolicyRoutingBook/ONLINE/TOC.html

[17] Rekhter, Y.; Li, T.
RFC 1771 – “A Border Gateway Protocol 4 (BGP-4)”, 1995
http://www.ietf.org/rfc/rfc1771.txt

[18] Rekhter, Y.; Moskowitz, B.; Karrenberg, D.; Groot, G. J. de; Lear, E.
RFC 1918 – “Address Allocation for Private Internets”, 1996
http://www.faqs.org/rfcs/rfc1918.html

[19] Reynolds, J.; Postel, J.
RFC 1700 – “Assigned Numbers”, 1994
http://www.faqs.org/rfcs/rfc1700.html

[20] Russel, Stuart J.; Norvig, Peter
Artificial Intelligence: a modern approach, Prentice-Hall, 1995

[21] Tanembaum, Andrew S.
Computer Networks, 3rd ed., Prentice-Hall, 1996

[22] ARIN
http://www.arin.net

[23] Configuração do roteador CIn/CESAR

[24] IANA
http://www.iana.org

[25] ISC Mailing List Archive
http://www.isc.org/ml-archives/

[26] Product Bulletin – No. 1195: Cisco IOS Network Address Translation (NAT), 2001
http://www.cisco.com/warp/public/cc/pd/iosw/ioft/ionetn/prodlit/1195_pp.htm

[27] Registro.Br
http://registro.br

[28] Understanding Policy Routing
http://www.cisco.com/warp/public/105/36.html