ecai

Universidade Federal de Santa Catarina
Centro Tecnológico
Curso de Engenharia de Controle e
Automação Industrial

ufsc

Estudo de protocolos de roteamento e implementação do protocolo RIP destinado à elaboração de um IPSwitch

Monografia submetida à Universidade Federal de Santa Catarina

como requisito para a aprovação da disciplina:

DAS 5511: Projeto de Fim de Curso

Alexandre Emanoel Camargo

Florianópolis, Outubro de 1998

Estudo de protocolos de roteamento e implementação do protocolo RIP destinado à elaboração de um IPSwitch

Alexandre Emanoel Camargo

Esta monografia foi julgada no contexto da disciplina
EEL 5901: Projeto de Fim de Curso
e aprovada na sua forma final pelo
Curso de Engenharia de Controle e Automação Industrial

Banca Examinadora:

Eng. Pedro Paulo da Silva
Orientador Empresa

Prof. Dr. Marcelo Ricardo Stemmer
Orientador do Curso

Prof. Augusto Humberto Bruciapaglia
Responsável pela disciplina e Coordenador do Curso

Prof. Roberto Wilrich, Avaliador

Clóvis Fernandes Júnior, Debatedor

Flávio Luis César Costa, Debatedor

Agradecimentos

À família, cujo apoio através de conselhos e palavras de incentivo foi fundamental no cumprimento de mais esta etapa da vida.

Aos colegas que, unidos pela amizade e também pelas inúmeras noites de estudos ao longo destes cinco anos, foram de grande apoio nos momentos mais difíceis longe da família.

À CIANET Ind. & Com. que, através de um ambiente de trabalho fértil para o desenvolvimento de novas idéias, ofereceu a oportunidade de aprendizado no que diz respeito à realização de projetos em grupo, criatividade e dinamismo.

Aos orientadores Eng. Pedro Paulo da Silva e Prof. Dr. Marcelo Ricardo Stemmer pelo capacidade em conduzir de modo simples e completo o projeto aqui realizado.

Resumo

O presente trabalho tem por objetivo o estudo de protocolos de roteamento dinâmico e a implementação do protocolo RIP - *Routing Information Protocol*. Tal protocolo é parte do projeto de desenvolvimento de um *IPSwitch* pela CIANET.

O *IPSwitch* é um dispositivo de conectividade baseado na recente tecnologia *IPSwitching*, que também é assunto deste trabalho.

O trabalho realizado baseou-se na implementação e testes do protocolo RIP e na simulação em *software* de um *IPSwitch* (aqui denominado *Switch de Nível 3*), usando a linguagem de programação C.

Deste modo, o projeto realizado na empresa CIANET Ind. & Com. pode ser divido em quatro fases:

- ✓ Estudo e escolha do melhor método de implementação da tecnologia *IPSwitching*;
- ✓ Estudo e escolha do protocolo de roteamento dinâmico que melhor se adapta à linha de produtos da empresa;
- ✓ Integração com a pilha de protocolos UDP/IP implementado pela equipe de *software* da empresa;
- ✓ Validação do projeto.

Abstract

This work has as its main goal the study of dynamic routing protocols and the implementation of the RIP - Rounting Information Protocol. This protocol is just a part of the project of development of a CIANET's IPSwitch.

The IPSwitch is a connectivity device based on the recent IPSwitching technology, which is also discussed in this work.

This work is based on the implementation and tests of the RIP protocol and simulation in software of a IPSwitch (here named *Switch de Nível 3*), using the C programming language.

In this way, the project made in the CIANET Ind. & Com. enterprise can be divided in four phases:

- ✓ Study and choice of the best implementation method of the IPSwitching technology;
- ✓ Study and choice of the dynamic routing protocol, which suits better to the line of products of the enterprise;
- ✓ Integration with the protocol stack UDP/IP implemented by the software team of the enterprise;
- ✓ Validation of the project.

Sumário

Lista de Figuras

Capítulo I: Introdução

O assunto abordado nesta monografia diz respeito ao estudo da tecnologia *IPSwitching* e de protocolos de roteamento dinâmico. Além disso, o projeto envolve a implementação e validação do protocolo RIP – *Routing Information Protocol*, um dos mais populares protocolos de roteamento dinâmico para redes corporativas. O projeto tem como finalidade a construção de um *IPSwitch* pela CIANET Ind. & Com.

A CIANET Ind. & Com. é uma empresa criada em 1994 por três estudantes de graduação da UFSC com o objetivo de aplicar uma arquitetura de barramento própria patenteada no Brasil e Estados Unidos. As vantagens desta arquitetura foram comprovadas através do desenvolvimento de um *Switch Ethernet* de 10/100 Mb/s, que apresenta um alto desempenho e um custo por porta extremamente competitivo a nível mundial. Este *Switch Ethernet* possui também gerenciamento SNMP que, a exemplo da pilha de protocolos UDP/IP, foi inteiramente desenvolvido pela equipe de *software* da CIANET.

A tecnologia *IPSwitching* surgiu para suprir a crescente demanda de elementos roteadores que apresentem um alto desempenho com alta confiabilidade e a um custo mais baixo. Esta demanda é resultante do crescente aumento no número de usuários em redes corporativas e também da chegada das *Intranets* a aplicativos multimídia (ex: teleconferência), que incrementam o tráfego destas redes.

O aumento no tráfego das redes solicita o desempenho dos roteadores ao extremo. Assim surge a necessidade dos elementos roteadores de alto desempenho evitando-se a criação de gargalos no sistema.

Por outro lado, com a popularização dos sistemas computacionais interligados através de rede, surge a necessidade de elementos roteadores de baixo custo para atenderem instalações de pequeno porte. Além disso, com o aumento da importância da informação, o sistema deve permanecer confiável.

Este tema localiza-se inteiramente dentro do contexto do Curso de Engenharia de Controle e Automação Industrial, mais especificamente na área de Informática Industrial. O estudo de tecnologias emergentes voltadas para redes de computadores é de grande importância para a automação, tanto industrial quanto comercial. Esta importância diz respeito à troca rápida e confiável de informações.

A escolha deste tema é resultado da constante procura mundial por técnicas que aumentam a velocidade de comunicação de dados em redes de computadores. Sendo uma das técnicas que apresentam a melhor relação CUSTO X BENEFÍCIO, a tecnologia *IPSwitching* adapta-se perfeitamente à linha de *switches* produzida pela CIANET Ind. & Com.

No desenvolvimento das funções de roteamento associadas ao protocolo IP, que constituem a parte essencial de um *IPSwitch*, buscou-se a implementação de um sistema dinâmico de troca de informações de roteamento. Tal sistema dinâmico é baseado em protocolos de troca de informações de roteamento, tais como RIP, OSPF, HELLO, etc.

Neste trabalho optou-se pela implementação do RIP (*Rounting Information Protocol*) que é o protocolo mais tradicional e, também, o que apresenta a melhor relação CUSTO X DESEMPENHO além de um tempo de desenvolvimento relativamente curto.

O trabalho baseou-se na pesquisa em livros, revistas do gênero (área de redes de computadores), *Internet* (rede mundial de computadores), listas de discussões e palestras. Além de contar com a experiência na área de comunicação de dados do grupo de engenheiros que compõem a empresa.

Esta monografia é dividida de modo a seguir uma seqüência lógica de aprendizado e também dos trabalhos realizados neste período. Assim, o capítulo II dá uma noção geral sobre redes de computadores, abordando assuntos como *frames*, endereçamento de computadores e interconexão de redes. O capítulo III explica o que vem a ser a tecnologia *IPSwitching* e o *IPSwitch*. O capítulo IV explica como o projeto de um sistema de comunicação de dados é feito, dando noções sobre camadas ou níveis de protocolos e o modelo conceitual utilizado neste projeto. O capítulo V faz uma descrição da pilha de protocolos UDP/IP e dos protocolos de roteamento dinâmico estudados. O capítulo VI descreve o protocolo RIP, analisando-o em detalhes. O capítulo VII diz respeito à implementação e a simulação de um *IPSwitch* básico, descrevendo os testes de validação realizados. Finalmente no capítulo VIII é feita uma análise global de todo o projeto, resultando nas conclusões e perspectivas.

Capítulo II: Redes de Computadores

A diminuição do custo de obtenção associada ao aumento do número de programas destinados à solução de problemas cotidianos têm determinado a efetiva popularização dos computadores.

Anteriormente confinados aos ambientes empresariais de grande ou médio porte, hoje podem ser encontrados em pequenas empresas e, não menos freqüentemente, em residências.

Um conseqüência do aumento do número de máquinas, bem como do aumento do número de aplicações às quais os computadores estão associados é o surgimento da necessidade de interligação entre estes computadores.

Outra conseqüência é o aumento da importância e, consequentemente, do valor da informação guardada e trocada pelos computadores. Surge então a necessidade de se criar um sistema eficiente para a troca de informação entre computadores.

Com este propósito foram criadas as redes de computadores. Elas consistem da interligação de dois ou mais computadores que passam a operar de forma integrada criando um sistema multiusuário multiprocessado, caracterizado pela facilidade de operação, pela comodidade e, principalmente, pelo aumento da confiabilidade do sistema.

Nas últimas décadas, diversas tecnologias de rede surgiram para prover a comunicação de dados. Dentre elas, cita-se a *Ethernet*, o *Token Ring* e o *X25NET*. Cada uma seguindo padrões próprios de comunicação o que impedia a perfeita integração entre elas.

O conceito de uma rede mundial de computadores consiste em conciliar as diversas tecnologias de comunicação de dados, de modo que elas operem de forma integrada, trocando dados entre si, sem que o usuário ou o administrador da rede se preocupe com os detalhes de comunicação a nível físico [COMER 91a]. Um exemplo é a *Internet*, a rede mundial de computadores.

Projetistas de redes de computadores utilizam basicamente dois métodos para esconder os detalhes de rede: o uso de programas de aplicação para manusear redes heterogêneas ou o uso de um sistema operacional que disponha de ferramentas de alto nível (escondem os detalhes de interface com o nível físico) [COMER 91a].

Entretanto, para descrever as redes de computadores com mais precisão é necessário que sejam conhecidos alguns conceitos fundamentais, como *frame* (quadro) e endereçamento.

2.1) *Frame* (Quadro)

Por questões de limitação tecnológica e econômica, a informação que trafega em uma rede é transmitida de forma serial. Sendo assim, é necessário que sejam estabelecidas regras para que uma máquina consiga detectar o início e o fim de uma transmissão.

Frame ou quadro corresponde ao conjunto de dados contidos entre o início e o final de uma transmissão serial através de uma rede. Cada *frame* contém campos de informações padronizados para que seja possível organizá-los e classificá-los dentro de uma rede. O formato do *frame* varia de acordo com a tecnologia de comunicação de dados utilizada. No caso deste projeto, as redes se baseiam na tecnologia *Ethernet*. Deste modo, um *frame Ethernet* possui o formato mostrado na figura 2.1.

Preâmbulo	*End. Destino*	*End. Fonte*	*Tipo do Frame*	*Dados do Frame*	*CRC*
64 bits	*48 bits*	*48 bits*	*16 bits*	*368-12000 bits*	*32 bits*

Figura 2.1 - Frame Ethernet.

O campo PREÂMBULO refere-se ao início do *frame*. Os campos referentes aos ENDEREÇOS DESTINO e FONTE dizem respeito à máquina a qual o *frame* se destina e à máquina que o enviou respectivamente. O campo TIPO DO FRAME indica o tipo de dados contidos no campo DADOS DO FRAME. Neste projeto, os dados podem ser do tipo ARP ou IP, estes tipos serão explicados no capítulo V. O campo CRC (*Ciclic Redundance Check*) corresponde a um campo de checagem onde são determinados possíveis erros de transmissão. O *frame* possui um tamanho mínimo de 64 bytes e um tamanho máximo de 1518 bytes.

O trabalho com *frames*, ao invés de arquivos ou grandes mensagens, resulta em muitas vantagens. Uma dessas vantagens é a possibilidade de se tornar as funções de comunicação independentes do aplicativo que as utiliza, ou seja, a máquina manuseia o tráfego da rede sem precisar conhecer a aplicação que está em uso.

Outra vantagem está em manter a flexibilidade do sistema viabilizando a construção de protocolos de redes de propósito geral.

2.2) Endereçamento

O endereçamento de máquinas dentro de uma rede de computadores é uma forma de garantir que a informação chegue ao seu destino. Ele pode ser comparado com o endereço de uma casa e os *frames*, com as cartas contendo os endereços do remetente e do destinatário e o dados a serem enviados.

O endereço de uma interface de rede é definido pelo fabricante, sendo denominado **ENDEREÇO FÍSICO**. Assim, numa rede de computadores, como a da figura 2.2, cada computador possui uma interface de rede e consequentemente um endereço físico. A comunicação entre dois computadores se dá da seguinte maneira. Se A quer se comunicar com B, este coloca o endereço físico de B no *frame* e o envia pela rede. Então o computador B analisa o *frame* e verifica que ele é o destino correto. Se o computador C receber o mesmo *frame*, este será ignorado.

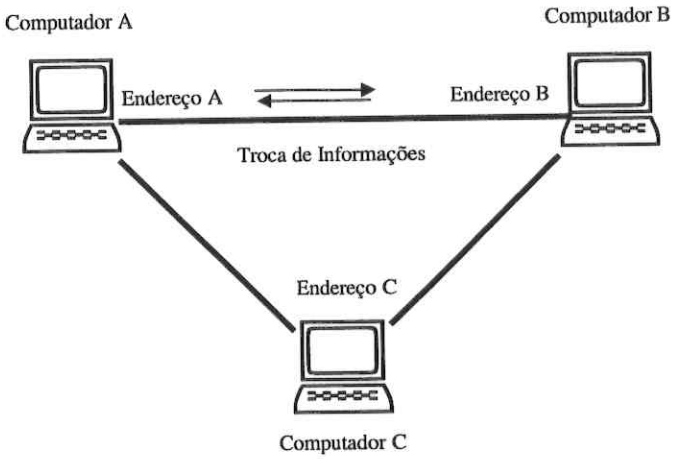


Figura 2.2 - Comunicação em uma rede de computadores.

O endereço físico varia de acordo com a tecnologia de rede. No caso da *Ethernet*, ele é composto por 48 bits ou 6 bytes.

Como o endereço físico depende da interface de rede, quando esta é mudada, ele também muda. Caso a tecnologia de rede mude, o formato do endereço físico também pode mudar. Esta falta de flexibilidade e compatibilidade é um empecilho ao conceito de uma rede mundial de computadores.

Para contornar esta falta de flexibilidade foram desenvolvidas soluções que definem sistemas de identificação alternativos independentes do endereço físico. Sendo assim, se uma máquina possui um identificador, a mudança da sua interface não irá alterar este identificador.

Os identificadores mais comumente utilizados são os **ENDEREÇOS IP (*IPAddress*)**. Tais endereços independem da tecnologia de rede adotada a não mudam quando uma interface de rede é trocada.

Os endereços IP são constituídos de 32 bits ou 4bytes e estruturados de tal modo a facilitar a formação de grupos de computadores denominados **REDES (*nets*)** e o roteamento da informação através destes grupos. Para que isso possa acontecer, um endereço IP se divide em duas partes: um identificador de rede (*netid*) e um identificador de máquina (*hostid*).

A partir desta divisão, os endereços IP são classificados em classes de acordo com o número de redes e máquinas dentro desta rede. A figura 2.3 mostra as classes dos endereços IP.

De acordo com a figura acima os roteadores R_1 e R_2 são os caminhos através dos quais a informação passa de uma rede para outra.

Os roteadores (*routers*) e *gateways* são dispositivos responsáveis pelo manuseio do tráfego e roteamento (definição do caminho) dos *frames* (informação) [ITD 98]. Ou seja, se um *frame* chegar no roteador R_1 e tiver como destino uma máquina na rede 3, então R_1 envia o *frame* para o roteador R_2, o qual serve como "porta" de entrada para a rede 3 e, consequentemente, para a máquina destino. O roteamento de pacotes é descrito mais detalhadamente no capítulo V.

Outra característica de uma *internetwork* é a existência de um *backbone* ou "espinha dorsal" da rede, onde as (sub-) redes são conectadas. O *backbone* é sempre um roteador ou um *gateway* e muitas vezes representam o núcleo da *internetwork* e a porta de saída para outras redes de computadores. Um exemplo de um *backbone* é mostrado na figura 2.5.

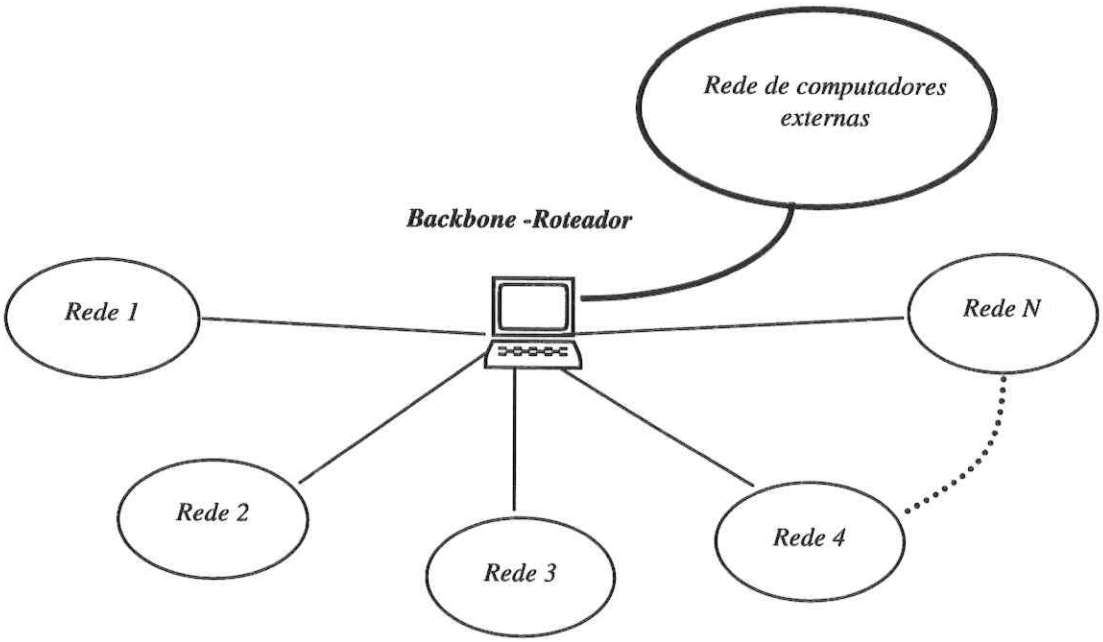


Figura 2.5 - Backbone.

Capítulo III: A Tecnologia *IPSwitching*

A tecnologia *IPSwitching* é uma solução inovadora para resolver um dos principais problemas presentes em redes corporativas: a baixa velocidade na troca de informações entre (sub-) redes.

Criado pela *Ipsilon Networks*, a tecnologia *IPSwitching* tem como filosofia unir em um só equipamento as principais características de duas classes de dispositivos de conectividade de rede, o *switch* e o roteador. Tais características são: a velocidade dos *switches* e a capacidade dos roteadores em rotear pacotes.

A seguir há uma breve descrição sobre os *switches* e roteadores.

3.1) O *Switch*

Switches são concentradores de rede, cuja principal função é fornecer um *link* dedicado de alta velocidade entre computadores ou segmentos de uma LAN [REDES 95], ao contrário dos *hubs*, que recebem um *frame* por uma porta e o enviam por todas as **n-1** portas restantes dividindo a banda de transmissão entre todos os segmentos. O *switches* estabelecem conexões dinâmicas exclusivas para cada par de máquinas (ou segmentos de rede) que desejam trocar informações [REDES 95]. Sendo assim, cada conexão estabelecida opera à mesma velocidade, como se as máquinas tivessem toda a banda dedicada à sua troca de informações.

Uma das principais vantagens dos *switches* é a sua baixa latência (tempo que um pacote de informações leva para passar através dele), uma vez que as "conexões" de rede são estabelecidas a nível físico. Deste modo, diz-se que os *switches* operam a uma velocidade *wire-speed*, ou seja, velocidade de transmissão a nível físico.

A garantia de uma baixa latência é importante para assegurar o tempo de resposta das aplicações através da rede. A latência na rede está relacionada com as aplicações cliente/servidor e as obrigações dos servidores. Para não ter que trocar as aplicações atuais, o processamento de *software* ou a base de dados, deve-se levar em conta uma ultrabaixa latência e o aproveitamento máximo da banda de transmissão. Em resumo, o desempenho *wire-speed* na rede significa tempos de resposta ótimos [ITD 98].

O *switches* acima descritos são também denominados *Switches de Nível 2*, pois eles trabalham a nível de camada 2 (*Link de Dados*) do modelo OSI. Assim, eles apenas transmitem a informação, não fazendo nenhuma alteração. A figura 3.1 mostra um esquema de um *Switch de Nível 2* e como a informação trafega por ele.

O modelo ISO/OSI diz respeito a um modelo de protocolos em camadas, composto de 7 camadas: Física, Enlace de Dados, Rede, Transporte, Sessão, Apresentação e Aplicação (ver apêndice B). Porém, este modelo não será abordado neste projeto. A definição de protocolos e camadas será melhor explicada no capítulo IV.

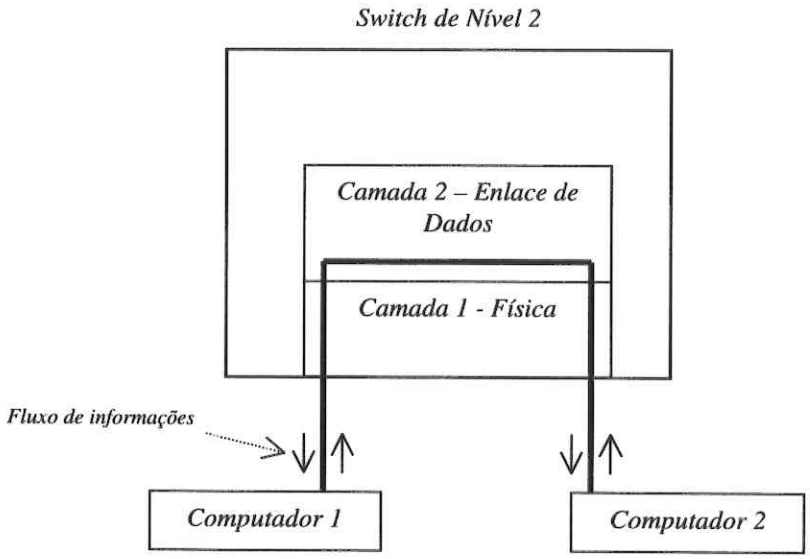


Figura 3.1 - Switch de Nível 2.

3.2) O Roteador

Os roteadores são concentradores de rede, que oferecem serviços de roteamento e conectividade confiáveis. Estes serviços são realizados a nível de *software* (camada de Rede – camada 3 do modelo OSI), resultando assim numa alta latência. Deste modo, os roteadores são geralmente usados em WANs (*Wide Area Networks*) e MANs (*Metropolitan Area Networks*), pois tais redes não exigem uma alta velocidade na troca de informações.

A figura 3.2 ilustra um esquema de um roteador e como a informação trafega por ele.

Por operar até a camada 3 do modelo OSI, os roteadores podem conectar tanto redes de mesma tecnologia, quanto redes de tecnologias diferentes. De acordo com a filosofia do modelo de protocolos em camadas, a camada 3 não interage com o meio físico, realizando assim apenas as tarefas a ela atribuídas.

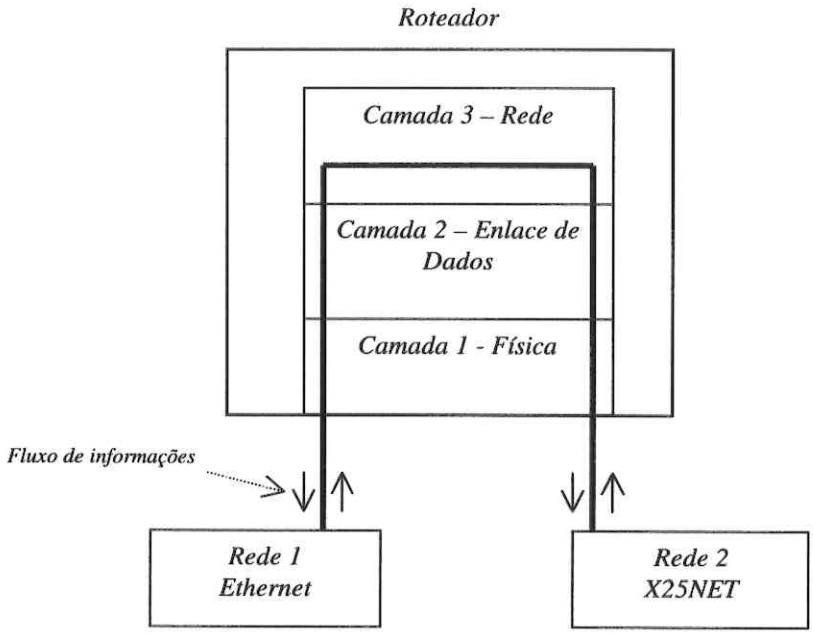


Figura 3.2 - Roteador.

Com o crescimento das redes de computadores, os problemas de congestionamento começaram a se agravar devido à intensificação da mensagens do tipo *broadcast* (direcionadas a todos os computadores de uma rede) e à utilização de aplicações mais "pesadas", como é o caso de aplicações multimídia e *intranets*.

Para reduzir a carga de informações em uma única rede, foi criada um esquema de sub-redes, que consiste em segmentar uma rede única em *LANs* menores.

Até há algum tempo, os administradores de rede não contavam com equipamentos específicos dedicados à conexão entre as diversas sub-redes. A única solução era o uso de roteadores para este fim.

Porém, devido à alta latência dos roteadores, estes entraram em conflito direto com a crescente demanda por aumento de velocidade, solicitada pelo aumento do tráfego nas redes corporativas.

Foi, portanto para resolver esse tipo de problema que surgiram os *IPSwitches*.

3.3) *IPSwitch*

O *IPSwitch* faz parte de uma nova geração de dispositivos de conectividade. Resultado da aplicação da tecnologia *IPSwitching*, o *IPSwitch* é basicamente a junção de um *Switch de Nível 2* e um roteador.

Existem muitas alternativas com diferentes nomes, como *routing switch, IP switch, L3 switch, fast IP, etc*, porém as características e vantagens imediatas são as mesmas: *wire-speed* em roteamento IP com protocolos padrões RIP, OSPF entre outras, suporte *Ethernet 10/100 Mb/s* (velocidade de transmissão), suporte a *Gigabit Ethernet*, desempenho entre 10 a 100 vezes superior aos roteadores convencionais [ITD 98].

Existem dois tipos de implementações básica.

1. ***Baseados em fluxo de informações***: através do primeiro pacote fecha-se um *link* dedicado, mantendo-se o fluxo através deste *link*. A figura 3.3, ilustra um *IPSwitch* baseado em fluxo de informações:

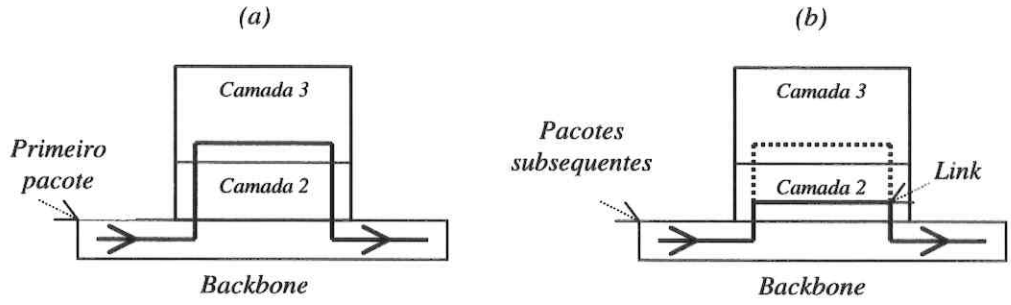


Figura 3.3 - IPSwitch baseado em fluxo de informações.

Neste caso, o *IPSwitch* roteia pela camada de rede o primeiro pacote e fecha um *link* entre as duas interfaces de rede. A partir de então os pacotes subsequentes com o mesmo destino não necessitam ser roteados novamente. Eles utilizam o *link*, para obter uma velocidade *wire-speed*. À medida que um novo pacote com um destino diferente chega, este *link* é desfeito, então o pacote é roteado e um novo *link* é gerado, repetindo assim o ciclo.

2. ***Baseado em nível físico***: neste caso, uma tabela é acessada a nível físico. Tal tabela contém informações sobre qual rota um específico pacote deve seguir. As informações contidas nesta tabela são fornecidas pelos protocolos de roteamento dinâmico, como por exemplo o RIP.

A utilização da segunda opção fornece maiores benefícios, tais como exigir menor processamento da CPU e menor latência, sendo a sua implementação uma excelente opção para *backbones* de alta velocidade (*Gigabit*).

Os *IPSwitches* são também denominados *Switches de Nível 3* pois utilizam funções de roteamento referentes à camada 3 (Rede) do modelo OSI. Neste projeto, o *IPSwitch* da CIANET será nomeado de *Switch de Nível 3*, e será descrito em detalhes no capítulo VII.

O objetivo de um *IPSwitch* é de ter ao menos melhores características de *thoughput* (quantidade de informações processadas) do que um roteador e uma latência de um *Switch* tradicional [ITD 98].

Em resumo, um *Switch de Nível 3* deve cumprir na rede local (LAN) funções de roteador com velocidade de *Switch de Nível 2* [ITD 98].

Capítulo IV: Protocolos de Rede [COMER 91a]

Como definido anteriormente, protocolo de rede é um conjunto de regras, que regem a comunicação entre duas máquinas. A seguir há uma descrição dos protocolos estudados para implementação deste trabalho.

4.1) A Pilha de Protocolos UDP/IP

Neste projeto foi utilizada a pilha de protocolos UDP/IP por duas razões principais: a equipe de *software* da CIANET já havia desenvolvido a pilha UDP/IP e esta faz parte de um padrão mundialmente conhecido e utilizado na grande maioria dos sistemas de transmissão de dados digitais através de redes do computadores.

Portanto, a pilha de protocolos UDP/IP foi estudada em função das razões acima apresentadas e, além disso, por constituir a base para que ocorra um roteamento dinâmico, uma vez que os protocolos de troca de mensagens de roteamento, incluindo o RIP, operam sobre este conjunto de protocolos.

A seguir são apresentados os protocolos que compõem a pilha UDP/IP.

4.1.1) ARP (*Address Resolution Protocol*)

Um dos objetivos do projeto de um sistema de comunicação é o de tornar os detalhes a nível físico transparentes ao usuário. Para tanto, uma máquina pode possuir um endereço IP qualquer, independente do endereço físico.

Porém, para que uma máquina possa enviar um pacote de informação através da camada física, ela deve conhecer o endereço físico da máquina destino (ver figura 2.1 - *Frame Ethernet*).

Sabendo que o endereço IP é conhecido a priori, como é possível associar o endereço IP com o endereço físico, mesmo quando o endereço IP muda, e além disso tornar o endereço físico transparente ao usuário? Esta pergunta corresponde ao ***problema de resolução de endereço***.

Uma maneira de resolver este problema é através do *ARP – Address Resolution Protocol*. Este protocolo disponibiliza um mecanismo que permite a uma máquina, que deseja se comunicar com outra, mas possui apenas o endereço IP dela, determinar o endereço físico da máquina destino.

O mecanismo é mostrado na figura 4.1.

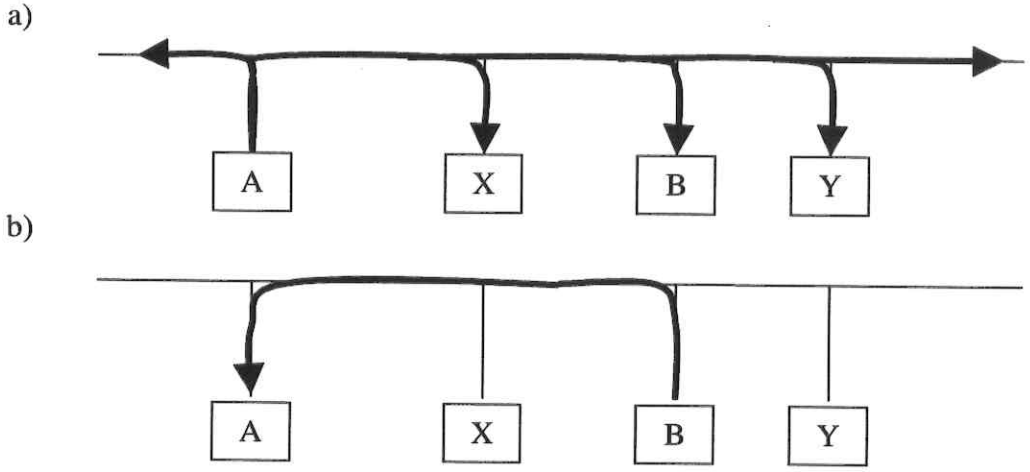


Figura 4.1 - Mecanismo do protocolo ARP.

O protocolo ARP oferece uma resolução de endereço dinâmica e simples. De acordo com a figura 4.1, quando a máquina A quer resolver* o endereço IP de B, ela envia pacotes de requisição, denominados *ARPrequests*, para todas as máquinas da rede (*broadcast*) e pergunta pelo endereço físico da máquina, cujo endereço IP é o endereço IP de B. Todas as máquinas recebem a pergunta, porém apenas a máquina B responde, enviando o seu endereço físico através de um pacote de resposta, denominado *ARPreply*.

O protocolo ARP possui uma tabela dinâmica, responsável pelo mapeamento entre o endereço IP e o endereço físico. Assim, toda vez que o protocolo recebe um pacote *ARPreply* ele armazena ambos os endereços na tabela.

Cada entrada da tabela possui um temporizador de *TIMEOUT*. No caso de o endereço IP de uma máquina mudar, após um período determinado de tempo, a entrada referente a este endereço IP é eliminada. O temporizador de uma entrada é resetado toda vez que um pacote ARP, referente a esta entrada, é recebido.

4.1.2) IP (*Internet Protocol*)

Todo usuário de rede imagina uma *internet* como sendo uma única rede virtual, que interconecta todas as máquinas, e através da qual a comunicação de dados é possível. A sua arquitetura a nível físico é irrelevante.

---

* Resolver, neste caso, significa mapear o endereço IP no endereço físico correspondente.

Conceitualmente uma rede TCP/IP provê três conjuntos de serviços como mostrado na figura 4.2.

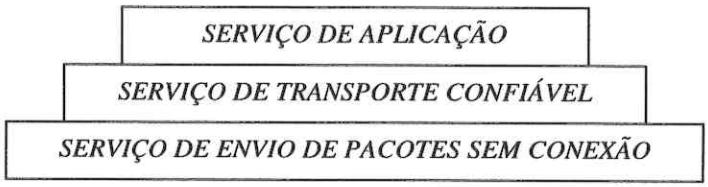


Figura 4.2 - Três camadas conceituais de serviços de uma rede TCP/IP.

Os serviços de envio de pacotes sem conexão correspondem aos serviços fundamentais de uma rede. Tecnicamente é definido como um sistema de envio de pacotes não confiável (*unreliable*), de melhor esforço (*best-effort*) e sem conexão (*connectionless*).

O sistema é não confiável, pois o pacote pode ser perdido, duplicado, atrasado ou enviado fora de ordem. O sistema não detecta tais incidentes. Ele é sem conexão, porque cada pacote é tratado de modo independente de todos os outros. E por fim, o sistema é de melhor-esforço, pois o programa de rede realiza uma tentativa confiante para enviar pacotes.

O protocolo IP fornece os serviços de envio de pacotes sem conexão, tornando-se assim a base de toda a grande rede virtual. O protocolo IP fornece três importantes definições:

1. protocolo IP define a unidade básica de transferência de dados através da rede TCP/IP. Esta unidade básica é denominada **datagrama**. Um datagrama é composto por um cabeçalho e um campo de dados. A figura 4.3 mostra um datagrama e como ele é encapsulado dentro de um *frame*.

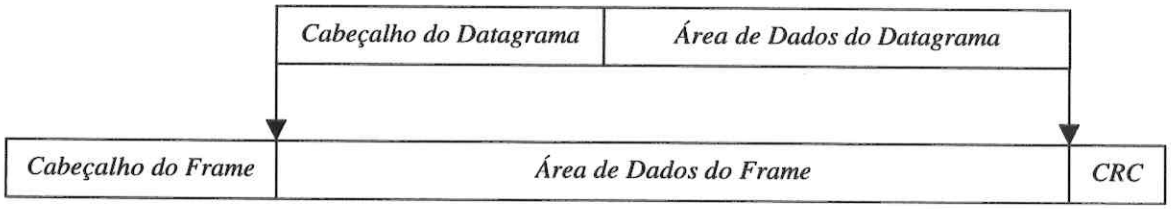


Figura 4.3 - Datagrama IP e como ele é encapsulado no campo de dados do frame.

2. O protocolo IP realiza a função de roteamento, escolhendo assim o caminho através do qual o pacote deve ser enviado.
3. O IP inclui um conjunto de regras, que caracterizam como os computadores, roteadores e *gateways* devem processar os pacotes, como e quando mensagens de erro devem ser geradas e em quais condições um pacote deve ser descartado.

Assim o IP se torna uma parte fundamental no projeto de uma rede TCP/IP, a qual às vezes é chamada rede de **tecnologia baseada em IP** (*IP-based tecnology*).

FRAGMENTAÇÃO DE PACOTES

O envio de pacotes IP possui como limitação a capacidade da rede em questão, ou seja, cada tecnologia de rede consegue transmitir um tamanho máximo de informação por pacote. Este tamanho é denominado MTU (*Maximum Transfer Unit*) – Unidade Máxima de Transferência. Por exemplo, para redes *Ethernet* o MTU é de 1518 bytes por *frame*, enquanto que em uma rede *proNET-10* o MTU é de 2044 bytes por *frame*. Como fazer então para enviar uma informação maior que o MTU da tecnologia de rede em questão?

A resposta está em uma das funções do protocolo IP, a **fragmentação**, que consiste em particionar a informação em **fragmentos** do tamanho do MTU e enviá-los pela rede. Junto aos pacotes seguem informações necessárias para que a máquina destino possa montá-lo corretamente.

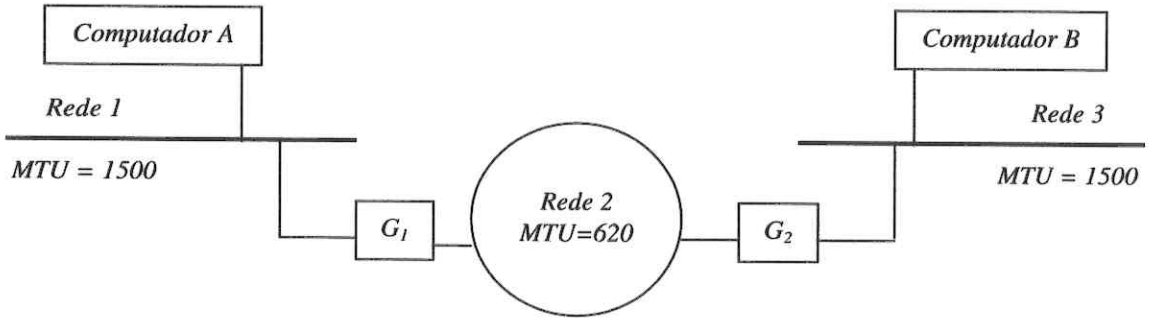


Figura 4.4 - Ambiente onde um pacote pode ser fragmentado.

A figura 4.4 mostra um ambiente onde seria necessário fragmentar datagramas. Para enviar um datagrama de tamanho 1400 bytes, do computador A para o computador B, o datagrama deverá passar pela Rede 2, cuja capacidade de transmissão (MTU = 620 bytes) é menor que o tamanho do datagrama. Assim sendo, o *gateway* G_1 deverá fragmentá-lo. O tamanho de cada fragmento deverá ser um múltiplo de 8, uma vez que o protocolo IP representa o *offset* da informação presente no pacote em múltiplos de oito (8) octetos. Assim o datagrama será dividido da seguinte maneira mostrado pela figura 4.5.

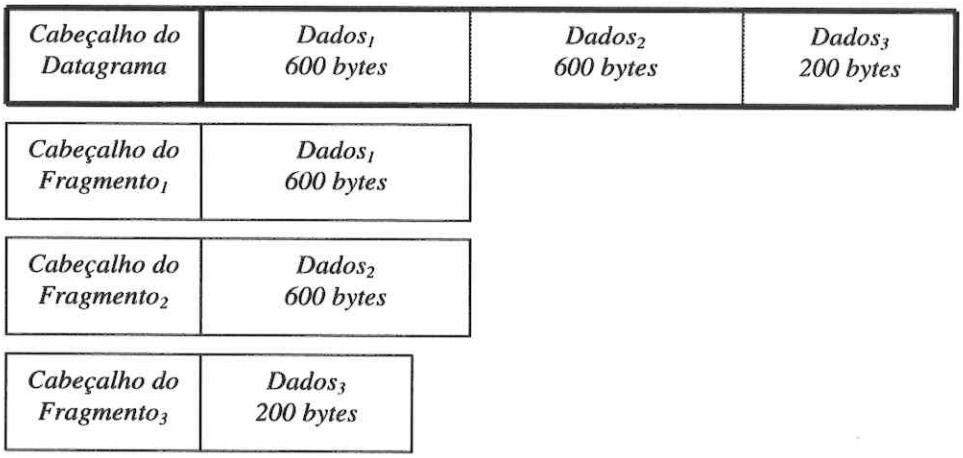


Figura 4.5 - Datagrama fragmentado.

Após passar pela rede 2, o *gateway* G_2 monta o pacote de acordo com as informações contidas no cabeçalho dos fragmentos e o envia para o computador B.

Caso um fragmento seja perdido ou por um erro de roteamento chegue na ordem errada, o *gateway* eliminará o pacote e enviará uma mensagem de erro à origem do pacote. Estas situações podem ocorrer devido às características dos serviços oferecidos pelo protocolo IP.

ROTEAMENTO DE PACOTES

A escolha do caminho que a informação deve seguir é uma das principais funções do protocolo IP. A máquina responsável por esta função se chama **roteador** (*router*).

É importante lembrar que o objetivo dos protocolos TCP/IP é o de prover uma rede virtual, onde é possível a comunicação de máquinas em redes físicas diferentes. O protocolo de roteamento IP deve escolher como enviar uma informação através de múltiplas redes físicas.

Para entender o roteamento IP, vamos primeiramente analisar a arquitetura de rede mostrada na figura 4.6. Todas as redes pertencem à mesma tecnologia, por exemplo *Ethernet*.

Uma máquina da rede 1 que necessite se comunicar com uma máquina da rede 5, deverá enviar a informação para o roteador R_1. Este deverá escolher o caminho certo, enviando assim a informação para o roteador R_2, que em seguida a enviará para a máquina da rede 5.

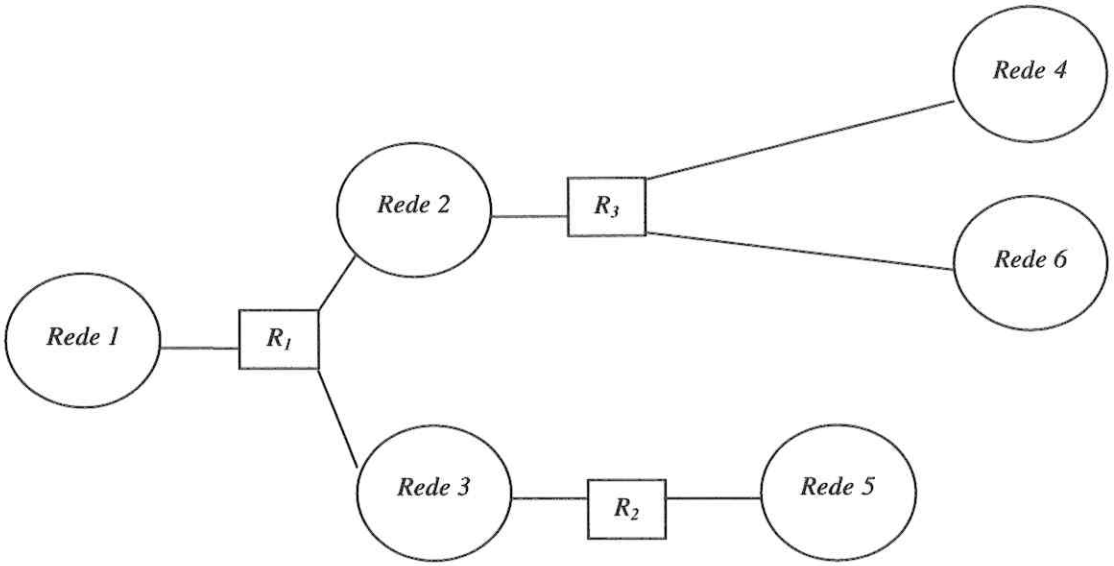


Figura 4.6 - Arquitetura de uma rede TCP/IP, onde pacotes são roteados pelos roteadores.

Porém o roteamento de pacotes não é tão simples assim. Primeiramente vamos dividir o roteamento em duas formas: **roteamento direto** e **roteamento indireto.**

Roteamento direto ocorre quando uma máquina quer se comunicar com outra máquina, estando ambas na mesma rede física. Desde modo, a fonte envia o pacote diretamente ao destino. Todas as máquinas em uma rede possuem a capacidade de roteamento direto.

No caso do roteamento indireto, as máquinas fonte e destino localizam-se em redes físicas diferentes. Assim a máquina fonte deve passar o pacote para o roteador, para assim ser enviado ao destino.

O Roteamento indireto é mais complexo que o direto, pois a máquina que for enviar um pacote deve saber para qual roteador deverá enviá-lo. Assim sendo, este tipo de roteamento é freqüentemente realizado pelos roteadores. Como exemplo, na figura 4.6 pega-se o roteador R_1, que deverá escolher entre os roteadores R_2 ou R_3; se quiser enviar um pacote para a rede 5 ou 6 respectivamente.

Conceitualmente, uma máquina, que não seja um roteador, não pode rotear pacotes. Porém, na prática, quando a máquina quer enviar um pacote, ela primeiramente verifica se o destino pertence à mesma rede física (roteamento direto), caso contrário, ela envia o pacote para um roteador padrão (roteamento indireto). O roteador se encarregará de escolher o caminho certo.

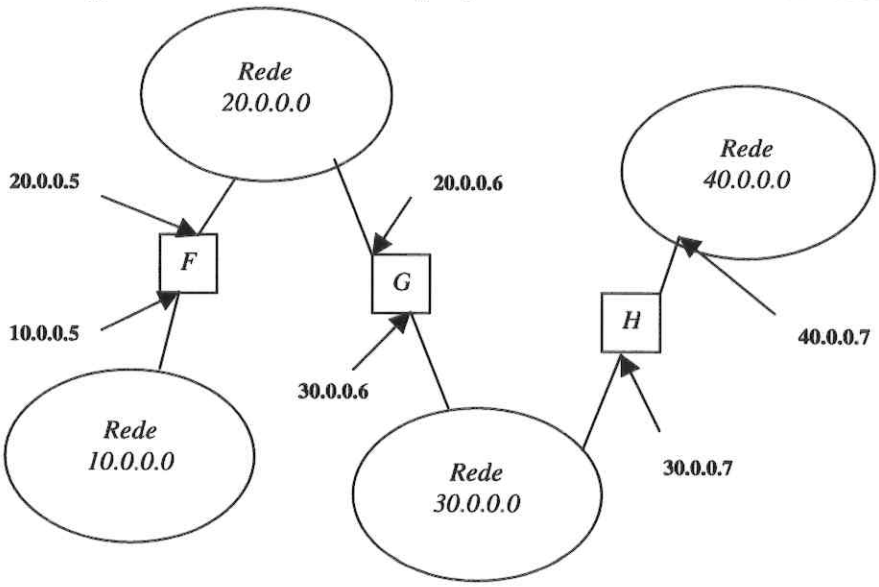
Tabela de Roteamento

O protocolo de roteamento IP se baseia em uma tabela para escolher o caminho correto por onde deverão trafegar os pacote destinados a uma máquina que não está na rede local. Esta tabela é denominada **Tabela de Roteamento de Rede**. Nesta tabela estão contidas as rotas a serem seguidas. Como existem milhões de endereços IP, armazená-los em uma tabela seria inviável, pois em pouco tempo não haveria mais memória disponível na máquina. Para evitar este inconveniente optou-se pelo uso do **princípio de ocultamento de informação** (*principle of information hiding*).

Este princípio permite às máquinas fazerem o roteamento com o mínimo de informação.

Como descrito anteriormente o endereço IP é composto por um identificador da rede (*netid*) e um identificador da máquina (*hostid*). Assim através do princípio de ocultamento de informação, o protocolo de roteamento IP armazena apenas o *netid* na tabela. Desde modo, a tabela de roteamento de rede é composta pelo par *(N, R)*, onde *N* corresponde ao endereço destino de rede (*net*), e *R* ao endereço do próximo roteador (*router*). A tabela torna-se assim consideravelmente menor.

A figura 4.7 mostra um exemplo prático do uso da tabela de roteamento de rede.



Para alcançar máquinas em redes a partir de G	Roteia-se os pacotes para o endereço IP
20.0.0.0	Envio direto
30.0.0.0	Envio direto
10.0.0.0	20.0.0.5
40.0.0.0	30.0.0.7

Figura 4.7 - Exemplo prático do uso da tabela de roteamento de rede do gateway G.

A tabela da figura 4.7 pertence ao roteador *G*. Pela tabela o roteador *G* pode alcançar diretamente qualquer máquina pertencente às redes 20.0.0.0 e 30.0.0.0, pois ele está conectado fisicamente a estas duas redes.

Pode-se observar que um roteador conectado a **n** redes diferentes possui **n** endereços IP associados a cada uma de suas interfaces.

Caso *G* tenha que enviar um pacote para a rede 10.0.0.0, ele terá que realizar o roteamento indireto, enviando o pacote para o endereço IP 20.0.0.5. Este endereço IP corresponde ao roteador F, que realizará o roteamento direto para a máquina destino.

A maioria dos roteadores possuem rotas padrão. Sendo assim, se a rota não for encontrada na tabela de roteamento, ele utiliza uma rota padrão. Esta técnica é muito usada quando uma rede possui um pequeno conjunto de endereços IP locais e uma única conexão para o resto da *Internet*.

Levando em conta tudo o que foi dito sobre roteamento, é possível fazer um algoritmo de roteamento IP.

Algoritmo de Roteamento IP

- Recebe o datagrama
- Retira do datagrama o identificador da rede (*netid*) E_{rede}
- Compara o E_{rede} com os *netids* da tabela de roteamento
- Se E_{rede} resulta em um roteamento direto
  então envia o datagrama diretamente à máquina destino
- Senão se E_{rede} resulta em roteamento indireto
  então envia o datagrama para o *gateway* especificado
- Senão se E_{rede} não é encontrado na tabela de roteamento
  então envia o datagrama para uma rota padrão
- Senão
  informa um erro de roteamento

A tabela de roteamento de rede pode ser preenchida manualmente, tornando-se assim estática, ou através de protocolos de roteamento dinâmico, como por exemplo o RIP – *Routing Information Protocol*, o qual monta a tabela dinamicamente. O RIP será melhor explicado no capítulo VI, constituindo-se a principal etapa deste projeto o seu estudo e implementação.

4.1.3) ICMP (*Internet Control Message Protocol*)

Como foi descrito anteriormente, o protocolo IP provê um serviço de comunicação sem conexão e não confiável, transmitindo datagramas de roteador em roteador até a máquina destino. Se por acaso um roteador não conseguir rotear ou transmitir um datagrama, ou se o roteador detectar uma condição anormal da rede, por exemplo um congestionamento que o impeça de transmitir datagramas, é necessário instruir a máquina fonte para tomar as devidas providências.

O mecanismo utilizado pelos roteadores para comunicar tais controles ou informar erros é conhecido como *ICMP (Internet Control Message Protocol)* - Protocolo de Mensagens de Controle de Rede.

O ICMP somente relata condições de erro à fonte original. A fonte deverá tomar as medidas necessárias para corrigir o problema.

Alguns tipo de mensagens que o ICMP provê são:

- *Echo Request* e *Echo Reply*: Este tipo de mensagem é utilizado para saber se uma máquina destino está ligada ou não. Primeiramente envia-se um *Echo Request* para a máquina destino em questão, se esta estiver ligada, ela responderá com um *Echo Reply*. Caso contrário ocorrerá um TIMEOUT, indicando que a máquina destino está desligada. Este tipo de função é mais conhecida como *ping*.
- *Destination Unreachable*: Quando um roteador não consegue transmitir um datagrama, ele envia uma mensagem de erro de destino não alcançável.
- *Source Quench*: Quando um roteador começa a receber informações mais rápido do que consegue processá-las, então ele envia, à origem, uma mensagem *Source Quench*. Este tipo de mensagem ICMP tem por objetivo reduzir a velocidade de transmissão da fonte original.

4.1.4) UDP (*User Datagram Protocol*)

O usuário da rede normalmente utiliza mais de um programa de aplicação ao mesmo tempo. Assim, como é possível identificar que um pacote enviado pertence a uma aplicação A ou a uma aplicação B, ou o contrário?

Uma possível resposta está na utilização de um protocolo que sirva como interface entre os aplicativos e o protocolo IP. Este protocolo é denominado *UDP – User Datagram Protocol* e ele pertence à camada de transporte.

A figura 4.8 mostra a posição do protocolo UDP na camadas conceituais.

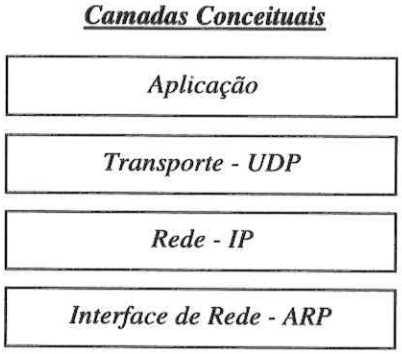


Figura 4.8 - Camadas conceituais e seus protocolos.

O mecanismo utilizado pelo UDP é o de alocar portas (*ports*) para cada aplicativo. Existem no entanto aquelas portas pré-definidas para diversas aplicações, tais como: porta 69, transferência de arquivos; porta 161, SNMP (*Simple Network Management Protocol*); porta 520, RIP; etc.

Assim, se um aplicativo em uma máquina A quer se comunicar com um aplicativo em uma máquina B, o UDP se encarrega de alocar uma porta para o aplicativo (ou utilizar umas das portas pré-definidas), indicar a porta de origem e a porta destino e enviar tais dados para o protocolo IP.

À medida que o protocolo UDP vai recebendo dados da camada de rede ele age como um demultiplexador, analisando as portas destino e enviando os dados para os aplicativos, alocados nas portas em questão.

A figura 4.9 mostra o processo de envio e recepção de dados pelo protocolo UDP.

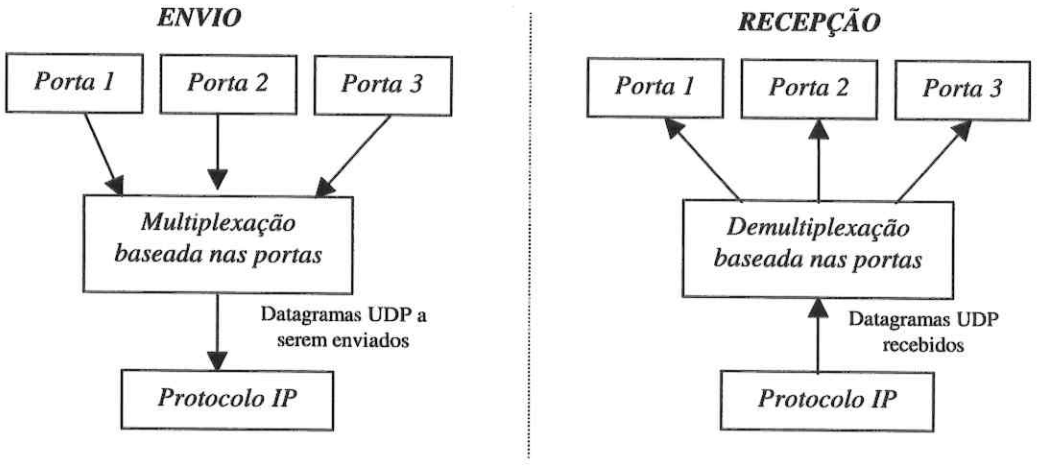


Figura 4.9 - Envio e recepção de datagramas UDP.

O protocolo UDP trabalha com as mesmas características do protocolo IP, ou seja, comunicação não confiável e sem conexão. A segurança, neste caso, deve ser provida pelos programas de aplicação.

4.2) Protocolos de Roteamento Dinâmico

Em uma rede de computadores com um ou mais elementos roteadores, para que a informação seja roteada corretamente é necessário que estes roteadores conheçam as rotas corretas e de menor custo. A função de informar aos roteadores as melhores rotas é, geralmente, atribuída ao administrador da rede. Assim, se uma rede é adicionada, eliminada ou muda de lugar, o administrador deve atualizar a tabela de roteamento de todos os roteadores.

Esta tarefa é viável em redes de computadores de pequeno porte, onde as mudanças são lentas e previsíveis. Porém, em redes de médio e grande porte, devido à quantidade de mudanças associadas à velocidade com que estas mudanças ocorrem, a atualização manual das tabelas se torna algo impossível. Para solucionar este problema foram criados protocolos de roteamento dinâmico.

O protocolo de roteamento dinâmico tem como principal função informar aos outros roteadores sobre as mudanças ocorridas na rede. Deste modo, as redes se tornam mais estáveis, pois as tabelas de roteamento dos roteadores são atualizadas mais rapidamente e de modo confiável. Além disto, tais protocolos também fornecem certa segurança com relação à eliminação de *loops* de roteamento.

Há diversos protocolos de roteamento dinâmico. Alguns foram estudados e descritos abaixo.

4.2.1) GGP (*Gateway to Gateway Protocol*)

A *Internet*, considerada a maior rede virtual do mundo, é formada por diversas redes, estas por (sub-) redes, e assim por diante. Esta gigantesca rede de computadores possui um núcleo, denominado *core*, o qual é composto por *core gateways*. Este núcleo é administrado pelo *INOC – Internet Network Operations Center*.

O *core* faz o papel de uma cola, que mantém toda a *Internet* unida. Assim os *core gateways* possuem as rotas de todas as redes, e como mencionado anteriormente, utilizam o mesmo princípio de preenchimento da tabela de roteamento de redes: o princípio de ocultamento de informação.

No caso dos *core gateways* não existe rota padrão, pois poderia gerar um *loop* desnecessário. Isto acontece devido ao mecanismo de roteamento já discutido. A figura 4.10 mostra a situação de um *loop*, gerado devido a presença de rota padrão nos *core gateways*.

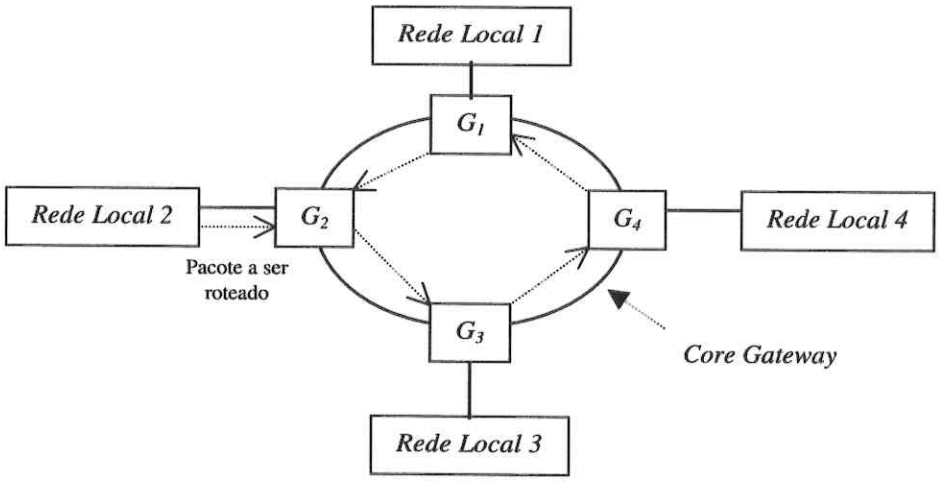


Figura 4.10 - Loop em um core, onde os core gateways possuem rota padrão.

Se um pacote chegar no *core gateway* G_2, este deve roteá-lo de acordo com a sua tabela de roteamento de rede. Porém, caso ele não encontre na tabela a rota em questão, o *gateway* envia o pacote pelo caminho fornecido pela rota padrão ao destino, que no caso é G_3. Se G_3 também não encontrar na tabela a rota destino do pacote, ele o envia para G_4, e assim por diante até voltar ao G_2. Este *loop* durará até o fim de vida do pacote, gerando um congestionamento desnecessário na rede.

Inicialmente, quando a *Internet* ainda era pequena, os administradores do INOC preenchiam a tabela de roteamento de rede manualmente. Atualmente, esta tarefa é impossível; neste caso utiliza-se o protocolo GGP – *Gateway to Gateway Protocol*.

O *GGP – Gateway to Gateway Protocol* é o protocolo inicialmente usado pelos *core gateways*. O protocolo GGP pertence a uma classe de algoritmos para roteamento denominado *vetor-distância*. A idéia por traz deste algoritmo é muito simples. Cada *gateway* possui a sua tabela de roteamento, e periodicamente envia aos *gateways* vizinhos as suas rotas. A mensagem é composta por uma lista de pares (vetor, distância), donde se originou o seu nome.

A figura 4.11 mostra um exemplo da tabela de roteamento e as informações enviadas pelo GGP. Quando o *gateway K* recebe as informações das rotas do *gateway J*, *K* examina os destinos e as suas respectivas distâncias. Se *J* conhece um caminho mais curto para alcançar um determinado destino, ou se *J* lista um destino que *K* não tem na sua tabela, ou se *K* possui uma rota que passe por *J* e a distância de *J* ao destino desta rota muda, então *K* faz as devidas mudanças na sua tabela de roteamento.

Tabela do gateway K

Destino	*Distância*	*Rota*
Rede 1	*0*	*Direta*
Rede 2	*0*	*Direta*
Rede 4	*8*	*Gateway L*
Rede 17	*5*	*Gateway M*
Rede 24	*6*	*Gateway J*
Rede 30	*2*	*Gateway Q*
Rede 42	*2*	*Gateway J*

Rotas do gateway J

	Destino	*Distância*
	Rede 1	*2*
✓	*Rede 4*	*3*
	Rede 17	*6*
✓	*Rede 21*	*4*
	Rede 24	*5*
	Rede 30	*10*
✓	*Rede 42*	*3*

Figura 4.11 - Tabela de roteamento e as informações de uma mensagem GGP.

Neste exemplo, os destinos assinalados serão modificados ou inseridos na tabela de roteamento *K*. A tabela possui uma outra coluna determinando assim a rota a ser seguida. Quando uma rota com uma distância **N,** vinda de *J*, é inserida ou modificada na tabela de *K*, a distância será acrescentada de **N+1**, ou seja, a distância de *J* até o destino mais a distância de *K* até *J*.

A distância é medida em *hops*, ou seja, em número de *gateway* que a informação deve passar para chegar ao seu destino. Se a máquina destino está na mesma rede física do *gateway* então a distância será 0 (zero), ocorrendo um roteamento direto.

Protocolos que utilizam o algoritmo *vetor-distância*, por exemplo o GGP, são fáceis de implementar, porém possuem desvantagens, tais como: quando uma rota muda, a informação desta mudança se propaga muito devagar de *gateway* em *gateway*, enquanto isso alguns *gateways* podem enviar pacotes com a rota incorreta; o tamanho da mensagem cresce à medida que o número de redes aumenta, gerando um congestionamento desagradável para a rede.

Atualmente o protocolo GGP não é mais utilizado pelo *core*, devido às suas desvantagens. O *core* utiliza um protocolo que se baseia no algoritmo conhecido como *SPF – Shortest Path First*. Este algoritmo será discutido logo em seguida.

4.2.2) SPF (*Shortest Path First*)

Uma alternativa para suprir as desvantagens geradas pelos algoritmos *vetor-distância* é a classe de algoritmos conhecida como *SPF – Shortest Path First*. O *SPF* requer que cada *gateway* participante tenha uma informação completa sobre a topologia da rede. É fácil pensar nesta topologia como um mapa, onde os *gateways* podem ser representados por nodos e as redes por *links* entre os nodos ou um simples ramo.

O mecanismo utilizado pelo *SPF* é simples: ele primeiramente analisa todos os *links* com os *gateways* vizinhos, verificando se estes estão ativos ou não. Então o *SPF* envia uma mensagem informando sobre o estados dos *links* analisados.

O *gateway* recebe estas informações, e se alguma rota foi modificada, ele aplica um algoritmo conhecido como *Dijkstra shortest path algorithm*, resultando em um novo gráfico ou mapa de toda a rede.

Dentre as vantagens do SPF podemos citar: como o estado dos *links* não muda quando a mensagem com esta informação está sendo enviada, fica fácil de resolver algum problema inesperado; como os *gateways* realizam a computação da rota localmente, a convergência é garantida; como as mensagem apenas carregam informações sobre os *links* vizinhos ao *gateway* fonte, o seu tamanho não depende do número de redes.

Como exemplo de protocolos que utilizam o algoritmo SPF, podemos citar o *OSPF – Open SPF protocol*, o qual será discutido posteriormente.

4.2.3) EGP (*Exterior Gateway Protocol*)

Como descrito anteriormente, a *Internet* possui um núcleo (*core*) composto por *core gateways*. Cada *core gateway* era conectado a uma rede e tinha como principal função realizar o roteamento de todas as informações de forma confiável. Para isso, dispunha-se de uma tabela de roteamento de rede com todas as rotas existentes na *Internet*. E utilizava-se de um protocolo para mantê-las atualizadas. Assim, para cada rede nova, haveria um *core gateway*. Infelizmente este tipo de topologia é impossível!

A topologia da *Internet* é constituída de múltiplas redes, uma conectada à outra por meio de *non-core gateways*. A figura 4.12 mostra um exemplo de uma topologia deste tipo:

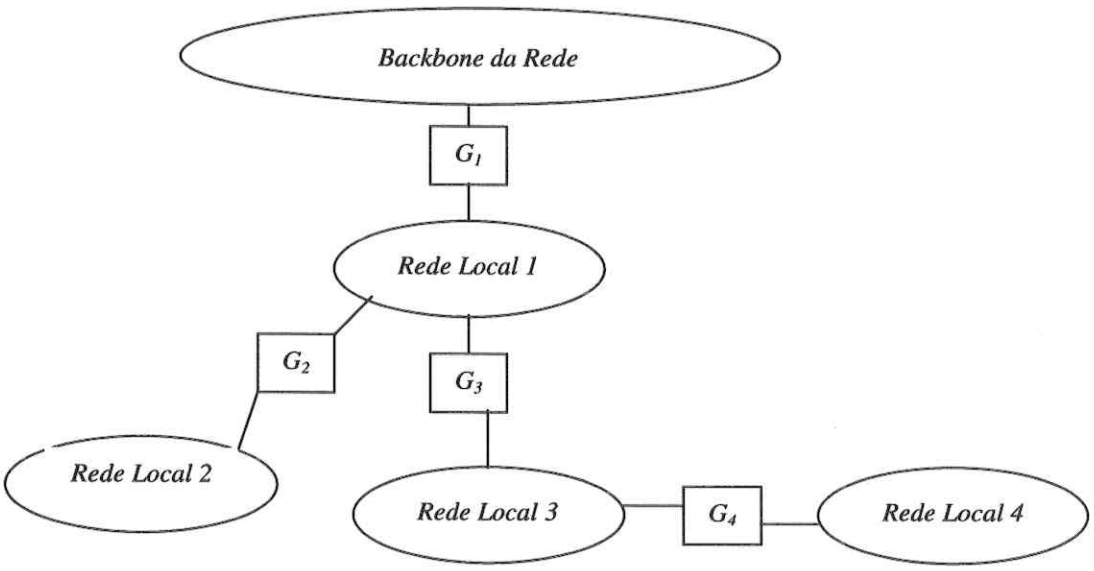


Figura 4.12 - Topologia de rede.

O *core gateway* G_1 conhece sobre as redes 1, 2 e 3. Supondo que uma nova rede seja instalada e denominada rede 4, G_1 não saberá como acessá-la, ou nem saberá que ela existe. Porém o G_3 conhece a rede 4. A solução está em designar um *gateway* responsável por avisar ao *core* sobre as rotas para as novas redes. A partir de então é necessário definir a idéia de um **Sistema Autônomo**.

Um Sistema Autônomo é um conjunto de redes conectadas por *gateways*, administradas por um orgão superior, como por exemplo o *Core*. O Sistema Autônomo possui um *gateway* responsável por informar ao *core gateway* ou a outro Sistema Autônomo sobre todas as rotas existentes. Assim surge o conceito de *exterior neighbors* (vizinhos externos), *gateways* localizados em Sistemas Autônomos diferentes e *interior neighbors* (vizinhos internos), localizados no mesmo Sistema Autônomo.

Os *gateways* e roteadores dentro dos Sistemas Autônomos ficam responsáveis pelos mecanismos de descoberta, propagação, validação e checagem da consistência das rotas. Os protocolos de roteamento interno serão discutidos mais adiante.

A figura 4.13 exemplifica o Sistema Autônomo e o *Core*.

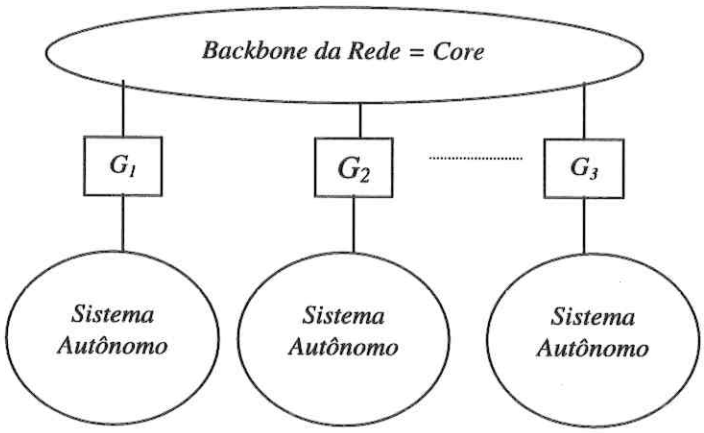


Figura 4.13 - Sistema Autônomo e o Core.

Para o *gateway* do Sistema Autônomo se comunicar com o *core gateway*, utiliza-se um protocolo denominado *EGP -Exterior Gateway Protocol*. Este utiliza o algoritmo *vetor-distância* para informar sobre as rotas. A figura 4.14 mostra um exemplo:

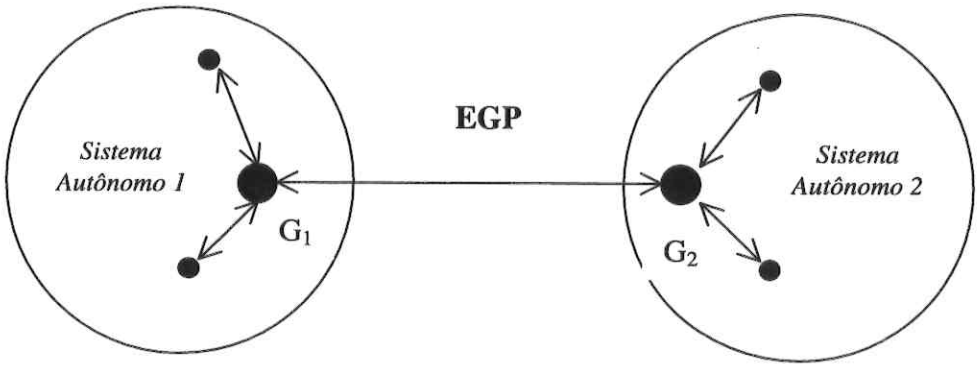


Figura 4.14 - Ilustração Conceitual de dois gateways vizinhos usando o protocolo EGP.

O EGP possui três principais características. Primeira, o mecanismo *neighbor acquisition* (aquisição do vizinho), que permite a um *gateway* requerer a outro uma confirmação de que os dois devem trocar informações sobre suas rotas. Segundo, um *gateway* testa constantemente se seus vizinhos EGPs estão respondendo. Terceiro, os vizinhos EGPs periodicamente trocam informações sobre as rotas, pelo envio de mensagens de atualização de rotas (*routing update messages* - mensagens de atualização de rotas).

Gateways com EGP estão restritos a informar somente as rotas pertencente ao seu Sistema Autônomo. Rotas que eles tenham aprendido e pertençam a outros Sistemas Autônomos estão proibidas de serem informadas aos vizinhos EGPs. Esta lei é também conhecida como *EGP third party restriction* (restrição a terceiros EGP).

4.2.4) IGP (*Interior Gateway Protocol*)

Vamos agora analisar os mecanismos utilizados pelos Sistemas Autônomos para descobrir, propagar, validar e checar a consistência das rotas internas. Estes mecanismos são providos por protocolos mais conhecidos como *IGPs - Interior Gateway Protocols*. Inicialmente a inexistência de uma padronização, levou ao surgimento de diversos protocolos IGPs proprietários. Dentre eles podemos citar o RIP, HELLO e o OSPF, que serão descritos a seguir.

HELLO

O protocolo HELLO é um típico protocolo *vetor-distância*, porém a distância não é mais medida em *hops,* mas sim em tempo.

O HELLO possui duas funções: sincronizar os relógios entre os conjunto de máquinas que compõem um Sistema Autônomo, e permitir que cada máquina compute o caminho cujo tempo é mais curto.

A idéia básica por trás do protocolo HELLO é simples. Cada máquina participante mantém uma tabela com as suas melhores estimativas do relógio das maquinas vizinhas. Antes de enviar um pacote, o protocolo insere o valor atual relógio. Quando um pacote chega, o receptor subtrai o valor do relógio encontrado no pacote com o valor estimado por ele, obtendo assim, o tempo que o pacote levou para chegar. Este tempo corresponde ao custo do caminho.

OSPF (OPEN SHORTEST PATH FIRST PROTOCOL)

Como descrito anteriomente, os protocolo que utilizam a classe de algoritmos conhecida como *SPF – Shortest Path First*, são muito melhores que a classe de algoritmos *vetor-distância*. Recentemente, um grupo de trabalho da IETF - *Internet Engineering Task Force* propôs um novo IGP, que utiliza o algoritmo SPF. Chamado de *Open SPF Protocol* (OSPF), o novo protocolo propõe-se a atingir alguns objetivos ambiciosos.

- ✓ *A especificação deverá estar disponível em literatura publicada*, possibilitando a implementação do protocolo por qualquer um, sem a necessidade de pagamento taxas;
- ✓ *OSPF deverá incluir tipo de serviço de roteamento*, ou seja, para cada tipo de serviço pode ser definida uma rota diferente para um mesmo destino. O OSPF é o primeiro dos protocolos de arquitetura *Internet TCP/IP* que utiliza este tipo de opção;
- ✓ *OSPF deverá prover balanceamento de cargas*. Se o gerente da rede especificar múltiplas rotas para um único destino, o OSPF distribui o tráfego pelas rotas igualmente;
- ✓ *Deverá permitir o crescimento e tornar os Sistemas Autônomos fáceis de gerenciar*; o OSPF permite organizar as suas redes e *gateways* em subconjuntos denominados **áreas**. Cada área é auto-suficiente, e a topologia de uma área permanece transparente para a outra;
- ✓ *O OSPF deverá especificar que todas as trocas de mensagens entre gateways sejam autenticadas*. A autenticação de uma mensagem aumenta a segurança da rede, evitando que pessoas maliciosas burlem o roteamento com mensagens de roteamento falsas;
- ✓ *OSPF deverá permiter o roteamento a um host específico, ou a uma rede específica*;
- ✓ *Deverá permitir um máximo de flexibilidade.* O OSPF permite gerentes de rede descreverem uma topologia de rede virtual, abstraindo assim os detalhes das conexões físicas;
- ✓ *OSPF deverá permitir gateway trocarem informações de roteamento aprendidas de outros Sistemas Autônomos*. O OSPF indica em suas mensagens, quais as conexões se referem a vizinhos exteriores e quais aos vizinhos interiores.

RIP (ROUTING INFORMATION PROTOCOL)

O RIP – *Routing Information Protocol* é um dos mais populares protocolos de roteamento dinâmico que utilizam o algoritmo vetor-distância. Utilizado principalmente em redes de computadores de pequeno e médio porte, o RIP foi escolhido para ser implementado neste projeto, principalmente, devido à sua boa relação CUSTO X BENEFÍCIO X TEMPO DE IMPLEMENTAÇÃO e também por se adaptar aos padrões utilizados pelos *Switches* da CIANET, além de estar presente em boa parte dos *switches* mais conhecidos.

Este protocolo é descrito em detalhes no capítulo VI.

Capítulo V: RIP (*Routing Information Protocol*) [RFC 1058][COMER 91b]

O protocolo de roteamento dinâmico RIP – *Routing Information Protocol* é um dos mais populares dentre os IGPs. O RIP surgiu do programa "*routed*", originalmente implementado na Universidade da California em Berkeley e usado no sistema operacional UNIX. Ele tem como objetivo prover informações de alcançabilidade consistentes entre as máquinas da rede local em questão.

A popularidade do RIP não é devido propriamente a seus méritos técnicos, mas sim à popularidade do sistema UNIX. Como todo protocolo que utiliza o algoritmo vetor-distância para propagar suas rotas, o RIP foi projetado principalmente para trabalhar com redes de tamanho moderado, utilizando tecnologias razoavelmente homogêneas. Por isso, ele é utilizado como IGP para campus, corporações, redes regionais, etc. O seu uso não está ligado a ambientes complexos, como por exemplo, redes com tecnologias variadas ou redes usando linhas seriais, cujas velocidades variam muito.

O protocolo RIP atribui o valor do custo do caminho a ser percorrido em *hops* (saltos). A métrica utilizada pelo RIP difere de outros protocolos vetor-distância, ou seja, a distância de uma rede de computadores a um roteador é definida como 1 *hop*, se esta rede está **diretamente** conectada a este roteador; 2 *hops* se a rede é alcançavel através de um outro roteador e assim por diante. Como exemplo, na figura 5.1 analisamos a métrica do RIP com relação ao roteador R_1.

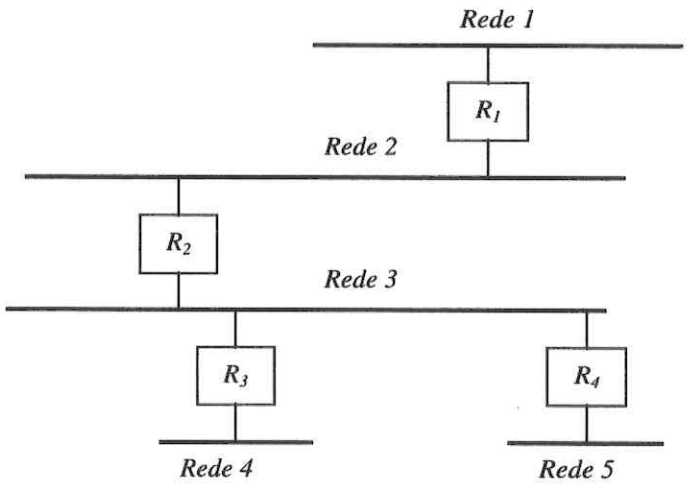


Figura 5.1 - Análise da métrica RIP com relação ao roteador R_1.

Baseando-se na métrica RIP, para o roteador R_1 as redes 1 e 2 estão a 1 *hop*, enquanto que a rede 3 está a 2 *hops* e as redes 4 e 5 estão a 3 *hops*. Assim:

> *"O número de hops ao longo do caminho entre uma máquina fonte e uma destino é dado pelo número de roteadores, que o datagrama encontra ao longo do caminho"*

Este tipo de métrica varia de 1 *hop* até 15 *hops*, sendo os valores fora deste campo considerados infinitos. Isto se dá devido à alguns problemas de instabilidade de rede que serão descritos posteriormente.

O protocolo RIP foi inicialmente projetado para ser utilizado em redes de computadores baseadas no protocolo IP. Isto se deu devido à utilização deste protocolo no sistema UNIX, o qual baseia suas comunicações na pilha de protocolos TCP/IP.

Devido às características até então descritas, o RIP possui algumas limitações:

- ✓ ***O protocolo é limitado quanto ao número de redes, pois o caminho mais longo é de até 15 hops***. Neste caso, considera-se que o custo de cada rede é de 1 *hop*. Assim, projetistas e administradores de redes acreditam que o RIP se torne inapropriado para redes de computadores muito grandes.
- ✓ ***O protocolo depende da "contagem para o infinito" para resolver certas situações incomuns***. Se um sistema autônomo possui muitas redes, estas podem formar *loops* entre si. Tais *loops* prejudicam o desempenho e a consistência de uma rede de computadores. Estas situações serão explicadas posteriormente.
- ✓ ***Este protocolo usa métricas fixas para comparar rotas alternativas***. Isso não é apropriado para situações onde rotas necessitam ser escolhidas baseadas em parâmetros de tempo real, tais como, o atraso de transporte, a confiabilidade ou carga.

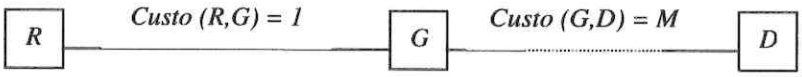
O protocolo RIP classifica as máquinas como **ativas** e **passivas**. As máquinas ativas são geralmente os roteadores, pois estes se encarregam de enviar e receber mensagens de roteamento. Já as máquinas passivas são melhor representadas pelos *hosts* (máquinas que não são roteadores ou *gateways*), estes apenas recebem (escutam) as mensagens de roteamento. Entretanto, há situações em que *hosts* com mais de uma interface de rede (denominados *multi-homed hosts*) assumem o papel de máquinas ativas.

Como todo protocolo de roteamento dinâmico, o RIP possui como principal objetivo suprir uma rede de computadores com informações necessárias para realizar o roteamento de informações. Deste modo, as máquinas que trabalham no modo ativo difundem periodicamente informações sobre suas rotas para a rede.

5.1) A Aritmética do RIP

Como outros protocolos vetor-distância, as mensagens RIP contém informações que correspondem a pares formados pela rede destino e a distância para atingi-la (medida em *hops*).

Assim a aritmética do RIP é simples e se baseia no seguinte exemplo ilustrado na figura 5.2.



$$Custo\ (R,D) = Custo\ (G,D) + Custo\ (R,G) = M + 1$$

Figura 5.2 - Aritmética do protocolo RIP.

Quando uma mensagem de atualização RIP chega em *R* proveniente do roteador *G* e possui uma métrica *M* para o destino *D*, então esta rota é comparada com as rotas existentes em *R*. Se esta rota ainda não existe, então ela é inserida na tabela de roteamento de *R* a um custo representado pela equação matemática

$$Custo\ (R,D) = Custo\ (G,D) + Custo\ (R,G)$$

ou seja, o custo para atingir o destino *D* é igual ao custo para atingir o roteador *G* mais o custo gasto entre *G* e *D*. Como a métrica da vizinhança é sempre 1, então

$$Custo\ (R,D) = Custo(G,D) + 1$$

logo para Custo (G,D) = M resulta em

$$Custo\ (R,D) = M + 1$$

o qual corresponde ao valor da métrica inserida na tabela de roteamento de *R*.

Se a rota já existe na tabela e especifica *G* como o próximo passo, então custo da rota é atualizado para o valor M+1.

Se o custo da rota atual é maior que M+1, então seta-se o novo custo para M+1 e o próximo passo para o roteador *G*.

Caso contrário nenhuma modificação é feita.

5.2) Prevenção de Instabilidades: O problema da convergência lenta

Os algoritmos vetor-distância compartilham um problema comum, que é a geração de *loops* temporários de roteamento. A geração destes *loops* de roteamento torna a rede de computadores instável e o roteamento de pacotes não confiável.

Para que seja possível contornar o problema da geração de *loops*, o RIP deve tratar 3 tipos de erros característicos dos algoritmos vetor-distância:

1. Uma vez que o algoritmo não detecta a ocorrência de *loops* de roteamento, o RIP deve tanto supor que os seus participantes sejam confiáveis como também deve tomar precauções para evitar tais *loops*;

2. Para evitar instabilidades o RIP deve usar um valor baixo para a máxima distância possível. (O RIP considera uma distância infinita como sendo igual a 16 o que limita a sua aplicação a redes onde o número de roteadores em seqüência não seja superior a 15);
3. O algoritmo vetor-distância usado pelo RIP cria um problema conhecido como convergência lenta (*slow convergence*) ou contagem para o infinito (*count to infinity*), que permitem o surgimento de inconsistências, uma vez que as mensagens de atualização se propagam lentamente pela rede. Escolhendo um valor baixo para infinito (16) ajuda a limitar o problema da convergência lenta, mas não o elimina.

O problema de convergência lenta ou contagem para o infinito é exemplificado pela figura 5.3.

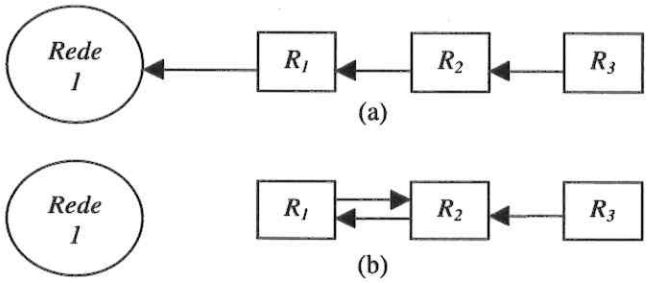


Figura 5.3 - Problema da convergência lenta.

Como mostrado na figura acima, em (a) R_1 possui uma conexão direta para a rede 1, assim a métrica é de 1 *hop*. R_2 aprende esta rota de R_1 e assume como métrica o valor de 2 *hops*. Do mesmo modo, R_3 aprende a rota de R_2 e assume uma métrica de 3 *hops*.

Agora supõe-se que a conexão de R_1 com a rede 1 falha (b). R_1 irá automaticamente atualizar esta rota e assumir como métrica o valor 16 *hops* (infinito). No próximo período R_1 difunde a nova informação e a rede se estabiliza. Porém, se R_1 primeiramente receber uma mensagem de atualização proveniente de R_2, R_1 irá comparar a rota atual com a recebida. Verificando que para atingir a rede 1 através de R_2 terá um custo de 2 *hops*, assim R_1 atualiza a rota existente colocando como próximo passo R_2 e uma métrica de 3 *hops*.

Após um período, R_2 receberá de R_1 uma nova métrica para a rede 1, igual a 3 *hops*, deste modo, R_2 atualiza a sua métrica para 4 *hops*. O mesmo acontece com R_1, no próximo período, e assim por diante até que o valor das métricas atinjam 16 *hops* (infinito), cessando assim a presença do *loop*.

Este processo é conhecido como convergência lenta ou contagem para o infinito. Devido a sua lentidão, pacotes ficam presos nos *loops* de roteamento. No caso da figura 5.3 (b), um pacote destinado à rede 1 ficará preso entre os roteadores R_1 e R_2, até que estes identifiquem que a rota para a rede 1 não existe mais, gerando um maior tráfego e inconsistências na rede de computadore.

Para redes com alta carga de informação estas inconsistências podem levar a problemas desagradáveis. O uso de um valor baixo para indicar a rede "não alcançavel" ameniza os efeitos da convergência lenta.

A seguir serão descritas 3 técnicas utilizadas na resolução do problema da convergência lenta: *Split Horizon Update*, *Hold Down* e *Poison Reverse with Triggered Updates*.

5.2.1) *Split Horizon Update*

Para explicar esta técnica, vamos nos basear na figura 5.3 (b). O roteador R_1 informa a rota para a rede 1 com um custo de 1 *hop*. O roteador R_2 também difunde a informação da mesma rota, porém a um custo de 2 *hops*. Quando R_1 recebe tal informação de R_2, ele verifica que o custo da rota recebida é maior que o custo presente na sua tabela de roteamento, deste modo R_1 não atualiza a rota.

Porém, se a conexão com a rede 1 falha, pode ocorrer o problema de convergência lenta. Este problema se dá, pois o roteador R_2 difunde a informação da rota, adquirida de R_1, de volta para o roteador R_1.

A técnica *Split Horizon Update* faz com que o roteador grave a interface sobre a qual uma determinada rota foi recebida e não a envie de volta pela mesma interface.

Com esta técnica R_1 não receberia informação da rota incorreta de R_2, estabilizando assim a rede de computadores.

Esta técnica, entretanto, não cobre todas as topologias de rede.

5.2.2) *Hold Down*

A técnica do *Hold Down* força o roteador participante a ignorar informações sobre a rede que acaba de se tornar inacessível, por um período determinado de tempo. Geralmente este período é do 60 segundos. Baseando-se na figura 5.3 (b), quando a conexão do roteador R_1 com a rede 1 falha, R_1 irá ignorar informações sobre a rede 1 por 60 segundos.

A idéia é esperar tempo suficiente para que todos os roteadores saibam da má notícia.

Esta técnica é praticamente o oposto da técnica *Split Horizon Update*.

A desvantagem desta técnica é que se ocorrerem *loops* de roteamento, eles serão preservados durante o período de *hold down*. Mais importante ainda, esta técnica preserva todas as rotas incorretas durante o período de *hold down*, mesmo que existam alternativas.

5.2.3) *Poison Reverse with triggered updates*

A técnica *Poison Reverse*, também conhecida como *Split Horizon with Poison Reverse* é uma das mais utilizadas em protocolos de roteamento dinâmico, que utilizam o algoritmo vetor-distância.

Nesta técnica, quando uma mensagem de atualização é enviada através de uma interface de rede, todas as rotas são incluídas, porém as métricas das rotas adquiridas por esta interface são setadas para infinito (16 *hops*). Assim, o *Poison Reverse* iria quebrar qualquer *loop* de roteamento que venha a surgir.

Uma das desvantagens do uso do *Poison Reverse* é o aumento do tamanho das mensagens de atualização, exigindo assim uma maior largura de banda da rede.

Triggered Updates é uma técnica adicional à do *Poison Reverse*, que torna o RIP mais robusto com relação à presença de grandes *loops* de roteamento. Esta técnica consiste em enviar mensagens de atualização imediatamente após a falha de uma conexão, informando rapidamente os outros roteadores que esta conexão não existe mais.

A geração de mensagens de atualização de rotas fora do ciclo periódico de atualização pode causar uma maior carga e tráfego nas redes que se utilizam desta técnica. Por exemplo, numa rede de computadores com vários roteadores. Uma simples mensagem de atualização pode modificar a tabela de roteamento dos roteadores, gerando assim várias outras mensagens. Em algumas rede, onde a largura de banda é pequena ou o tráfego já é intenso, esta técnica, apesar de eficiente, se torna inadequada.

No caso deste projeto, a técnica do *Poison Reverse with triggered updates* foi a escolhida, devido à sua eficiência e confiabilidade, mantendo assim a estabilidade da rede de computadores. Com relação às desvantagens acima descritas, este projeto está voltado para aplicação em redes de alta velocidade, 100 Mb/s e 1Gb/s. Assim, a eficiência da rede é garantida.

5.3) Formato da Mensagem RIP

O protocolo RIP baseia-se no protocolo UDP para se comunicar, utilizando a porta UDP 520. Deste modo, as mensagens RIP são encapsuladas dentro da mensagem UDP e enviadas para as camadas inferiores. A figura 5.4 ilustra o encapsulamento da mensagem RIP.

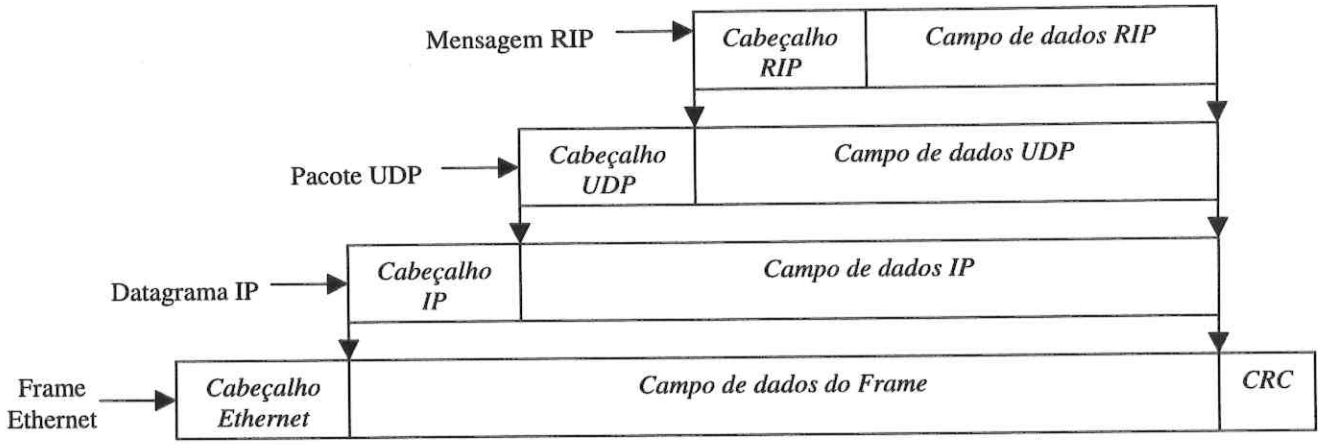


Figura 5.4 - Encapsulamento da mensagem RIP.

As mensagens RIP podem ser genericamente classificadas em dois tipos: mensagens de informação de roteamento e mensagens usadas para requerer informações de roteamento. Ambos os tipos de mensagens possuem o mesmo formato.

Toda mensagem RIP é composta por um cabeçalho de 4 bytes e um campo de dados, onde as rotas são enviadas, possuindo um tamanho máximo de 512 bytes. A figura 5.5 ilustra o formato de uma mensagem RIP.

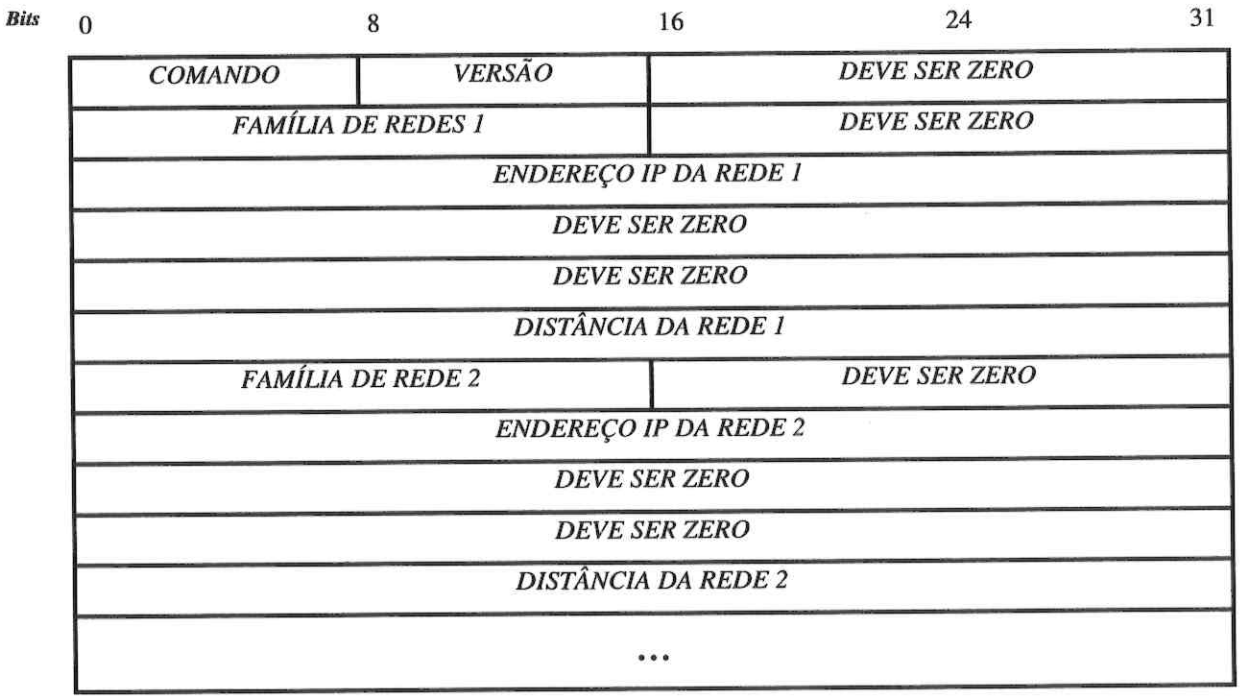


Figura 5.5 - Formato de uma mensagem RIP.

O cabeçalho é composto pelos campos: **COMANDO, VERSÃO** e **DEVE SER ZERO**. O campo VERSÃO determina a versão do protocolo RIP utilizado. Neste projeto utilizou-se a versão 1. O campo COMANDO define o tipo de mensagem enviada. Para a versão 1 deste protocolo, os tipo de comandos implementados são:

Comando	*Significado*
1	***Requisição para informação de roteamento parcial ou total***
2	***Resposta contendo as informações sobre as rotas***
3	***Liga o modo trace (obsoleto)***
4	***Desliga o modo trace (obsoleto)***
5	***Reservado pela Sun Microsystems para uso interno***

Assim, roteadores podem requerer informações de roteamento a outros roteadores, enviando uma mensagem de comando 1 (*request*). Como resposta, os roteadores enviam uma mensagem de comando 2 (*response*) contendo as rotas requisitadas ou toda a tabela de roteamento.

O campo de dados contém as informações sobre as rotas. Cada informação sobre rotas ocupa um espaço de 20 bytes e é composto pelos campos: **FAMÍLIA DE REDE, ENDREÇO IP DA REDE** e **DISTÂNCIA DA REDE**. O protocolo RIP tem a intenção de permitir a troca de informações de roteamento entre diferentes protocolos. Assim, através do campo FAMÍLIA DE REDE é possível definir o tipo de protocolo a que as informações sobre a rota pertencem. Neste caso, para redes baseadas no protocolo IP, o valor deste campo será 2. O campo do ENDEREÇO IP DA REDE é composto pelo identificador da rede. A métrica para alcançar tal rede estará no campo DISTÂNCIA DA REDE.

Assim, o tamanho máximo de uma mensagem RIP será de 512 bytes. Isto inclui somente os campos descritos acima, sem contar com os cabeçalhos IP e UDP.

5.4) Considerações de Endereçamento

O protocolo de roteamento IP, até então implementado, não utiliza rotas para máquinas específicas ou sub-redes, mas sim, rotas para redes específicas. Deste modo, rotas cujos destinos são máquinas, sub-redes e redes se tornam ambíguos, pois o algoritmo de roteamento não se baseia no endereço IP em si, mas apenas numa parte deste, correspondente ao identificador de rede.

O formato do pacote RIP não distingue entre vários tipos de endereços. O campo ENDEREÇO IP DA REDE pode conter:

Endereço IP de uma máquina

Endereço IP de uma sub-rede

Código de uma rede

0.0.0.0, indicando uma rota padrão

Endereço IP de uma rede

No caso deste projeto, somente as duas últimas opções serão consideradas.

O endereço especial 0.0.0.0 é usado para descrever uma rota padrão. Uma rota padrão é usada quando não é conveniente listar todas as possíveis redes em uma mensagem RIP de atualização e quando um ou mais roteadores estão preparados para manusear rotas para redes que não estão explicitamente listadas. Normalmente, o administrador da rede coloca um roteador, que utiliza também o EGP, como roteador padrão. Isto se dá, pois este tipo de roteador, que utiliza tanto protocolos IGP e EGP, servem como porta de saída de um sistema autônomo.

5.5) Temporizadores

Os temporizadores existentes no protocolo RIP regem todo o processo de manutenção das rotas na tabela de roteamento e o processo de troca de informações de roteamento entre os roteadores.

A cada 30 segundos o RIP se encarrega de enviar uma mensagem de atualização de rotas pelas suas interfaces.

Cada rota possui 2 temporizadores, um referente ao tempo de vida, ***TIMEOUT***, e outro referente ao tempo em que a rota ao ser eliminada, ainda permanece na tabela de roteamento, ***GARBAGE-COLLECTION TIME***. O *timeout* é de 180 segundos. Se o roteador não receber alguma informação sobre a rota, a sua métrica é definida com infinita e a rota entra no *garbage-collection time*.

O *garbage-collection time* é de 120 segundos e corresponde ao tempo para que esta informação seja difundida para todos os outros roteadores participantes. Após este tempo, a rota é eliminada da tabela de roteamento. Esta é uma das características da técnica *Poison Reverse with triggered updates*.

Caso chegue alguma informação da rota enquanto o *garbage-collection time* está em processo, esta rota é atualizada, o *garbage-collaction time* é interrompido e o *timeout* é inicializado.

5.6) Processo de Entrada de Mensagens RIP

A implementação do protocolo RIP se divide em duas partes: processos de entrada e saída de mensagens RIP.

Nesta seção será descrito o processo de entrada de mensagens RIP.

Como citado anteriormente existem dois tipos de mensagens RIP, *request* e *response*. Ambas utilizam a mesma porta UDP 520 e o mesmo formato de mensagem.

O fluxograma da figura 5.6 esquematiza o processo de entrada de uma mensagem RIP.

À medida que uma mensagem RIP chega, verifica-se a versão do protocolo RIP utilizado. Se a versão é igual a zero, a mensagem é descartada. Se a versão é igual a 1, então verifica-se os campos DEVE SER ZERO da mensagem. Se estes forem nulos, a mensagem é aceita. Se a versão do protocolo RIP é maior que 1, a verificação dos campos DEVE SER ZERO não é feita.

As mensagens cujas versões são maiores que 1, dizem respeito as versões futuras dos protocolos RIP. Por exemplo, a versão 2, que implementa o roteamento dinâmico a nível de subredes.

Se a mensagem não é descartada, então é verificado o tipo de mensagem RIP. A verificação é feita a partir da análise do campo COMANDO. Se o valor deste campo é 1, a mensagem é do tipo *request*. Se o valor do campo é 2, a mensagem é do tipo *response*. A análise dos dois tipo de mensagens é feita nas próximas seções.

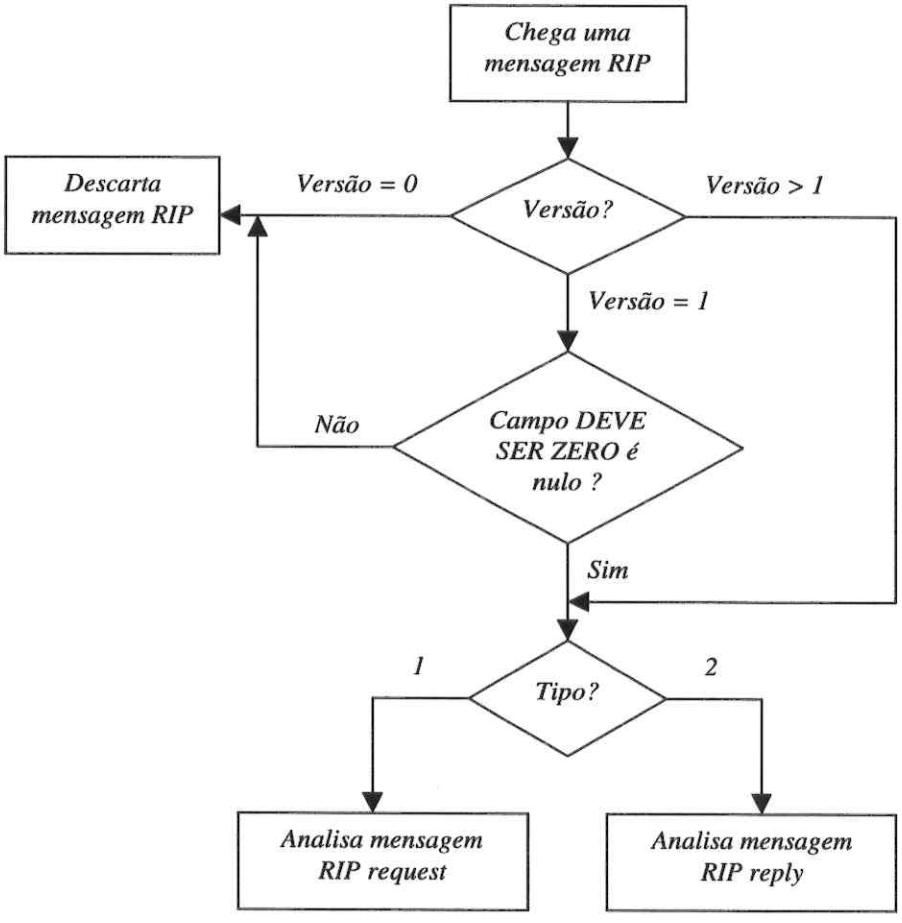


Figura 5.6 - Fluxograma do processo de entrada de mensagens RIP.

5.6.1) *Request*

Uma mensagem *request* é usada para requerer aos roteadores vizinhos informações sobre rotas. Esta mensagem é difundida pelas interfaces do roteador e pode requerer informações sobre determinadas rotas ou sobre toda a tabela de roteamento.

Na requisição das informações de toda a tabela de roteamento, o formato da mensagem RIP é o mostrado na figura 5.7.

COMANDO = 1	*VERSÃO = 1*	*DEVE SER ZERO*
FAMÍLIA DE REDES = 0		*DEVE SER ZERO*
ENDEREÇO IP DA REDE = Qualquer		
DEVE SER ZERO		
DEVE SER ZERO		
DISTÂNCIA DA REDE = 16 hops (infinito)		

Figura 5.7 - Formato de uma mensagem RIP para requisição das informações de toda a tabela.

Neste caso especial de requisição de informações de roteamento, a mensagem RIP deve conter apenas uma rota, cuja métrica é infinita (16 *hops*) e a família de endereços é nula. Deste modo, receptor enviará, como resposta, uma mensagem com todas as rotas de sua tabela de roteamento.

Geralmente este tipo de mensagem ocorre quando o roteador é inicializado, pois ele necessita obter informações sobre as rotas disponíveis na rede de computadores onde ele se encontra.

Para as requisições normais, a mensagem RIP é semelhante à da figura 5.7. Porém, ela é composta pelas diversas rotas requeridas e o campo FAMÍLIA DA REDE é igual a 2.

O processamento de uma mensagem de requisição RIP é ilustrada pelo fluxograma da figura 5.8.

Dando continuidade ao fluxograma da figura 5.6, a mensagem RIP do tipo *request* é processada do seguinte modo. Primeiramente é obtido o número de rotas contidas na mensagem. Em seguida, o protocolo verifica se esta corresponde ao caso especial de uma requisição de toda a tabela de roteamento, ou seja, se o número de rotas for igual a 1, a família de endereços igual a zero e a métrica infinita. Então uma mensagem RIP *response* de toda tabela é gerada.

Caso contrário, para cada rota da mensagem verifica-se se a família de endereços é igual a 2, se afirmativo, então procura-se a mesma rota da mensagem na tabela de roteamento. Caso esta rota seja encontrada, a métrica da rota da mensagem RIP é igualada à métrica da rota obtida da tabela. Se a rota não for encontrada, então a métrica da rota da mensagem RIP é setada para infinito (16 *hops*).

Após percorrer todas as rotas, o campo COMANDO da mensagem RIP é setado para 2, o qual corresponde a uma mensagem RIP *response*. Então, esta é enviada para a máquina que originou a requisição.

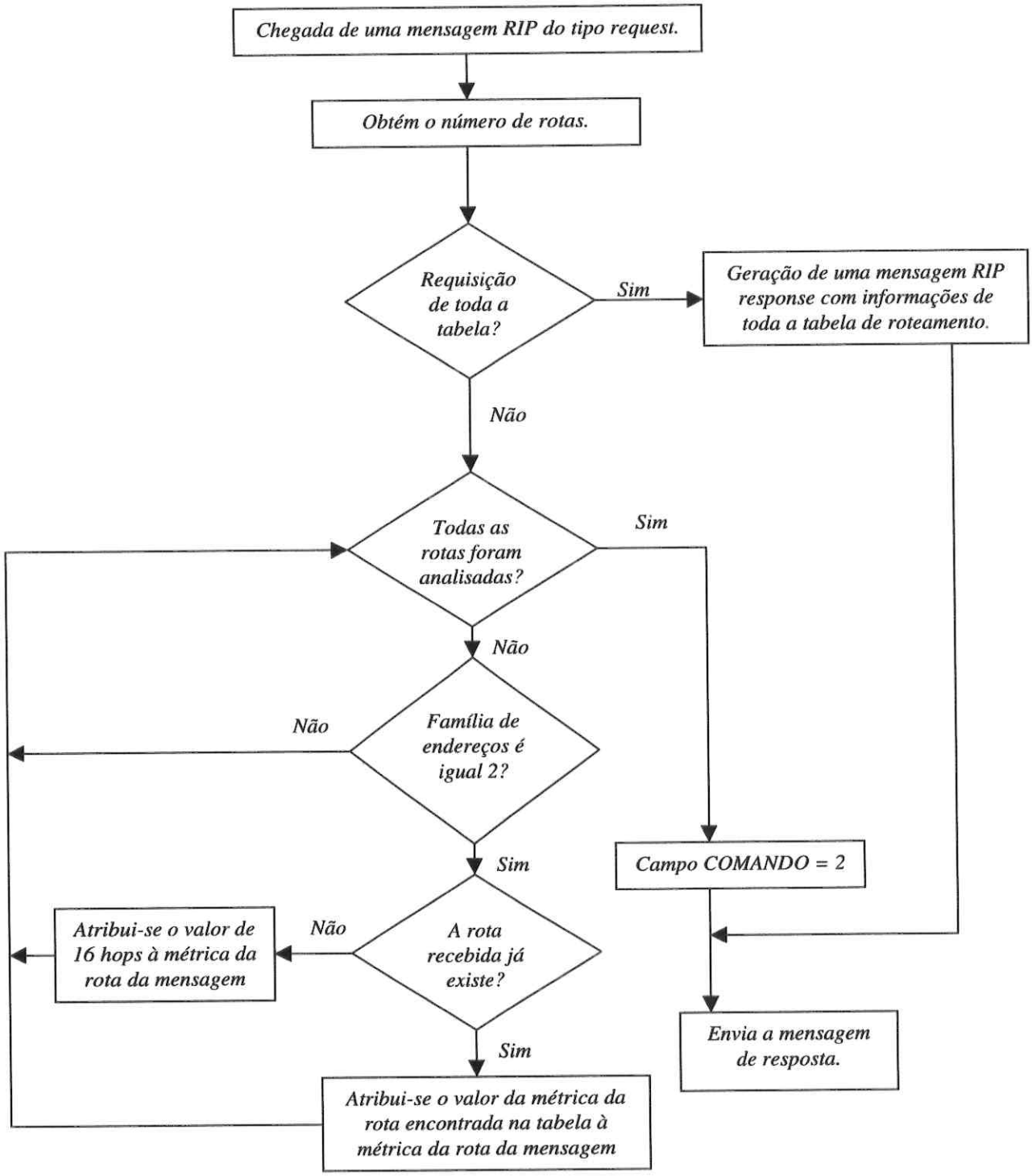


Figura 5.8 - Processamento de uma mensagem RIP request.

5.6.2) *Response*

Uma mensagem *response* pode ser recebida por duas diferentes razões: resposta a uma requisição específica ou atualização de mudança de rota. O formato da mensagem RIP é o mesmo da figura 5.5, mas com o campo COMANDO igual a 2.

O processamento da mensagem *response* é ilustrado pelo fluxograma da figura 5.9.

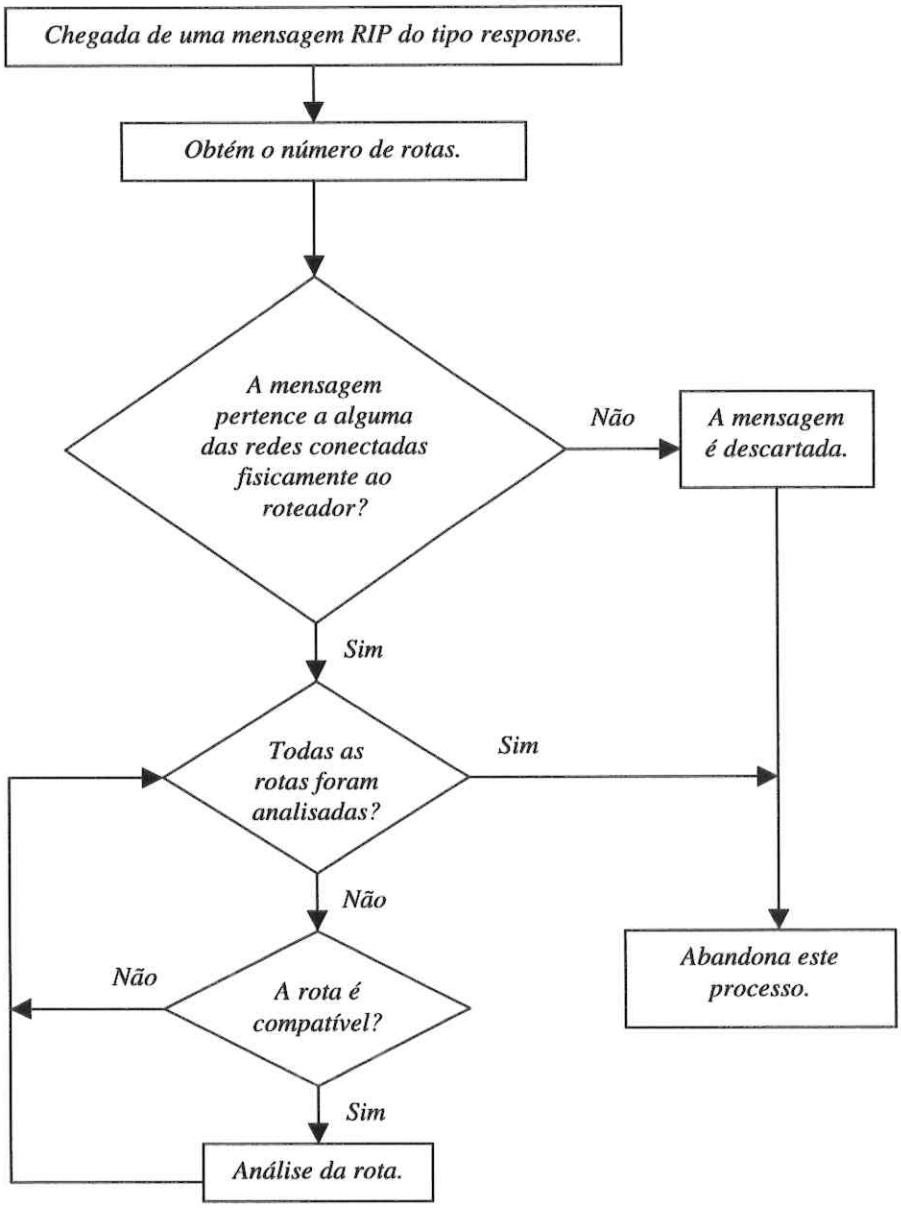


Figura 5.9 - Processamento de uma mensagem RIP response.

À medida que a mensagem RIP chega, ela passa pela análise inicial, mostrada no fluxograma da figura 5.6. Se ela for do tipo *response*, o processamento da mensagem mostrada no fluxograma da figura 5.9 se dá do seguinte modo:

Obtém-se o número de rotas contidas na mensagem RIP. Em seguida, o protocolo verifica se a mensagem pertence a uma das redes, as quais o roteador destino está conectado fisicamente. Caso afirmativo, percorre-se todas as rotas contidas na mensagem. Para cada rota, verifica-se a compatibilidade desta com o tipo de rotas aceita pelo protocolo RIP versão 1.

Nesta verificação, o protocolo confere se a família de endereços é igual 2, em seguida ele verifica se a métrica é menor que infinito (16 *hops*). Depois, verifica se o endereço IP não pertence às classes D e E. Finalmente, o protocolo confere se os campos DEVE SER ZERO são nulos e se o endereço IP não é do tipo *loopback*, ou seja, igual a 127.0.0.0.

Se a rota for compatível, ela é então analisada para saber se será adicionada na tabela de roteamento ou se esta última será apenas atualizada.

Após percorrer todas as rotas, abandona-se este processo.

O processo de análise da rota é ilustrado pelo fluxograma da figura 5.10.

A partir do fluxograma da figura 5.9, dá-se início à análise da rota contida na mensagem RIP. O primeiro passo é verificar se a rota é uma rota padrão. Se afirmativo, atualiza-se a rota padrão e abandona-se esta função. Caso contrário, verifica-se a existência desta rota na tabela de roteamento. Se a rota ainda não existe, então ela é adicionada e uma mensagem de atualização de rotas é enviada (*triggered updates*).

Se esta rota já existe na tabela de roteamento, então o protocolo verifica se o próximo passo da rota (o roteador, através do qual é possível acessar a rede destino) é o mesmo que o da rota existente na tabela. Se afirmativo, atualiza-se o tempo de vida da rota da tabela e verifica-se se a métrica de ambas são diferentes. Se elas forem diferentes, verifica-se se a métrica da rota recebida é infinita. Se afirmativo o tempo de vida da rota da tabela é setada para o *garbage-collection time* e a métrica da rota é atualizada. Em seguida, envia-se uma mensagem de atualização de rotas (*triggered updates*).

Se o próximo passo da rota recebida não é o mesmo da rota da tabela, então verifica-se se a métrica da rota da tabela é menor que a da rota recebida. Se afirmativo, a rota recebida é descartada e a função de análise de rota é abandonada. Caso contrário, a métrica e o próximo passo da rota da tabela são atualizados e uma mensagem de atualização de rotas é enviada (*triggered updates*). Em seguida a função é abandonada.

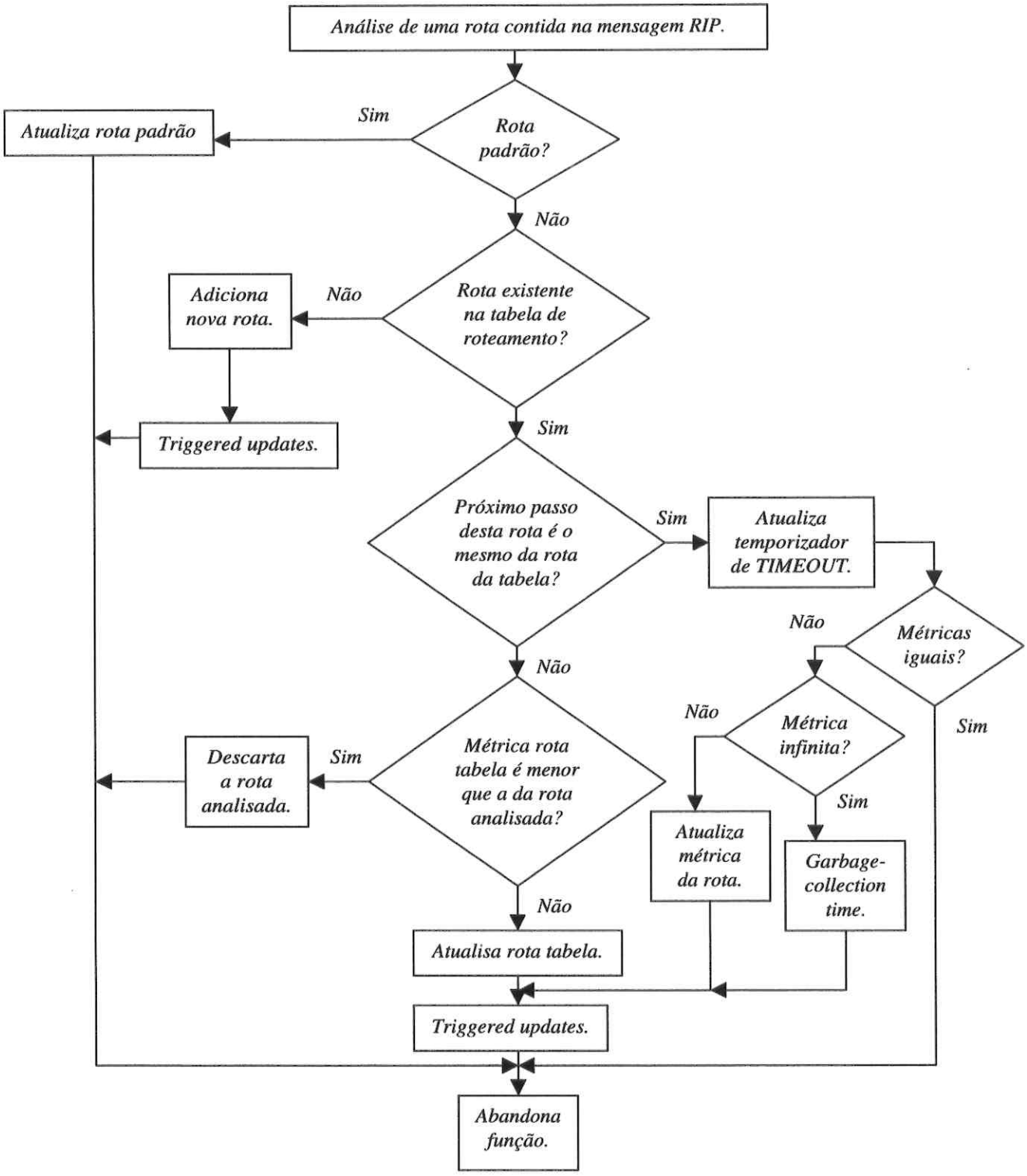


Figura 5.10 - Processo de análise de uma rota.

5.7) Processo de Saída de Mensagens RIP

O processo de saída de mensagens RIP diz respeito à geração de mensagens RIP do tipo response, ou seja, mensagens que contém informações sobre rotas. Tais mensagens são geradas devido à três situações:

- *Resposta a uma mensagem RIP do tipo request*: Neste caso, a mensagem resultante é enviada apenas a um único destino.
- *Atualização regular de rotas*: A cada 30 segundos, uma mensagem *response* contendo toda a tabela de roteamento é difundida por todas as interfaces de rede.
- *Triggered updates*: Toda vez que uma rota da tabela de roteamento é alterada, excluída ou adicionada, uma mensagem de atualização de rotas é gerada e difundida por todas as interfaces de rede.

O primeiro caso já foi descrito anteriormente. Os dois últimos caso serão descritos a seguir.

A geração de mensagens de atualização de rotas é de certo modo simples. O fluxograma da figura 5.11 ilustra este processo.

A geração de uma mensagem RIP do tipo *response* está associada a cada interface de rede. Ou seja, devido à técnica do *Poison Reverse* é necessário gerar uma mensagem RIP *response* para cada interface de rede em particular.

Então, primeiramente verifica-se se a mensagem destina-se a uma determinada interface de rede ou se ela deve ser difundida através de todas as interfaces. Se ela deve ser enviada por uma determinada interface de rede, preenche-se o cabeçalho de apenas uma mensagem RIP *response*. Caso contrário, uma mensagem para cada interface de rede é gerada. No cabeçalho de cada mensagem, o campo COMANDO possui o valor 2, que corresponde ao tipo *response*.

Em seguida, percorre-se toda a tabela de roteamento e cada rota é adicionada na(s) mensagem(s) RIP *response*. Terminada esta tarefa, o protocolo verifica se existe uma rota padrão. Se afirmativo, esta é adicionada à(s) mensagem(s).

Finalmente, cada mensagem é enviada pela sua respectiva interface de rede.

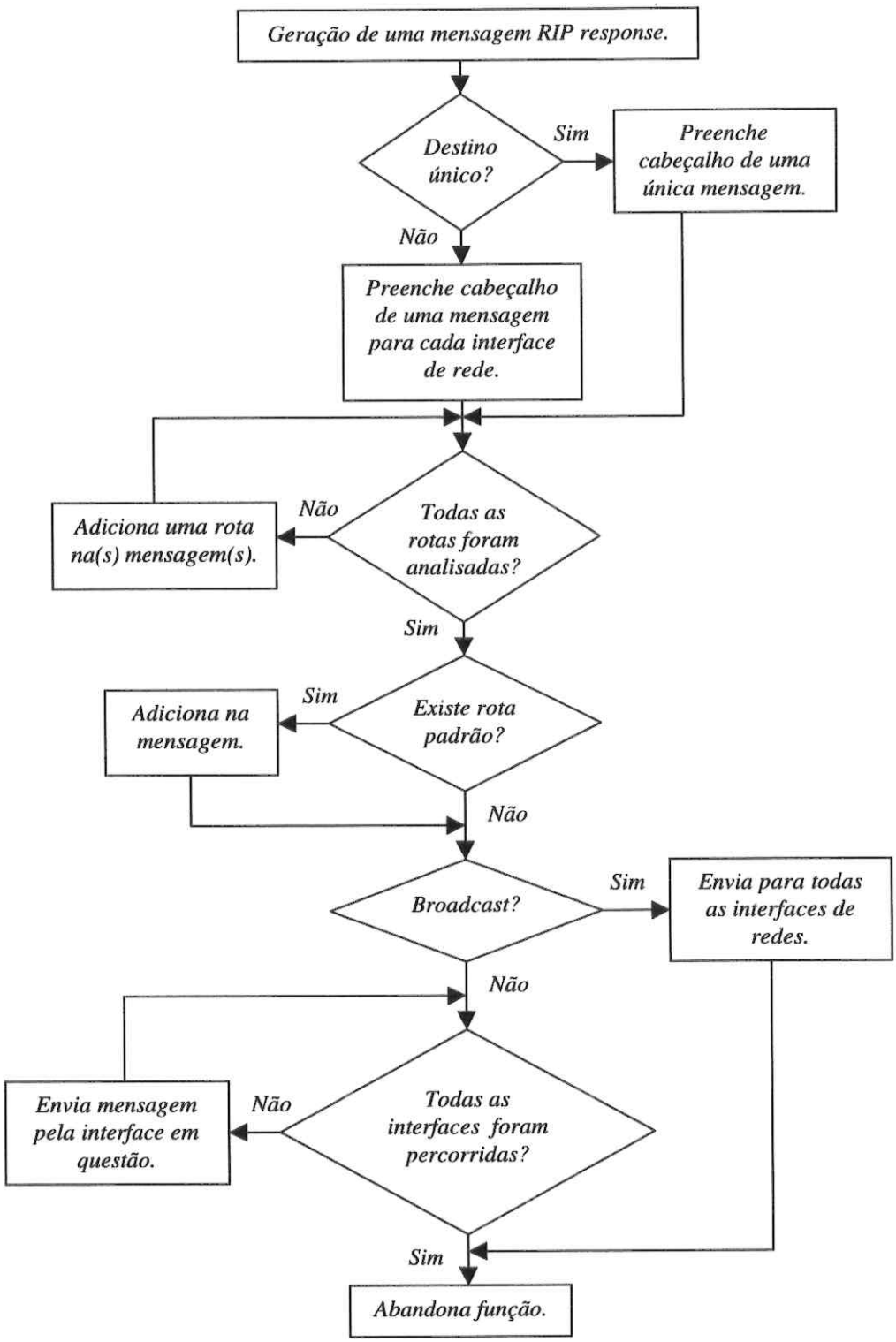


Figura 5.11 - Processo de geração de um mensagem RIP response.

A dependência da mensagem RIP *response* com a interface de rede é devido à técnica *Poison Reverse* utilizada para evita *loops* de roteamento.

Deste modo o processo de adição de rotas nas diversas mensagens varia dependendo da interface pela qual a mensagem será enviada. Lembrando a técnica *Poison Reverse*, se uma rota será enviada pela mesma interface pela qual ela foi obtida, a sua métrica será infinita (16 *hops*).

O fluxograma da figura 5.12 ilustra o processo de adição de rotas em uma mensagem RIP *response*.

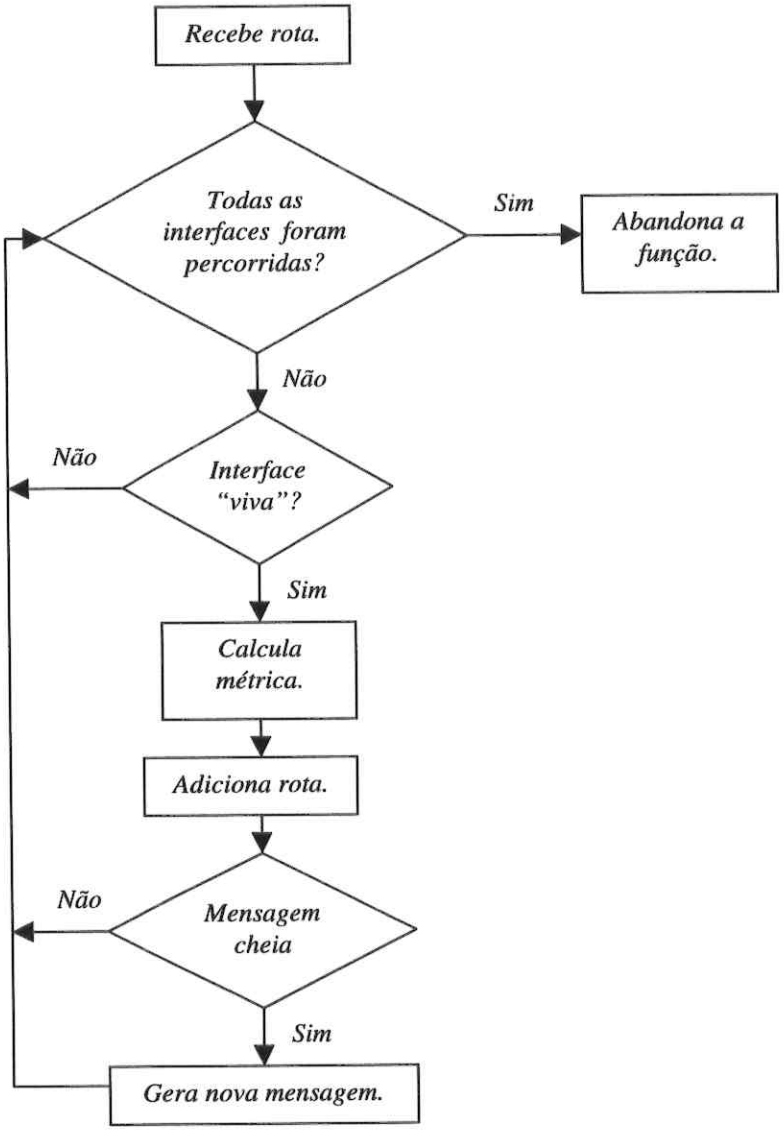


Figura 5.12 - Processo de adição de rotas.

Ao receber uma rota para ser adicionada, o protocolo analisa a rota para cada interface de rede. Primeiramente, ele verifica se a interface de rede esta "viva" (funcionando). Então, calcula a métrica da rota. Neste cálculo aplica-se a técnica *Poison Reverse*. Caso a métrica não esteja definida como infinita, o protocolo incrementa esta métrica em 1 *hop*. A partir de então a rota analisada é adicionada na mensagem pertencente à interface analisada.

Se o tamanho máximo de rotas em uma mensagem RIP é atingido, ou seja, 25 rotas por mensagem, então uma nova mensagem RIP é inicializada para serem inseridas as rotas restantes.

O protocolo RIP estudado e implementado neste projeto tem como objetivo difundir informações de roteamento. O algoritmo deverá interoperar com o programa *routed* implementado no sistema operacional UNIX.

Capítulo VI: *IPSwitch* da CIANET

O *IPSwitch* da CIANET, neste projeto denominado de *Switch de Nível 3*, acompanha a onda tecnológica da nova geração de dispositivos de conectividade. Tido como sucessor dos *switches*, os *IPSwitches* vêm invadindo o mercado de redes de computadores e prometendo ser a base das futuras tecnologias e aplicativos para redes.

O *switch de nível 3* da CIANET apresenta a mesma arquitetura de barramento de seus antecessores e as características da tecnologia *IPSwitching*. Porém, este produto ainda está em fase de implementação e simulações a nível de *software*, tendo como previsão da sua primeira versão para o segundo semestre do ano de 1999.

Sendo assim, o presente capítulo descreve a implementação de um *switch de nível 3* a nível de *software*. No capítulo III foram descritos dois tipos de implementação de um *IPSwitch*: baseada em fluxo de informações e baseada em nível físico. O *switch de nível 3* deriva da segunda implementação, seguindo a linha dos *switches* da CIANET.

Para um melhor entendimento será explicado na próxima seção como o *switch* da CIANET, aqui denominado de *Switch de Nível 2*, funciona.

6.1) *Switch de Nível 2*

Como descrito no capítulo III, um *switch de nível 2* é um concentrador de redes de computadores, cuja principal função é fornecer um *link* dedicado à velocidade *wire-speed* entre computadores e segmentos de uma *LAN*.

Para realizar este *link* o *switch de nível 2* deve analisar o endereço físico destino contido em cada *frame* e associá-lo à porta correta. Porém, como o *switch* saberá qual é a porta correta? Neste caso, é necessário analisar também o endereço físico fonte de cada *frame*. O seu procedimento será explicado adiante.

O *switch de nível 2* possui uma tabela a nível físico, denominada **Tabela de Nível 2,** onde é feito o mapeamento entre o endereço físico e a porta (interface) do *switch*. A tabela de nível 2 é uma tabela *HASH*, cujos índices são obtidos pelos 12 bits menos significativos do endereço físico. Deste modo, é possível mapear um total de 16.384 endereços físicos. Ou seja,

$$\begin{cases} 12 \text{ bits} = 2^{12} = 4096 \text{ índces} \\ 1 \text{ índice possui } 4 \text{ entradas} \end{cases} \Rightarrow 4096 \times 4 = 16384 \text{ endereços mapeados}$$

As entradas são compostas por três campos: o endereço físico, a porta associada e um *flag*. O *flag* é responsável por indicar se os dados desta entrada são válidas ou não; se o *flag* for nulo esta entrada é inválida, caso contrário (se ele for 1), a entrada é válida. Para cada índice há 4 entradas. Deste modo o custo de memória exigido para esta tabela é de 131.072 *bytes* ou 128 *Kbytes*, ou seja

$$\begin{cases} \text{Endereço físicos} = 6 \text{ bytes} \\ \text{porta} = 1 \text{ byte} \\ \text{flag} = 1 \text{ byte} \\ 16384 \text{ entradas} \end{cases} \Rightarrow 16384 \times (6+1+1) = 131072 \text{ bytes de memória}$$

A figura 6.1 mostra a tabela de nível 2, as entradas e como é obtido o índice da tabela *HASH*.

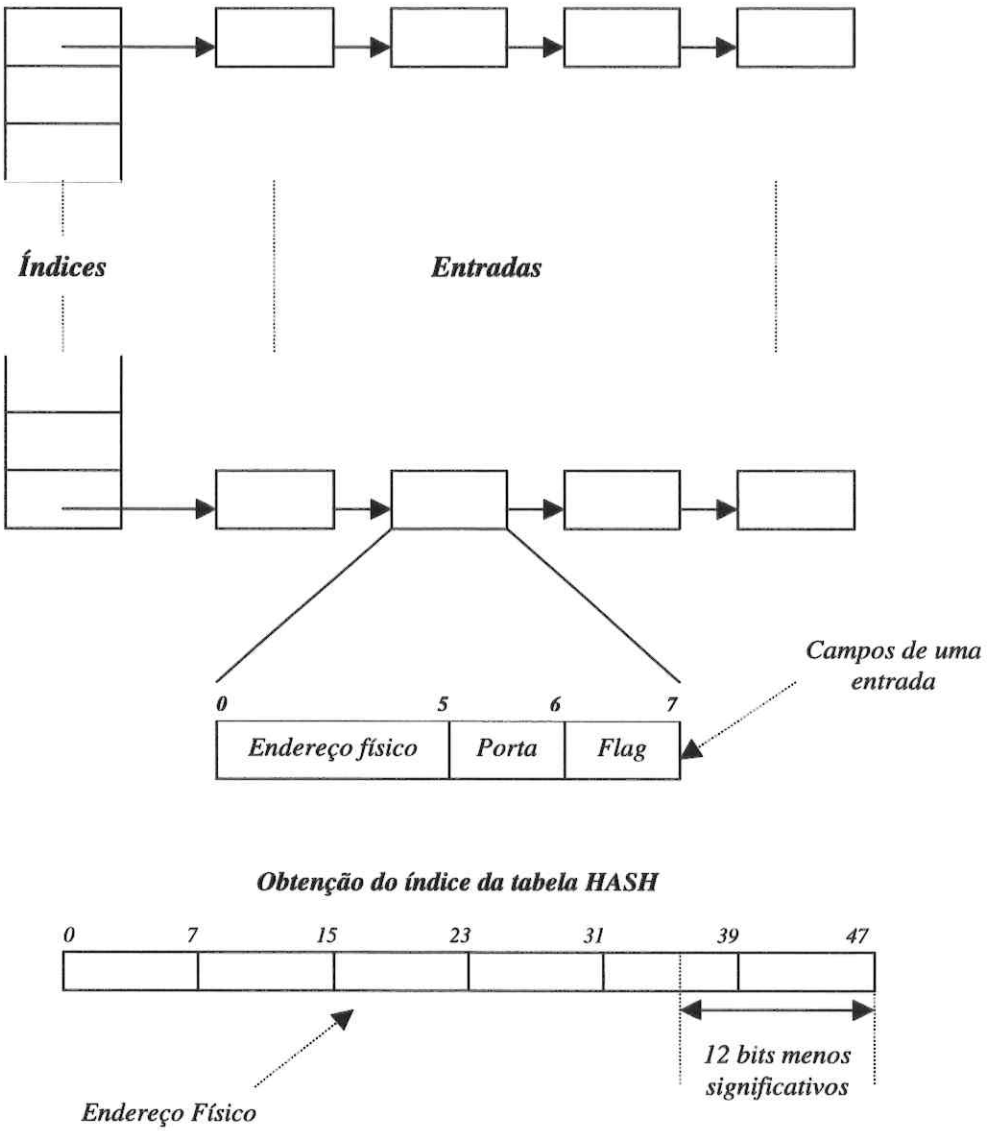


Figura 6.1 - Tabela de Nível 2.

Quando o *switch de nível 2* é inicializado (ligado), a tabela de nível 2 encontra-se totalmente vazia. Baseando-se no fluxograma da figura 6.2 será descrito o funcionamento de um *switch de nível 2* e como a tabela de nível 2 é preenchida.

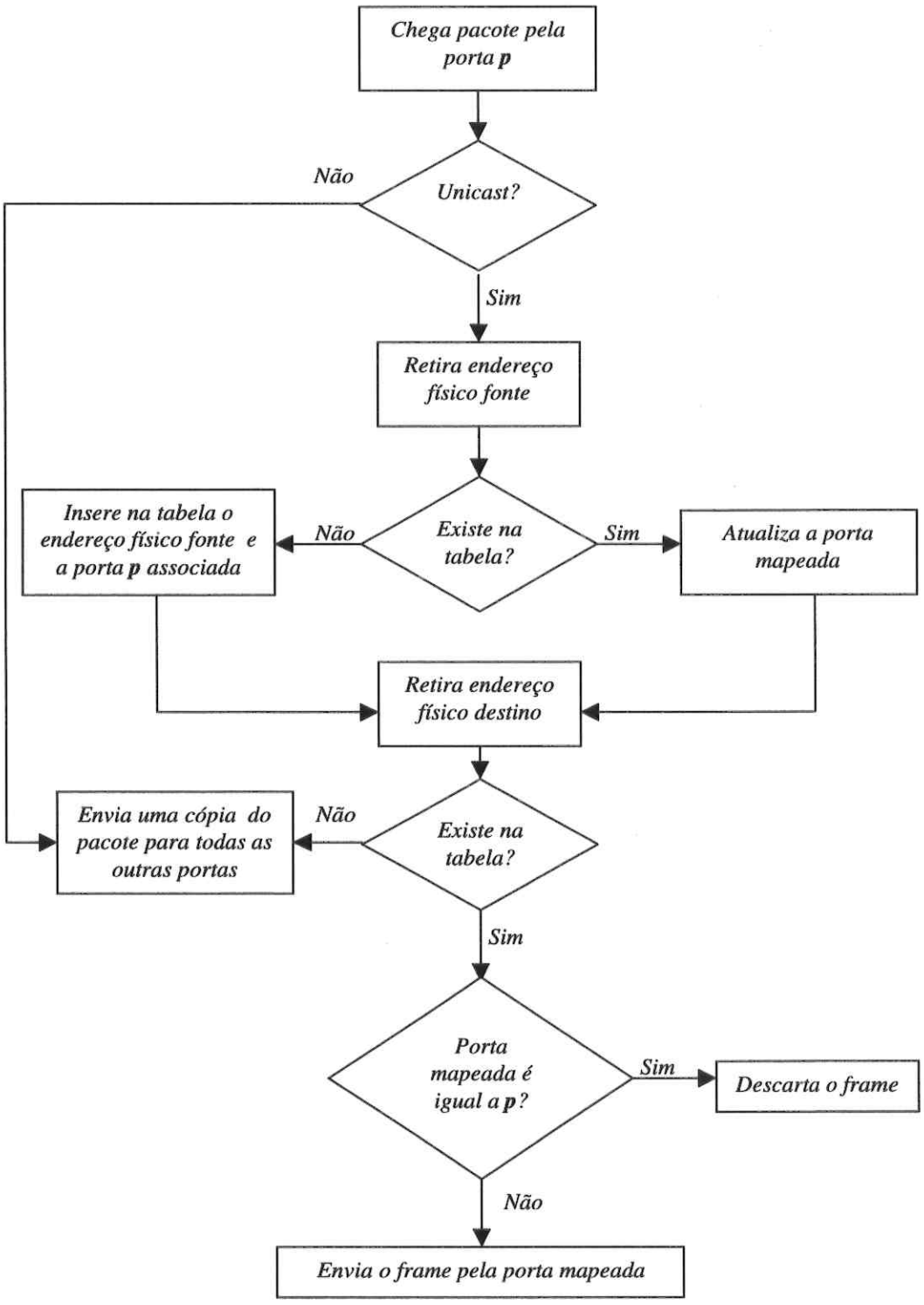


Figura 6.2 - Fluxograma das funções de um Switch de Nível 2.

Quando um *frame* chega por uma porta **p**, primeiramente verifica-se se ele é *unicast* ou um *broadcast*. Um *frame* é *unicast* se ele possui um endereço físico destino de uma máquina; ele será *broadcast* se o seu endereço físico destino for do tipo *broadcast*, ou seja, todos os *bits* setados para 1. Caso o *frame* seja um *broadcast*, o *switch* envia uma cópia dele para todas as outras portas.

Se o *frame* for *unicast*, o *switch* retira o endereço físico fonte e verifica se ele existe na tabela de nível 2. Se não existe ele é associado à porta **p** e ambos são inseridos na tabela e o *flag* da entrada em questão é setado para 1 (entrada válida). Se o endereço já existe, então a porta a ele mapeada é atualizada.

Quando a tabela de nível 2 fica cheia, as entradas antigas são sobrepostas por novas entradas, como uma fila circular.

O próximo passo é descobrir para qual porta deve-se enviar o *frame*. Para isto retira-se o endereço físico destino e verifica-se se ele existe na tabela. Se for encontrada uma tabela com este endereço, obtém-se assim a porta mapeada. Então, verifica-se se a porta mapeada é igual à porta pela qual o *frame* chegou (porta **p**). Se afirmativo, o *frame* é descartado. Caso contrário, ele é enviado pela porta mapeada.

Caso o endereço físico destino não seje encontrado na tabela de nível 2, ele será enviado para todas as outras portas, de forma semelhante a um pacote *broadcast*.

As funções acima descritas foram todas implementadas a nível físico, o que faz com que os *switches de nível 2* operem a uma velocidade *wire-speed*. Neste projeto, tais funções foram implementadas a nível de *software*, porém com algumas modificações com relação à tabela de nível 2. No caso do *software*, o índice da tabela se baseia apenas nos 10 *bits* menos significativos fornecendo assim um número de 4096 entradas e 32 *Kbytes* de memória. Esta escolha se deu devido à limitação de memória do computador.

6.2) *Switch de Nível 3*

Como citado anteriormente, o *Switch de Nível 3* deriva dos *IPSwitches*, cuja implementação é baseada em nível físico, ou seja, uma tabela de roteamento é acessada à nível físico. Porém, esta tabela não é diretamente preenchida por um protocolo de roteamento dinâmico (por exemplo: RIP).

O *switch de nível 3*, neste projeto implementado e simulado, mantém as mesmas características do *switch de nível 2* mais a tecnologia *IPSwitching*. Assim, a tabela de nível 2 e as funções do *switch de nível 2* foram mantidas.

No *switch de nível 3* foi implementada uma outra tabela, denominada **Tabela de Nível 3,** onde ocorre o mapeamento do endereço IP com o endereço físico e a porta em questão. Além desta tabela, que é implementada a nível físico, há a presença da pilha de protocolos UDP/IP e do protocolo de roteamento dinâmico RIP, implementados à nível de *software*.

A tabela de nível 3 é uma tabela *HASH* cujos índices são obtidos pelos 12 *bits* menos significativos do endereço IP. Cada índice é também composto por 4 entradas, onde

cada uma é composta por 4 campos. O endereço IP, o endereço físico e a porta mapeados e um *flag*. O *flag* possui a mesma função do *flag* da tabela de nível 2. Assim, o custo de memória exigido por esta tabela é de 196.608 *bytes* ou 192 *Kbytes*, ou seja

$$\begin{cases} \text{Endereço IP} = 4 \text{ bytes} \\ \text{Endereço físico} = 6 \text{ bytes} \\ \text{porta} = 1 \text{ byte} \\ \text{flag} = 1 \text{ byte} \\ 16384 \text{ entradas} \end{cases} \Rightarrow 16384 \times (4+6+1+1) = 196608 \text{ bytes de memória}$$

A figura 6.3 ilustra a tabela de nível 3, as entradas e como é obtido o índice da tabela *HASH*.

O processo de preenchimento da tabela de nível 3 ocorre em duas situações.

A primeira independe do protocolo RIP. Toda vez que um pacote ARP chega, analisa-se os endereços IP e físico da máquina fonte e a interface de rede pela qual o pacote ARP chegou.

A segunda situação ocorre quando, após rotear um pacote para o destino correto, o protocolo analisa os endereços IP e físico destino e a porta (interface) pela qual o pacote será enviado. Neste caso, o roteamento é feito à nível de *software*, pelo protocolo de roteamento IP, e baseia-se na tabela de roteamento preenchida pelo protocolo de roteamento dinâmico RIP.

A análise dos endereços é ilustrada pelo fluxograma da figura 6.4.

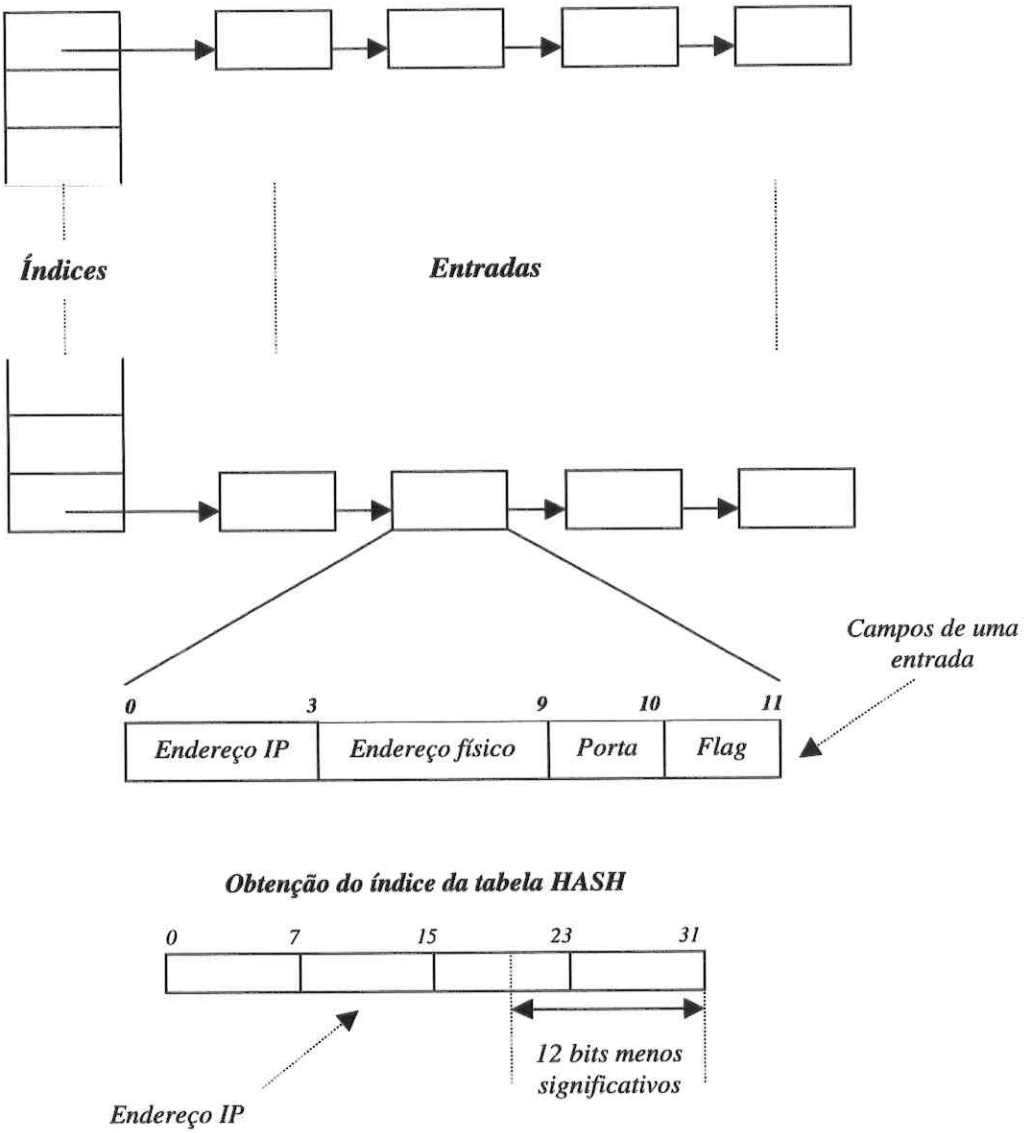


Figura 6.3 - Tabela de Nível 3.

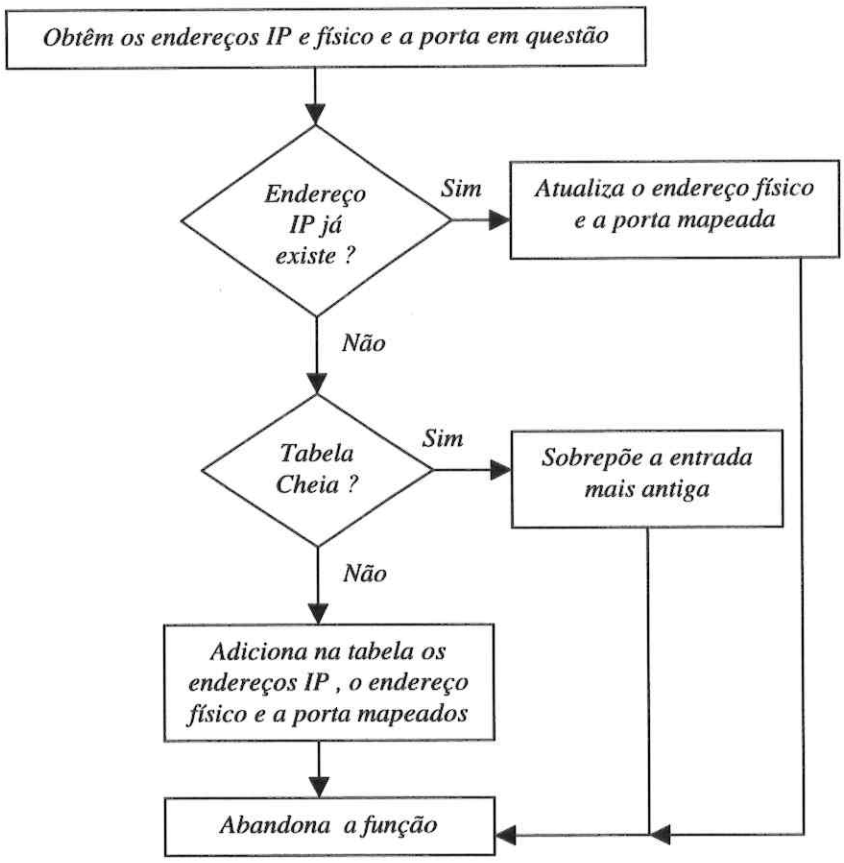


Figura 6.4 - Processo de análise dos endereços IP e físico e a porta em questão.

Primeiramente, verifica-se se o endereço IP existe na tabela de nível 3. Se afirmativo, então os endereços físico e a porta mapeada são atualizados. Caso contrário, verifica-se se a tabela está cheia, ou seja, se as quatro entradas pertencentes ao mesmo índice já estão ocupadas. Se verdadeiro, então a entrada mais antiga é sobreposta com os novos dados. Neste caso as entradas funcionam como uma fila circular. Caso a tabela não esteja cheia, então adiciona-se os endereços IP, o endereço físico e a porta mapeados.

O roteamento a nível físico, executado pelo *switch de nível 3*, é feito baseando-se no endereço de uma máquina destino e não apenas no endereço de uma rede destino. Analisa-se todo o endereço IP (*netid* + *hostid*) e, a partir da tabela de nível 3, obtêm-se o endereço físico destino e a interface pela qual o *frame* será enviado.

Ocorrem situações em que o endereço físico destino não corresponde à máquina, a qual o endereço IP pertence. Os endereços IP destino e físico destino correspondem à mesma máquina somente quanto esta pertence à uma das redes, que o *switch de nível 3* está diretamente conectado. Nesta situação o *switch de nível 3* faz um roteamento direto. Caso contrário, o *switch de nível 3* executa o roteamento indireto, ou seja, envia o *frame* para o próximo passo (outro elemento roteador).

As funções definidas e implementadas no *switch de nível 3* são apresentadas no fluxograma da figura 6.5.

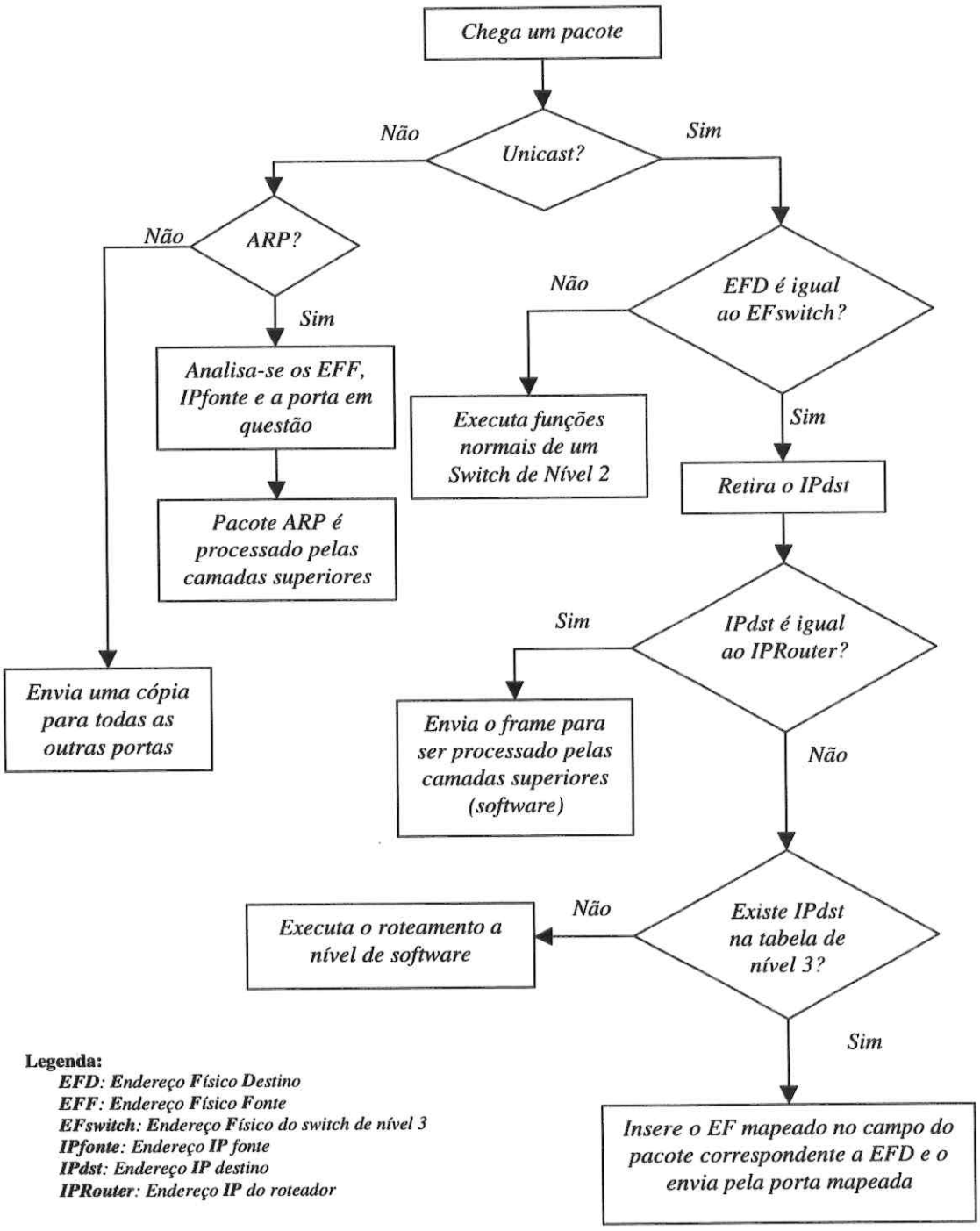


Figura 6.5 - Fluxograma das funções de um Switch de Nível 3.

Quando o *switch de nível 3* recebe um *frame*, verifica se este é do tipo *unicast*. Se o *frame* não for *unicast*, ou seja, se ele for do tipo *broadcast*, então verifica se ele é um pacote ARP.

Se verdadeiro, analisa os endereços IP e físico da máquina fonte e a porta, pela qual o *frame* chegou. Esta análise decidirá como a tabela de nível 3 será preenchida, ou seja, se uma nova entrada será adicionada ou se a tabela será apenas atualizada. Após esta análise, o pacote ARP é processado pelas camadas superiores (protocolo ARP).

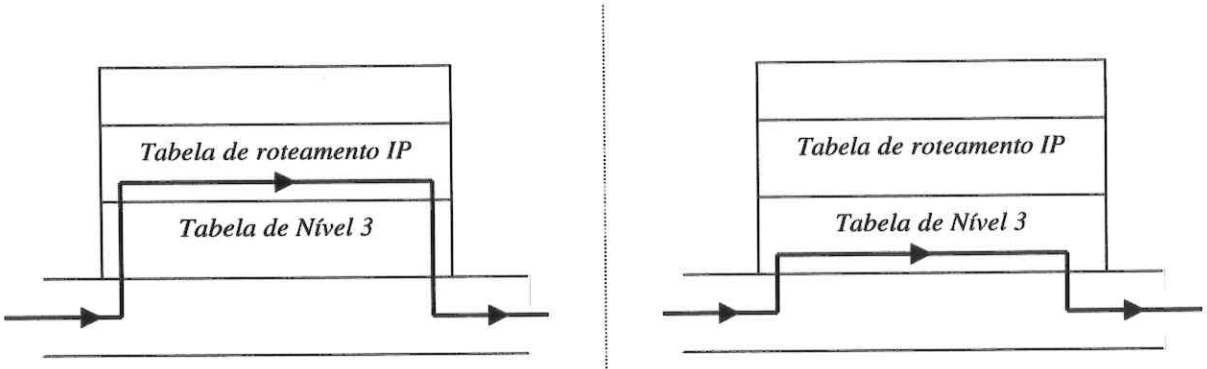
Nesta situação, onde ocorre o preenchimento da tabela de nível 3, os endereços IP mapeados correspondem às máquinas pertencentes às redes, às quais o *switch de nível 3* está diretamente conectado.

Se o *frame* for *unicast*, verifica se o endereço físico destino é igual ao endereço físico do *switch de nível 3*, ou seja, o endereço físico da porta pela qual o *frame* chegou. Se não forem iguais, o *switch de nível 3* executa as mesmas funções do *switch de nível 2*.

Caso contrário, verifica se o endereço IP destino é igual à um dos endereços IP do *switch de nível 3* (como roteador, possui mais de um endereço IP). Se verdadeiro, o *frame* é processado pelas camadas superiores (à nível de *software*). Um exemplo desta situação é a chegada de mensagens ICMP do tipo *ECHOrequest* (*ping*), ou de pacotes ARP do tipo *ARPreply*.

Caso contrário, verifica se o endereço IP destino se encontra na tabela de nível 3. Se for encontrado, então o endereço físico destino do *frame* é trocado pelo endereço físico mapeado e o *frame* é enviado pela porta mapeada. Se o endereço IP destino não for encontrado na tabela de nível 3, o *frame* será roteado pelo protocolo de roteamento IP.

Nesta situação, também ocorre o preenchimento e atualização da tabela de nível 3. Quando o *frame* não é roteado pela tabela de nível 3, ele é então roteado a nível de *software*, baseando-se na tabela de roteamento IP, lembrando que a tabela é preenchida pelo protocolo de roteamento dinâmico RIP. Após o *frame* ser roteado, o *software* analisa o endereço IP destino , o endereço físico da máquina destino ou do próximo passo (roteador) e a porta, através da qual o *frame* será enviado. Assim, os próximos *frames* que possuírem o mesmo endereço IP destino serão roteados pela tabela de nível 3. A figura 6.6 ilustra este processo da tabela de nível 3.



1) O primeiro frame é roteado a nível de software (tabela de roteamento IP) e a tabela de nível 3 é preenchida.

2) Os frames subsequentes são roteados a nível físico, baseado-se na tabela de nível 3.

Figura 6.6 - Processo de preenchimento da Tabela de Nível 3.

Nesta situação, os endereços IP mapeados correspondem às máquinas pertencentes às redes às quais o *switch de nível 3* não está diretamente conectado.

Quando o *switch de nível 3* é inicializado (ligado), o roteamento dos primeiros *frames* é feito a nível de *software*, com um desempenho relativamente baixo, próprio do processamento de roteamento por *software*. Porém o baixo desempenho dura pouco, pois à medida que a tabela de nível 3 é preenchida, as informações das redes de computadores passam a ser roteados a uma velocidade *wire-speed*.

Todas as funções do *switch de nível 3*, com exceção das pertencentes às camadas superiores, serão implementadas a nível físico na primeira versão do produto.

A realização deste projeto foi dividida em etapas, a fim de facilitar a resolução de problemas e a realização de testes e consequentemente a validação de todo o projeto. A seção seguinte descreve os testes de validação realizados.

6.3) *Validação do Projeto*

O projeto do *Switch de Nível 3* da CIANET foi dividido em 5 etapas.

1ª Etapa : Estudo dos protocolos que serão utilizados neste projeto;

2ª Etapa : Implementação de um roteador "estático";

3ª Etapa : Implementação de um *Switch de Nível 2*;

4ª Etapa : Implementação de um *Switch de Nível 3* "estático";

5ª Etapa : Implementação de um *Switch de Nível 3* "dinâmico".

Todas as implementações realizadas nas etapas acima citadas foram feitas a nível de *software*, utilizando a linguagem de programação C.

Os testes de validação foram realizados no final de cada etapa (com exceção da 1ª etapa) e serão descritos a seguir.

➢ *Estudo dos protocolos que serão utilizados neste projeto*

O estudo dos protocolos utilizados no projeto, principalmente os protocolos IP e RIP, promoveram um entendimento melhor sobre o funcionamento da *Internet*. No caso deste projeto, a utilização destes protocolos e das diversas definições anteriormente descritas, restringem-se a redes de alcance local (*LAN*), tais como corporações, campus universitários, etc. Sendo assim, o grau de desempenho e confiabilidade existente nestes tipos de redes são muito maiores, evitando assim os incidentes que foram descritos, e que geralmente ocorrem em redes de médio (*MAN*) e grande (*WAN*) alcance.

Esta etapa caracterizou-se por uma elevada carga de informações teóricas. Um resumo do que foi abordado encontra-se descrito neste documento, nos capítulos IV, V e VI.

➢ *Implementação de um roteador "estático"*

Nesta etapa foi implementado um roteador estático. A palavra estático significa que uma vez definidas as rotas, não é possível alterá-las em tempo real. Deste modo, a atualização da tabela de roteamento é feita, em geral, pelo administrador da rede. Assim, após a sua implementação utilizou-se da topologia de rede da figura 6.7 para testar o roteador.

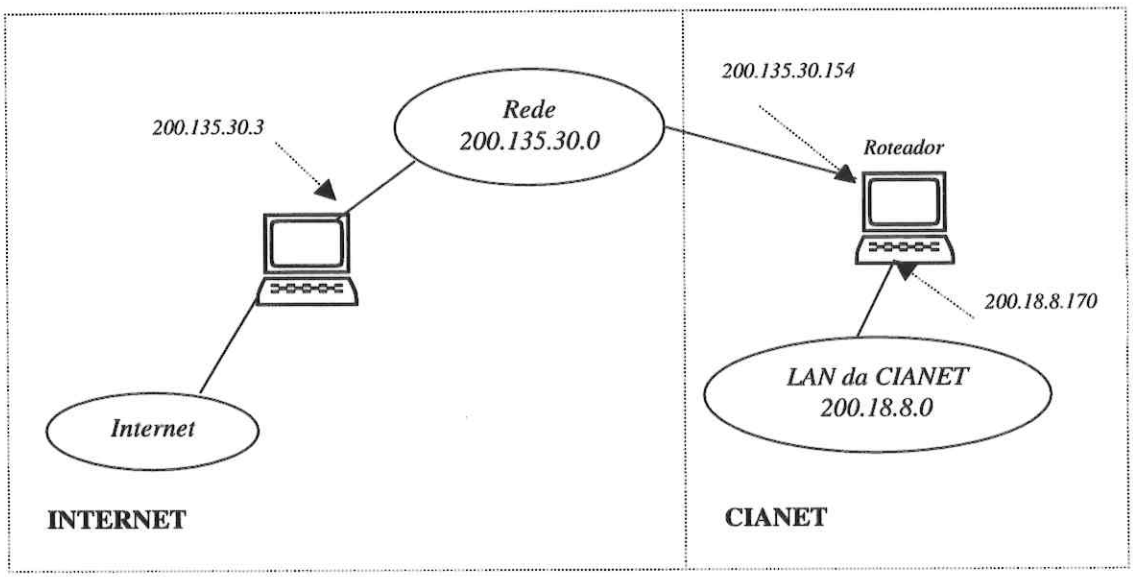


Figura 6.7 : LAN da CIANET com o roteador "estático".

A CIANET possui apenas um endereço IP que possibilita o acesso à *Internet*. Assim, existia apenas um computador conectado à grande rede virtual. As outras máquinas não podiam se comunicar, pois pertenciam a uma rede diferente.

Na inserção do roteador "estático", uma interface foi conectada à *Internet* e a outra à rede local da CIANET. Deste modo foram inseridas uma rota para a rede local, cujo *netid* é 200.18.8.0, e uma rota para a rede, cujo *netid* é 200.135.30.0. Para as informação não destinadas à ambas as redes, adicionou-se uma rota padrão, cujo próximo passo é o roteador ou *gateway* de endereço 200.135.30.3.

A figura 6.8 ilustra a tabela de roteamento do roteador da CIANET.

Para alcançar máquinas em redes	***Roteia-se os pacotes para o endereço IP***
200.18.8.0	*Envio direto*
200.135.30.0	*Envio direto*
Qualquer outra rede	*200.135.30.3 (rota padrão)*

Figura 6.8 - Tabela de roteamento do roteador "estático".

O teste realizado foi o seguinte: todos os computadores da rede local da CIANET acessaram ao mesmo tempo a *Internet* com diversos aplicativos, por um período de aproximadamente 2 horas. Os aplicativos utilizados foram o *browser Netscape*, para

acessar *Home Pages*; *EUDORA*, para verificar *mails* em caixas postais; *CuteFTP*, para transferência de arquivos.

O resultado foi que apesar da alta latência proporcionado pelo computador e pelo *software* de roteamento, a máquina roteou as informações com um alto grau de confiabilidade.

➢ *Implementação de um Switch de Nível 2*

A rede local da CIANET é composta de 8 microcomputadores interligados por um *switch de nível 2* de 12 portas. Após a implementação do *switch de nível 2* em *software*, equipou-se um microcomputador com 2 interfaces de rede. Conectou-se uma máquina em uma interface e a outra interface foi conectada ao *switch*. A figura 6.9 ilustra a topologia montada.

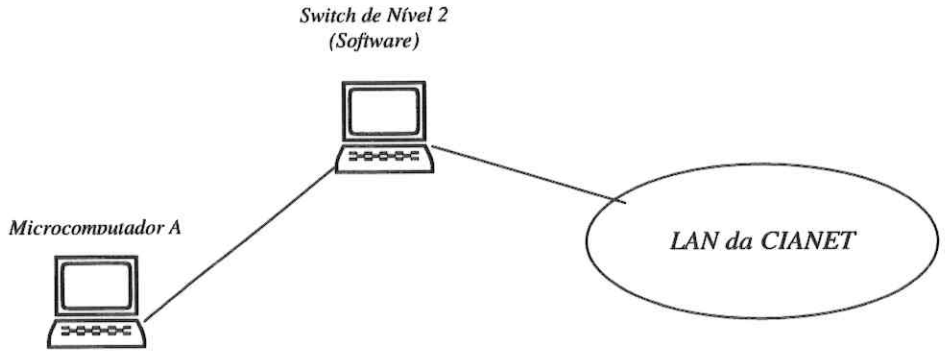


Figura 6.9 - Topologia para testar as funções do Switch de Nível 2.

Então, através do microcomputador A acessou-se a LAN da CIANET. Pelo próprio computador, onde foi implementado o *switch de nível 2*, monitorou-se a tabela de nível 2.

O resultado foi o esperado, o *software* simulou as funções de um *switch de nível 2*. Logo, o teste foi um sucesso.

➢ *Implementação de um Switch de Nível 3 "estático"*

A primeira fase de desenvolvimento de um *Switch de Nível 3* baseou-se da implementação de todas as suas funções, realizadas a nível físico, e a integração com a pilha de protocolos UDP. O protocolo de roteamento dinâmico RIP não foi implementado neste primeiro estágio. Deste modo o *switch de nível 3* operou através de um roteamento estático. O teste foi o mesmo realizado na validação do roteador "estático". Utilizou-se da

mesma topologia da figura 6.7 e da mesma tabela de roteamento da figura 6.8. Através do computador, onde o *switch de nível 3* foi simulado, monitorou-se a tabela de nível 3.

Com este teste observou-se no primeiro instante que todo o roteamento era feito a nível de protocolo de roteamento IP (Camada de Rede – Protocolo IP). Porém, à medida que a tabela a nível 3 foi sendo preenchida, o roteamento passou a ser realizado sem a intervenção da pilha de protocolos UDP.

No produto final, será possível notar a diferença de latência entre o início, quando o *switch de nível 3* é inicializado e a tabela de nível 3 está vazia, e os períodos subsequentes, quando a tabela de nível 3 é preenchida.

O próximo passo foi a validação do protocolo de roteamento dinâmico RIP, descrito a seguir.

➢ *Implementação de um Switch de Nível 3 "dinâmico"*

Esta fase final de desenvolvimento do *Switch de Nível 3* baseou-se na implementação e validação do protocolo de roteamento dinâmico RIP. Após implementado o protocolo RIP, o *switch de nível 3* adquiriu algumas características essenciais, tais como:

1. Quando o *switch de nível 3* é inicializado, ele envia uma mensagem RIP de requisição de todas as informações contidas nas tabelas de roteamento dos roteadores vizinhos participantes;
2. A cada 30 segundos ele envia uma mensagem RIP de atualização de rotas à todos os roteadores vizinhos participantes;
3. Depois de 180 segundos, a rota que não foi atualizada entra no *garbage-collection time*. Após a expiração de tempo (120 segundos) a rota é eliminada;
4. O *switch de nível 3* fornece informações de determinadas rotas.

Os testes finais consistiram na validação de tais características.

Primeiramente era necessário saber da veracidade das mensagens RIP enviadas pelo *switch de nível 3*. Então criou-se um programa denominado **LADRÃO**, o qual era responsável por "roubar" todas as mensagens RIP de uma rede de computadores. Em seguida configurou-se um roteador UNIX, para que, através do programa *routed*, trocasse mensagens de roteamento dinâmico. Com o LADRÃO capturou-se as mensagens RIP do *switch de nível 3* e comparou-as com as obtidas da máquina UNIX (ambas com as mesmas configurações de rotas e topologia de rede). O resultado foi o esperado, ambas eram idênticas, tanto as mensagens RIP *request* como as mensagens RIP *response*.

O próximo passo foi a verificação das 3 primeiras características citadas. Para isso, montou-se a topologia de rede mostrada na figura 6.10.

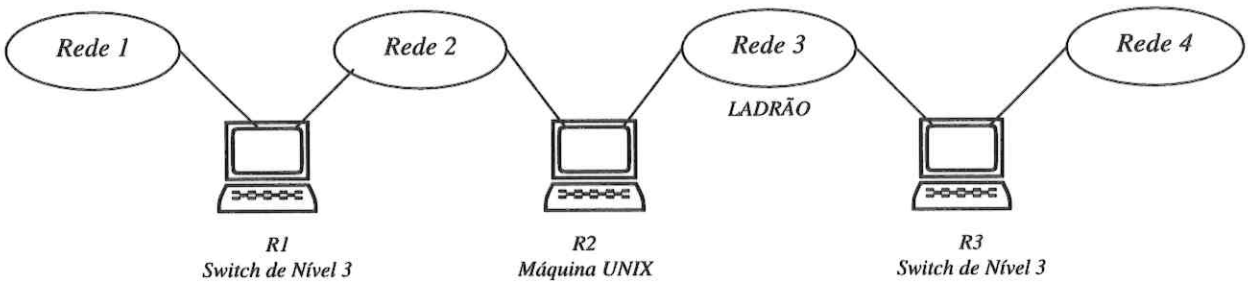


Figura 6.10 - Topologia utilizada para testar o Switch de Nível 3 "dinâmico".

Baseando-se na topologia da figura 6.10, as tabelas de roteamento dos *switches de nível 3* R1 e R3 e da máquina UNIX R2 foram configuradas com as rotas estáticas (para redes, nas quais ele estão diretamente conectados) e rota padrão. Para as rotas padrão tivemos a seguinte configuração:

Switch de Nível 3 ou máquina UNIX	***Rota padrão (próximo passo)***
R1	*R2*
R2	*R3*
R3	*Nenhuma rota padrão*

A partir de cada microcomputador, onde o *switch de nível 3* foi simulado, monitorou-se a tabela de roteamento. Assim ao serem inicializados, as tabelas de roteamento de R1 e R3 eram compostas pelas seguintes rotas:

Switch de Nível 3 – R1

Destino	***Próximo passo***	***Custo***
Rede 1	*Direto*	*1*
Rede 2	*Direto*	*1*
Outro	*R2*	*2*

Switch de Nível 3 – R2

Destino	***Proximo passo***	***Custo***
Rede 3	*Direto*	*1*
Rede 4	*Direto*	*1*
Outro	*Inexistente*	-

Ao serem inicializados, cada *switch de nível 3* envia uma mensagem RIP de requisição de todas as informações contidas nas tabelas de roteamento do *switch de nível 3* vizinho e do roteador UNIX. Esta requisição tem por objetivo obter informações sobre todas as redes de um sistema autônomo.

Nos primeiros momentos de troca de informações foi observado o preenchimento das tabelas de roteamento. Este período foi de certo modo rápido devido à técnica *triggered*

updates. Após este período as tabelas não se modificam mais, a não ser que alguma mudança ocorra na topologia da rede. As tabelas de roteamento dos *switches de nível 3* ficaram do seguinte modo:

Switch de Nível 3 – R1

Destino	*Próximo passo*	*Custo*
Rede 1	*Direto*	*1*
Rede 2	*Direto*	*1*
Rede 3	*R2*	*2*
Rede 4	*R2*	*3*
Outro	*R2*	*2*

Switch de Nível 3 – R2

Destino	*Proximo passo*	*Custo*
Rede 3	*Direto*	*1*
Rede 4	*Direto*	*1*
Rede 1	*R2*	*3*
Rede 2	*R2*	*2*
Outro	*Inexistente*	-

Pelo *software* LADRÃO, localizado em um computador da rede 3 (figura 6.10), capturou-se mensagens RIP de atualização de rotas das máquinas R2 e R3.

O próximo passo foi desconectar R1 da rede 2 e observar a mudança automática da tabela de roteamento. Após 180 segundos sem receber informações de R1, R2 coloca a rota para a rede 1 no *garbage-collection time* e os seus custos são setados para infinito (16 *hops*). Deste modo, há uma troca intensa de mensagens RIP de atualização de rotas (*triggered updates*), colocando a rota para a rede 1 das outras tabelas no *garbage-collection time* e seus custos para infinito.

Após expirado o *garbage-collection time* (120 segundos) rota para a rede 1 foi eliminada.

O mesmo processo ocorreu na tabela de roteamento do R1, com relação à rota para a rede 4.

Depois de estabilizada a nova topologia de rede, através de um computador localizado na rede 4, tentou-se dar um *ping* em um computador da rede 1. Porém, como R3 não possui mais a rota para a rede 1 e também não possui rota padrão, uma mensagem de advertência (*"destination unreachable"* – protocolo ICMP) foi recebida. Então conectou-se novamente R1 à rede 2 e esperou-se o preenchimento automático das tabelas de roteamento com as informações da nova topologia. Em seguida tentou-se novamente acessar uma máquina da rede 1 através do *ping*, recebendo desta vez uma resposta positiva.

Há momentos em que o administrador da rede ou o próprio roteador necessitam obter informações de rotas específicas. Para tanto, foi implementado um programa que requisitasse determinadas rotas e, através do mesmo computador localizado na rede 4, foram obtidas informações sobre determinadas rotas ou grupo delas.

Deste modo, os testes realizados foram bem sucedidos.

Capítulo VII: Conclusões e Perpectivas

A construção de um dispositivo de conectividade utilizando a tecnologia *IPSwitching* está entre as mais recentes soluções para suprir a demanda de elementos roteadores aplicados às redes corporativas, e que apresentem um alto desempenho com alta confiabilidade e a um baixo custo.

Assim sendo, o trabalho aqui apresentado tem por objetivo a implementação de uma parte do projeto, o qual se refere ao roteamento de pacotes. Mais especificamente ao estudo e implementação do protocolo de roteamento dinâmico RIP - *Routing Information Protocol*. Além disso, foram implementadas a nível de *software* as características básicas de um dispositivo *IPSwitch*.

Inicialmente foi estudada a tecnologia *IPSwitching* e as principais variações dos *IPSwitches*. Deste modo, foi possível escolher o método de implementação da tecnologia *IPSwitching* mais apropriado à linha de *Switches* da CIANET.

A fase seguinte consistiu no estudo e familiarização dos protocolos de comunicação de dados que compõem a pilha UDP/IP, pois tais protocolos são a base para os protocolos de roteamento dinâmico.

Em seguida foram estudados os principais protocolos de roteamento dinâmico e, dentre eles, optou-se pelo RIP. O protocolo RIP foi implementado baseando-se nos protocolos de comunicação de dados acima citados, objetivando a implementação da tecnologia *IPSwitching* escolhida.

Por fim, implementou-se à nível de *software* um *IPSwitch*, aqui denominado de *Switch de Nível 3*, o qual serviu de base para a depuração do protocolo RIP.

O uso do protocolo de roteamento dinâmico RIP tem como principais benefícios:

- ✓ A sua popularidade e conseqüente compatibilidade com as demais versões do protocolo;
- ✓ A facilidade de implementação , depuração e posterior mudança de versão, resultando assim numa boa relação CUSTO X BENEFÍCIO X TEMPO DE IMPLEMENTAÇÃO para redes corporativas (pequeno e médio porte);
- ✓ Fácil integração à pilha de protocolos UDP/IP implementada pela CIANET;
- ✓ Confiabilidade necessária para redes corporativas.

Do ponto de vista da experiência profissional adquirida, este projeto proporcionou o contato e aprendizado com o que há de mais recente em técnicas de roteamento e em dispositivos roteadores. Além disso, favoreceu o desenvolvimento da responsabilidade profissional em tomar decisões com relação à melhor tecnologia, protocolo e métodos de programação, baseando-se no mercado de redes de computadores e na experiência de profissionais altamente capacitados.

Em vista disso foi possível aplicar conhecimentos adquiridos no curso de Eng. de Controle e Automação Industrial, mais especificamente, conhecimentos relacionados às áreas de Informática Industrial e Engenharia de Software.

Na perspectiva de dar continuidade ao projeto, outros temas seriam ainda abordados. Entre eles:

- ✓ RIPv2 - *Routing Information Protocol*: Roteamento dinâmico a nível de sub-redes;
- ✓ NAT - *Network Address Translation*: Tradução de endereços de rede.

Nesse sentido a próxima etapa seria o estudo e a implementação do RIPv2, o qual facilitaria o roteamento de informações para somente uma parte de uma rede, denominada sub-rede.

A utilização deste protocolo promoveria a utilização de *virtual LANs*, ou seja, redes virtuais. Neste caso, a divisão de uma rede em sub-redes se dá apenas virtualmente (a nível de *software*).

Com a escassez de endereços IP e também como questão de segurança, a implementação do protocolo NAT seria de grande utilidade. Tal protocolo possibilita que uma corporação possua somente um endereço IP globalmente único em uma máquina e endereços IP arbitrários nas outras máquinas.

O NAT, localizado na máquina com o endereço IP válido, simplesmente traduz os outros endereços, possibilitando o acesso das máquinas à *Internet*. Deste modo, qualquer pessoa localizada fora da corporação e que quiser acessar uma máquina dela não conseguirá, pois não saberá o endereço IP correto.

Apêndice A: Projeto de um Sistema de Comunicação de Dados

Para que haja uma comunicação de dados entre dois computadores é necessário definir um conjunto de regras, denominado **PROTOCOLO**. Tais regras definem o formato da informação trocada, o teste de veracidade desta, a correção de erros, a velocidade de transmissão, etc. Ou seja, duas máquinas que obedecem a um mesmo protocolo conseguem "conversar" entre si.

Assim, um protocolo deveria conter todas as características de uma comunicação confiável. Porém, o projeto de um sistema de comunicação de dados baseado em um único protocolo seria uma tarefa muito árdua para qualquer projetista. E mesmo se realizada, a depuração deste seria complexa e demorada. A solução está no desenvolvimento de um conjunto de protocolos, cada um trabalhando em cooperação com o outro [TANENBAUM 94].

Deste modo, o projeto de um sistema de comunicação de dados se torna altamente estruturado, tendo como objetivo reduzir a complexidade do projeto e os problemas que venham a surgir [TANENBAUM 94].

Para um melhor entendimento, tais sistemas são organizados em **CAMADAS** ou **NÍVEIS**, cada uma construída sobre a sua predecessora. Cada camada possui suas próprias atribuições e comunica-se com as camadas vizinhas através de interfaces.

Um exemplo de um sistema organizado em camadas é a *Internet*, a qual obedece o modelo de camadas da pilha de protocolos TCP/IP. O desenvolvimento do *IPSwitch* baseia-se neste modelo. A figura 4.1 ilustra o modelo em camadas conceituais da pilha de protocolos TCP/IP.

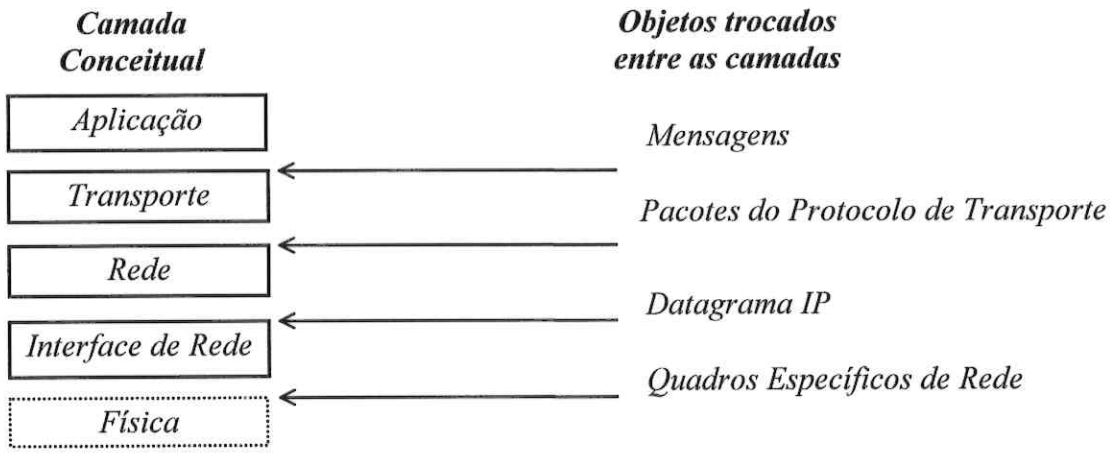


Figura A.1 - Camadas conceituais do modelo Internet TCP/IP.

Este modelo é dividido em 4 camadas conceituais, as quais são construídas sobre uma quinta camada: a de nível físico. A definição de camadas conceituais é a principal característica que torna viável a integração de diferentes tecnologias de rede.

Uma breve descrição de cada camada é feita a seguir.

- ***Application Layer***. Corresponde à camada de aplicação e é o nível mais alto. Usuários utilizam programas de aplicação para acessar os serviços disponíveis na rede TCP/IP. Esta camada interage com a camada de transporte para mandar e receber dados [COMER 91a]. Ex. SNMP (*Simple Network Management Protocol*), Telnet, etc.
- ***Transport Layer***. Corresponde à camada de transporte, cuja principal função é a de garantir a comunicação entre um programa em uma máquina e outro em outra máquina. Tal comunicação é freqüentemente denominada *end-to-end*. Esta camada deve regular o fluxo de transmissão, assim como garantir o envio e a recepção de dados sem erro. A camada de transporte deve garantir o acesso de múltiplos programas de aplicação à *Internet*. Assim, ela se encarrega de enviar os dados à camada de rede e fornecer as informações necessárias para identificar qual aplicativo está enviando os dados [COMER 91a]. Ex. UDP (*User Datagram Protocol*), TCP (*Transmission Control Protocol*).
- ***Internet Layer***. Corresponde à camada de rede e tem como funções empacotar os dados corretamente, roteá-los e, se necessário, fragmentá-los. As informações recebidas são roteadas pela camada de rede. No caso de o destino ser a própria máquina, ela envia os dados necessários para a camada de transporte. Finalmente, a camada de rede se encarrega de enviar mensagens ICMP de erro e controle [COMER 91a]. Ex. IP (*Internet Protocol*), ICMP (*Internet Control Message Protocol*).
- ***Network Interface Layer***. Corresponde à camada de interface de rede. Serve como uma interface entre o protocolo de rede e o meio físico em questão [COMER 91a]. Ex. ARP (*Address Resolution Protocol*).

No caso deste projeto os protocolos estudados foram:

- ✓ *Camada de Inteface de Rede:* ARP.
- ✓ *Camada de Rede:* IP, ICMP.
- ✓ *Camada de Transporte:* UDP.
- ✓ *Protocolos de Gerenciamento de Redes*: GGP, SPF, EGP, RIP, OSPF e HELLO.

Apêndice B: Modelo ISO/OSI [CYCLADES]

O modelo OSI - *Open System Interconnect* foi criado em 1977 pela ISO - *International Organization for Standardization* - com o objetivo de criar padrões de conectividade para interligar sistemas de computadores locais e remotos. Os aspectos gerais da rede estão divididos em 7 camadas funcionais e fundamentais, apesar de tal modelo não ser adotado para fins comerciais.

A figura A.1 mostra o modelo ISO/OSI e a atuação dos produtos de comunicação em cada uma das camadas desse modelo. O modelo ISO/OSI tem uma divisão muito clara das camadas de um sistema de comunicação. Este é um grande auxílio para o entendimento dos diversos protocolos de mercado.

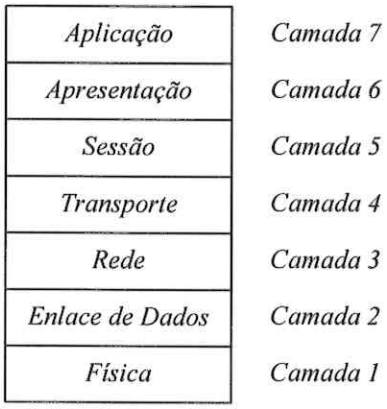


Figura B.1 - Modelo ISO/OSI.

- ***Camada Física***: A camada 1 compreende as especificações do *hardware* utilizado na rede (compreendido em aspectos mecânicos , elétricos e físicos – todos documentados em padrões internacionais). Exemplos: *Ethernet 802.3*, *RS-232*, *RS-449*, *V.22*, *V.35*, *X.21*.
- ***Camada de Enlace de Dados***: A "visão" da camada de enlace restringe-se a dois nós da rede somente. Os protocolos desta camada devem garantir que os dados transmitidos de um computador cheguem ao outro diretamente ligado a ele com integridade (controle de fluxo, correção de erros, ...).
- ***Camada de Rede***: Na camada de rede o "conhecimento" da rede passa a existir (topologia, como os nós estão conectados entre si). Protocolos desta camada tratam de encaminhar as mensagens na rede segundo algoritmos de roteamento, disciplinas de controle de fluxo e endereçamento. Exemplo: *IP*, *ISO Internetworking Protocol*.
- ***Camada de Transporte***: Os protocolos de transporte possuem uma "visão fim-a-fim" de comunicação. Eles devem garantir que os dados transmitidos de um computador a outro (não necessariamente ligados entre si) cheguem com integridade: controle de fluxo, correção de erros, ...
- ***Camada de Sessão***: A camada de sessão trata do "diálogo" entre dois computadores da rede. Detalhes como: tipo comunicação *half-duplex*, *full-duplex* ou *one-way*, estabelecimento de pontos de sincronismo na comunicação (por exemplo para recuperação de um conexão de transferência de arquivos) são tratados nessa camada.

- ***Camada de Apresentação***: Trata da sintaxe e semântica dos dados transmitidos entre dois computadores. Criptografia, conversão entre caracteres ASCII e EBCDIC, compreensão e descompressão de dados são algumas funções acumuladas nesta camada. Exemplo: ASN.1.
- ***Camada de Aplicação***: Trata da definição dos protocolos de aplicação propriamente ditos. É importante observar que esta camada **não** define como a aplicação final deve ser, mas sim o protocolo de aplicação correspondente. Exemplo: uma aplicação de transferência de arquivos em um computador estabelece comunicação com o usuário (para obter dados como nome dos arquivos, endereço do nó destino, etc.), com os gerenciadores de arquivos do sistema operacional (de onde e para onde ler e gravar os arquivos com computador e como) e com um gerenciador de comunicação que implementa as funções da camada 7 do modelo ISO/OSI (protocolo de transferência de arquivos no caso).

Glossário

ARP:

Address Resolution Protocol. (ver capítulo V)

Backbone:

Ou "espinha dorsal" corresponde a um roteador ou *gateway* onde as (sub-) redes estão conectadas. O *backbone* representa, muitas vezes, o núcleo de uma *internetwork* corporativa e a porta de saída para redes externas.

Core:

Corresponde ao núcleo de toda a *Internet*. O *core* forma o centro do roteamento da *Internet* para onde todos os *gateways* externos devem difundir rotas para suas respectivas redes. Estas rotas são enviadas aos *Core Gateways* utilizando o EGP

Core Gateway:

Gateway pertencente a um conjunto de *gateways* operado pelo Centro de Operacões da Rede Internet (INOC - *Internet Network Operations Center*). Os *gateways* pertencentes ao *core* ("núcleo") trocam informações de roteamento periodicamente entre si para assegurar que as tabelas de roteamento permaneçam consistentes.

Datagrama:

Unidade básica de tranferência de dados através da rede TCP/IP.

EGP:

Exterior Gateway Protocol. (ver capítulo V)

Ethernet:

Criado pela Xerox, corresponde a um padrão que define os níveis 1 e 2 (físico e lógico) especificados respectivamente pelas normas 802.3 e 802.2 da IEEE.

Extranet:

São empresas que oferecem acesso via *Internet* à sua *Intranet.*

Frame:

Corresponde ao conjunto de dados contidos entre o inicio e o final de uma transmissão serial através de uma rede. Cada *frame* contém campos de informações padronizados para que seja possível organizá-los e classificá-los dentro de uma rede.

Full-duplex:

Método de comunicação no qual uma estação pode enviar e receber dados simultaneamente.

Gateway:

Dispositivo usado para permitir a conexão de diferentes redes de computadores. Normalmente opera até a camada 4 do modelo ISO/OSI, convertendo assim informações entre diferentes redes, computadores e aplicações.

GGP:

Gateway to Gateway Protocol. (ver capítulo V)

Half-duplex:

Método de comunicação no qual uma estação pode tanto enviar quanto receber dados, porém não pode realizar estas duas operações simultaneamente.

ICMP:

Internet Control Message Protocol. (ver capítulo V)

INOC:

Internet Network Operations Center - Centro de Operações da Rede Internet.

Internet:

Conjunto de redes de computadores interligadas pelo mundo inteiro, que têm em comum um conjunto de protocolos e serviços, de forma que os usuários a ela conectados podem usufruir de serviços de informação de alcance mundial.

Internetworking:

Conjunto de várias redes de computadores comunicando-se entre si.

Intranet:

É uma rede corporativa que utiliza a tecnologia da *Internet*, ou seja, coloca um servidor *Web* para que funcionários possam acessar as informações da empresa através de um *browser* (navegador *Web*).

IP:

Internet Protocol. (ver capítulo V)

IPSwitch:

Dispositivo de conectividade que utiliza da tecnologia *IPSwitching*. Dependendo do fabricante possui diferentes nomes: *Routing Switch, L3 Switch, Fast IP,* etc.

IPSwitching:

Recente tecnologia que tem como filosofia unir característica de dois dispositivos de conetividade: a velocidade de transmissão de dados do *Switch* e a capacidade e confiabilidade em rotear informações do roteador . Esta tecnologia é aplicada em redes de pequeno e médio porte (LANs ou MANs).

ISO:

International Organization for Standardization.

Latência:

Tempo que um pacote de informações leva para passar através de um dispositivo de conectividade.

Link:

Corresponde à conexão entre dois computadores ou entre um computador e um dispositivo de conectividade (*Switch*, roteador, etc...).

Bibliografia

[COMER 91a] COMER, Douglas E. *Internetworking with TCP/IP Volume I* : Principles, Protocols, and Architecture . 2. ed . Englewood Cliffs , New Jersey : Prentice Hall , 1991.

[COMER 91b] COMER, Douglas E. ; STEVENS, David L. *Internetworking with TCP/IP Volume II* : Design, Implementation, and Internals . 1. ed . Englewood Cliffs , New Jersey : Prentice Hall , 1991.

[CYCLADES] CYCLADES. Guia Internet da Conectividade. São Paulo . 4. ed , SP : Cyclades Brasil, 1997.

[ITD 98] ITD Latinoamericana. Switching de nível 3 e Gigabit Ethernet : Soluções de backbone de alto rendimento : CCIP, v.3, n.1, 1998.

[LAN TIMES 97] LAN TIMES BRASIL. Fabricantes apostam em nova tecnologia para comutação. São Paulo : Rever, v.3, n.5, 1997.

[REDES 95] KEE, Eddie. *Redes de Computadores Ilustrada.* 1. ed . Rio de Janeiro , RJ : Axcel Books , 1995.

[RFC 1058] HEDRICK, C. *Routing Information Protocol.* RFC 1058 : Rutgers University, 1988.

[TANENBAUM 94] TANENBAUM, Andrew S. *Redes de Computadores.* 2. ed. Rio de Janeiro : Campus , 1994.