

# **Apostila TCP/IP**

**Wandreson Luiz Brandino**  
**[wandreson.com](http://wandreson.com)**  
**Wandreson@wandreson.com**  
**Setembro/1998**

# Índice

<b>1</b>	<b><i>Introdução ao TCP/IP</i></b>	<b>4</b>
1.1	Histórico	4
1.2	Documentação	5
<b>2</b>	<b><i>Endereço de Rede</i></b>	<b>6</b>
2.1	Endereço IP	7
2.2	Classes de Endereços	8
2.2.1	Endereço Loopback	9
2.2.2	Endereços IP reservados	10
2.3	Roteadores	10
2.4	Sub-rede	13
2.4.1	Máscara de uma Sub-Rede	13
2.5	Formato do datagrama IP	17
2.5.1	Algumas <i>OPTIONS</i> interessantes	20
<b>3</b>	<b><i>Mapeamento de endereço de rede em endereço do nível físico</i></b>	<b>22</b>
3.1	Protocolo ARP	22
3.1.1	Formato do Pacote ARP	24
3.2	Protocolo RARP	25
3.3	Fragmentação de endereço IP em endereço físico	25
<b>4</b>	<b><i>ICMP – Internet Control message Protocol</i></b>	<b>28</b>
4.1	Entrega de Mensagens ICMP	29
4.2	Tipos de Mensagem ICMP	29
4.2.1	Echo Request & Echo Reply	30
4.2.2	Reports unreachable destination	31
4.2.3	Controle de Fluxo e de Congestionamento	32
4.2.4	Route Change Request (Redirect)	32
4.2.5	Detecta rota circular ou excessivamente longa	33
4.2.6	Reporta outros problemas	34
4.2.7	Sincronização de relógio e estimativa de tempo de transito	34
<b>5</b>	<b><i>Camada de Transporte</i></b>	<b>35</b>
5.1	Aplicações Cliente-Servidor	35
5.1.1	O conceito de Porta	36
5.1.2	Utilizando a Arquitetura Cliente-Servidor	38
5.2	O Protocolo TCP (Transmission Control Protocol)	39
5.2.1	Formato do pacote TCP	40
5.2.2	Portas bem conhecidas do TCP	42
5.3	O Protocolo UDP (User Datagram Protocol)	43
5.3.1	Formato do Pacote UDP (User Datagram Protocol)	43
5.3.2	Portas bem conhecidas do UDP	44
<b>6</b>	<b><i>Protocolos de Roteamento</i></b>	<b>45</b>

<b>6.1</b>	<b>Roteamento Direto</b>	<b>45</b>
<b>6.2</b>	<b>Roteamento Indireto</b>	<b>45</b>
<b>6.3</b>	<b>Tabela de Roteamento</b>	<b>47</b>
<b>6.4</b>	<b>Rota Default</b>	<b>49</b>
<b>6.5</b>	<b>Alguns exemplos práticos</b>	<b>50</b>
6.5.1	Exemplo 1	50
6.5.2	Exemplo 2	55
<b>7</b>	<b>IGP – Interior Gateway Protocol</b>	<b>59</b>
<b>7.1</b>	<b>RIP - Routing Information Protocol (RFC 1058)</b>	<b>61</b>
7.1.1	Problemas do Protocolo RIP	61
7.1.2	Convergência Lenta do RIP	62
7.1.2.1	Método Split Horizon	64
7.1.2.2	Método Hold Down	64
7.1.2.3	Método Poison Reverse	64
<b>7.2</b>	<b>O Protocolo Hello</b>	<b>64</b>
<b>7.3</b>	<b>OSPF - Open Shortest Path First (RFC 1131)</b>	<b>65</b>
7.3.1	Formato das Mensagens	66
7.3.1.1	Database Description	67
7.3.1.2	Link Status Request	68
7.3.1.3	Link Status Update	69
7.3.1.4	Link Status Acknowledgement	70
7.3.2	Exemplo de Funcionamento OSPF	71
<b>7.4</b>	<b>IGRP (Interior Gateway Routing Protocol)</b>	<b>74</b>
<b>8</b>	<b>EGP - Exterior Gateway Protocol (RFC 904)</b>	<b>75</b>
<b>8.1</b>	<b>Cabeçalho Padrão do EGP</b>	<b>76</b>
<b>8.2</b>	<b>Mensagem de Aquisição de Vizinho</b>	<b>77</b>
<b>8.3</b>	<b>Teste Contínuo de Funcionamento de Vizinho</b>	<b>78</b>
<b>8.4</b>	<b>Mensagem POLL REQUEST</b>	<b>78</b>
<b>8.5</b>	<b>Mensagem de Atualização de Rotas</b>	<b>78</b>
<b>9</b>	<b>DNS - Domain Name System</b>	<b>82</b>
<b>9.1</b>	<b>Nomes Hierárquicos</b>	<b>83</b>
<b>9.2</b>	<b>Ferramenta Nslookup</b>	<b>85</b>
<b>9.3</b>	<b>Configuração do DNS</b>	<b>86</b>
<b>9.4</b>	<b>Resolução de Nomes</b>	<b>88</b>
9.4.1	Formato da Mensagem	88
<b>10</b>	<b>Aplicações</b>	<b>91</b>
<b>10.1</b>	<b>Telnet – Terminal Remoto</b>	<b>91</b>
<b>10.2</b>	<b>FTP - Transferência de Arquivos</b>	<b>93</b>
<b>10.3</b>	<b>E-mail - Mensagens Eletrônicas</b>	<b>97</b>
<b>10.4</b>	<b>News Group</b>	<b>99</b>

<b>10.5</b>	<b>WWW (Word Wide Web)</b>	<b>100</b>
<b>10.6</b>	<b>Ping</b>	<b>102</b>
<b>10.7</b>	<b>Finger</b>	<b>103</b>
<b>11</b>	<b><i>Evolução e Conclusão</i></b>	<b>104</b>
<b>12</b>	<b><i>Bibliografia</i></b>	<b>105</b>

# 1 Introdução ao TCP/IP

Com o crescimento cada vez mais acentuado das redes de computadores surge a necessidade de interconecata-las haja visto que uma grande rede é formada por pequenas unidades de LAN (Local Area Network ), MAN (Metropolitan Area Network) ou WAN (Wide Area Network).

Como existem diversos tipos de usuários e diversos tipos de aplicações, existem tecnologias de redes que se adequam melhor a cada perfil de usuário. O problema começa a surgir quando precisamos conectar diferentes tecnologias de redes de forma transparente.

Torna-se então necessário um protocolo (ou linguagem) comum que independente da tecnologia de rede utilizada permita uma comunicação (ou *internetworking*) de forma transparente. Neste contexto, o protocolo TCP/IP (Transport Control Protocol / Internet Protocol) vem suprir esta necessidade dando total transparência aos usuários finais das diversas tecnologias de rede empregadas pelas diversas LANs, MANs e WANs existentes, mascarando todos os detalhes da tecnologia de Hardware utilizada.

Na Internet é usado o conceito de *Open System* (Sistemas Abertos), onde as especificações são públicas, não têm dono, ou seja, qualquer pessoa pode produzir software para esta tecnologia sem que seja preciso autorização ou pagamento de *royalties* para terceiros.

A Internet já possui milhares de aplicações, e outras tantas estão diariamente sendo criadas. Dentre todas as aplicações algumas merecem destaque especial, como:

- Correio Eletrônico (SMTP)
- WWW (HTTP)
- Resolução de Nomes (DNS)
- Transferência de Arquivos (FTP)
- Terminal Remoto (TELNET)
- Gerenciamento (SNMP)

## 1.1 Histórico

A plataforma TCP/IP surgiu através dos trabalhos do DARPA (Defense Advanced Research Projects Agency) dos Estados Unidos, em meados da década de 70, constituindo a ARPANET, que mais tarde se desmembrou em ARPANET, para pesquisa, e MILNET, voltada para as instituições militares.

Vale ressaltar que desde o princípio a arquitetura TCP/IP foi concebida em um contexto de guerra (Guerra Fria), onde uma das grandes preocupações era interligar os diversos computadores (independente da tecnologia de rede utilizada), de forma simples e não centralizada, ou seja, se determinados computadores fossem eventualmente destruídos a rede continuasse funcionando independente daqueles computadores, o que inclui um conceito muito forte de descentralização, característica essa que não era comum na época.

Para encorajar os pesquisadores universitários a adotar o TCP/IP, o DARPA fez uma implementação de baixo custo, integrando-o ao sistema operacional UNIX da Universidade de Berkeley (BSD) já em uso em todas as universidades americanas. Além disso, teve-se o cuidado de definir aplicações de rede similares às já conhecidas em Unix, como *rusers* e *rcp*.

Mais tarde a NSF (National Science Foundation) estimulou o seu crescimento criando a NSFNET, que ligava centros de supercomputação espalhados por todo o Estados Unidos, numa rede de longa distância, também o utilizando o protocolo TCP/IP para interligar as diferentes tecnologias de redes.

Devido a sua grande facilidade de implementação, baixo custo e as vantagens que esta rede oferecia, ela cresceu rapidamente e se espalhou por diversos países, constituindo o que hoje conhecemos como Internet.

Quando alguém nos fala do protocolo TCP/IP logo nos vem a cabeça a palavra Internet, porque a Internet só é o que é graças a este protocolo, vale observar que você pode utilizar o TCP/IP independente de estar ligado a Internet. A palavra que usamos atualmente para definir uma rede que utiliza o TCP/IP mas não esta ligada à Internet é Intranet. Neste contexto, é possível ter todas as facilidades das aplicações Internet, ou seja, do protocolo TCP/IP, dentro de um ambiente fechado.

Como o TCP/IP é um **sistemas aberto**, não existe uma pessoa ou instituição responsável por ele. Existe sim, organismos como o IAB (Internet Activites Board) que coordena os esforços de pesquisa na área, através de vários grupos de trabalho, tal como o IETF (Internet Engineering Task Force). Todas estas especificações são descritas nas RFC (Request for Comments), que detalham o conjunto de padrões para comunicação entre os computadores, assim como as convenções de interconexão, roteamento, tráfego e etc.

## 1.2 Documentação

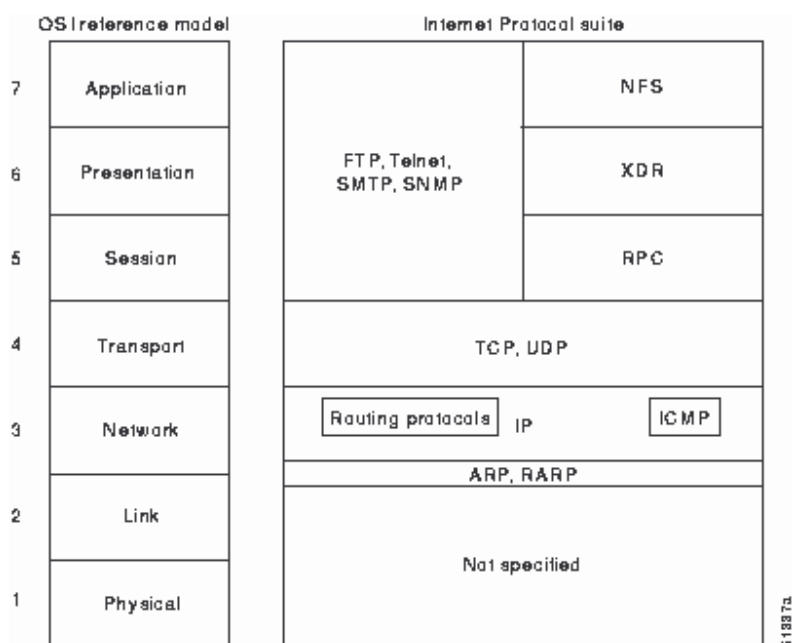
Qualquer pessoa pode sugerir um novo padrão ou alteração de um já existente, para isso ela escreve a especificação deste padrão e o envia na forma de um *draft* (Rascunho) de uma RFC, para o NIC (Network Information Center), para ser julgado da viabilidade de aceitação ou não do padrão, caso o padrão seja aceito, ele recebe um número sequência e é publicado na Internet para que as pessoas tenham conhecimento dele. Um exemplo de RFC seria portanto a RFC1092, RFC734 e assim por diante. Caso o *draft* não seja aceito, ele é simplesmente descartado.

Existem vários lugares na Internet que se publicam as RFCs, entre eles citamos:

- <http://www.ietf.org/>
- <http://andrew2.andrew.cmu.edu/rfc/rfc1160.html> , vai trazer a RFC1160, por exemplo

## 2 Endereço de Rede

O nível de rede da arquitetura Internet TCP/IP é exatamente o protocolo IP (Internet Protocol). Como visto anteriormente, este protocolo tem como funcionalidade básica **rotear pacotes** de uma máquina para outra, dentro de uma mesma rede ou entre redes diferentes (baseado na informação de endereço contida no pacote), utilizando a tecnologia de chaveamento de pacotes com **datagrama não-confiável**. Isto significa dizer que o nível IP não faz nenhum tipo de verificação de entrega dos pacotes, nem tão pouco estabelece conexão antes de transmitir qualquer dado (Connectionless Packet Delivery Service), ficando esta responsabilidade para as camadas superiores, no caso, o TCP (Transmission Control Protocol) para dados com garantia de entrega ou UDP (User Datagram Protocol) para dados sem garantia de entrega. A Figura 2.1 mostra uma comparação entre o modelo de referência OSI e a arquitetura Internet, nele é possível ver que o protocolo IP correspondente exatamente ao nível de rede do modelo OSI.



**Figura 2.1 - Modelo OSI versus Arquitetura TCP/IP**

Além de rotear os pacotes pela rede, o nível IP também define o *endereçamento universal da Internet*, ou seja, é neste nível que as máquinas são diferenciadas uma das outras, através do seu endereço IP.

## 2.1 Endereço IP

Para duas máquinas se comunicarem utilizando o protocolo TCP/IP, cada uma destas máquinas precisa ter um endereço IP diferente, pois é através do endereço IP que é possível identificar uma determinada máquina.

Agora imagine que ao invés de você ligar somente duas máquinas, você queira ligar milhões delas em diversas partes do mundo. É natural pensar que a quantidade de endereços também seja enorme. Pensando nisso, o endereço IP foi criado como um conjunto de 32 bits para ser utilizado por todas as aplicações que utilizem o protocolo TCP/IP. A notação desta representação é mostrada a seguir:

**X.X.X.X**

Onde o valor de X varia de 0 à 255, ou seja,  $2^8 = 256$  possibilidades, como mostrado abaixo:

**0.0.0.0 à 255.255.255.255**

Observe, portanto, que o número máximo de computadores e elementos de rede utilizando esta forma de endereçamento seria: 4.294.967.296 ( $256 \times 256 \times 256 \times 256$ ), o que é um número bastante representativo, mas que já está ficando saturado para os dias atuais<sup>1</sup>. É por isso que soluções como Proxy, DHCP ou o próprio IPv6 (nova versão do IP) estão sendo largamente utilizados para resolver este problema de escassez de números IP, como veremos nos próximos capítulos. Alguns exemplos de endereço IP seriam:

- 200.241.16.8
- 30.10.90.155
- 197.240.30.1

De forma a facilitar a compreensão ao homem, o endereço IP é escrito como quatro números **decimais** separados por ponto. Cada decimal dá o valor de um octeto do endereço IP (em binário). A figura 2.2 mostra um endereço IP com a sua representação binária e a sua representação decimal. A representação binária é separada em quatro blocos de oito bits, já na sua forma decimal, estes blocos são agrupados e separados por ponto.

<b>10000000 00001010 00000010 00011110</b>	<b>128.10.2.30</b>
<b>11001000 11110001 00010000 00001000</b>	<b>200.241.16.8</b>

**Figura 2.2 – Representação IP binária e decimal**

<sup>1</sup> Esta conta é só ilustrativa, pois a quantidade de máquinas é ainda menor. Pois não estamos considerando endereços de Loopback, IPs reservados entre outros.



Observe que mesmo o endereçamento IP sendo bastante eficiente, as pessoas tendem a Ter dificuldade para decorar números, ainda mais um número tão grande. Para resolver este problema foi criado o serviço de DNS (Domain Name System), que associa um nome, geralmente mais fácil de memorizar ao respectivo número IP da máquina, isto porque a comunicação entre qualquer dois usuários na Internet sempre é feita através de endereços IPs. O serviço de DNS será abordado mais adiante.

Quando a Internet surgiu os endereços IPs eram distribuídos de forma centralizada pelo NIC (Network Information Center), à medida que a Internet foi crescendo tornou-se impossível uma única entidade distribuir os endereços IP. O NIC então tomou a decisão de descentralizar este serviço, passando a responsabilidade para determinadas instituições de cada país. No Brasil a instituição responsável pela distribuição dos endereços IPs e pelo gerenciamento dos domínios é a FAPESP. Os endereços IPs são gratuitos, antigamente bastava-se fazer uma solicitação para receber um endereço classe C, com a crescente escassez destes endereços os critérios mudaram um pouco, passando também o provedor de acesso a fornecer alguns endereços classe C para um determinado cliente. Maiores informações sobre a política atual da FAPESP podem ser encontradas no *site* [www.fapesp.br](http://www.fapesp.br).

## 2.2 Classes de Endereços

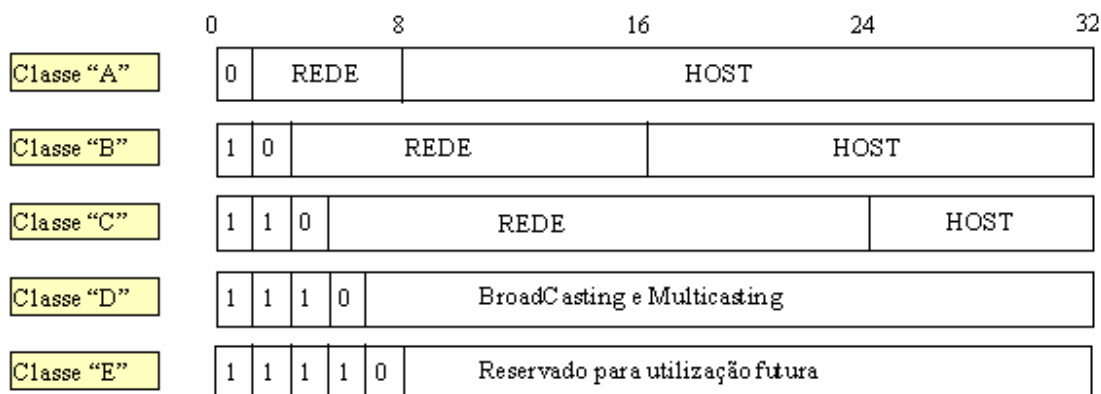
Os endereços IP foram divididos em classes para facilitar o roteamento de pacotes. Nesta divisão, um endereço de classe A por exemplo tem o seu primeiro octeto reservado para o endereço de rede e os demais são utilizados para as máquinas, já o endereço classe B, tem os dois primeiros octetos reservados para a rede e os demais para as máquinas, no endereço de classe C os três primeiros octetos são reservados para a rede e somente o último octeto para as máquinas. Isto significa dizer que os endereços de classe C são usados por pequenas redes, até o limite de 256 computador (utilizando somente um endereço de classe C), já os endereços de classe B, são para redes maiores suportando até 65.536 computadores na mesma rede e os de classe A suportam até 1.6777.216<sup>2</sup>. Existe ainda os endereços de Classe D que são utilizados para enviar mensagens *multicasting* (uma mensagem enviada através de um único endereço IP para vários destinatários) e o *broadcasting* (uma mensagem enviada através de um único endereço IP para todos os destinatários de uma determinada rede). A faixa de endereços IP de cada classe é mostrada na Tabela 2.1

De	À	Classe de Endereço
0	126	A
128	191	B
192	223	C
224	239	D
240	247	E

**Tabela 2.1 – Faixa de endereços IP de cada classe**

<sup>2</sup> Novamente esta conta não leva em consideração diversos aspectos, sendo o número de computadores menor do que o especificado.

A Figura 2.3 ilustra cada classe com a sua respectiva representação binária e a quantidade de Hosts (computadores) que podem estar conectados dentro daquela classe de endereço.



**Figura 2.3 - Divisão dos endereços IP em classes**

A Tabela 2.2 ilustra as classes de endereços com suas faixas decimal e sua representação binária.

Classe	Faixa de Endereços	Representação Binária	Utilização
A	1-126.X.X.X	0nnnnnnn.hhhhhhhh.hhhhhhhh.hhhhhhhh	
B	128-191.X.X.X	10nnnnnnn.nnnnnnnn.hhhhhhhh.hhhhhhhh	
C	192-223.X.X.X	110nnnnnn.nnnnnnnn.nnnnnnnn.hhhhhhhh	
D	224-239.X.X.X	1110xxxx.xxxxxxxxx.xxxxxxxxx.xxxxxxxxx	Multicast / Broadcast
E	240-247.X.X.X	11110xxx.xxxxxxxxx.xxxxxxxxx.xxxxxxxxx	Reservado

**Tabela 2.2 – Faixa de Endereços decimais e binários**

Onde,

**X** é um número que varia de 0 à 255

**n** é o número de bits da rede

**h** é o número de bits do host

**x** é o número de bits da rede e do host

### 2.2.1 Endereço Loopback

Observe que o endereço **127.X.X.X** não é mostrado na tabela acima, isto porque este endereço foi reservado para *Loopback*.

Por convenção, toda máquina rodando TCP/IP possui uma interface de *Loopback*, além das interfaces de rede que possui. Esta interface não conecta a rede alguma. Seu objetivo é permitir teste de comunicação inter-processos dentro da mesma máquina.

Quando um programa usa o endereço de *loopback* para enviar dados, o software do protocolo retorna o dado sem gerar tráfego na rede (isto é muito utilizado para se testar programas, inclusive quando não se tem placa de rede). A comunicação vai pelo caminho normal, saindo do nível de aplicação, passando pelo nível de transporte (TCP ou UDP) e chegando ao nível IP, que retorna a comunicação de volta ao nível de aplicação de um outro processo.

A especificação determina que um pacote enviado para a rede 127 nunca deve aparecer em nenhuma rede. O endereço de *loopback* utilizado pela quase totalidade das implementações é 127.0.0.1.

## 2.2.2 Endereços IP reservados

Assim como a classe de endereços 127.0.0.0 é reservada para *loopback*, existem outros endereços reservados que não podem ser utilizados em nenhuma máquina conectada a Internet.

Esses endereços são reservados para redes que não se ligarão nunca à Internet ou que se ligarão através de um *proxy*<sup>3</sup> (assim como as Intranets).

Nenhum destes endereços pode ser anunciado, o que quer dizer que se uma máquina for conectada a Internet com algum endereço reservado ela não conseguirá passar pelos gateways core (Core serão vistos mais a frente).

Os endereços reservados da Internet são especificados pela RFC1597 e são os seguintes:

Rede	Máscara
10.0.0.0	255.0.0.0
172.16.0.0	255.240.0.0
192.168.0.0	255.255.0.0

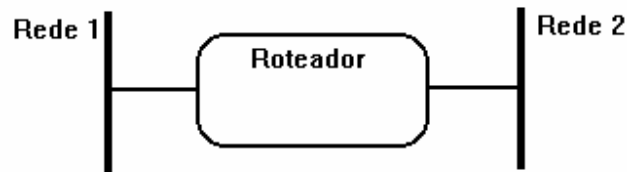
## 2.3 Roteadores

A função básica do protocolo IP é o transporte dos blocos de dados dentro de uma mesma rede ou entre sub-redes. Quando uma máquina **A** deseja “falar” com uma máquina **B** que esteja na mesma rede, esta simplesmente joga a mensagem na rede, para ser mapeada no protocolo de enlace e depois no físico para chegar à seu destinatário.

Entretanto, às vezes a máquina **A** deseja “falar” com a máquina **C**, que esta em uma outra sub-rede. Neste caso o tráfego é transferido de uma rede para outra através de equipamentos chamados Roteadores ou Gateways, estes equipamentos tem por finalidade interconectar duas redes, como mostra a Figura 2.4.

---

<sup>3</sup> Proxy e Core serão vistos mais adiante.



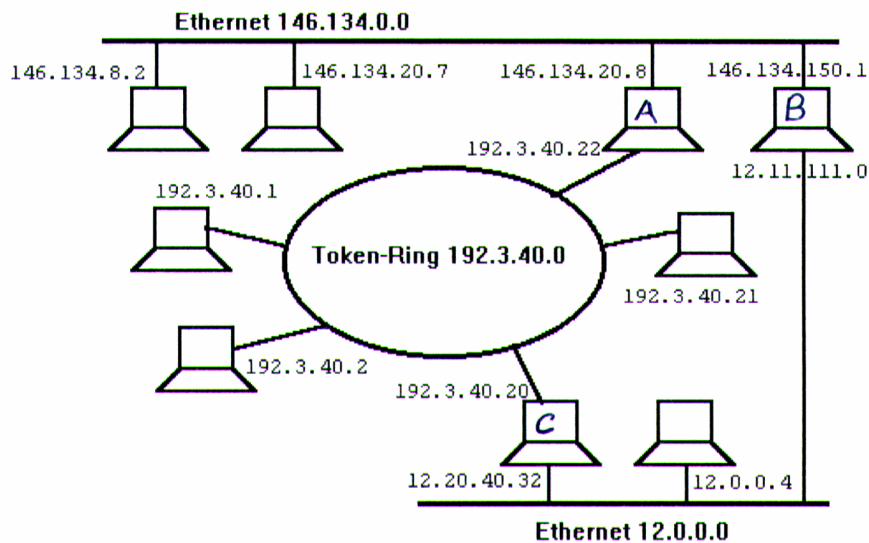
**Figura 2.4 – Roteador entre duas Redes**

Como vimos, o endereço IP contém informações sobre a rede e a máquina. Como resolver então o problema do Roteador, já que este está ligado a duas redes distintas?

Este problema é resolvido atribuindo dois endereços IP ao roteador, um endereço referente à *Interface* de rede da **Rede 1** e outro endereço IP referente à *Interface* de rede da **Rede 2**. O roteador terá portanto no mínimo dois endereços IPs. Haverão casos que o roteador estará ligado a mais redes, logo um número maior de endereços IPs será requerido.

Um primeiro problema que surge é que se uma máquina **A**, da Rede 1, for transferida para a Rede 2, o endereço IP desta máquina terá que ser alterado. A Figura 2.5 é um exemplo de conexão de redes diferentes (2 redes Ethernet e 1 Token Ring ) utilizando o protocolo TCP/IP. Vale ressaltar que temos três classes de endereços IP, como mostrado abaixo:

- Token Ring 192.3.40.0 (classe C)
- Ethernet 146.134.0.0 (classe B)
- Ethernet 12.0.0.0 (classe A)



**Figura 2.5 – Interconexão de três redes**

Na rede Token Ring (192.3.40.0), temos as máquinas:

- **192.3.40.1**
- **192.3.40.2**
- **192.3.40.20** (Máquina C)
- **192.3.40.21**
- **192.3.40.22** (Máquina A)

Na rede Ethernet, 146.134.0.0, temos as máquinas:

- **146.134.8.2**
- **146.134.20.7**
- **146.134.20.8** (Máquina A)
- **146.134.150.1** (Máquina B)

Na rede Ethernet, 12.0.0.0, temos as máquinas:

- **12.11.111.0** (Máquina B)
- **12.0.0.4**
- **12.20.40.32** (Máquina C)

Observe que a **Máquina A**, esta fazendo o papel de roteador (ou gateway) entre a rede Token-Ring (192.3.40.0) e à rede Ethernet (146.134.0.0), da mesma forma que a **Máquina B**, entre a rede Ethernet (146.134.0.0) e à rede Ethernet (12.0.0.0) e a **Máquina C** entre a rede Ethernet (12.0.0.0) e a rede Token-Ring (192.3.40.0). Isto é facilmente implementado colocando duas placas de rede em cada uma dessas máquinas e configurando para cada placa de rede o seu respectivo endereço IP.

## 2.4 Sub-rede

A classificação dos endereços IP por classes tem como objetivo facilitar o roteamento. Pela classe do endereço sabe-se quantos bits representam a rede e quantos a máquina. O endereço classe C é geralmente usado para redes pequenas. No entanto, a alocação de um endereço classe C para uma rede implica na alocação de 256 endereços, na verdade 254 (são desconsiderados o último número igual a 0 e a 255), mesmo que eles não sejam todos usados. O crescimento elevado da Internet fez escassear o número de endereços disponível, principalmente porque muitas empresas tem endereço IP classe C para 254 máquinas e usam bem menos que esta quantidade. Para minimizar este problema, foi introduzido o conceito de sub-rede, que procura utilizar outros bits para identificar a rede ao invés da estação. O exemplo da Figura 2.6 mostra uma rede 128.10.0.0 que é sub-dividida internamente em duas redes distintas, a rede 128.10.1.0 e a rede 128.10.2.0.

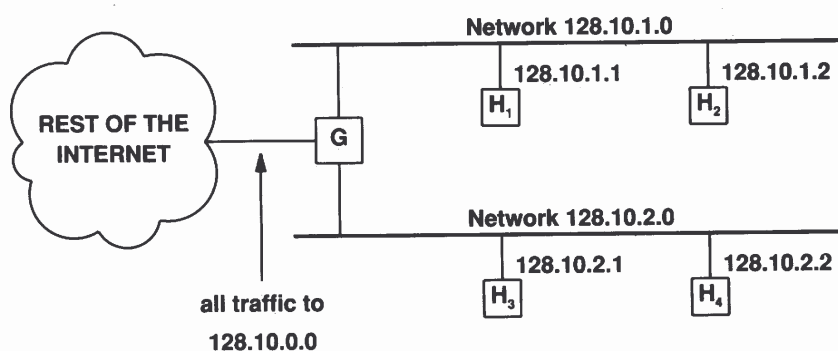


Figura 2.6 – Sub-Divisão de uma rede

Para toda a Internet, existe o endereço de rede 128.10.0.0, que é o endereço que chega até o **Gateway (G)**, quando um pacote chega a este *Gateway* é que ele vai ser roteado para uma ou outra rede dependendo do IP de destino (no caso irá analisar o terceiro octeto de destino). Em vários casos a separação em sub-rede traz outras vantagens como melhoria na *performance* da própria rede interna, como veremos a seguir.

Vale ressaltar que se eu não fizesse a divisão em sub-redes, o endereço 128.10.1.0 e o endereço 128.10.2.0 serão endereços de estações e não endereços de rede como esta agora.

### 2.4.1 Máscara de uma Sub-Rede

Para definirmos se um determinado endereço IP é um endereço de rede ou um endereço de máquina utilizamos o conceito de máscara. A máscara de uma rede vai nos permitir dizer quais endereços são da rede e quais são de máquinas e dentro de qual rede. O formato de escrita da máscara é o mesmo do número IP.

A lógica é bastante simples: Vamos definir que se o bit da máscara for igual a 1 significa que é um endereço de rede e se for igual a 0 (zero) é endereço de máquina. Neste sentido a máscara **255.255.255.0** por exemplo, que na sua forma binária equivaleria a

**11111111 11111111 11111111 00000000**

estaria indicando que os três primeiros octetos estão sendo utilizados para rede e o último para máquina. Isto significa dizer que dentro desta rede podemos ter 254 máquinas ( $2^8 - 2$  dos endereços 0 e 255). Isto porque gastamos os três primeiros octetos para dizer qual a rede, então só nos sobrou o último para especificar as máquinas.

A Tabela 2.3 mostra qual é a máscara utilizada para as classes A, B e C, sem sub-redes.

O padrão especifica que um *site* usando subrede deve escolher uma máscara de 32 bits para cada rede. Bits na máscara são 1 se o bit correspondente no endereço IP for referente a rede, e 0 se o bit correspondente no endereço IP for referente a máquina.

Por exemplo, para redes classe A, B e C sem subrede teríamos as seguintes máscaras (Tabela 2.3):

Classe	Máscara Binária	Máscara Decimal
A	11111111 00000000 00000000 00000000	255.0.0.0
B	11111111 11111111 00000000 00000000	255.255.0.0
C	11111111 11111111 11111111 00000000	255.255.255.0

**Tabela 2.3 – Máscara Binária e Decimal por classe, sem sub-rede**

Pelo padrão, os bits usados para sub-rede não precisam ser contíguos. A seguinte máscara é válida:

**11111111 11111111 00011000 01000000**

Esta máscara, no entanto, traria endereços de máquinas na sub-rede bastante confusos, de forma que é boa política usar bits de rede contíguos.

### Exemplo:

Vamos fazer agora um exemplo real. Suponha que tenhamos um endereço IP classe C, digamos 200.241.16.X. Quantas sub-redes eu posso ter? E quais os endereços de máquina?

No caso mais simples, temos apenas 1 rede, não há sub-redes, logo a máscara seria:

Binário	11111111	11111111	11111111	00000000
Decimal	255	255	255	0

E os endereços IPs seriam de: 200.241.16.0 à 200.241.16.255

Só que devemos lembrar **SEMPRE** de retirar os endereços IP extremos. Portanto, nossos endereços válidos são:

**200.241.16.1 à 200.241.16.254**

Agora suponha que tenhamos **2 sub-redes**. Nossa máscara então seria:

Binário	11111111	11111111	11111111	<b>1</b> 0000000
Decimal	255	255	255	128

Neste caso estamos utilizando mais um bit para identificar a rede (o primeiro bit igual a 1 do quarto octeto). Logo temos duas redes possíveis. A rede **0** e a rede **1**.

Na sub-rede **0** teremos os endereços IPs de: 200.241.16.**0** à 200.241.16.**127**

Como devemos eliminar sempre os extremos, os nossos endereços válidos nessa sub-rede serão:

**200.241.16.1 à 200.241.16.126**

Já na sub-rede **1** teremos os endereços IPs de: 200.241.16.**128** à 200.241.16.**255**

Aplicando também a eliminação dos extremos, ficamos com os endereços:

**200.241.16.129 à 200.241.16.254**

**Observação Importante:** De acordo com o padrão a primeira sub-rede (todos os bits de rede igual a zero) e a última sub-rede (todos os bits de rede igual a um) tem que ser descartadas. *Logo, pelo padrão, a rede acima não teria nenhuma máquina já que descartaríamos a primeira e a última sub-rede.* Alguns fabricantes, entretanto, como a Cisco por exemplo aceitam a máscara acima, mas vale ressaltar que isto não faz parte do padrão e portanto não é recomendado.

Agora suponha que tenhamos **4 sub-redes**. Nossa máscara então seria:

Binário	11111111	11111111	11111111	<b>11</b> 000000
Decimal	255	255	255	192

Neste caso estamos utilizando mais dois bits para identificar a rede (os dois primeiros bits igual a 1 do quarto octeto). Logo temos quatro redes possíveis. A rede **00**, **01**, **10** e **11**.

Na sub-rede **00** teremos os endereços IPs de: 200.241.16.**0** à 200.241.16.**63**

Como devemos eliminar sempre os extremos, os nossos endereços válidos nessa sub-rede serão:

**200.241.16.1 à 200.241.16.62**

A explicação acima é só ilustrativa, porque como já vimos esta sub-rede deve ser desconsiderada.



Na sub-rede **01** teremos os endereços IPs de: 200.241.16.**64** à 200.241.16.**127**

Aplicando também a eliminação dos extremos, ficamos com os endereços:

**200.241.16.65 à 200.241.16.126**

Na sub-rede **10** teremos os endereços IPs de: 200.241.16.**128** à 200.241.16.**191**

Aplicando também a eliminação dos extremos, ficamos com os endereços:

**200.241.16.129 à 200.241.16.190**

Na sub-rede **11** teremos os endereços IPs de: 200.241.16.**192** à 200.241.16.**255**

Aplicando também a eliminação dos extremos, ficamos com os endereços:

**200.241.16.193 à 200.241.16.254**

Pelo mesmo motivo mencionado anteriormente esta sub-rede também será descartada.

Para saber qual é a faixa de endereços IP de cada sub-rede basta dividir 256 pelo número de sub-redes. No exemplo anterior teríamos 256 dividido por 4 igual a 64. Isto significa dizer que cada rede teria 64 máquinas (sem considerar a eliminação dos extremos). Ficando então os endereços distribuídos da seguinte forma:

- X.X.X.**0** à X.X.X.**63**
- X.X.X.**64** à X.X.X.**127**
- X.X.X.**128** à X.X.X.**191**
- X.X.X.**192** à X.X.X.**255**

Lembrando sempre que o primeiro e o último conjunto de endereços deverá ser descartado. No caso os endereços de X.X.X.0 à X.X.X.63 e X.X.X.192 à X.X.X.255

O processo para achar todas as outras sub-redes é o mesmo. A tabela 2.4 faz um resumo deste processo para um endereço IP classe C até 8 sub-redes. As sub-redes grifadas não poderão ser usadas de acordo com o padrão.

Sub-Redes	Máscara	No. Da Sub-Rede	IPs Válidos	Tot. Maq.
1	255.255.255.0	-	x.x.x.1 à x.x.x.254	254
2	255.255.255.128	0	x.x.x.1 à x.x.x.126	126
	255.255.255.128	1	x.x.x.129 à x.x.x.254	126
4	255.255.255.192	00	x.x.x.1 à x.x.x.62	62
	255.255.255.192	01	x.x.x.65 à x.x.x.126	62
	255.255.255.192	10	x.x.x.129 à x.x.x.190	62
	255.255.255.192	11	x.x.x.193 à x.x.x.254	62
8	255.255.255.224	000	x.x.x.1 à x.x.x.30	30

Sub-Redes	Máscara	No. Da Sub-Rede	IPs Válidos	Tot. Maq.
	255.255.255.224	001	x.x.x.33 à x.x.x.62	30
	255.255.255.224	010	x.x.x.65 à x.x.x.94	30
	255.255.255.224	011	x.x.x.97 à x.x.x.126	30
	255.255.255.224	100	x.x.x.129 à x.x.x.158	30
	255.255.255.224	101	x.x.x.161 à x.x.x.190	30
	255.255.255.224	110	x.x.x.193 à x.x.x.222	30
	255.255.255.224	111	x.x.x.225 à x.x.x.254	30

Tabela 2.4 – N° de sub-redes versus classes

A Tabela 2.5 mostra a quantidade de sub-redes, a quantidade de máquinas de cada sub-rede e a quantidade total de máquinas.

N° de Bits	N° Sub-Redes	Quantidade de Máquinas por Sub-Rede	Total Máquinas
0	0	254	254
1	2-2=0	0	0
2	4-2=2	62	124
3	8-2=6	30	180
4	16-2=14	14	196
5	32-2=30	6	180
6	64-2=62	2	124

Tabela 2.5 – N° de sub-redes e máquinas de um endereço classe C

## 2.5 Formato do datagrama IP

O datagrama IP é a unidade básica de dados no nível de rede (Internet Protocol – IP). Este protocolo consiste em um serviço de entrega de pacotes, não confiável, sem reconhecimento e sem conexão de um datagrama IP. Isto significa dizer que um pacote IP pode ser perdido, duplicado atrasado ou entregue fora de ordem, mas o nível IP não detectará tais condições ficando esta responsabilidade para os níveis superiores. É função do nível de rede (IP) também efetuar o roteamento do pacote escolhendo o caminho que ele deverá seguir.

Um datagrama está dividido em duas áreas, uma área de cabeçalho e outra de dados. O cabeçalho contém toda a informação necessária para identificar o conteúdo do datagrama, bem como o seu endereço origem e destino, entre outros. Na área de dados está encapsulado o pacote do nível superior, ou seja um pacote TCP ou UDP, que contém os dados propriamente ditos. A Figura 2.7 mostra o formato do datagrama.

0	4	8	16	19	24	31
VERS	HLEN	SERVICE TYPE	TOTAL LENGTH			
IDENTIFICATION			FLAGS	FRAGMENT OFFSET		
TIME TO LIVE		PROTOCOL	HEADER CHECKSUM			
SOURCE IP ADDRESS						
DESTINATION IP ADDRESS						
IP OPTIONS (IF ANY)					PADDING	
DATA						

Figura 2.7 – Datagrama IP

A Tabela 2.6 fornece a descrição de cada campo.

Campo IP	Descrição																
VERS	Versão do protocolo IP que foi usada para criar o datagrama																
HLEN	Comprimento do cabeçalho, medido em palavras de 32 bits																
SERVICE-TYPE	<p>Este campo especifica como o datagrama poderia ser manipulado pelo sistema de comunicação. Algumas opções são:</p> <p><i>Precedence</i>: indica a importância do pacote, com valores desde 0 (precedência normal) até 7 (controle da rede). Este bits permite-se ao transmissor indicar a importância de cada datagrama que ele está enviando, sem contudo <b>garantir</b> que estas exigências serão cumpridas pela rede. Existem ainda as opções abaixo:</p> <table><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr><tr><td colspan="3">PRECEDÊNCIA</td><td>D</td><td>T</td><td>R</td><td></td><td></td></tr></table> <ul style="list-style-type: none"><li>• Precedência Indica a importância do pacote</li><li>• D: Baixo Delay</li><li>• T: Alto throughput</li><li>• R: Alta confiabilidade</li></ul>	0	1	2	3	4	5	6	7	PRECEDÊNCIA			D	T	R		
0	1	2	3	4	5	6	7										
PRECEDÊNCIA			D	T	R												
TOTAL-LENGTH	Este campo proporciona o comprimento do datagrama medido em bytes, incluindo cabeçalho e dados. Se o pacote for fragmentado este campo indicará o tamanho do fragmento e não do pacote original																
IDENTIFICATION	Valor inteiro que identifica o datagrama. Este campo é usado para ajudar na remontagem de pacotes fragmentados. Este campo é muito importante porque quando um gateway fragmenta um datagrama, ele copia a maioria dos campos do cabeçalho do datagrama em cada fragmento, então a identificação também deve ser copiada, com o																

Campo IP	Descrição																										
	propósito de que o destino saiba quais fragmentos pertencem a quais datagramas. Cada fragmento tem o mesmo formato que um datagrama completo.																										
FRAGMENT OFF SET	Especifica o início do datagrama original dos dados que estão sendo transportados no fragmento. É medido em unidades de 8 bytes																										
FLAG	Controla a fragmentação. Os dois bits de mais baixa ordem controlam a fragmentação. Um bit especifica se o pacote pode ou não ser fragmentado e o outro bit especifica se o pacote é o último fragmento																										
TTL (Time to Live)	Especifica o tempo em segundos que o datagrama está permitido a permanecer no sistema Internet. Gateways e hosts que processam o datagrama devem decrementar o campo TTL cada vez que um datagrama passa por eles e devem removê-lo quando seu tempo expirar.																										
PROTOCOL	Especifica qual protocolo de alto nível foi usado para criar a mensagem que está sendo transportada na área de dados do datagrama. Exemplo: FTP, HTTP, SMTP, etc																										
HEADER-CHECKSUM	Assegura a integridade dos valores do cabeçalho. Observe que não existe CheckSum para os dados, somente para o cabeçalho, isto porque não pode haver erro na hora da entrega de um pacote ao endereço de destino, mesmo que os dados estejam com problemas. Isto significa dizer que o endereço da carta estará sempre correto, não significando que esta chegue ao seu destinatário ou caso chegue, que os dados estejam corretos.																										
SOURCE IP	Específica o endereço IP de origem																										
DESTINATION IP	Específica o endereço IP de destino																										
OPTIONS	<p>Consiste de um único octeto com opções diversas sobre um determinado pacote. Algumas delas são:</p> <table><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr><tr><td>COPY</td><td>Option Class</td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table> <p>Copy – controla como os gateways tratam as opções quando há fragmentação: Se Copy = 1, então Opções devem ser copiadas em todos os fragmentos</p> <table><tr><th>Option Class</th><th>Significado</th></tr><tr><td>0</td><td>Datagrama de controle</td></tr><tr><td>1</td><td>Reservado para uso futuro</td></tr><tr><td>2</td><td>Debug</td></tr><tr><td>3</td><td>Reservado para uso futuro</td></tr></table> <p>Para cada uma das opções existe ainda um <b>Option Number</b> para aumentar mais ainda a diversidade de opções.</p> <p>A próxima sessão traz alguns exemplo interessantes.</p>	0	1	2	3	4	5	6	7	COPY	Option Class							Option Class	Significado	0	Datagrama de controle	1	Reservado para uso futuro	2	Debug	3	Reservado para uso futuro
0	1	2	3	4	5	6	7																				
COPY	Option Class																										
Option Class	Significado																										
0	Datagrama de controle																										
1	Reservado para uso futuro																										
2	Debug																										
3	Reservado para uso futuro																										
PADDING	Este campo pode ou não ser usado dependendo do campo <i>OPTIONS</i> . É um campo complementar, usado para armazenar um determinado																										

Campo IP	Descrição
	valor dependendo da opção anterior.
DATA	É o pacote da camada de transporte (TCP ou UDP)

Tabela 2.6 – Opções do Datagrama IP

### 2.5.1 Algumas *OPTIONS* interessantes

Estas opções referem-se ao campo OPTION do datagrama IP.

- **Option Number 7, Option Class 0 - Record Route Option**

Esta opção permite a origem criar uma lista vazia de endereços IPs e a cada *gateway* que manusear o datagrama colocará o seu IP nesta lista. Esta opção então traria toda a rota utilizada que um determinado pacote fez entre o seu endereço de origem e o seu endereço de destino. A Figura 2.8 ilustra como os endereços IP seriam armazenados.

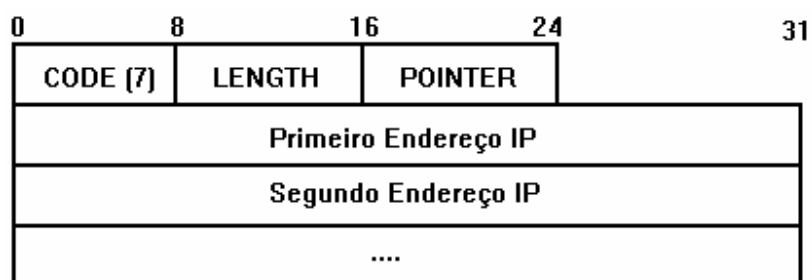
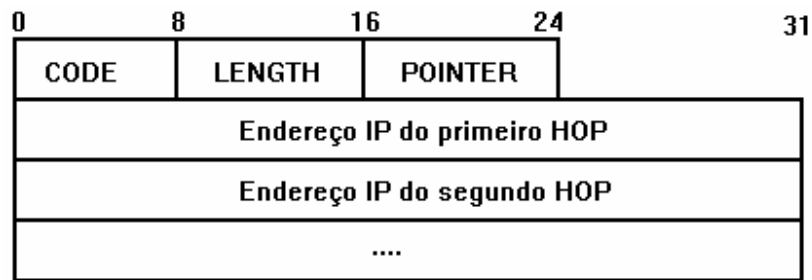


Figura 2.8 – Endereços IP de uma rota armazenados

Os endereços IP dos Gateways intermediários são colocados na posição indicada pelo campo POINTER.

- **Option Number 9 ou 3, Option Class 0 - Source Route Option**

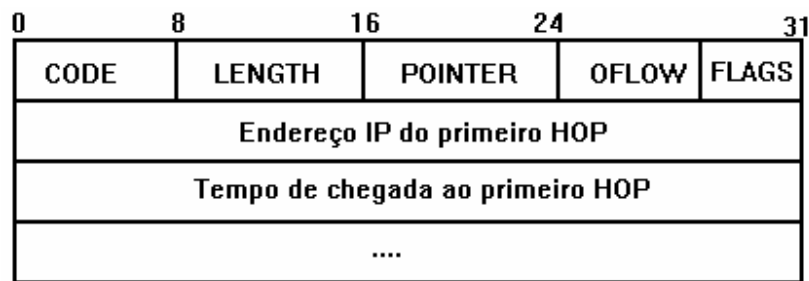
Permite a origem indicar a rota a ser seguida pelo pacote IP. Pode ser *strict* (option 9) ou *loose* (option 3). No modo *strict*, o caminho entre dois endereços sucessivos deve consistir de uma única conexão física de rede. Na opção *loose*, o datagrama deve seguir a sequência de endereços, mas dois endereços sucessivos na lista podem ter vários gateways entre eles. Esta opção é utilizada quando se deseja que um datagrama passe por um caminho específico independente do protocolo de roteamento. A Figura 2.9 ilustra este fato.



**Figura 2.9 – Definindo uma rota no datagrama**

- **Option Number 4, Option Class 2 - Timestamp Option**

Esta opção é similar ao **Record Route**, porém cada gateway coloca na lista, além do seu IP, o tempo em que o pacote chegou à este gateway. A Figura 2.10 ilustra este fato.



**Figura 2.10 – Armazenando uma rota e o seu tempo**

O campo OFLOW conta o número de gateways que não suportam esta opção.

### 3 Mapeamento de endereço de rede em endereço do nível físico

Duas máquinas numa mesma rede física só podem se comunicar se elas souberem o endereço físico uma da outra. Observe que até agora estávamos falando de endereço de rede, agora já abaixamos mais um nível no modelo de referência OSI e no TCP/IP. Este capítulo tratará basicamente de transformar um endereço de rede em um endereço físico, para isto veremos os protocolos ARP e RARP.

#### 3.1 Protocolo ARP

Como dito anteriormente, para duas máquinas numa mesma rede se comunicarem, elas precisam saber o endereço físico uma da outra. A Figura 3.1 ilustra uma rede com topologia em barra.



Figura 3.1 – Computadores em uma mesma rede

Considere as máquinas A e B da Figura 3.1, com seus endereços IP  $I_A$  e  $I_B$  e endereços físicos  $F_A$  e  $F_B$ . Suponha agora, que a máquina A deseja enviar um pacote para a máquina B, mas A sabe apenas o endereço IP de B ( $I_B$ ). Será necessário portanto, um mapeamento entre o endereço IP de B ( $I_B$ ) no seu endereço físico, como ilustra a Figura 3.2. Mas como a máquina A fará isso?

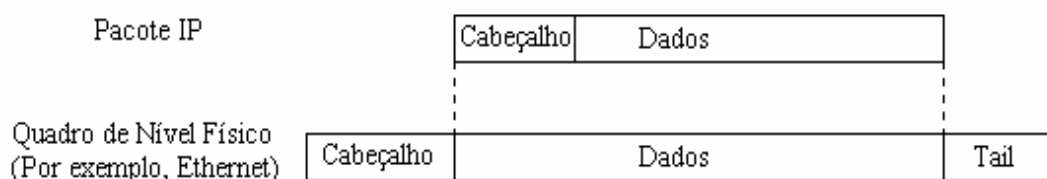


Figura 3.2 – Encapsulamento nível de rede no nível físico

Primeiramente, A envia um *broadcast* (uma mensagem que parte de uma estação e atinge todas as outras estações da mesma rede), pedindo para a máquina cujo endereço IP seja  $I_B$  responder com seu endereço físico  $F_B$ .

Todas as máquinas recebem este *broadcast*, mas só **B** responde à máquina que solicitou, no caso a máquina **A**, só a partir deste momento que **A** pode enviar a mensagem para **B**. Este protocolo é chamado de **ARP** (Address Resolution Protocol).

De forma a otimizar a resolução de endereços IP em físico, o protocolo ARP usa uma cache da seguinte forma:

- Quando A recebe a resposta de B, guarda numa tabela os endereços IP e físico de B (**I<sub>B</sub>** e **F<sub>B</sub>**)
- Quando B recebe o *broadcast* de A pedindo seu endereço físico, B guarda em sua cache **I<sub>A</sub>** e **F<sub>A</sub>**, supondo que se A deseja falar com ela (a máquina B), certamente ela (a máquina B) terá de falar com A.
- Esta informação residirá no cache por um período apropriado.

A seguir é apresentado um exemplo de como saber quais são os endereços IPs de uma rede e os seus respectivos endereços físicos. O comando **arp -a**, pode ser usado tanto para máquinas UNIX quanto na linha de comando do DOS.

```
cairo.inf.ufes.br> arp -a
milao.inf.ufes.br (200.241.16.86) at b4-66-56-ed-98-81
router.inf.ufes.br (200.241.16.1) at 08-00-02-09-71-90
xareu.16.241.200.in-addr.arpa (200.241.16.212) at 00-00-e8-31-8e-eb stale
helsinki.inf.ufes.br (200.241.16.253) at 00-a0-4b-03-5b-52
barcelona.inf.ufes.br (200.241.16.31) at 52-54-00-db-05-fe
seoul.inf.ufes.br (200.241.16.20) at (incomplete) stale not responding
novell07.inf.ufes.br (200.241.16.57) at b4-66-48-90-98-81
berlim.inf.ufes.br (200.241.16.9) at 08-00-2b-e7-74-e0
baleia.16.241.200.in-addr.arpa (200.241.16.220) at 00-60-08-df-4a-62 stale
paris.inf.ufes.br (200.241.16.2) at 08-00-2b-3d-ef-4e stale
novell15.inf.ufes.br (200.241.16.65) at b4-66-59-85-98-81 stale
albany.inf.ufes.br (200.241.16.28) at 00-00-e8-08-71-1d stale
londres.inf.ufes.br (200.241.16.6) at 00-00-3b-80-33-94 stale
novell19.inf.ufes.br (200.241.16.69) at b4-66-52-9e-98-81 stale
leeds.inf.ufes.br (200.241.16.10) at 10-00-5a-fa-37-10
novell09.inf.ufes.br (200.241.16.59) at 00-00-b4-33-9f-4b stale
vенеza.inf.ufes.br (200.241.16.11) at 10-00-5a-fa-50-83
marlin.inf.ufes.br (200.241.16.200) at 10-00-5a-fa-4d-b2
camburi.inf.ufes.br (200.241.16.130) at 08-00-20-0e-9a-d3
otawa.inf.ufes.br (200.241.16.23) at 00-00-e8-08-6b-f6
```

Neste exemplo, é possível ver que a máquina de endereço hierárquico **otawa.inf.ufes.br**, possui endereço IP **200.241.16.23** e endereço físico **00-00-e8-08-6b-f6**. Seguindo a mesma leitura, sabe-se o endereço físico de todas as outras máquinas da rede.



É importante ressaltar que diferentes redes (Ethernet, Token Ring, ATM, FDDI), tem diferentes implementações do ARP. Sendo isto totalmente transparente para arquitetura TCP/IP, já que tendo o endereço físico a preocupação de entrega deste pacote nesta rede fica por conta dos protocolos de acesso ao meio e de entrega desta rede.

Quando iniciadas, as estações não possuem a tabela endereço IP / endereço físico. Esta tabela vai sendo criada e manipulada dinamicamente dependendo da sua necessidade.

### 3.1.1 Formato do Pacote ARP

A Figura 3.3 mostra o formato do pacote ARP.

Hardware Type		Protocol Type
HLEN	PLEN	Operation
Sender HA [octetos 0-3]		
Sender HA [octetos 4-5]		Sender IP [octetos 0-1]
Sender IP [octetos 2-3]		Target HA [octetos 0-1]
Target HA [octetos 2-5]		
Target IP [octetos 0-3]		

**Figura 3.3 – Formato do pacote ARP**

A Tabela 3.1 descreve os campos do pacote ARP.

Nome do Campo	Descrição
Hardware Type	Especifica a interface de hardware pela qual o usuário aguarda uma resposta. No caso da rede Ethernet o valor é 1.
<b><u>Protocol Type</u></b>	Especifica o tipo de endereço que o usuário está procurando (0800H se for IP).
HLEN	Tamanho do endereço de Hardware
PLEN	Tamanho do endereço do protocolo de alto nível. As opções HLEN e PLEN em conjunto permitem que o ARP possa ser usado para uma rede qualquer.
Operation	1 - ARP request (Requisição do endereço físico) 2 - ARP response (Resposta do endereço físico) 3 - RARP request (Requisição do endereço IP) 4 - RARP response (Resposta do endereço IP)
Sender HÁ	Endereço de Hardware (Hardware Address) do remetente
Sender IP	Endereço IP do remetente
Target HÁ	Endereço de Hardware (Hardware Address) do destinatário

Nome do Campo	Descrição
Target IP	Endereço IP do destinatário

Tabela 3.1 – Descrição dos campos do pacote ARP

## 3.2 Protocolo RARP

O ARP converte um endereço IP em um endereço físico, já o RARP (*Reverse Address Resolution Protocol*) faz exatamente o processo inverso, que é o de converter um endereço físico em um endereço IP.

Talvez você esteja se perguntando mas para que ser isso? Pois bem, quando se usa máquinas *diskless* (sem disco) que dão *boot* via rede e deseja-se usar um protocolo para este *boot* baseado em TCP/IP, será necessário que a máquina disponha de seu endereço IP ainda no processo de boot. O problema é onde guardar este endereço se a máquina não possui disco local?

O protocolo RARP é usado exatamente para este propósito. Ele mapeia o endereço físico no endereço IP. Uma máquina (com disco) deverá ser designada servidor de RARP. Quando uma máquina *diskless* entrar em processo de boot, o software de boot acionará o RARP. Este enviará um *broadcast* na rede perguntando se alguém sabe o endereço IP dela (o endereço físico é automaticamente enviado no quadro). O servidor responde com o endereço IP daquela máquina, a qual pode falar TCP/IP normalmente daí pra frente, até que ela seja desligada e o processo se repita novamente.

Repare que inicialmente o administrador da rede precisa identificar os endereços físicos das máquinas *diskless* e criar uma tabela de mapeamento no servidor.

O protocolo RARP é idêntico ao ARP, mudando apenas o tipo de mapeamento feito.

## 3.3 Fragmentação de endereço IP em endereço físico

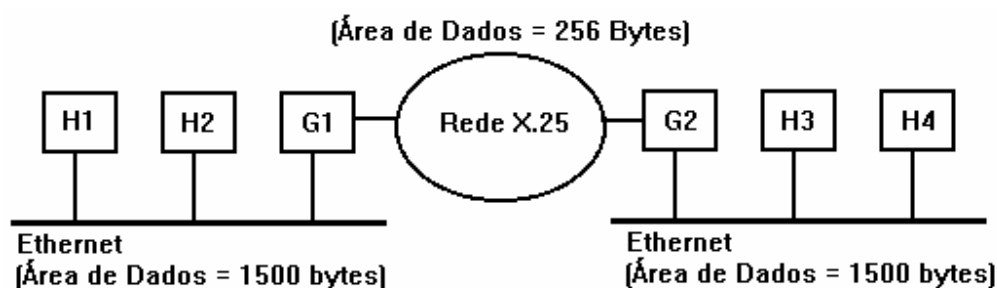
Para o nível IP (de rede), um pacote IP é uma unidade única e indivisível. Idealmente, o pacote IP deveria caber inteiro no quadro do nível de enlace. No entanto, nem sempre isso acontece. Pode acontecer de que num certo caminho existam níveis de enlace que tenham tamanho de área de dados bastante reduzidos. Pode acontecer também do usuário configurar o tamanho do pacote IP maior que a área de dados da própria rede onde a máquina está conectada.

Quando um pacote IP não couber inteiro na área de dados do quadro físico, ele deve ser **fragmentado**. Isto significa dizer, que o pacote deve ser quebrado em diversos pedaços que caibam no nível físico. Ao chegar no seu destino, o pacote deve ser reconstituído, já que o nível IP só poderá processar o pacote se este estiver inteiro. A fragmentação/remontagem é feita pelo próprio nível IP.

Na configuração do TCP/IP o usuário pode configurar o **MTU** (*Maximum Transfer Unit*), parâmetro que define o tamanho máximo do pacote IP inteiro. Se o tamanho máximo da área de dados do quadro de enlace for menor que o MTU o pacote IP será fragmentado. Esta fragmentação poderá ser feita em qualquer gateway no caminho do pacote até o seu destino, e não apenas na origem. No entanto, uma vez fragmentado, o pacote seguirá viagem assim (podendo ainda ser fragmentado novamente, caso em algum caminho a unidade de dados do nível físico seja menor que o tamanho deste fragmento).

A responsabilidade pela remontagem dos fragmentos é da máquina destino. A perda de um fragmento implica na perda do pacote inteiro.

Para exemplificar o processo de fragmentação, observe a Figura 3.4, nela existem duas redes Ethernet, cada uma com área de dados de 1500 bytes, ligando as duas redes temos uma rede X.25 com área de dados de 256 bytes.



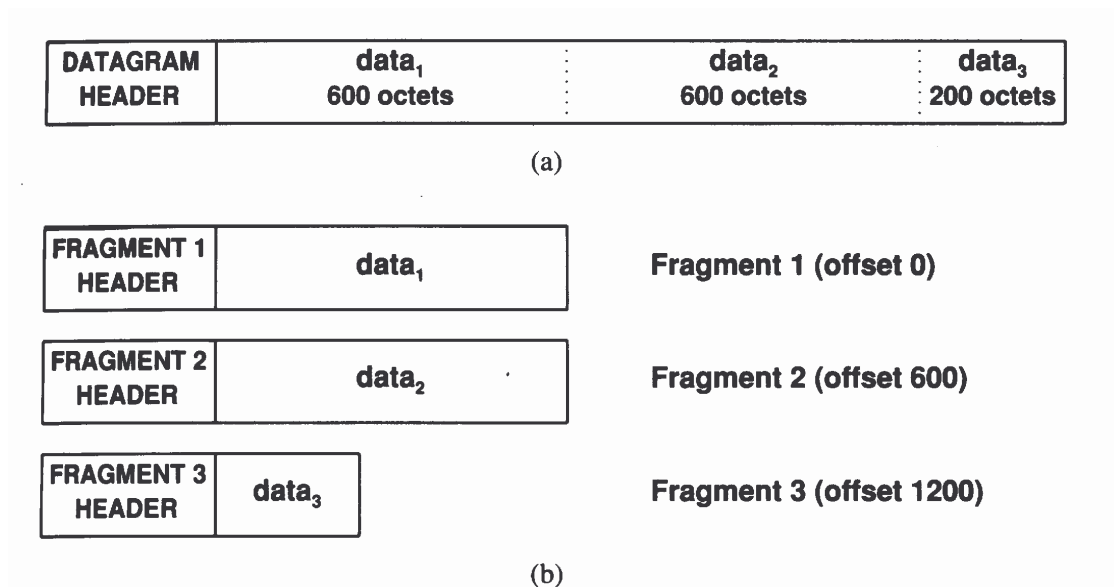
**Figura 3.4 – Redes com áreas de dados diferentes**

Observe que caso H1 queira enviar um dado para H2 não haverá fragmentação (desde que o pacote IP seja menor que 1500 bytes), o mesmo ocorre entre as máquinas H3 e H4. Agora imagine que H1 queira enviar um dado para a máquina H3 ou H4.

Será de responsabilidade de G1 fragmentar este dado para que ele possa passar pela rede X.25, e será de responsabilidade de H3 ou H4 remontar este pacote. O mesmo acontece caso H3 ou H4 queiram enviar um pacote para H1 ou H2.

Uma forma de evitar esse problema seria definir a MTU das duas redes Ethernet para 256 (mesmo tamanho da rede X.25). Neste caso não haveria fragmentação, entretanto, operações dentro da mesma rede estariam sub-utilizando o *frame* Ethernet.

Na Figura 3.5 temos um pacote IP de 1400 bytes, sendo transmitido em uma rede com MTU igual a 620 bytes e 20 bytes de cabeçalho. Será portanto necessário fragmentar este pacote em três outros. Pacote 1: 620 bytes (600 de dados e 20 de cabeçalho), Pacote 2: 620 bytes (600 de dados e 20 de cabeçalho) e Pacote 3: 220 bytes (200 de dados e 20 de cabeçalho). É claro que este processo aumenta o *Overhead* da rede, incluindo vários cabeçalhos, desperdiçando assim recursos do sistema de computação.



**Figura 3.5 – Fragmentação de um pacote**

O campo **offset** ajudará à máquina de destino na remontagem do pacote.

## 4 ICMP – Internet Control message Protocol

A Internet funciona bem se todas as máquinas operam corretamente e os roteadores estejam coerentes, o que na verdade é uma utopia. O protocolo IP pode falhar na entrega de datagramas, estas falhas podem ser ocasionadas por:

- Falha nas linhas de comunicação;
- Máquina destino desconectada da rede;
- TTL (Time-to-Live) do pacote IP expirar;
- Gateway intermediários congestionados entre outros

O ICMP (Internet Control Message Protocol) permite que os Gateways reportem erros ou forneçam informações sobre circunstâncias inesperadas (mensagens de controle). Assim como qualquer outro tráfego na Internet, ICMP viaja na área de dados de um pacote IP. **O destino final de uma mensagem ICMP é o nível IP e não uma aplicação.**

O ICMP apenas informa a máquina que enviou a mensagem que houve um erro ou uma situação inesperada. Protocolos de nível superior que vão interpretar estes erros e tomar as devidas providências. O ICMP não faz correção de erros.

Algumas mensagens reportados pelo ICMP são:

- Network Unreachable (rede não alcançável)
- Host Unreachable (host não alcançável)
- Port Unreachable (port não alcançável)
- Destination Host Unknown (Host destino desconhecido)
- Destination Network Unknown (rede destino desconhecida)
- Echo Request e Echo Reply (Solicitação de Eco e Resposta de Eco);
- Time Exceeded for Datagram – TTL (Tempo do pacote excedido);
- Entre outros

ICMP somente reporta condições de erros à fonte original. A fonte deve relatar os erros aos programas de aplicação individuais e tomar ação para corrigir o problema. Uma fragilidade do ICMP é a de só reportar erros à fonte original, não a intermediários.

Suponha que um datagrama siga uma rota através de uma série de gateways  $G_1, G_2, \dots, G_K$ . Se  $G_K$  tem rotas incorretas e envia o datagrama para o gateway  $G_E$ ,  $G_E$  reporta o erro de volta para a origem do datagrama. Porém a origem não tem nenhuma responsabilidade sobre esse erro nem controle sobre o gateway problemático. Pode inclusive nem saber qual é esse gateway. Além disso, o próprio pacote ICMP pode se perder como qualquer datagrama IP, não informando a ninguém do problema.

O comando PING (ver capítulo de aplicações) por exemplo, faz uso de um dos tipos de pacote ICMP. A máquina origem envia pacote ICMP do tipo **echo request**. A máquina que recebe essa mensagem responde com ICMP do tipo **echo reply**. Algumas versões de PING

enviam vários pacotes e devolvem estatísticas. Se o PING tem sucesso, significa que as principais partes do sistema de transporte estão funcionando.

## 4.1 Entrega de Mensagens ICMP

Uma mensagem ICMP requer dois níveis de encapsulamento, como mostra a Figura 4.1. Apesar de ser encapsulada no nível IP, o ICMP não é considerado um protocolo de alto nível. O ICMP é parte do nível IP.

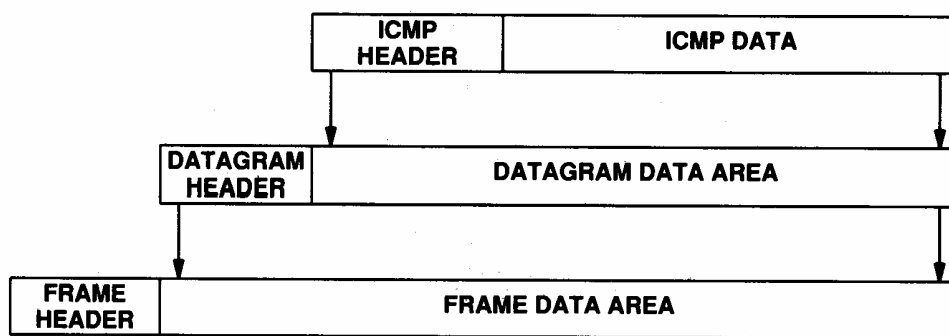


Figura 4.1 – Encapsulamento do ICMP em um pacote IP

Observe que o pacote ICMP (Dados + Cabeçalho) é colocado na área de dados do pacote IP (Dados + Cabeçalho), e este novo dado é por sua vez colocado no frame, que é a parte física propriamente dita de transmissão.

## 4.2 Tipos de Mensagem ICMP

Cada mensagem ICMP tem seu próprio formato, mas todas elas começam com os campos abaixo:

- **TYPE** (8 bits), que identifica a mensagem
- **CODE** (8 bits), que fornece mais informações sobre a mensagem
- **CHECKSUM** (16 bits)

A Figura 4.2 relaciona os possíveis tipos para o campo **TYPE**.

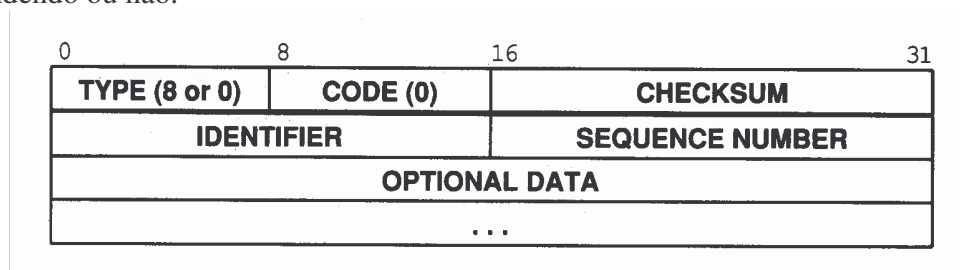
Type Field	ICMP Message Type
0	Echo Reply
3	Destination Unreachable
4	Source Quench
5	Redirect (change a route)
8	Echo Request
11	Time Exceeded for a Datagram
12	Parameter Problem on a Datagram
13	Timestamp Request
14	Timestamp Reply
15	Information Request (obsolete)
16	Information Reply (obsolete)
17	Address Mask Request
18	Address Mask Reply

**Figura 4.2 – Possíveis tipos de mensagens ICMP**

Além destes campos, o ICMP que reporta erro sempre inclui o cabeçalho e os primeiros 64 bits de dados do pacote causador do problema. Isto permite ao remetente descobrir que protocolo e que aplicação são responsáveis pelo datagrama.

### 4.2.1 Echo Request & Echo Reply

A Figura 4.3 descreve o campo de solicitação (Echo Request) e resposta (Echo Reply) do tipo eco. Essa mensagem é usada pelo programa PING para verificar se uma máquina esta respondendo ou não.



**Figura 4.3 – Mensagem do tipo Echo Request & Echo Reply**

Os campos TYPE, CODE e CHECKSUM são os mesmos descritos acima. A Tabela 4.1 relaciona os outros campos com as suas funcionalidades.

Campo	Descrição
TYPE=8	Echo Request (Requisição de resposta)
TYPE=0	Echo Reply (Resposta à solicitação anterior)
IDENTIFIER e SEQUENCE NUMBER	São usados para identificar qual mensagem foi enviada e qual esta sendo recebida
OPTIONAL DATA	Campo opcional, que dependendo da implementação pode retornar dados ao remetente, como por exemplo o tempo gasto para se alcançar a máquina.

Tabela 4.1 – Descrição dos campos da mensagem Echo Request &amp; Echo Reply

### 4.2.2 Reports unreachable destination

A Figura 4.4 ilustra a mensagem que é enviada para informar que uma máquina ou rede não esta alcançável (unreachable).

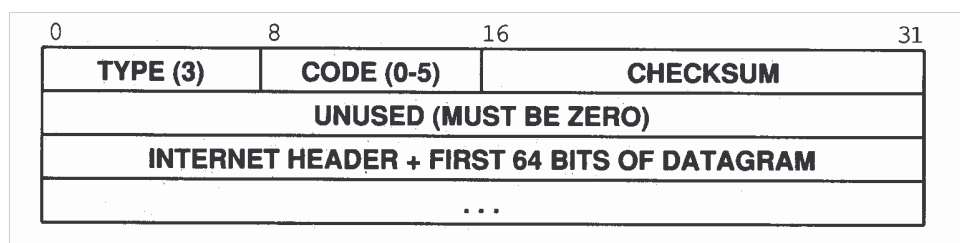


Figura 4.4 – Cabeçalho padrão para endereço não alcançável

Para identificar uma mensagem deste tipo basta verificar o campo **TYPE=3**. O campo **CODE** pode receber valores de 0 à 12. O significado de cada campo é mostrado na Figura 4.5

Code Value	Meaning
0	Network Unreachable
1	Host Unreachable
2	Protocol Unreachable
3	Port Unreachable
4	Fragmentation Needed and DF set
5	Source Route Failed
6	Destination network unknown
7	Destination host unknown
8	Source host isolated
9	Communication with destination network administratively prohibited
10	Communication with destination host Administratively prohibited
11	Network unreachable for type of service
12	Host unreachable for type of service

Figura 4.5 – Valores dos campos CODE para mensagens com TYPE=3



### 4.2.3 Controle de Fluxo e de Congestionamento

Quando um Gateway está com suas filas de envio de pacotes cheias e não tem mais espaço para armazenar novos pacotes, ele terá que descartar os novos pacotes que chegarem. De forma a avisar aos remetentes que parem de mandar pacotes, o Gateway pode enviar uma mensagem ICMP do tipo QUENCH para que a origem pare de transmitir dados. A Figura 4.6 ilustra mostra os cabeçalho padrão desta mensagem.

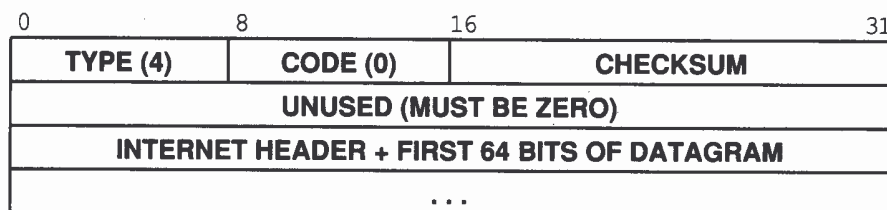


Figura 4.6 – Cabeçalho padrão para controle de congestionamento

Uma mensagem de congestionamento é identificada pelo campo **TYPE=4**, **CODE=0**, e o campo **INTERNET HEADER** conterá o endereço do gateway problemático, mais **64 bits do datagrama**.

### 4.2.4 Route Change Request (Redirect)

Quando um Gateway detecta que um host está usando uma rota não otimizada, ele envia um ICMP de REDIRECT (Figura 4.7) requisitando que o host mude sua tabela de rotas. O host mudará sua tabela de rotas caso ele esteja configurado para trabalhar com roteamento dinâmico.

Esta mensagem está restrita a um Gateway e um host fisicamente conectados (na mesma rede).

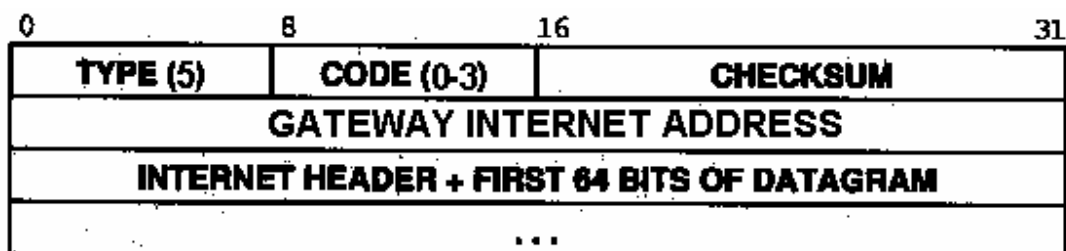


Figura 4.7 – Mensagem de mudança de rota

Uma mensagem de mudança de rota é identificada pelo campo **TYPE=5** e o campo **CODE=0 à 3**. O campo **INTERNET HEADER** conterá o endereço do gateway problemático, mais **64 bits do datagrama**.

Suponha por exemplo que eu tenha um Host conectado a dois Gateways (G1 e G2), nas redes 1 e 2 respectivamente (Figura 4.8).

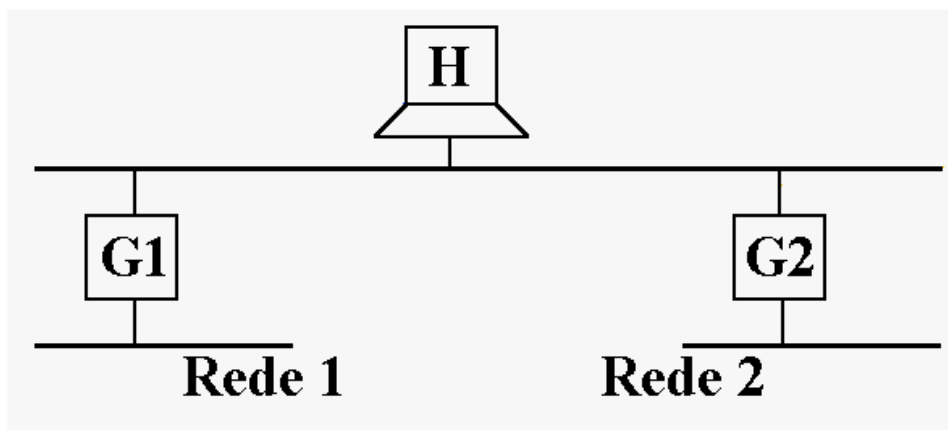


Figura 4.8 – Exemplo de mensagem de redirecionamento de rota

Se o Host H mandar um pacote para a Rede 1, via o Gateway G2. Este Gateway (G2), retornará uma mensagem de ICMP REDIRECT para o Host H. O Host H mandará então o pacote pelo Gateway G1, e este será entregue corretamente à Rede 1.

#### 4.2.5 Detecta rota circular ou excessivamente longa

Quando um Gateway detecta que o campo TTL (Time-to-Live) do datagrama IP está com o valor ZERO, ele manda o ICMP TIME EXCEEDED para a origem (Figura 4.9).

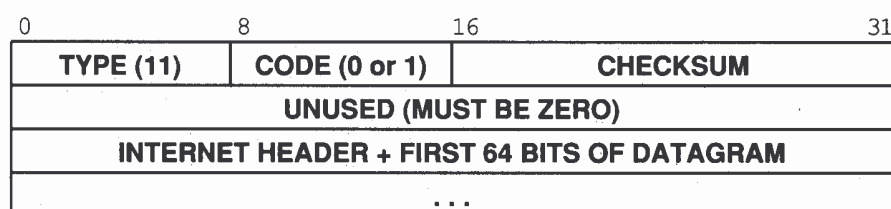


Figura 4.9 – Cabeçalho ICMP TIME EXCEEDED

Caso o campo **CODE** retorne o valor **0** indicará que o TTL zerou, caso retorne **1**, indicará que a remontagem de fragmentos excedeu o tempo máximo.

## 4.2.6 Reporta outros problemas

Quando um Gateway detecta um problema diferente dos citados anteriormente (por exemplo, erro no cabeçalho), ele envia um ICMP **PARAMETER PROBLEM** para a origem (Figura 4.10).

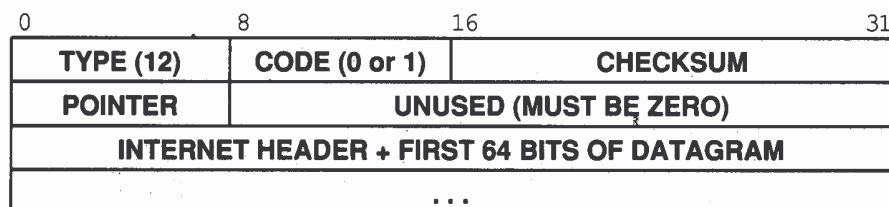


Figura 4.10 – Cabeçalho ICMP para outros problemas

O campo **POINTER** é usado para apontar para o octeto do datagrama original que causou o problema. Já o campo **CODE** virá com o valor **1** caso o campo **POINTER** esteja sendo usado ou com o campo **0** se o campo **POINTER** não estiver sendo utilizado.

## 4.2.7 Sincronização de relógio e estimativa de tempo de transito

Uma máquina pode usar o ICMP **REQUEST TIMESTAMP** para requisitar a “hora do dia” de outra máquina. A máquina endereçada retorna um ICMP **TIMESTAMP REPLY** (Figura 4.11).

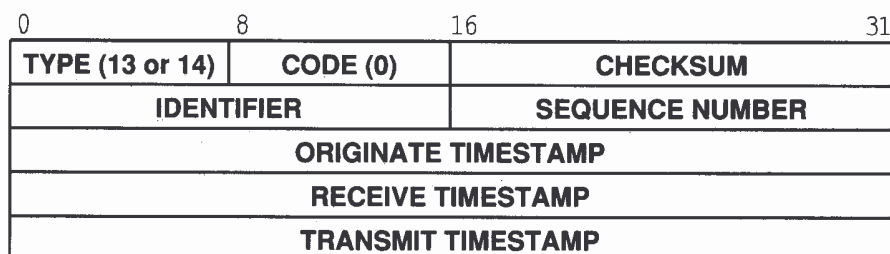


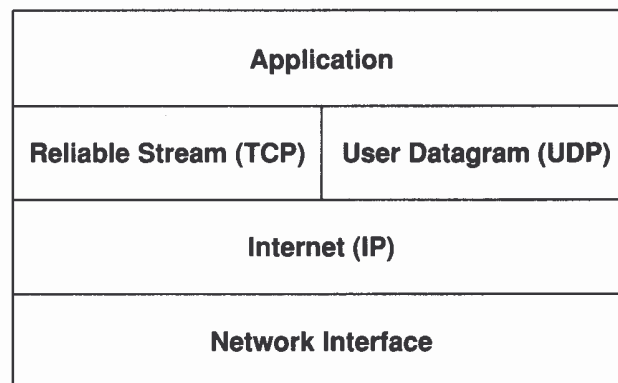
Figura 4.11 – Cabeçalho ICMP para mensagem de sincronização de relógios

O campo **TYPE = 13** indica usa mensagem de solicitação (request) e o campo **TYPE = 14** indica a resposta (reply). O campo **ORIGINATE TIMESTAMP** é preenchido pelo remetente e os campos **RECEIVE TIMESTAMP** e **TRANSMIT TIMESTAMP** são preenchidos pelo destinatário no momento da chegada do *request* e no momento da saída do *reply* respectivamente. Todos os tempos são em milisegundos, a partir da meia noite.

## 5 Camada de Transporte

A camada de transporte da arquitetura TCP/IP tem por responsabilidade transferir um dado fim-a-fim entre duas máquinas, ou seja, é de responsabilidade desta camada transferir um dado de uma máquina A, até uma máquina B, independente da infra-estrutura de comunicação que se tenha entre elas.

A arquitetura Internet define dois protocolos para esta camada: o TCP (Transmission Control Protocol – Protocolo com controle de transmissão) e o UDP (User Datagram Protocol – Protocolo sem controle de transmissão) como ilustra a Figura 5.1.



**Figura 5.1 – Camada de Transporte da Arquitetura TCP/IP**

A diferença básica entre os dois é que o TCP é um protocolo confiável, ele garante a entrega de informações corretas entre duas estações, já o UDP não faz isso. Por ter que se preocupar com diversos detalhes, tais como sequencialização, CheckSum, etc o TCP é muito mais lento que o UDP. Veremos que existem aplicações que necessitam da confiabilidade do TCP e outras da velocidade e particularidades do UDP. As próximas sessões discutem com mais detalhes estes dois protocolos, antes porém é preciso entender o conceito de aplicação cliente-servidor.

### 5.1 Aplicações Cliente-Servidor

As aplicações Cliente-Servidor são caracterizadas basicamente pela presença de um **Servidor**, responsável por prover determinadas facilidades e um **Cliente** que irá acessar as facilidades que são providas pelo servidor.

Basicamente o Cliente solicita uma determinada tarefa ao Servidor, este a processa e devolve o resultado ao Cliente, diminuindo substancialmente a troca de mensagens na rede, já que as perguntas e respostas costumam ser bem curtas.

Na maioria dos casos, temos um Servidor, em geral uma máquina melhor que as demais, servindo a vários Clientes simultaneamente. Dentro deste contexto para uma mesma

aplicação, por exemplo o **telnet** (Terminal Remoto) existe o **telnet do servidor**, conhecido também como *daemon* e o **telnet do cliente**.

Quando um usuário chama a aplicação telnet no seu terminal, ele está usando o telnet do cliente, só que para esta aplicação funcionar ele precisa se conectar a um servidor, que **precisa** estar rodando o telnet do servidor. A este conjunto de aplicações do cliente e do servidor damos o nome de arquitetura cliente-servidor ou *client-server*.

A Figura 5.2 mostra algumas aplicações do modelo TCP/IP. Observe que o Telnet, FTP e SMTP utilizam o TCP como protocolo de transporte, já aplicações tipo NSF, SNMP e TFTP utilizam o UDP.

Telnet	FTP	SMTP	NSF	SNMP	TFTP	Nivel de Aplicacoes
Fluxo Confiavel(TCP)			Datagrama de Usuario (UDP)			Nivel de Transporte
Internet Protocol						Nivel da Rede
Interface Fisica						Nivel de Enlace e Fisico

Figura 5.2 – Algumas aplicações do modelo TCP/IP

### 5.1.1 O conceito de Porta

Quando uma aplicação cliente deseja conectar-se à aplicação servidor, esta deve fornecer o endereço IP ou nome hierárquico (que será convertido em endereço IP pelo serviço de DNS, pois na Internet só trafegam endereços IPs) do servidor. Neste caso teremos um endereço IP de origem e um endereço IP de destino que identificará uma comunicação. Entretanto, estes dois campos não são suficientes para identificar univocamente uma conexão, pois o servidor poderá estar *rodando* diversas aplicações simultaneamente. Como então eu conseguiria identificar que eu desejo me conectar à aplicação A ou à aplicação B?

Para resolver este problema, foi criado o conceito de **Porta**. A ideia é bastante simples: Para cada aplicação é definido um número de porta único que a distingue das demais aplicações, chamaremos à porta do servidor de *Porta de Destino*.

É possível agora informar o *endereço IP de destino* e o *endereço da Porta de Destino*, para identificar uma determinada aplicação. Aplicações como Telnet, FTP (Transferência de Arquivos), HTTP (Home Pages), entre outras tem suas portas bem definidas e em geral não precisam ser informadas pelas aplicações, já são colocadas nos pacotes de Transporte automaticamente por estes programas.

Observe que o endereço IP de origem, endereço IP de destino e Porta de destino ainda não definem unicamente uma aplicação. Pois dentro da máquina cliente (o mesmo endereço IP de origem) eu posso querer me conectar com o mesmo servidor (endereço IP de destino) na mesma aplicação (porta de destino). Como eu conseguiria identificar quais das duas aplicações da minha máquina deve receber determinada resposta?

Isto acontece com frequência quando abrimos diversas instâncias de um mesmo programa. Quando por exemplo estamos navegando na Internet através de um *browser* (Internet Explorer ou Netscape) e abrimos várias janelas que tem como endereço o mesmo servidor de destino. Para resolver este problema é incluindo à nossa lista o conceito de *Porta de origem*. A porta de origem vai identificar qual das instâncias de um mesmo programa esta se conectando a um mesmo serviço, numa mesma máquina destino.

A Figura 5.3 ilustra este fato, mostrando o formato de um pacote TCP (observe que o pacote UDP também terá que ter porta de origem e porta de destino também). O campo *Source Port*, identifica a porta de origem e o campo *Destination Port*, identifica a porta de destino. Os campos de IP de origem e IP de destino são colocados no cabeçalho do pacote IP e não do TCP ou UDP. Os demais campos serão explicados na sessão **Protocolo TCP**.

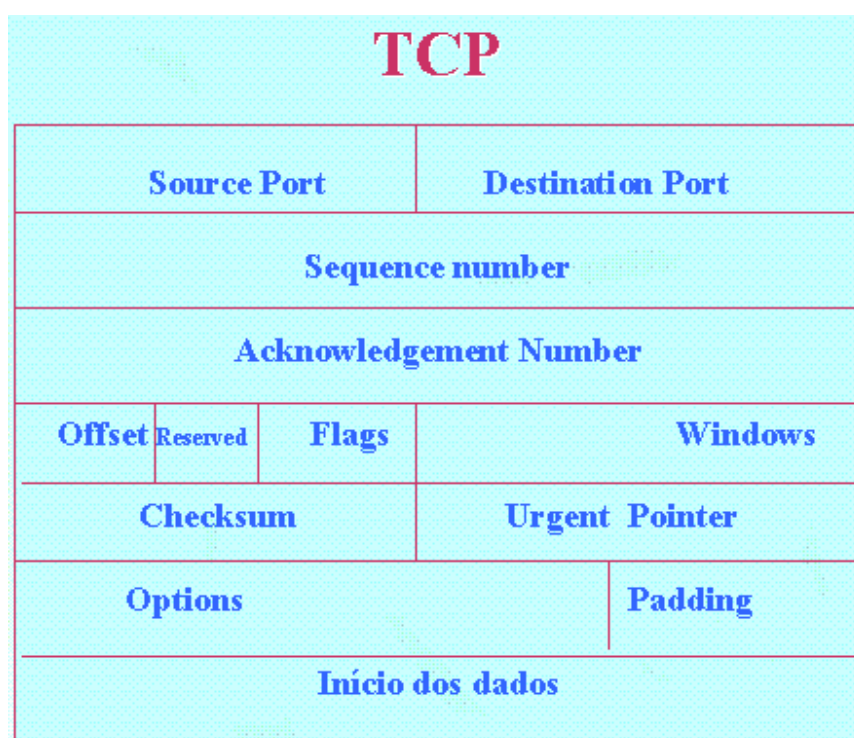


Figura 5.3 – Porta de Origem e Destino no pacote TCP

### 5.1.2 Utilizando a Arquitetura Cliente-Servidor

Para utilizarmos desta arquitetura primeiramente precisamos instalar um determinado serviço em um determinado servidor. Suponha que iremos instalar o serviço de WWW (World Wide Web) que utiliza o protocolo HTTP no servidor **www.uol.com.br (200.246.5.65)**. O serviço de WWW é um serviço bastante difundido e por isso já possui uma porta padrão, a porta 80. Temos portanto que do ponto de vista de um cliente:

**IP de destino: 200.246.4.65**  
**Porta de destino: 80**

Suponha agora que você esteja na máquina 200.241.16.8 e deseja acessar este *site* através de um *Browser* qualquer. Você precisa de uma porta de origem. Este número de porta é fornecido dinamicamente pelo sistema operacional, digamos que num dado momento a porta de origem seja: 3478. Logo já temos todo o cabeçalho para estabelecer a comunicação

**IP de origem: 200.241.16.8**  
**Porta de Origem: 3478**

Caso você abra uma nova janela do seu *Browser*, e se conecte ao mesmo lugar, utilizando o mesmo serviço o sistema operacional lhe dará uma nova Porta de Origem, para que não se confunda com as outras portas em uso. Digamos que a nova porta seja 4312. Teremos portanto:

**IP de origem: 200.241.16.8**  
**Porta de Origem: 4312**

Desta forma é possível endereçar corretamente uma determinada informação.

Se outra pessoa, numa outra máquina, digamos 200.30.20.83 quiser acessar o mesmo site, pode ocorrer do sistema operacional dar a ela o mesmo número de porta, digamos 4312. Logo teríamos:

**IP de origem: 200.30.20.83**  
**Porta de Origem: 4312**

Observe que o IP de origem é outro e portanto não haverá confusão na hora da entrega dos pacotes.

Os servidores de aplicação estão sempre ativos e esperando conexões. As conexões acontecem assincronamente e por um período de tempo indeterminado. Nas máquinas UNIX é possível ver quais são os serviços ativos, e quais os programas que os manipulam através dos respectivos comandos:

```
% cat /etc/services
```

```
% cat /etc/inetd.conf
```

Nas máquinas Windows NT é possível ver quais são os serviços ativos através da sequência: Iniciar / Configuração / Painel de Controle / Serviços, sendo possível aí inicializar ou parar um determinado serviço.

## 5.2 O Protocolo TCP (Transmission Control Protocol)

O TCP é um protocolo do nível de transporte da arquitetura Internet. Este protocolo garante que um dado enviado de um computador A para um computador B em qualquer rede (que esteja ligada via TCP/IP) vai ser entregue corretamente.

O TCP é um protocolo **confiável e orientado a conexão**. Este protocolo provê uma *Interface* para a camada de aplicação que deixa totalmente transparente os procedimentos de retransmissão, sequencialização, correção e verificação de erros, controle de fluxo, entre outros, sendo de sua responsabilidade estas tarefas.

Também é de responsabilidade do TCP especificar como duas máquinas iniciam uma conexão, transferem dados e terminam uma conexão. Mostraremos a seguir as principais responsabilidades do protocolo TCP.

1. **Estabelecimento e Liberação de conexão** — Como o TCP é orientado à conexão, antes de começar qualquer processo de transferência de dados é preciso estabelecer uma conexão entre a máquina origem e a máquina destino. Após o término da transferência de dados a conexão é desfeita.
2. **Transferência de dados** — Após o estabelecimento da conexão é possível começar o processo de transferência de dados, que pode ser por mensagens variáveis ou de tamanho fixo e no formato *full-duplex* (as duas máquinas podem estar transmitindo dados simultaneamente).
3. **Transferência de dados urgentes** — Ainda na transferência de dados é possível definir se uma mensagem será normal ou urgente. Esta última opção é usada geralmente para informações de controle.
4. **Multiplexação** — A multiplexação é um processo de colocar diversas conexões de transporte, no caso o TCP, em uma conexão de rede, no caso o IP, este processo é feito na máquina origem. Na máquina destino é feito o processo inverso, chamado de de-multiplexação, que consiste em separar uma conexão de rede em várias conexões de transporte.

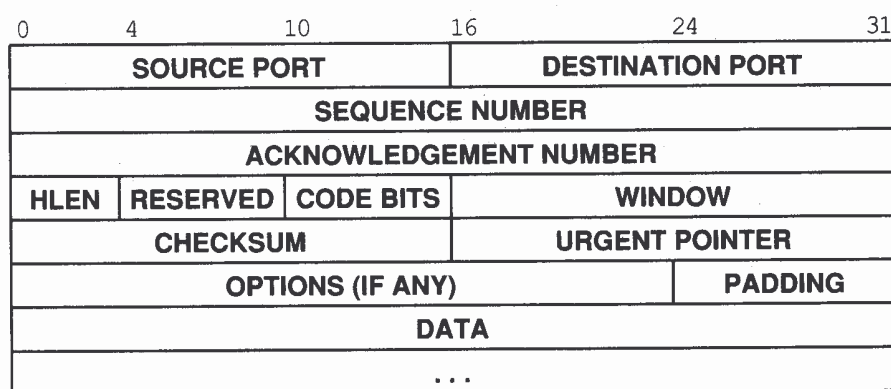


5. **Segmentação** — Acontecerá segmentação quando a área de dados do pacote IP for menor que a área de dados do pacote TCP. Neste caso o pacote TCP será fragmentada em vários pacotes IP para que possa ser trafegado na rede. Sendo depois remontado na máquina destino.
6. **Controle do fluxo** — Através de um sistema de *bufferização* denominada *Janela deslizante*, o TCP envia uma série de pacotes sem aguardar o reconhecimento de cada um deles. Na medida em que recebe o reconhecimento de cada bloco enviado, atualiza o *buffer* (caso reconhecimento positivo) ou reenvia os dados ao destinatário (caso reconhecimento negativo ou não reconhecimento após um *timeout* pré estipulado).
7. **Controle de erros** — Além da numeração dos segmentos transmitidos, vai junto com o cabeçalho (header) uma soma verificadora dos dados transmitidos (checksum), assim o destinatário verifica a soma com o cálculo dos dados recebidos e consegue identificar se os dados estão corretos ou não.

### 5.2.1 Formato do pacote TCP

A unidade de transferência entre o protocolo TCP de duas máquinas é chamado de **Segmento**. Os segmentos são trocados para estabelecer conexões, transferir dados, enviar reconhecimentos e fechar conexões. Dado que TCP usa a técnica de *Piggybacking*, um reconhecimento viajando de uma máquina A para B pode ir no mesmo segmento de dados que estão sendo enviados de A para B, embora o reconhecimento refere-se a dados enviados da máquina B para A.

A Figura 5.4 mostra o formato de um pacote TCP e suas opções.



**Figura 5.4 – Formato do pacote TCP**

A seguir são apresentados os objetivos de cada campo do pacote TCP.

1. **SOURCE PORT** – Identifica a porta de origem
2. **DESTINATION PORT** – Identifica a porta de destino
3. **SEQUENCE NUMBER** – Especifica o número de sequência de dados no mesmo segmento. É através deste número que é possível fazer a sequencialização.
4. **ACKNOWLEDGEMENT NUMBER** – Identifica o número de bytes que a fonte esperar receber na próxima transmissão.
5. **HLEN** – Este campo especifica a partir da onde irá começar o campo de dados. Isto é necessário pois dependendo do tamanho do campo de opções (OPTION), o campo de dados irá começar em uma determinada posição.
6. **RESERVED** – Campo reservado para uso futuro.
7. **CODE BITS** – Determina o propósito e conteúdo do segmento, codificado na forma da Tabela 5.1.

Opção	Significado
URG	Campo de ponteiro Urgente é válido
ACK	Campo de Reconhecimento é válido
PSH	Este segmento solicita um <b>Push</b>
RST	<i>Resetar</i> a conexão
SYN	Sincroniza números de Sequências
FIN	O transmissor chega ao fim do fluxo de bytes

**Tabela 5.1 – Significado do Code Bits**

8. **WINDOW** – Através deste campo o protocolo TCP indica quantos dados ele tem capacidade de receber em seu *buffer*.
9. **CHECKSUM** – É usado para verificar a integridade tanto do cabeçalho como dos dados do segmento TCP.
10. **URGENT POINT** – É possível, através deste campo especificar que alguns dados devem ser entregues de forma urgente. Esta informação é repassada ao nível IP que tenta executar esta operação. Esta opção é totalmente dependente dos recursos do sistema de comunicação.
11. **OPTION (IF ANY)** – Campo reservado para opções do TCP (caso haja algum).
12. **PADDING** – Este campo serve para colocar alguns parâmetros que podem ser requeridos na escolha de uma determinada opção no campo OPTION.

### 13. DATA – Campo de dados do TCP.

#### 5.2.2 Portas bem conhecidas do TCP

Como já foi dito anteriormente, uma mesma máquina pode prover diversos serviços concorrentemente, bastando para isso que cada serviço (ou aplicação) esteja colocado em uma porta diferente. Alguns serviços são bastante utilizados na Internet, e por isso suas portas são bem conhecidas. A Figura 5.5 mostra algumas portas conhecidas. Nela é possível ver, por exemplo, que o serviço de terminal remoto (telnet) está na porta 23, que o serviço de correio eletrônico (SMTP) está na porta 25, e que o serviço de transferência de arquivos (FTP) está na porta 21, entre outros. Observe que o serviço de WWW (HTTP) não está listado na Figura 5.5, entretanto pela popularização cada vez maior deste serviço, ele é colocado na maioria dos casos na porta 80, podendo eventualmente ser colocado em outra porta. As portas acima de 1024 podem ser usadas por qualquer aplicação cliente-servidor

Decimal	Keyword	UNIX Keyword	Description
0			Reserved
1	TCPMUX	-	TCP Multiplexor
5	RJE	-	Remote Job Entry
7	ECHO	echo	Echo
9	DISCARD	discard	Discard
11	USERS	systat	Active Users
13	DAYTIME	daytime	Daytime
15	-	netstat	Network status program
17	QUOTE	qotd	Quote of the Day
19	CHARGEN	chargen	Character Generator
20	FTP-DATA	ftp-data	File Transfer Protocol (data)
21	FTP	ftp	File Transfer Protocol
23	TELNET	telnet	Terminal Connection
25	SMTP	smtp	Simple Mail Transport Protocol
37	TIME	time	Time
42	NAMESERVER	name	Host Name Server
43	NICNAME	whois	Who Is
53	DOMAIN	nameserver	Domain Name Server
77	-	rje	any private RJE service
79	FINGER	finger	Finger
93	DCP	-	Device Control Protocol
95	SUPDUP	supdup	SUPDUP Protocol
101	HOSTNAME	hostnames	NIC Host Name Server
102	ISO-TSAP	iso-tsap	ISO-TSAP
103	X400	x400	X.400 Mail Service
104	X400-SND	x400-snd	X.400 Mail Sending
111	SUNRPC	sunrpc	SUN Remote Procedure Call
113	AUTH	auth	Authentication Service
117	UUCP-PATH	uucp-path	UUCP Path Service
119	NNTP	nntp	USENET News Transfer Protocol
129	PWDGEN	-	Password Generator Protocol
139	NETBIOS-SSN	-	NETBIOS Session Service
160-223	Reserved		

Figura 5.5 – Algumas portas TCP bem conhecidas

## 5.3 O Protocolo UDP (User Datagram Protocol)

O protocolo UDP usa o mesmo princípio de portas e de aplicações distribuídas do TCP. A grande diferença entre o TCP e o UDP é que o UDP é muito mais *leve* que o TCP, pois ele não garante a entrega dos dados, sequencialização, fluxo de mensagens, etc. O UDP não é orientado à conexão, ele é *Connectionless*, ou seja, não é necessário estabelecer uma conexão para começar a transferir os dados e depois se preocupar em terminar a conexão.

No UDP, você simplesmente envia o dado de uma máquina (IP Origem, Porta Origem) para uma máquina de destino (IP Destino, Porta Destino). Caso a rede consiga enviar o pacote ao seu destino, sem erro, este chegará perfeitamente. Caso contrário não haverá retransmissão deste pacote. Talvez você esteja se perguntando para que serve um serviço que não garante que a informação vai chegar do outro lado, e caso chegue também não garante que a ordem é correta ou que os próprios dados estejam corretos?

### 5.3.1 Formato do Pacote UDP (User Datagram Protocol)

A Figura 5.6 mostra o formato do pacote UDP.

<b>PORTA UDP ORIGEM</b>	<b>PORTA UDP DESTINO</b>
<b>Tamanho da mensagem UDP</b>	<b>UDP Checksum</b>
<b>Dados</b>	
....	

**Figura 5.6 – Formato do pacote UDP**

A estrutura do pacote UDP é muito simples, os campos que compõe o pacote são:

1. **PORTA UDP ORIGEM** – Este campo contém a porta UDP usado pela origem (opcional).
2. **PORTA UDP DESTINO** – Este campo contém a porta UDP de destino.
3. **TAMANHO DA MENSAGEM UDP** – Este campo contém o tamanho de todo o pacote UDP (Dados + Cabeçalho)
4. **UDP CHECKSUM** – Este campo é utilizado na recepção dos dados para verificar se o dado recebido esta certo ou não. É um campo opcional, para tanto basta colocar o valor

0 (zero) no campo. Caso o valor zero seja informado o receptor não tem como saber se o dado está correto ou não.

5. **DADOS** – Este campo contém os dados propriamente ditos.

### 5.3.2 Portas bem conhecidas do UDP

Tal como no TCP, o UDP também possui um conjunto de portas bem conhecidas, como ilustra a Figura 5.7. As portas reservadas vão de 1 à 1000, e as portas livres vão de 1001 à 65535.

Na Figura 5.7 é possível ver, por exemplo, que o serviço de *DayTime*, que informa o dia e a hora daquela máquina é utilizado na porta 13, o serviço de DNS (Domain Name Service) está na porta 53, o serviço de Gerenciamento da rede (SNMP) está na porta 161 e 162, entre outros serviços.

Decimal	Keyword	UNIX Keyword	Description
0	-	-	Reserved
7	ECHO	echo	Echo
9	DISCARD	discard	Discard
11	USERS	systat	Active Users
13	DAYTIME	daytime	Daytime
15	-	netstat	Who is up or NETSTAT
17	QUOTE	qotd	Quote of the Day
19	CHARGEN	chargen	Character Generator
37	TIME	time	Time
42	NAMESERVER	name	Host Name Server
43	NICNAME	whois	Who Is
53	DOMAIN	nameserver	Domain Name Server
67	BOOTPS	bootps	Bootstrap Protocol Server
68	BOOTPC	bootpc	Bootstrap Protocol Client
69	TFTP	tftp	Trivial File Transfer
111	SUNRPC	sunrpc	Sun Microsystems RPC
123	NTP	ntp	Network Time Protocol
161	-	snmp	SNMP net monitor
162	-	snmp-trap	SNMP traps
512	-	biff	UNIX comsat
513	-	who	UNIX rwho daemon
514	-	syslog	system log
525	-	timed	Time daemon

**Figura 5.7 – Portas UDP bem conhecidas**

## 6 Protocolos de Roteamento

A arquitetura TCP/IP é uma arquitetura de interconexão de redes. Neste sentido, se duas máquinas A e B estão conectadas a uma rede TCP/IP, é possível que as duas máquinas conversem entre si, independente de quantas redes existam entre as duas máquinas.

Quando a máquina A deseja falar com a máquina B surge um pequeno problema. Como podem existir diversas redes entre A e B, qual o caminho o pacote IP deve percorrer para sair da máquina A e ir até a máquina B? A este processo chamamos de **Roteamento**. O roteamento consiste em encontrar **um caminho** (pode não ser o melhor caminho) de tal forma que as informações saiam de A e consigam atingir a máquina B e vice-versa. O nome **Roteador** é dado à máquina ou equipamento que toma tal decisão.

Chamaremos de **Roteador** ou **Gateway** ao equipamento que conecta duas ou mais redes fisicamente. E denotaremos por **Host** a uma máquina que esta conectada a uma rede física. As decisões de roteamento são tomadas tanto por **Hosts** ou **Gateways**.

O roteamento IP consiste em decidir para onde enviar um datagrama baseando-se no endereço IP destino contido neste datagrama. Existem dois tipos de roteamento: O roteamento direto e o roteamento indireto.

### 6.1 Roteamento Direto

O roteamento direto ocorre quando as duas máquinas envolvidas **estão na mesma rede física**. Neste caso basta o transmissor encapsular o pacote IP em uma quadro físico, com o endereço do destino, que é conseguido via o protocolo ARP e enviar este pacote diretamente na rede. A máquina destino receberá o pacote físico descapsulará o pacote e tratará o datagrama IP. Observe que não há a necessidade da figura do Gateway, haja visto que as duas máquinas envolvidas estão na mesma rede física, por exemplo um barramento Ethernet.

Para saber se a máquina destino está na mesma rede se faz uma comparação entre os endereços IP fonte e destino, especificamente entre os campos que identificam a rede. Se ambos campos são iguais (todos igual a 1 binário), significa que o datagrama pode ser enviado diretamente sem ter que passar por um gateway.

### 6.2 Roteamento Indireto

O roteamento indireto ocorre quando as duas máquinas envolvidas **não estão na mesma rede física**. Neste caso, gateways intermediários terão que manusear este pacote até que ele chegue à rede física de destino, para ai ser tratado como um roteamento direto.

No roteamento indireto o transmissor deve identificar um Gateway para o qual o datagrama deve ser enviado, para que este Gateway tome a decisão de continuar enviando o pacote para a rede física que o conecta ou para um outro Gateway. Observe que o endereço deste Gateway deve estar na mesma rede física da máquina de origem.

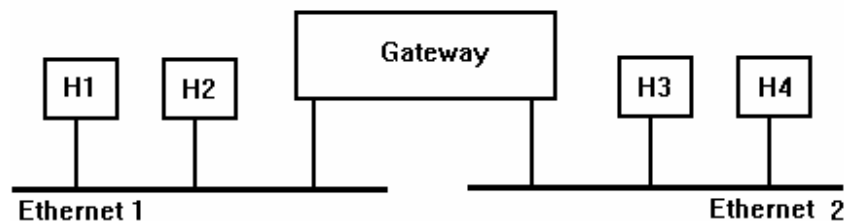
Cada máquina na Internet, seja ela um Host ou um Gateway, deverá possuir uma *tabela de rotas*.

A decisão que um Host deverá tomar, baseado na sua tabela de rotas é:

- O roteamento será direto ou não?
- Caso não seja, para qual Gateway deverá enviar o pacote?

Já a decisão do Gateway é:

- Se a máquina destino está na mesma rede física e pode ser alcançada diretamente? (Neste caso ele é o Gateway Final)
- Caso contrário, para qual outro Gateway entregar o pacote?



**Figura 6.1 – Duas redes Ethernet ligadas por um Gateway**

Na Figura 6.1, se H1 desejar se comunicar com H2, isto será feito utilizando o **roteamento direto**, ou seja, H1 terá que mapear o endereço IP de H2 em seu endereço físico (via ARP), encapsular o pacote IP no quadro físico e enviar na rede para que H2 possa recebê-lo. O mesmo ocorre se H3 quiser se comunicar com H4 ou vice-versa.

Agora se H1 desejar se comunicar com H3, será feito um **roteamento indireto**. H1 escolherá o Gateway como a “máquina no caminho para H3”. Observe que a princípio, H1 não sabe por quantos gateways o pacote terá que passar até atingir a máquina H3. A única informação que ele tem é que como o endereço de destino não está conectado à sua rede física ele precisa enviar este pacote para o Gateway, para que este tome uma decisão.

Para que H1 consiga então mandar o pacote para H3 ele terá que enviá-lo antes ao Gateway. Neste caso, H1 fará o mapeamento do endereço IP do gateway no seu endereço físico (via ARP) e enviará o pacote para o gateway. Ao receber o pacote, o Gateway verifica que precisa fazer uma entrega direta para H3, e também via ARP o faz.

Observe que apesar de H1 direcionar o pacote para o Gateway, este direcionamento é apenas no nível físico. O pacote não será alterado, ou seja, continuará com o endereço IP de destino H3. Desta forma o Gateway poderá então roteá-lo.

Este tipo de roteamento é mais difícil que o roteamento direto, já que o remetente deve identificar um gateway ao qual o datagrama pode ser enviado, depois o gateway deve enviar o datagrama a rede destino, ou a outro gateway, e assim sucessivamente.

Vamos supor que tenham muitas redes interconetadas por gateways, mas só tenham dois hosts em cada extremo da interconexão das redes, quando um host quer enviar ao outro, ele encapsula o datagrama e o envia ao gateway mais próximo. Uma vez que o quadro chega ao gateway, o software de IP extraí o datagrama encapsulado, e a rotina de roteamento IP seleciona o próximo gateway que formará parte do caminho que levará o datagrama ao Host destino.

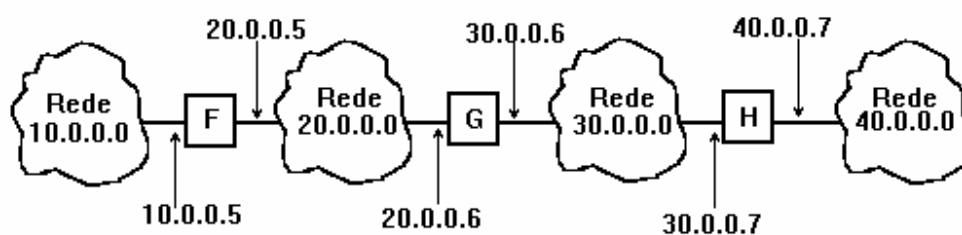
Talvez você esteja se perguntando, como que eu vou ter certeza que as escolhas de gateways levarão o pacote ao endereço IP de destino correto?

Este processo é feito através da manipulação de um banco de dados distribuído. Onde cada rede tem que informar somente os dados referentes à sua rede e manipular o que chamamos de **Tabela de Roteamento**.

### 6.3 Tabela de Roteamento

Tipicamente, uma tabela de roteamento (ou tabela de rotas), contém um par (N,G), onde N é o endereço IP da rede destino e G é o endereço IP do próximo Gateway no caminho da rede N.

Os Gateways não conhecem o caminho completo até a máquina destino e sim o próximo passo em direção àquela rede. As tabelas de rota sempre apontam para Gateways que estão na mesma rede física da máquina que possui esta tabela. Na Figura 6.2 vemos quatro redes interligadas por três gateways.



**Figura 6.2 – Configuração de quatro redes e três gateways**

Veremos a seguir quais deveriam ser as tabelas de rotas do gateway F (Tabela 6.1), G (Tabela 6.2) e H (Tabela 6.3).



Rede Destino	Mandar Para
10.0.0.0	Direto
20.0.0.0	Direto
30.0.0.0	20.0.0.6
40.0.0.0	20.0.0.6

**Tabela 6.1 – Tabela de rotas do Gateway F**

Observe na Tabela 6.1 que para enviar uma mensagem para a rede 10.0.0.0 ou para a rede 20.0.0.0 o envio é feito de forma direta, pois as duas redes estão conectadas ao Gateway. Já para enviar uma mensagem para as redes 30.0.0.0 e 40.0.0.0 este envia a mensagem para o endereço IP 20.0.0.6 que é um dos endereços do Gateway G. A este gateway caberá a responsabilidade de continuar o roteamento, como visto na Tabela 6.2

Rede Destino	Mandar Para
10.0.0.0	20.0.0.5
20.0.0.0	Direto
30.0.0.0	Direto
40.0.0.0	30.0.0.7

**Tabela 6.2 – Tabela de rotas do Gateway G**

Ainda em relação ao exemplo anterior, observe que na Tabela 6.2, quando um datagrama é enviado para a rede 30.0.0.0 o roteamento é feito de forma direta e quando é enviada para a rede 40.0.0.0 é encaminhado para o endereço IP 30.0.0.7 (Gateway H), para este tomar a decisão de roteamento. E naturalmente quando um pacote é enviado para a rede 20.0.0.0 ele é enviado de forma direta, pois o Gateway esta fisicamente conectado a este rede, e quando um pacote for enviado para a rede 10.0.0.0 ele será roteado primeiramente para o endereço IP 20.0.0.5 (Gateway F), para que este possa tomar a decisão do que fazer com o pacote. Observe agora a Tabela 6.3 do Gateway H

Rede Destino	Mandar Para
10.0.0.0	30.0.0.6
20.0.0.0	30.0.0.6
30.0.0.0	Direto
40.0.0.0	Direto

**Tabela 6.3 – Tabela de rotas do Gateway H**

Nesta tabela é possível observar que quando um datagrama for enviado para a rede 10.0.0.0 ou para a rede 20.0.0.0 ele será roteado para o endereço IP 30.0.0.6 (Gateway G) e quando for enviado para a rede 30.0.0.0 ou 40.0.0.0 o roteamento será direto.

Observe que cada Gateway não guarda toda a rota que o pacote deverá seguir, e sim o próximo Gateway ou rede para o qual ele deverá ser roteado ou entregue diretamente.

É importante entender que a tabela de roteamento sempre aponta aos gateways que podem ser alcançados através da rede a qual esse gateway está conectado. Isso significa que todos os gateways listados na tabela de roteamento de uma máquina M devem conectar-se as redes às quais M está conectada diretamente. Quando um datagrama está pronto para sair de M, o protocolo IP localiza o endereço IP destino e extrai a porção da rede. Logo M usa a identificação da rede para fazer uma decisão de roteamento, selecionando um gateway que possa ser alcançado diretamente.

Nas tabelas de roteamento não é possível armazenar as informações de cada máquina destino, seria impossível manter as tabelas atualizadas, além do que as máquinas teriam problemas com armazenamento para toda a informação.

## 6.4 Rota Default

Suponha que a rede do exemplo anterior se expandisse para um dos lados, como mostra a Figura 6.3.

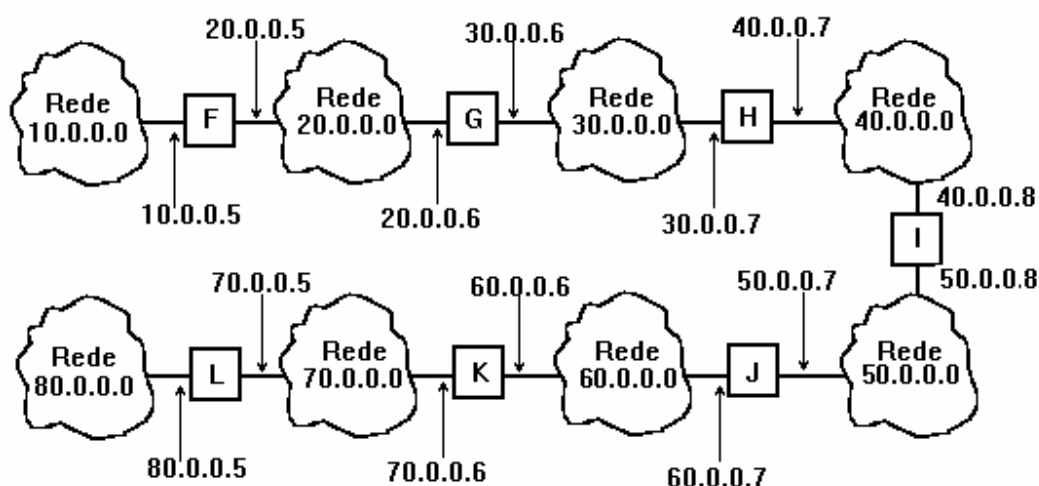


Figura 6.3 – Várias redes interconectadas por Gateways

A Tabela de Rotas do Gateway G ficaria na forma da Tabela 6.4.

Rede Destino	Mandar Para
10.0.0.0	20.0.0.5
20.0.0.0	Direto
30.0.0.0	Direto
40.0.0.0	30.0.0.7
50.0.0.0	30.0.0.7

Rede Destino	Mandar Para
60.0.0.0	30.0.0.7
70.0.0.0	30.0.0.7
80.0.0.0	30.0.0.7

**Tabela 6.4 – Tabela de Rotas do Gateway G**

Podemos notar claramente que existem várias rotas apontando para o mesmo Gateway H (endereço IP 30.0.0.7). Além disso, se a rede continuar crescendo para este lado, a tabela de rotas de G aumentaria também, e sempre apontando para o mesmo gateway H.

Para não ficarmos desperdiçando espaço em memória, a especificação TCP/IP permite que seja definida uma rota *default*, que será usada sempre que nenhuma outra rota for encontrada na tabela. Utilizando a rota *default* a tabela de rotas de G seria apenas (Tabela 6.5):

Rede Destino	Mandar Para
10.0.0.0	20.0.0.5
20.0.0.0	Direto
30.0.0.0	Direto
Default	30.0.0.7

**Tabela 6.5 –Tabela de Rotas do Gateway G usando rota *Default***

Observe que esta forma de especificar as rotas de destino é bem mais fácil que a anterior, pois o número de linhas na tabela de rotas é reduzido bruscamente.

## 6.5 Alguns exemplos práticos

Nesta sessão faremos alguns exercícios práticos de estabelecimento de rotas de *Gateways* e *Hosts*.

Faremos agora um exemplo prático, para demonstrar como é feita a tabela de rotas estáticas de três redes conectadas entre si e à Internet.

### 6.5.1 Exemplo 1

Estabeleça a tabela de rotas das máquinas G1, G2, G3, G4, H1, H2 conforme a Figura 6.4

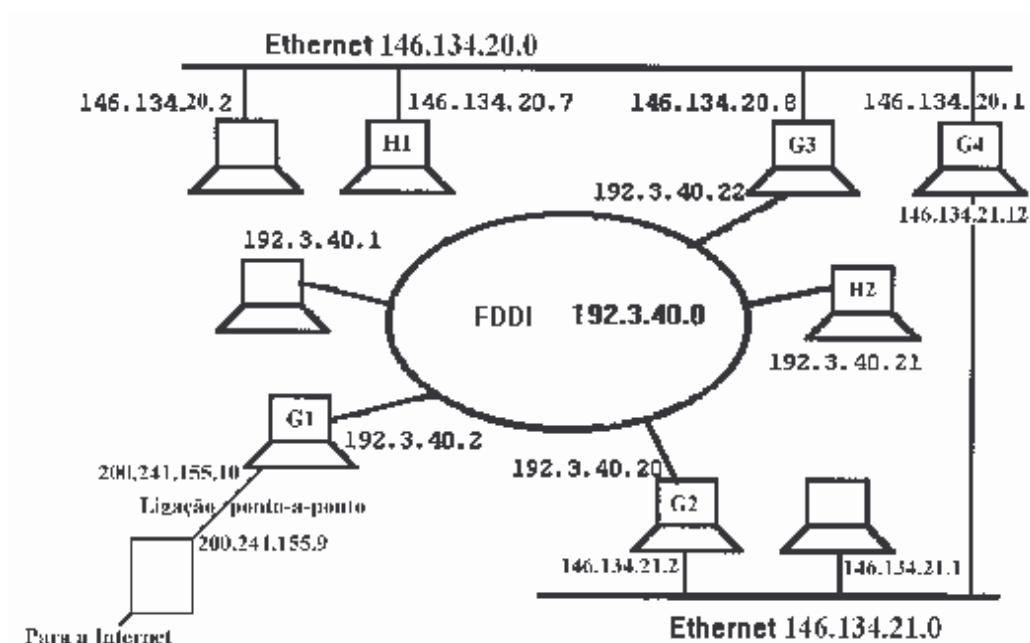


Figura 6.4 – Três redes conectadas entre si e à Internet

## G1

Rede Destino	Máscara	Gateway Destino	Interface	Métrica
192.3.40.0	255.255.255.0	192.3.40.2	192.3.40.2	1
146.134.20.0	255.255.255.0	192.3.40.22	192.3.40.2	1
146.134.21.0	255.255.255.0	192.3.40.20	192.3.40.2	1
0.0.0.0	0.0.0.0	200.241.155.9	200.241.155.10	1

Observe que na tabela de rotas do Gateway G1, quando um pacote for endereçado para a rede FDDI (192.3.40.0) com máscara 255.255.255.0 ele será enviado através do próprio gateway G1 (192.3.40.2), que neste caso também é gateway destino, através de sua interface FDDI (192.3.40.2).

Já para enviar pacotes para a rede Ethernet (146.134.20.0) com máscara 255.255.255.0, os pacotes deverão ser enviados para G3 (192.3.40.22) através da própria Interface com a rede FDDI de G1 (192.3.40.2).

Para enviar dados para a rede Ethernet (146.134.21.0) com máscara 255.255.255.0, os pacotes deverão ser enviados para G2 (192.3.40.20) através da Interface FDDI (192.3.40.2).

E por último, qualquer pacote que não for enviado para nenhuma destas redes deverá ser enviado para a rede default (Rede destino: 0.0.0.0 e máscara: 0.0.0.0), que é o gateway destino 200.241.155.9 através da Interface ponto-a-ponto (200.241.155.10).

O campo métrica é utilizado para identificar, no caso de múltiplas rotas, qual delas será a rota prioritária. No caso da rota prioritária (menor métrica) falhar a próxima rota com menos métrica será utilizada.

## G2

Rede Destino	Máscara	Gateway Destino	Interface	Métrica
192.3.40.0	255.255.255.0	192.3.40.20	192.3.40.20	1
146.134.21.0	255.255.255.0	146.134.21.2	146.134.21.2	1
146.134.20.0	255.255.255.0	192.3.40.22	192.3.40.20	1
146.134.20.0	255.255.255.0	146.134.21.12	146.134.21.2	2
0.0.0.0	0.0.0.0	192.3.40.2	192.3.40.20	1

A configuração de G2 segue o mesmo princípio de G1, ou seja, um dado enviado para a rede FDDI (192.3.40.0) será enviado pelo próprio gateway G1 através de sua interface 192.3.40.20.

Para alcançar a rede Ethernet (146.134.21.0) os dados serão enviado pelo próprio Gateway G2, só que através da interface 146.134.21.2 para esta rede.

Para alcançar a rede Ethernet (146.134.20.0) existem duas possibilidades, uma indo por G3 e outra por G4. Neste caso, definimos que ir por G3 é melhor que ir por G4 (métrica para ir por G3 menor que por G4).

E por último, a nossa *default* é enviada para G1 (192.3.40.2) através da interface FDDI (192.3.40.20)

A configuração de G3 e G4 segue o mesmo princípio da de G2 e G3

## G3

Rede Destino	Máscara	Gateway Destino	Interface	Métrica
146.134.20.0	255.255.255.0	146.134.20.8	146.134.20.8	1
192.3.40.0	255.255.255.0	192.3.40.22	192.3.40.22	1
146.134.21.0	255.255.255.0	192.3.40.20	192.3.40.22	1
146.134.21.0	255.255.255.0	146.134.20.1	146.134.20.8	2
0.0.0.0	0.0.0.0	192.3.40.2	192.3.40.22	1

## G4

Rede Destino	Máscara	Gateway Destino	Interface	Métrica
146.134.20.0	255.255.255.0	146.134.20.1	146.134.20.1	1
146.134.21.0	255.255.255.0	146.134.21.12	146.134.21.12	1
192.3.40.0	255.255.255.0	146.134.20.8	146.134.20.1	1
192.3.40.0	255.255.255.0	146.134.21.2	146.134.21.12	2
0.0.0.0	0.0.0.0	146.134.20.8	146.134.20.1	1
0.0.0.0	0.0.0.0	146.134.21.2	146.134.21.12	2

A observação que se faz na configuração de G4 é que a rota para a rede FDDI (192.3.40.0) não é necessária, haja visto que G3 e G2 já conseguiram alcançar esta rede. Entretanto é interessante coloca-la a fim de mostrar que esta rede esta presente.

## H1

Rede Destino	Máscara	Gateway Destino	Interface	Métrica
146.134.20.0	255.255.255.0	146.134.20.7	146.134.20.7	1
146.134.20.7	255.255.255.255	127.0.0.1	127.0.0.1	1
0.0.0.0	0.0.0.0	146.134.20.8	146.134.20.7	1
0.0.0.0	0.0.0.0	146.134.20.1	146.134.20.7	2

Na configuração de H1, cujo endereço IP é 146.134.20.7 informamos que para a rede 146.134.20.0 (a rede que H1 pertence), será usado a própria máquina como gateway, enviando pela sua interface.

A Segunda linha da tabela de rotas é extremamente importante pois ela esta informando que quando o endereço IP de destino for o próprio endereço da máquina (comunicação inter-processos dentro da mesma máquina) É importante definirmos um endereço de LoopBack (127.0.0.1) com máscara 255.255.255.255 com o objetivo de que a mensagem seja enviada de um processo para outro sem chegar no nível de rede.

Caso esta linha seja omitida, um dado para a mesma máquina será enviado para a rede, de acordo com a primeira linha e voltará para a própria máquina, causando um tráfego desnecessário na rede.

Observe ainda que existem duas possíveis rotas *default* com diferentes métricas, para o caso de uma delas falhar.

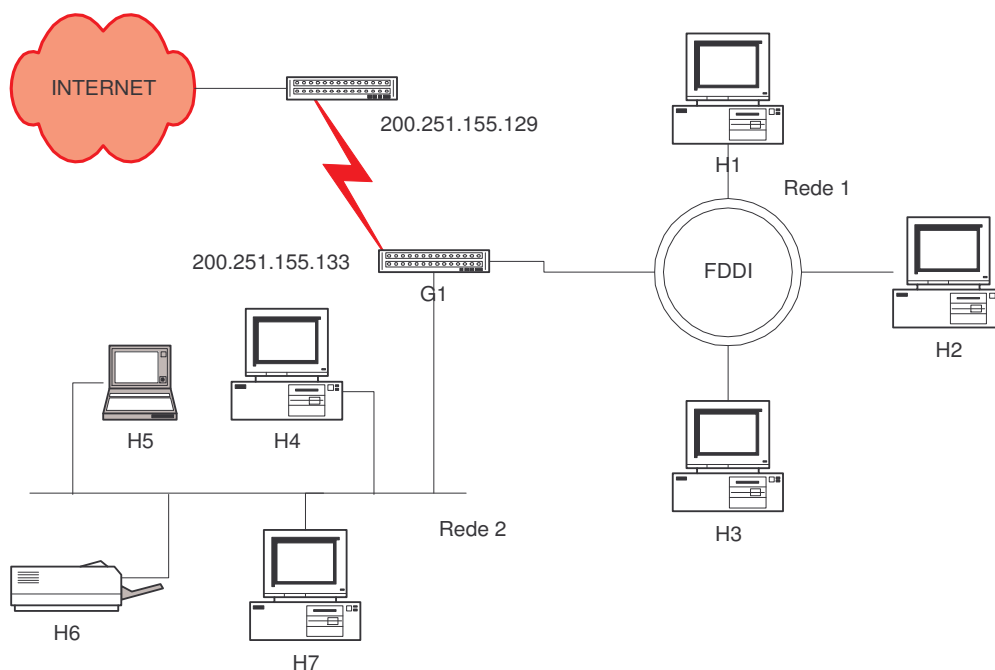
## H2

Rede Destino	Máscara	Gateway Destino	Interface	Métrica
192.3.40.0	255.255.255.0	192.3.40.21	192.3.40.21	1
192.3.40.21	255.255.255.255	127.0.0.1	127.0.0.1	1
0.0.0.0	0.0.0.0	192.3.40.2	192.3.40.21	1
146.134.20.0	255.255.255.0	192.3.40.22	192.3.40.21	1
146.134.21.0	255.255.255.0	192.3.40.20	192.3.40.21	1

A configuração de H2 é bastante parecida com a que H1. Segue o mesmo princípio. As duas últimas rotas são opcionais.

## 6.5.2 Exemplo 2

Você dispõe de apenas um endereço classe C (200.241.14.x) para as duas redes (Figura 6.5). Divida a sua rede em sub-redes, identificando a faixa de endereços IP para cada sub-rede e faça a tabela de rotas para G1, H1, H2, H3, H4, H5, H6 e H7.



**Figura 6.5 – Topologia da Rede**

A máscara da rede será **255.255.255.192**

Teremos a rede **200.241.14.64** com endereços IP entre **200.241.14.65** e **200.241.14.126** e a rede **200.241.14.128** com endereço IP variando de **200.241.14.129** à **200.241.14.190**

Vamos definir também que:

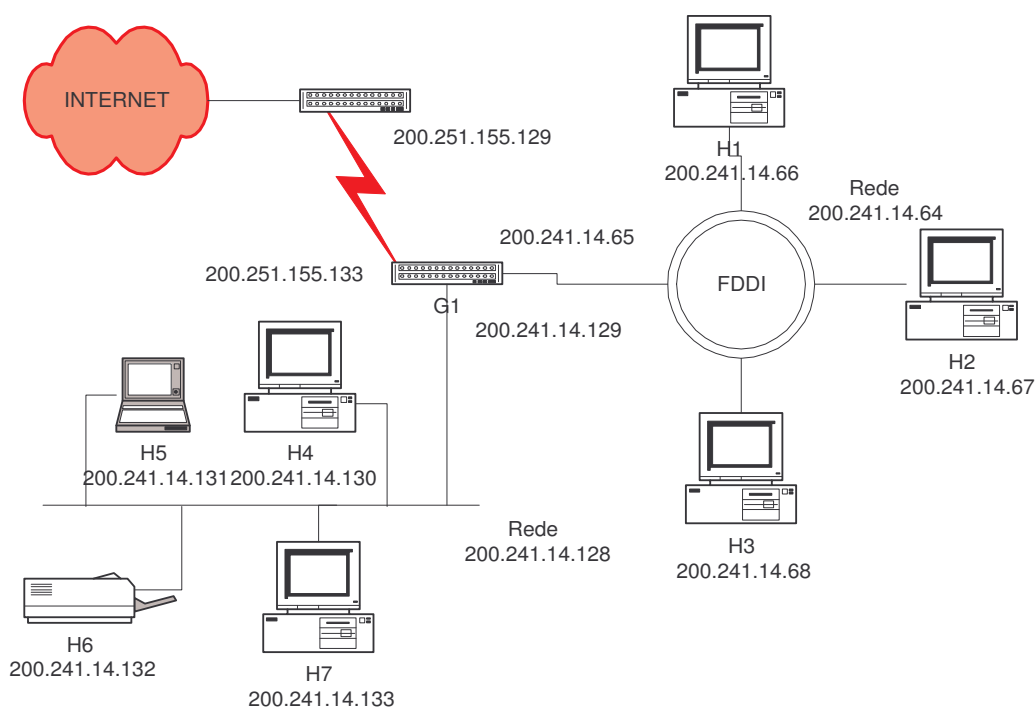
Máquina	Endereço IP
G1 (Interface FDDI)	200.241.14.65
G1 (Interface Ethernet)	200.241.14.129
H1	200.241.14.66
H2	200.241.14.67
H3	200.241.14.68
H4	200.241.14.130
H5	200.241.14.131
H6 (Impressora)	200.241.14.132
H7	200.241.14.133



Observe que esta definição pode diferir de uma outra. A única coisa que importa é os endereços IP estarem dentro da faixa especificada.

O roteador terá dois endereços IP por causa das suas duas interfaces, uma com a rede FDDI e a outra com a rede Ethernet

É interessante ressaltar também que a impressora (H6) também possui um endereço IP na rede. Esta solução é utilizada em algumas redes e em outras não. Depende da situação. A Figura 6.6 ilustra como ficará a topologia da rede



**Figura 6.6 – Topologia da rede com os endereços IP**

# G1

Rede Destino	Máscara	Gateway Destino	Interface	Métrica
200.241.14.64	255.255.255.192	200.241.14.65	200.241.14.65	1
200.241.14.128	255.255.255.192	200.241.14.129	200.241.14.129	1
0.0.0.0	0.0.0.0	200.251.155.129	200.251.155.133	1

# H1

Rede Destino	Máscara	Gateway Destino	Interface	Métrica
200.241.14.64	255.255.255.192	200.241.14.66	200.241.14.66	1
200.241.14.66	255.255.255.255	127.0.0.1	127.0.0.1	1
0.0.0.0	0.0.0.0	200.241.14.65	200.241.14.66	1

# H2

Rede Destino	Máscara	Gateway Destino	Interface	Métrica
200.241.14.64	255.255.255.192	200.241.14.67	200.241.14.67	1
200.241.14.67	255.255.255.255	127.0.0.1	127.0.0.1	1
0.0.0.0	0.0.0.0	200.241.14.65	200.241.14.67	1

# H3

Rede Destino	Máscara	Gateway Destino	Interface	Métrica
200.241.14.64	255.255.255.192	200.241.14.68	200.241.14.68	1
200.241.14.68	255.255.255.255	127.0.0.1	127.0.0.1	1
0.0.0.0	0.0.0.0	200.241.14.65	200.241.14.68	1

# H4

Rede Destino	Máscara	Gateway Destino	Interface	Métrica
200.241.14.128	255.255.255.192	200.241.14.130	200.241.14.130	1
200.241.14.130	255.255.255.255	127.0.0.1	127.0.0.1	1
0.0.0.0	0.0.0.0	200.241.14.129	200.241.14.130	1

# H5

Rede Destino	Máscara	Gateway Destino	Interface	Métrica
200.241.14.128	255.255.255.192	200.241.14.131	200.241.14.131	1
200.241.14.131	255.255.255.255	127.0.0.1	127.0.0.1	1
0.0.0.0	0.0.0.0	200.241.14.129	200.241.14.131	1

## H6

Rede Destino	Máscara	Gateway Destino	Interface	Métrica
200.241.14.128	255.255.255.192	200.241.14.132	200.241.14.132	1
200.241.14.132	255.255.255.255	127.0.0.1	127.0.0.1	1
0.0.0.0	0.0.0.0	200.241.14.129	200.241.14.132	1

## H7

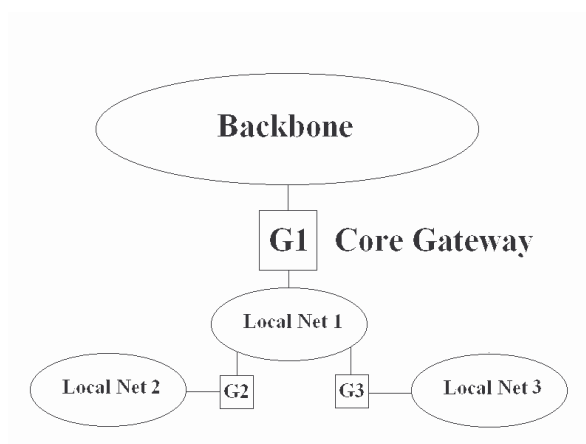
Rede Destino	Máscara	Gateway Destino	Interface	Métrica
200.241.14.128	255.255.255.192	200.241.14.133	200.241.14.133	1
200.241.14.133	255.255.255.255	127.0.0.1	127.0.0.1	1
0.0.0.0	0.0.0.0	200.241.14.129	200.241.14.133	1

Obs.: Em máquinas Windows e UNIX é possível ver a tabela de rotas através do comando **netstat -rn**. O aplicativo **route** também pode ser utilizado para imprimir as rotas (**route print**) e para adicionar uma rota à tabela de rotas (**route add**)

## 7 IGP – Interior Gateway Protocol

Vimos no capítulo 6 alguns esquemas de roteamento bastante simples, com um roteador ligando somente duas redes, tendo sempre uma rota *default* bem definida. Não tínhamos também a preocupação em reconfigurar o sistema no caso de um roteador esta conectando mais de uma rede e um determinado *link* parar de funcionar.

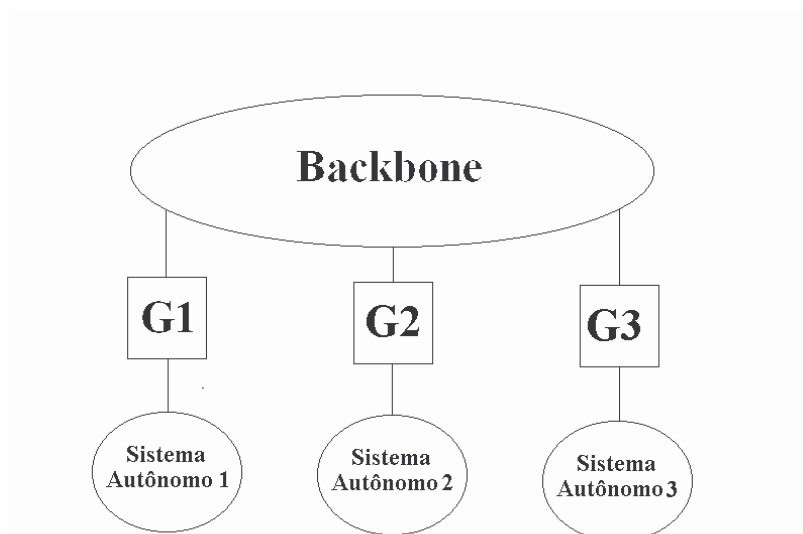
Neste e nos próximos capítulos trataremos do problema de roteamento de forma mais complexa e completa. Neste capítulo estudaremos os **Sistemas Autônomos**. Estes sistemas são caracterizados pela presença de uma entidade centralizadora responsável por uma determinada região (um conjunto de redes interligadas por gateways). A Figura 7.1 ilustra este conceito.



**Figura 7.1 – Sistema Autônomo**

Observe que temos três redes locais interligadas pelos gateways G2 e G3. Neste caso o gateway G1 é que interliga estas redes ao Backbone. A este gateway (G1), damos o nome de **Core Gateway**, sendo ele o responsável pelo sistema autônomo.

A Figura 7.2 mostra a interligação de três sistemas autônomos pelos gateways G1, G2 e G3 a um Backbone. Cada um destes gateways é responsável pelo seu próprio sistema autônomo.

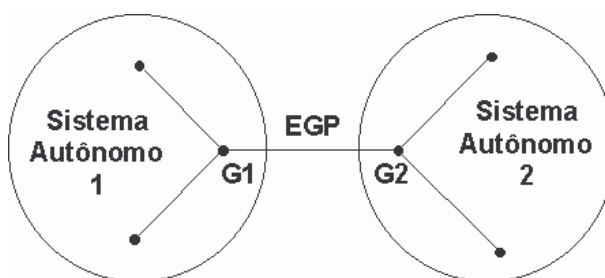


**Figura 7.2 – Interligação de três sistemas autônomos ao Backbone**

Cada sistema autônomo tem a liberdade de escolher o protocolo que melhor lhe convier para manter, descobrir e divulgar as rotas dentro do seu universo.

Os Gateways responsáveis pelo sistema autônomo tem a incumbência de manter coerentes as tabelas de rotas de seus gateways internos, mesmo na presença de erros em alguns links, bem procurar definir a menor rota para que um determinado pacote alcance o seu alvo destino.

Gateways que trocam informações de roteamento com outros gateways que não pertencem ao mesmo Sistema Autônomo, são considerados "Vizinhos Exteriores" e utilizam o protocolo **EGP** (Exterior Gateway Protocol) para se comunicarem. A Figura 7.3 ilustra este conceito.



**Figura 7.3 – Troca de informações entre sistemas autônomos**

Gateways que trocam informações de roteamento somente com gateways do mesmo Sistema Autônomo são considerados "Vizinhos Interiores" e utilizam diversos protocolos denominados genericamente **IGP** (*Interior Gateway Protocols*). Entre eles encontram-se o

RIP (Routing Information Protocol), o HELLO, o OSPF (Open Shortest Path First), o IGRP (Internal Gateway Routing Protocol), entre outros. Neste capítulo veremos os protocolos IGP e no próximo capítulo os protocolos EGP.

## 7.1 RIP - Routing Information Protocol (RFC 1058)

Um dos protocolos IGP mais difundidos, conhecido também como *Routed* é o RIP – Routing Information Protocol (Protocolo com informações de roteamento), este protocolo foi desenvolvido pela Universidade de Berkeley na Califórnia, e sua grande popularidade deve-se ao fato dele ser distribuído junto com o 4BSD UNIX.

O RIP utiliza a técnica "Vector Distance" para atualização de tabelas de roteamento. Em seu método de atuação ele particiona as máquinas envolvidas em **ativas** e **passivas** (ou silenciosas). Máquinas ativas informam suas rotas para outros, já as máquinas passivas escutam e atualizam suas rotas baseadas nas informações transmitidas pelas máquinas ativas, mas não informam. Tipicamente, gateways rodam RIP em modo ativo, enquanto hosts usam modo passivo.

Um gateway utilizando RIP envia mensagens de atualização de rotas em *broadcasting* a cada 30 segundos, mencionando as redes e suas respectivas distâncias (em *hops*). Os hosts, assim que recebem mensagens RIP, efetuam a atualização de suas tabelas de rotas.

Na métrica do RIP, um host é dito com *hop*=1 quando é diretamente conectado e *hop*=2 quando é alcançado através de um gateway.

Nem sempre o menor número de saltos (*hops*) significa a melhor rota, pode ser que uma rota mais longa propicie melhor qualidade de linhas. Para compensar esta diferença de tecnologia algumas implementações usam alta contagem artificial de *hops* quando informados de conexões lentas.

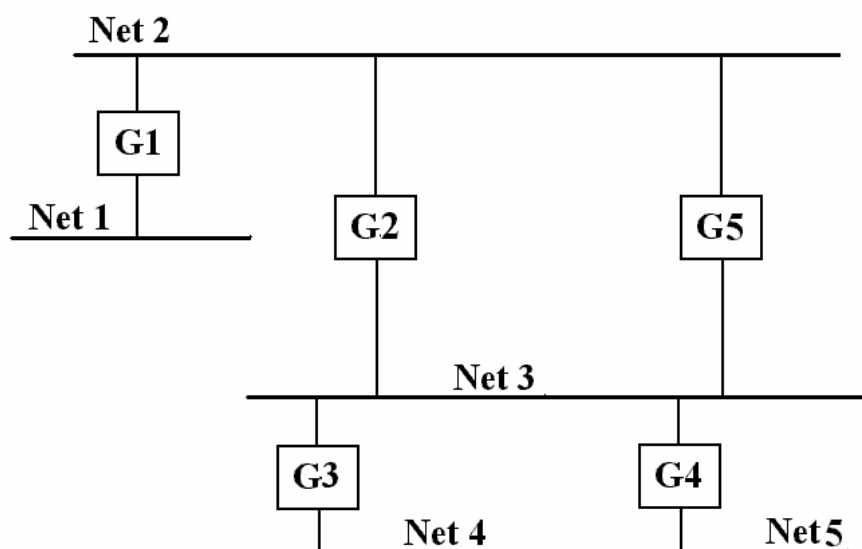
Devido à morosidade da divulgação de rotas, o RIP implementa um número máximo de *hops* igual a 16. Quando uma rota é divulgada com valor de **16 hops**, esta é considerada inatingível. A tabela de roteamento do RIP pode ser estática ou dinâmica.

Nas máquinas UNIX o comando **routed** aciona o *daemon* responsável pelo roteamento. Este *daemon* deve ser executado quando a máquina é ligada. O arquivo **/etc/gateways** indica a rota para uma determinada rede e a métrica para ela. Já o comando **netstat -r**, mostra a tabela de roteamento e a sua respectiva métrica. O RIP utiliza a porta UDP 520 para transmissão dos dados de roteamento.

### 7.1.1 Problemas do Protocolo RIP

O RIP especifica poucas regras para melhorar a performance e confiabilidade. Por exemplo, quando um gateway aprende uma rota de um outro gateway, ele deve manter esta rota até que aprenda uma melhor. Se dois gateways anunciam uma rota com o mesmo custo, será

usada aquela que for anunciada primeiro. A Figura 7.4 ilustra a interligação de cinco redes (Net1 à Net5) através de cinco Gateways (G1 à G5).



**Figura 7.4 – Interligação de redes**

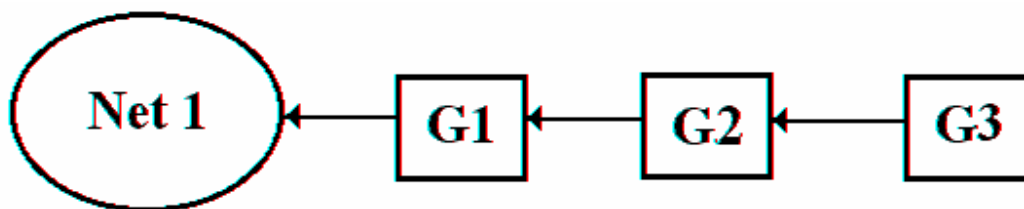
Na Figura 7.4, suponha que G4 aprendeu que a rota para a rede 1 (Net1) é pelo Gateway G2. Caso o gateway G2 falhe, G4 manteria esta informação “errada” por um determinado tempo, não conseguindo rotear os pacotes para Net1. Observe entretanto que G4 poderia mandar os pacotes através do gateway G5. Esta rota sempre existiu, mas G4 simplesmente não enxergava.

Para evitar este tipo de problema, o RIP estabelece que todos os ouvintes devem colocar um temporizador (*time-out*) para cada rota aprendida. O temporizador é reinicializado sempre que o gateway receber um novo anúncio para aquela rota. Caso este tempo passe de 180 segundos a rota é eliminada.

No caso acima, a partir do momento que o gateway G2 para de responder, levará pelos menos 180 segundos para que a rota em G4 seja excluída. Após este momento levará algum tempo ainda para que G4 aprenda a rota por G5. Observe que o tempo para aprendizado de rotas é muito grande no RIP.

### 7.1.2 Convergência Lenta do RIP

A Figura 7.5 ilustra a interligação de uma rede (Net1) ao gateway G1, que por sua vez esta conectado ao gateway G2 que esta conectado a G3.

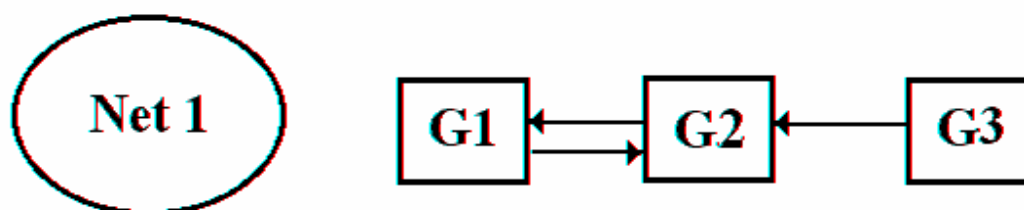


**Figura 7.5 – Interligação de gateways a uma rede**

Na Figura 7.5, o gateway G1 tem conexão direta a rede 1 (Net1), por isso anuncia esta rota com distância **1** (hop=1). O gateway G2 aprende a rota para a Rede 1 (Net1), através de G1 e coloca a distância para **2** (hop=2). Da mesma forma G3 coloca distância para **3** (hop=3).

Se a conexão de G1 a Rede 1 cair, G1 coloca que a distância a esta rede é 16, ou seja, inatingível. Como isso, na próxima vez que G1 divulgar a rota para a Rede 1, os gateways que já receberam esta informação de G1 anteriormente atualizarão a métrica para 16, fazendo a rota para a Rede 1 inatingível e distribuindo esta informação ao longo de todos os outros gateways.

Suponha entretanto que G1 receba a tabela de rotas de G2 **antes** de divulgar a sua tabela de rotas com a métrica para Rede 1 marcada com **hop=16**. G1 recebe a informação de G2 e compara com a sua e vê que G2 tem um caminho mais curto para a rede 1. Isto não deveria acontecer pois a rede 1 está inatingível. Esta falha vai causar um *loop* entre G1 e G2 como ilustra a Figura 7.6



**Figura 7.6 – Loop entre G1 e G2**

G1 colocará em sua tabela de rotas que alcança a rede 1 com distância 3 via G2 (G2 havia informado que alcançava a custo 2). Na próxima atualização, G2 recebe a informação de G1 de que a rede 1 é alcançável a distância 3. Como ele tem na sua tabela que a rede 1 é alcançável por G1 com distância 2, ele atualiza esta distância para 4. Este incremento de *hop count* acontecerá **lentamente** até que chegue a 16, o que define a inalcançabilidade da rede 1. Veremos agora três métodos para resolver este problema.



### 7.1.2.1 Método Split Horizon

No uso da técnica de *Split Horizon* um gateway recorda a interface sobre a qual ele recebeu uma rota particular e não propaga esta informação a respeito da rota anterior sobre a mesma interface. No exemplo anterior, G2 não informaria a G1 que tem uma rota para Rede 1, pois esta informação foi obtida exatamente de G1.

### 7.1.2.2 Método Hold Down

Se um gateway informa uma rota curta de acesso a uma rede, todas os gateways recebem esta informação e rapidamente atualizam suas tabelas de rotas. Agora se um gateway para de informar uma rota o protocolo depende de um mecanismo de *time-out* antes de considerar a rota inalcançável. Uma vez que o *time-out* ocorre, o gateway pode encontrar uma rota alternativa, atualizar sua tabela e começar a propagar esta informação.

Infelizmente, um gateway não pode saber se a rota alternativa depende da rota que esta interrompida (por exemplo, se foi propagação de um outro gateway que ainda não estourou o *time-out*). Esta negativa de informação não se propaga rapidamente.

Na técnica de *Hold Down* força-se um gateway participativo a ignorar a informação sobre uma rede por um período fixo de tempo (tipicamente 60 segundos) após o recebimento de uma mensagem que informa que a rede está inalcançável ou após o estouro do *time-out*.

O problema do Hold Down é que ele preserva rotas incorretas durante seu tempo de duração.

### 7.1.2.3 Método Poison Reverse

Quando uma conexão é removida, o gateway responsável pela propagação desta rota retém as entradas por vários períodos de atualização, e inclui um custo infinito no seu broadcast. Esta técnica é chamada de Poison Reverse.

Para o Poison Reverse ser mais eficiente ele deve ser combinado com "triggers updates". Triggered Updates força um gateway a enviar uma mensagem de broadcast imediatamente após receber uma notícia de falha de conexão, ao invés de esperar o broadcast periódico.

## 7.2 O Protocolo Hello

O protocolo HELLO é muito parecido com o RIP. A diferença básica é que o RIP usa a métrica de *hops* e o Hello usa a métrica de *delay* (atraso) entre uma rede e outra. As funções básicas são:

- manter o sincronismo dos relógios de todos os gateways envolvidos
- divulgar alcance por tempo referente a cada rede.

Cada máquina participante do Hello mantém uma tabela com a estimativa do relógio em cada um de seus vizinhos. Antes de transmitir um pacote Hello, a máquina coloca neste pacote o seu relógio. Quando o pacote chega, a máquina receptora computa o **delay** no enlace baseado na estimativa do relógio da máquina que enviou. Periodicamente as máquinas atualizam seus vizinhos para reestabelecer as estimativas de relógios destes.

### 7.3 OSPF - Open Shortest Path First (RFC 1131)

O protocolo OSPF foi desenvolvido pelo grupo de trabalho IETF (Internet Engineering Task Force) e é baseado na tecnologia **link state** para manter as tabelas de rotas.

O OSPF também provê autenticação das mensagens de atualização de roteamento. O OSPF roteia pacotes IP baseando-se no endereço de destino e no **TOS** (*Type of Service*), ambos pertencentes ao protocolo IP.

O protocolo OSPF detecta rapidamente mudanças no sistema autônomo, como falhas na interface de roteamento, e calcula novas rotas, livres de loops, após o período de convergência. Este período de convergência é pequeno e envolve o mínimo de tráfego de roteamento.

O OSPF calcula separadamente rotas para cada “tipo de serviço” (TOS - *type of service*). Quando várias rotas de custos iguais existem para um mesmo destino, o tráfego é distribuído igualmente sobre elas, ou seja, existe **balanceamento de carga**. O custo de uma rota é descrito por uma métrica.

O OSPF permite que um conjunto de redes sejam agrupados. A este agrupamento damos o nome de **Área**. A topologia de uma Área não é vista pelo resto do Sistema Autônomo. Esta informação oculta permite uma redução significativa no tráfego de roteamento. Da mesma forma, o roteamento de uma Área é determinado apenas pela topologia da mesma, dando proteção a Área de dados de roteamentos errados.

O protocolo OSPF permite uma configuração flexível de subredes IP. Cada rota distribuída pelo OSPF possui um destino e uma máscara. Duas subredes diferentes em um mesmo número IP de rede podem ter diferentes tamanhos (máscaras). Isto é comumente referenciado como tamanho variável de subredes. Um pacote é roteado para melhor combinação. Esta característica é uma grande vantagem em relação ao RIP, que não trata bem o conceito de máscaras.

**Toda troca do protocolo OSPF é autenticada.** Isto significa que apenas gateways confiáveis podem participar do roteamento de um sistema autônomo. Uma variedade de esquemas de autenticação pode ser usado. Um esquema simples de autenticação é configurado para cada Área. Isto permite que algumas Áreas usem autenticação mais restrita que outras. A autenticação é uma característica importante pois garante que um pacote não vai ser desviado para um destino que não o correto.

Em um protocolo de roteamento baseado no SPF (*shortest path first*), cada roteador mantém uma base de dados descrevendo a topologia do Sistema Autônomo. Cada roteador participante possui uma base idêntica. Cada parte individual desta base de dados é um estado particular do roteador local (a interface usável do roteador e vizinhos alcançáveis).

Todos roteadores executam o mesmo algoritmo em paralelo. Sobre a base de dados topologica, cada roteador constrói uma árvore dos menores caminhos alcançáveis, com ele próprio de raiz. Esta árvore mostra a rota para cada destino de um sistema autônomo. Informações para roteamento externo aparecem como folhas na árvore. Este esquema reduz drasticamente a comunicação entre os *gateways*, além de permitir uma tomada de decisão muito mais rápida e eficiente a respeito do caminho a ser escolhido.

### 7.3.1 Formato das Mensagens

Descreveremos nesta sessão o formato básico das mensagens que são trocadas através do protocolo OSPF. A Figura 7.7 mostra o cabeçalho padrão de uma mensagem OSPF.

VERSION(1)	TYPE	MESSAGE LENGTH
SOURCE GATEWAY IP ADDRESS		
AREA ID		
CHECKSUM	AUTHENTICATION TYPE	
AUTHENTICATION (octets 0-3)		
AUTHENTICATION (octets 4-7)		

Figura 7.7 – Cabeçalho padrão mensagem OSPF

A Tabela 7.1 descreve os campos do cabeçalho padrão.

Nome do Campo	Descrição
VERSION	Especifica a versão do protocolo OSPF
TYPE	Tipo de Mensagem. As possíveis mensagens são: <b>1 – Hello</b> (usado para testar alcance) <b>2 – Database Description</b> (Topologia) <b>3 – Link Status Request</b> (Requisição de status do link) <b>4 – Link Status Update</b> (Atualização do status do link) <b>5 – Link Status Acknowledgement</b> (Reconhecimento do status do link)
MESSAGE LENGHT	Tamanho da mensagem OSPF
SOURCE GATEWAY IP ADDRESS	Utilizado para identificação do gateway gerador da mensagem OSPF, onde é introduzido o seu respectivo

Nome do Campo	Descrição
	endereço IP
CHECKSUM	Utilizado na verificação de erros da mensagem OSPF
AUTHENTICATION TYPE	Utilizado para garantir a idoneidade das informações de roteamento trocadas. Quando "0", não será utilizada a autenticação. Quando "1", será utilizada uma "senha" de autenticação
AUTHENTICATION	Senha de validação, caso AUTHENTICATION TYPE = 1

Tabela 7.1 – Descrição dos campos da mensagem OSPF

Veremos a seguir quais são as mensagens trocadas para cada tipo de serviço. O tipo **Hello** não será apresentado por ser tratar de um tipo que só trata se a máquina esta alcançável ou não.

### 7.3.1.1 Database Description

O OSPF trabalha com o conceito de **Gateway Mestre** e **Gateway Escravo**. O Gateway Mestre é responsável por repassar aos Gateways Escravos a topologia e as alterações da rede. Já os Gateways escravos somente se alimentam dessas informações. A Figura 7.8 mostra a mensagem padrão enviada do Gateway Mestre para os Escravos.

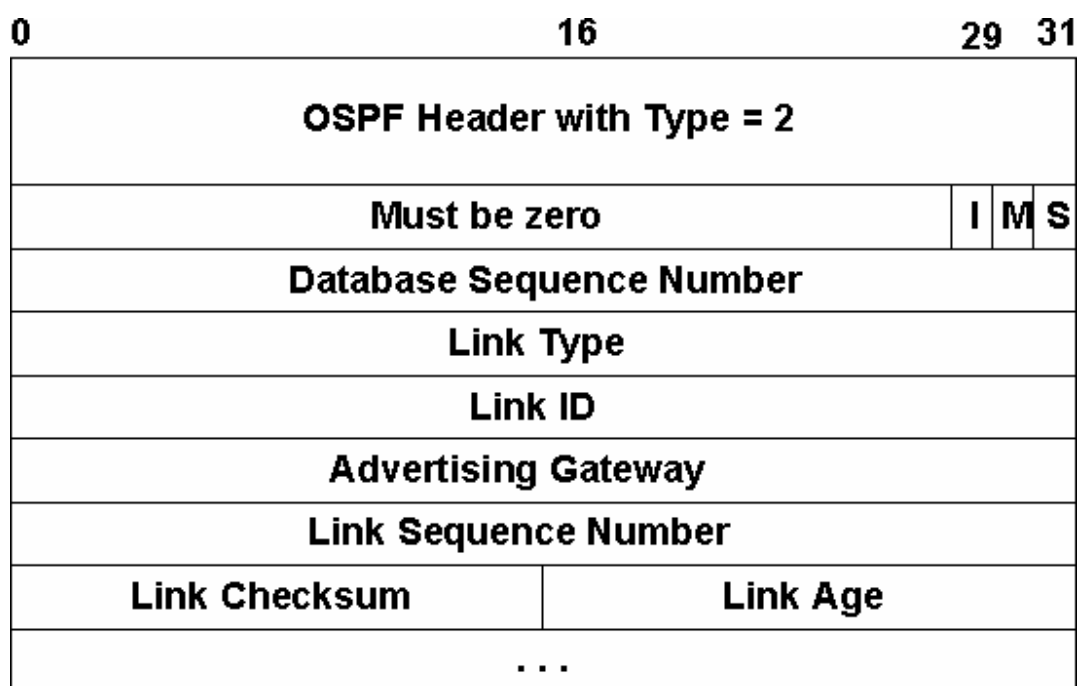


Figura 7.8 – Formato da Mensagem enviada pelo Gateway Mestre aos Escravos

A Tabela 7.2 descreve os campos da Figura 7.8

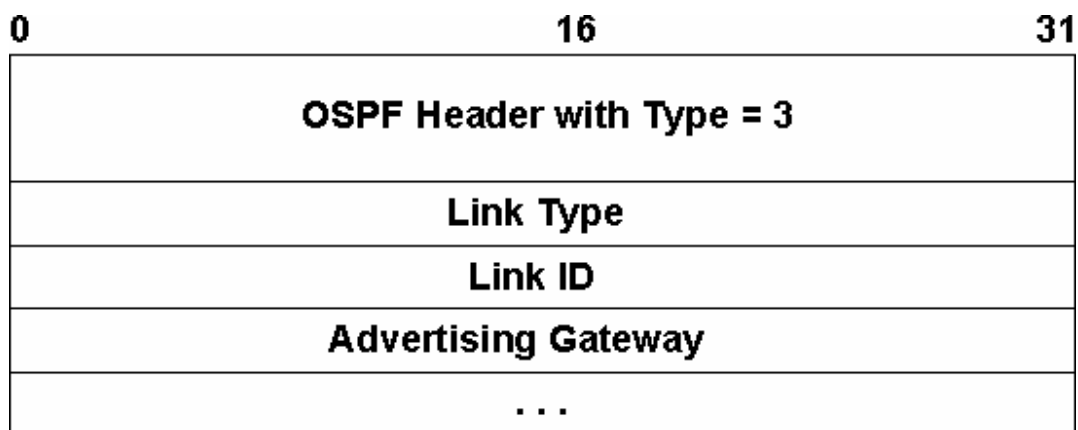
Nome do Campo	Descrição
I	Controle de fragmentação. Quando "0" identifica segmentos posteriores de uma mensagem. Se "1", indica o fragmento inicial
M	Se "0" indica que a mensagem não foi segmentada ou que este é o último fragmento. Se "1" indica que a mensagem foi fragmentada e mais fragmentos serão enviados
S	Se "0" indica que a mensagem foi enviada por um gateway escravo, e se "1", foi enviada por um gateway mestre.
DATABASE SEQUENCE NUMBER	Utilizado para manter o sincronismo das mensagens enviadas, onde um número é gerado inicialmente e acrescido a cada mensagem enviada.
LINK TYPE	Descreve o tipo do link mencionado
LINK ID	Identifica o endereço IP de um gateway ou de uma rede
ADVERTISING GATEWAY	Endereço IP do gateway que divulga o link
LINK SEQUENCE NUMBER	Cada link divulgado na <i>database</i> tem seu próprio número de sequência, assegurando que as mensagens não se percam ou cheguem fora de ordem
LINK CHECKSUM	Visa garantir que os dados referentes a cada link não sejam corrompidos
LINK AGE	Utilizado para medir a permanência de cada link em estado ativo

**Tabela 7.2 – Descrição dos campos Database Description**

Os campos de **LINK TYPE** até **LINK AGE** são repetidos na mensagem para cada link especificado.

### 7.3.1.2 Link Status Request

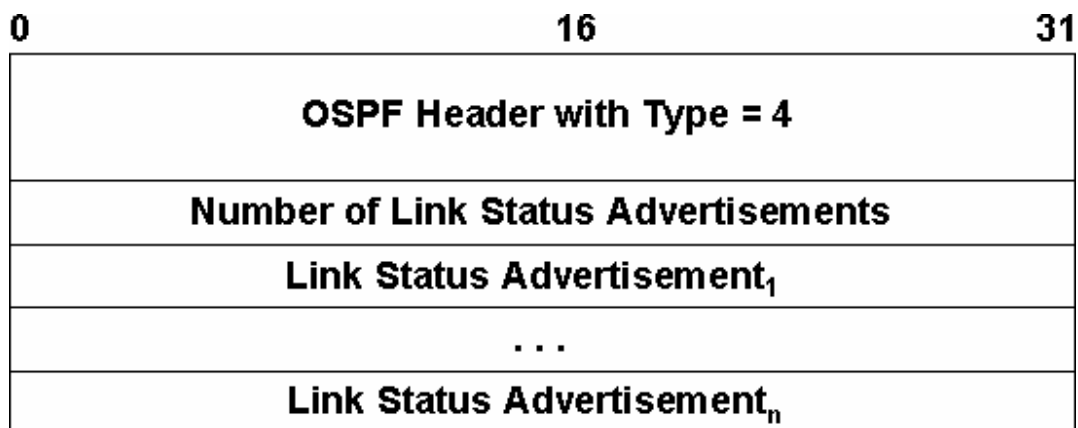
Após a troca de mensagens Tipo 2 (Database Description), os gateways podem descobrir que existem partes de seus bancos de dados desatualizadas, para atualizar suas bases de dados, um gateway envia uma mensagem de requisição de estado ao seu vizinho, especificando quais os links que o emissor deseja atualizar. O receptor envia uma resposta o mais atualizado possível. A Figura 7.9 especifica o cabeçalho do **Link Status Request**. Os campos Link Type, Link ID e Advertising Gateway são os mesmos do **Database Description**.



**Figura 7.9 – Cabeçalho de uma mensagem Link Status Request**

### 7.3.1.3 Link Status Update

As mensagens de **Link Status Update**, como o próprio nome sugere atualizam a situação de um determinado Link. Esta mensagem é enviada por um gateway escravo ao gateway mestre para que este atualize sua tabela de status daquele(s) link(s) e repasse esta informação para os demais gateways escravos. Cada Update consiste de uma lista de anúncios, como mostra a Figura 7.10



**Figura 7.10 – Mensagem de atualização de um link**

Na Figura 7.10 o campo **Number of Link Status Advertisements** indica quantos links estão sendo enviados para atualização, já o campo **Link Status Advertisement** (Situação do Link divulgado) possui o formato da Figura 7.11

<b>0</b>	<b>16</b>	<b>31</b>
<b>Link Age</b>	<b>Link Type</b>	
<b>Link ID</b>		
<b>Advertising Gateway</b>		
<b>Link Sequence Number</b>		
<b>Link Checksum</b>	<b>Lenght</b>	

Figura 7.11 – Status do Link

Esta mensagem possui vários campos de controle para identificar o link (Link Age, Link Type e Link ID), com o nome do Link (Advertising Gateway) e um número de sequência do Link (Link Sequence Number), bem como os campos de controle de erro (Link Checksum) e tamanho (Lenght). É através do campo Link Type que o receptor consegue identificar quais links estão dentro do mesmo “site” ou são externos a ele.

#### 7.3.1.4 Link Status Acknowledgement

Todo link divulgado ou atualizado precisa de uma confirmação de recebimento desta informação (lembre-se que a divulgação é através do protocolo UDP). Esta confirmação é feita através da mensagem **Link Status Acknowledgement** (Reconhecimento da situação do link). A Figura 7.12 ilustra a mensagem enviada.

<b>0</b>	<b>16</b>	<b>31</b>
<b>OSPF Header with Type = 5</b>		
<b>Link Age</b>	<b>Link Type</b>	
<b>Advertising Gateway</b>		
<b>Link ID</b>		
<b>Link Sequence Number</b>		
<b>...</b>		

Figura 7.12 – Confirmação da situação do link

Os campos são os mesmo da Figura 7.11. É através do **Link Sequence Number** (número de sequência do link) que receptor e transmissor fazem o controle entre mensagens enviadas e confirmações das mesmas.

### 7.3.2 Exemplo de Funcionamento OSPF

A Figura 7.13 mostra um mapa simples de um Sistema Autônomo. A nomenclatura utilizada é  $H_i$  para Host,  $N_i$  para Network (Rede) e  $RT_i$  para Router (Roteador ou Gateway). Desta forma, o retângulo H1 indica um host, que possui uma conexão SLIP com o roteador RT12. Linhas entre os roteadores indicam uma rede física ponto a ponto entre eles. Os roteadores RT5 e RT7 possuem conexões EGP para outros Sistemas Autônomos. Um conjunto de rotas aprendidas pelos roteadores EGP, são informadas para estes roteadores.

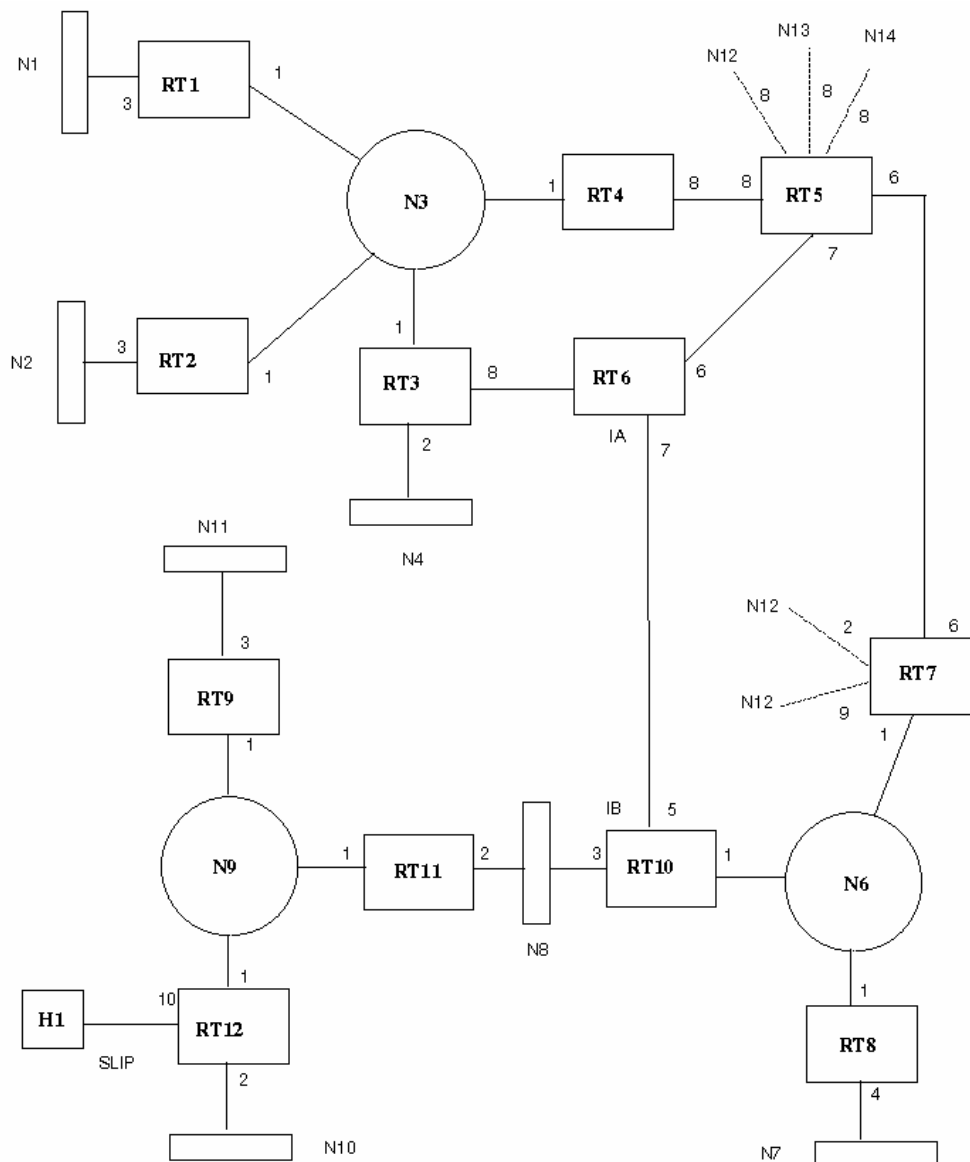


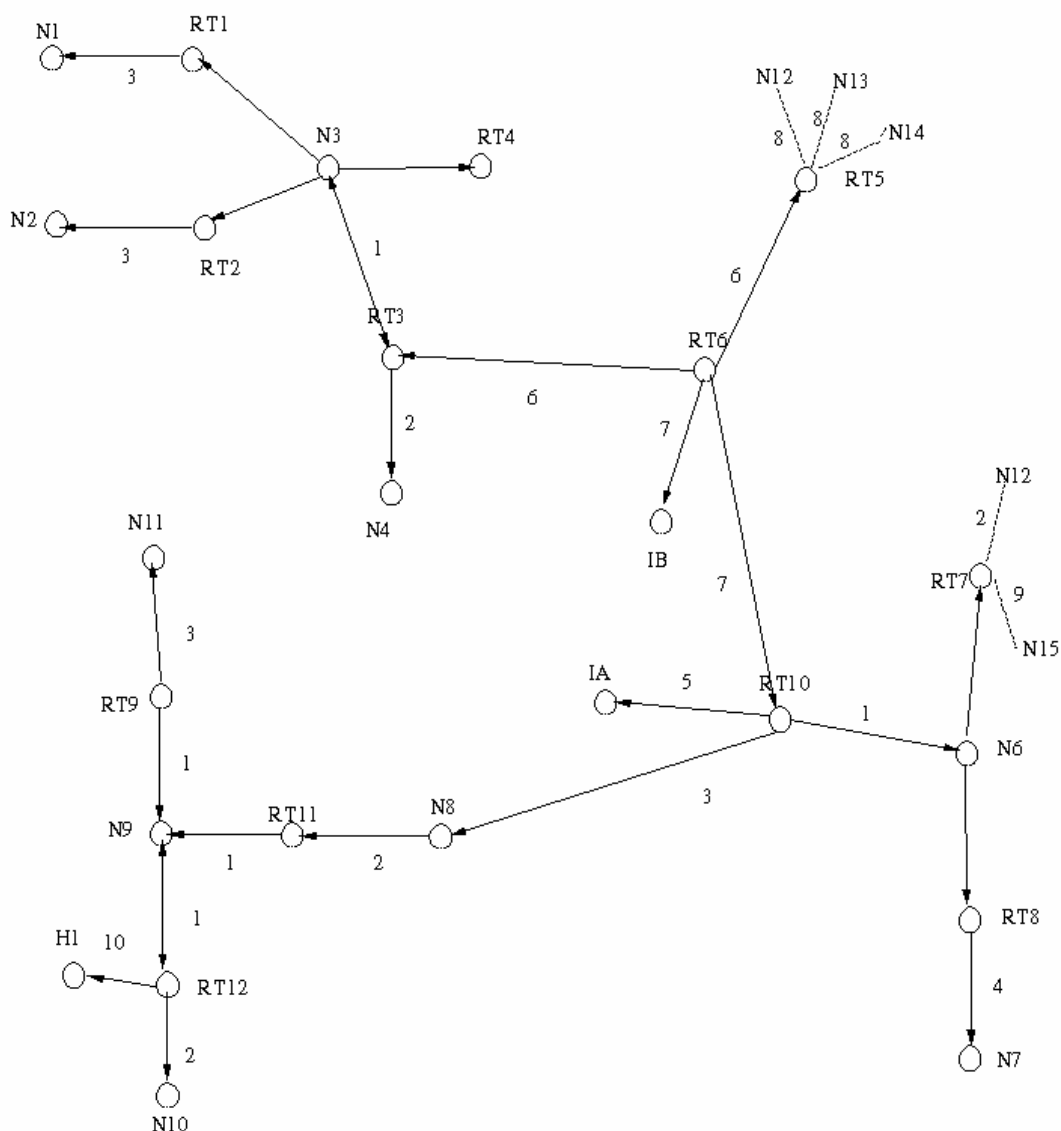
Figura 7.13 – Mapa de um Sistema Autônomo



A cada saída de *interface* do roteador é associado um custo. Este custo é configurado pelo administrador do sistema. Além da configuração manual os custos são também associados com dados de roteamento externo (rotas EGP).

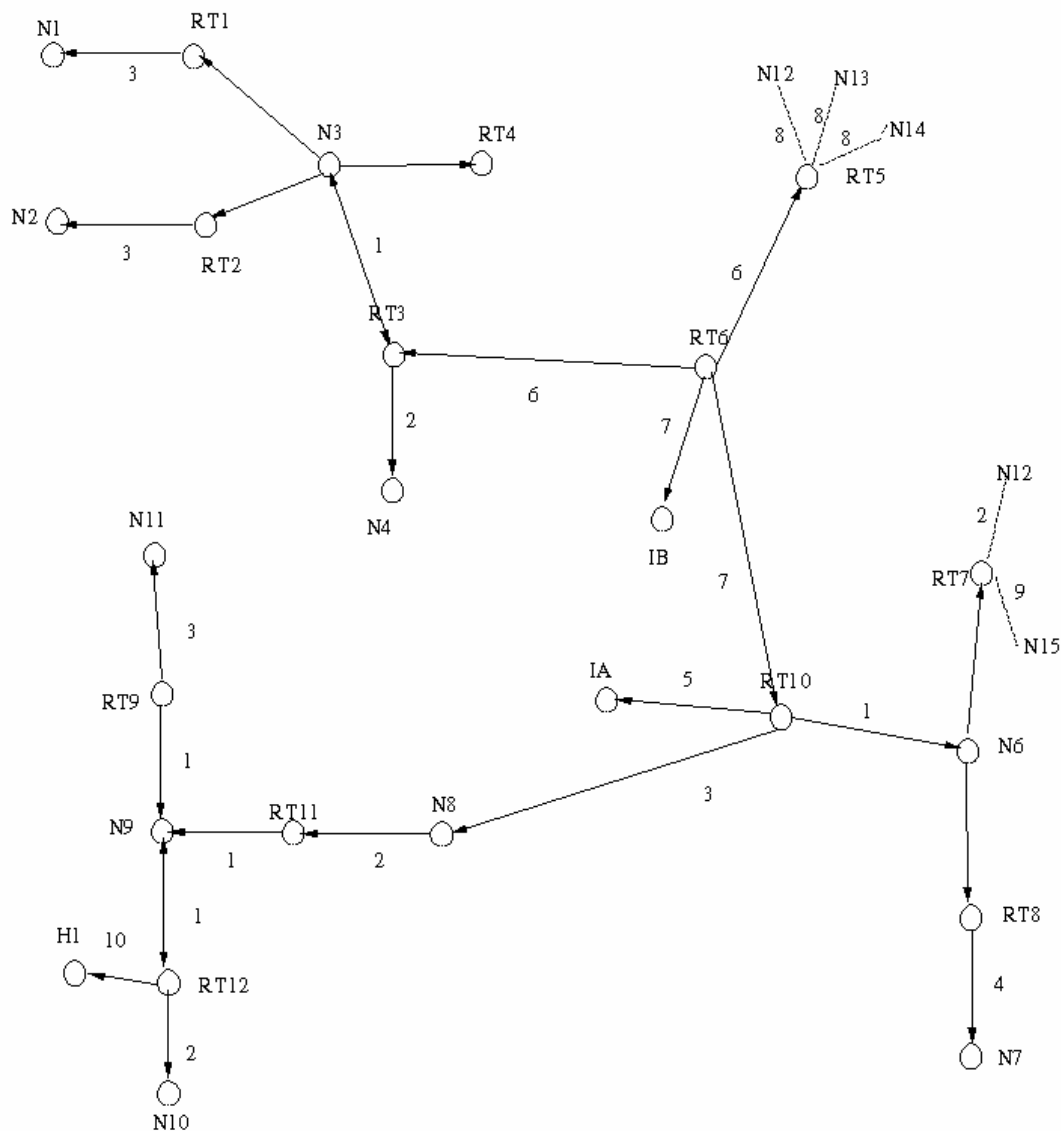
A partir da Figura 7.13, e os seus respectivos custos é possível especificar o **grafo** da Figura 7.14. Observe que cada reta indica o custo associado a ela. Para as retas que não possuem custo associado é entendido que o seu custo é igual a zero.

O grafo gerado através da Figura 7.13 (Banco de dados da topologia) deve ser analisado conjuntamente com as informações do *link state* geradas pelos roteadores.



**Figura 7.13 – Gráfico de custos de um sistema autônomo**

Quando as áreas OSPF não são configuradas, cada roteador no Sistema Autônomo possui uma mesma base de dados topológicas. O roteador gera a sua tabela de roteamento a partir deste grafo, gerando uma árvore do caminho mais curto com ele mesmo de raiz. Obviamente o caminho mais curto depende do roteador que está fazendo o cálculo. A árvore do caminho mais curto para o RT6 está exemplificada na Figura 7.14.



**Figura 7.14 – Tabela de Rotas para o Roteador RT6**

Observe que a árvore dá toda a rota para qualquer destino (rede ou host). Contudo, apenas o próximo passo para o destino é usado para enviar a mensagem.

## 7.4 IGRP (Interior Gateway Routing Protocol)

O protocolo IGRP é um protocolo de roteamento proprietário da Cisco System, desenvolvido em meados dos anos 80. O principal objetivo do IGRP é prover um mecanismo robusto de roteamento dentro de um sistema autônomo com topologia complexa e meios de comunicação com diferentes bandas passantes e atrasos.

O IGRP usa o protocolo Vector Distance e sua métrica é um **conjunto de parâmetros** que definem o quão bom é um caminho até o destino desejado. IGRP escolherá, então, o caminho com a melhor métrica. Também poderá escolher dois ou mais caminhos desde que suas métricas sejam próximas. A métrica do IGRP inclui:

- Delay Topológico (delay supondo que o caminho está totalmente livre de tráfego);
- A banda passante do enlace mais lento no caminho;
- A ocupação dos canais no caminho (percentagem da banda passante que está sendo gasta naquele momento);
- A confiabilidade do caminho (taxa média de erros).

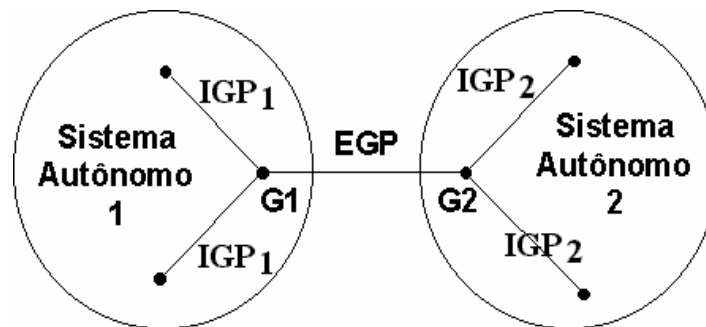
No IGRP, o protocolo Bellman-Ford (Vector Distance) é modificado em três aspectos críticos:

1. Ao invés de métrica simples, um vetor de métricas é usado para caracterizar o caminho.
2. Ao invés de escolher o caminho com menor métrica, o tráfego é dividido em vários caminhos cujas métricas estejam dentro de uma faixa especificada.
3. Várias características são introduzidas para prover estabilidade em situações onde a topologia muda. Split Horizons, Hold Down, reverse Poison e triggered Update.

Todas estas características são na verdade a fusão de várias idéias dos protocolos anteriormente apresentados. Neste sentido o IGRP tem grande vantagem sobre os demais. O ponto que mais pesa contra este protocolo é ele ser proprietário, diferente dos demais protocolos de roteamento apresentados anteriormente.

## 8 EGP - Exterior Gateway Protocol (RFC 904)

Como visto no capítulo 7, os protocolos de IGP (Interior Gateway Protocol) atuam dentro de um determinado sistema autônomo. Quando conectamos dois sistemas autônomos usamos o protocolo EGP (Exterior Gateway Protocol) para termos acesso as informações de roteamento e conseguirmos alcançar um gateway dentro de um sistema interior. A Figura 8.1 ilustra dois sistemas autônomos interligados por um protocolo EGP. Observe que cada sistema autônomo poderia estar rodando um protocolo de roteamento diferente, Sistema Autônomo 1 executando RIP e Sistema Autônomo 2 rodando OSPF.



**Figura 8.1 – Dois sistemas autônomos interligados por um protocolo EGP**

Em particular, gateways que rodam EGP usualmente usam IGP para obter informações de seu sistema autônomo. Um mesmo Gateway pode usar dois tipos diferentes de protocolos de roteamento simultaneamente, um para comunicar para fora do sistema autônomo e outro para dentro.

O protocolo EGP possui três características principais:

- Suporta mecanismo de aquisição de vizinho, que permite que um gateway requisiite a outro que eles devem trocar informações de alcançabilidade;
- Faz testes contínuos para ver se os vizinhos estão respondendo;
- Divulgação de informação entre vizinhos utilizando mensagens de atualização de rotas.

Para acomodar estas três funções básicas, o protocolo EGP define nove tipos de mensagens, mostrados na tabela 8.1.

<b>Tipo de Mensagem EGP</b>	<b>Descrição</b>
ACQUISITION REQUEST	Requisita um gateway para tornar-se vizinho.
ACQUISITION CONFIRM	Resposta positiva para Acquisition Request.
ACQUISITION REFUSE	Resposta negativa para Acquisition Request.
CEASE REQUEST	Requisita o término da relação de vizinhança.
CEASE CONFIRM	Resposta de confirmação para Cease Request.
HELLO	Requisita uma resposta ao vizinho para verificar se ele está operante.
I HEARD YOU	Resposta da mensagem Hello.
POLL REQUEST	Requisita a atualização de informações de roteamento da rede.
ROUTING UPDATE	Atualiza as informações de roteamento.
ERROR	Resposta a mensagens incorretas.

Tabela 8.1 – Mensagens do protocolo EGP

Veremos a seguir algumas dessas mensagens.

## 8.1 Cabeçalho Padrão do EGP

Todas as mensagens EGP começam com um cabeçalho fixo que identifica o tipo de mensagem enviada. O cabeçalho padrão é mostrado na Figura 8.2 com a descrição dos seus campos sendo apresentados na Tabela 8.2

<b>0</b>	<b>8</b>	<b>16</b>	<b>31</b>
<b>Version</b>	<b>Type</b>	<b>Code</b>	<b>Status</b>
<b>Checksum</b>		<b>Autonomous System Num.</b>	
<b>Sequence Number</b>			

Figura 8.2 – Cabeçalho Padrão EGP

<b>Campo</b>	<b>Descrição</b>
VERSION	Utilizado para identificar a versão corrente do protocolo
TYPE	Utilizado junto ao campo CODE, indica o tipo da mensagem
CODE	Identifica o código da mensagem (subtipo)
STATUS	Contém informações adicionais do estado, dependendo do tipo da mensagem
CHECKSUM	Utilizado para verificação de erros
AUTONOMOUS SYSTEM NUM	Indica o número do Sistema Autônomo do gateway que enviou a mensagem
SEQUENCE NUMBER	Contém um número utilizado pelo remetente para sincronismo de mensagens e respostas. É inicializado quando um gateway inicia a

Campo	Descrição
	comunicação com um "Vizinho Exterior" e incrementado a cada mensagem enviada.

Tabela 8.2 – Descrição dos campos do cabeçalho padrão EGP

## 8.2 Mensagem de Aquisição de Vizinho

Para um "Gateway Exterior" obter informações sobre um "Vizinho Exterior", este envia uma mensagem do tipo **Neighbor Acquisition** para adquirir estas informações. O formato da mensagem é mostrado na Figura 8.3

0	8	16	31
<b>Version</b>	<b>Type (3)</b>	<b>Code (0-4)</b>	<b>Status</b>
<b>Checksum</b>		<b>Autonomous System Num.</b>	
<b>Sequence Number</b>		<b>Hello Interval</b>	
<b>Poll Interval</b>		<b>Unused</b>	

Figura 8.3 – Formato da Mensagem de Aquisição do Vizinho

Os campos **Version**, **Type**, **Status**, **Checksum**, **Autonomous System Num.** e **Sequence Number** são os campos do cabeçalho padrão mostrado na Figura 8.2. Nesta mensagem especificamente o campo **Code** pode receber os seguintes valores (Tabela 8.3)

CODE	Significado
0	comando de requisição
1	resposta de confirmação
2	resposta de rejeição
3	comando de terminação
4	resposta de encerramento

Tabela 8.3 – Opções do campo CODE para a mensagem Neighbor Acquisition

Já os outros campos são descritos na Tabela 8.4

Campo	Descrição
HELLO INTERVAL	Utilizado para implementar um <i>timer</i> a ser utilizado para o envio de mensagens periódicas com o intuito de testar a resposta do vizinho (saber se o vizinho está vivo).
POLL INTERVAL	Máxima frequência de atualizações de rotas
UNUSED	Não usado

Tabela 8.4 – Demais campos da mensagem Neighbor Acquisition

### 8.3 Teste Contínuo de Funcionamento de Vizinho

A mensagem para descobrir se um vizinho esta respondendo é a mensagem EGP básica, com o campo **Type = 5** e o campo **Code = 0** (Comando **HELLO**) ou **1** (Resposta **I HEAD YOU** – Eu te escuto). A forma da mensagem é mostrada na Figura 8.4

<b>0</b>	<b>8</b>	<b>16</b>	<b>31</b>
<b>Version</b>	<b>Type (5)</b>	<b>Code (0 ou 1)</b>	<b>Status</b>
<b>Checksum</b>		<b>Autonomous System Num.</b>	
<b>Sequence Number</b>		<b>Unused</b>	

Figura 8.4 – Formato da Mensagem confirmação se vizinho esta respondendo

EGP usa o algoritmo *k-out-of-n* para declarar que o parceiro está UP ou DOWN. Ou seja, só quando *k* dentre as últimas *n* mensagens forem perdidas é que o parceiro é declarado DOWN.

### 8.4 Mensagem POLL REQUEST

As mensagens de *Poll request* e *poll response* permitem a um gateway obter informações de alcançabilidade de redes. Somente o campo **IP Source Network** é incluído no cabeçalho padrão (Figura 8.5).

Version	Type (2)	Code (0)	Status
Checksum		Autonomous System Num.	
Sequence Number		Reserved	
IP Source Network			

Figura 8.5 – Cabeçalho padrão POLL REQUEST

O campo **IP SOURCE NETWORK** especifica a rede comum ao sistema autônomo a qual ambos os gateways se ligam. A resposta a esta mensagem conterá rotas que tem distâncias medidas com respeito aos gateways na referida rede.

### 8.5 Mensagem de Atualização de Rotas

Um gateway exterior envia uma mensagem de *routing update* com informações de roteamento para o EGP vizinho. A mensagem enviada é a da Figura 8.6

8		16		31
Version	Type (1)	Code (0)	Status	
Checksum		Autonomous System Num.		
Sequence Number		#Int. Gwys	#Ext. Gwys	
IP Source Network				
Gateway 1 IP address (without net prefix)			Unused	
#Distances	Unused			
Distance $D_{11}$	#Nets at $D_{11}$	Unused		
Network 1 at Distance $D_{11}$			Unused	
Network 2 at Distance $D_{11}$			Unused	
...				
Distance $D_{12}$	#Nets at $D_{12}$	Unused		
Network 1 at Distance $D_{12}$			Unused	
Network 2 at Distance $D_{12}$			Unused	
...				
Gateway N IP address (without net prefix)			Unused	
#Distances	Unused			
Distance $D_{n1}$	#Nets at $D_{n1}$	Unused		
Network 1 at Distance $D_{n1}$			Unused	
Network 2 at Distance $D_{n1}$			Unused	
...				
Last Network at last distance from gateway N				

Figura 8.6 – Formato da mensagem de atualização de rotas do EGP

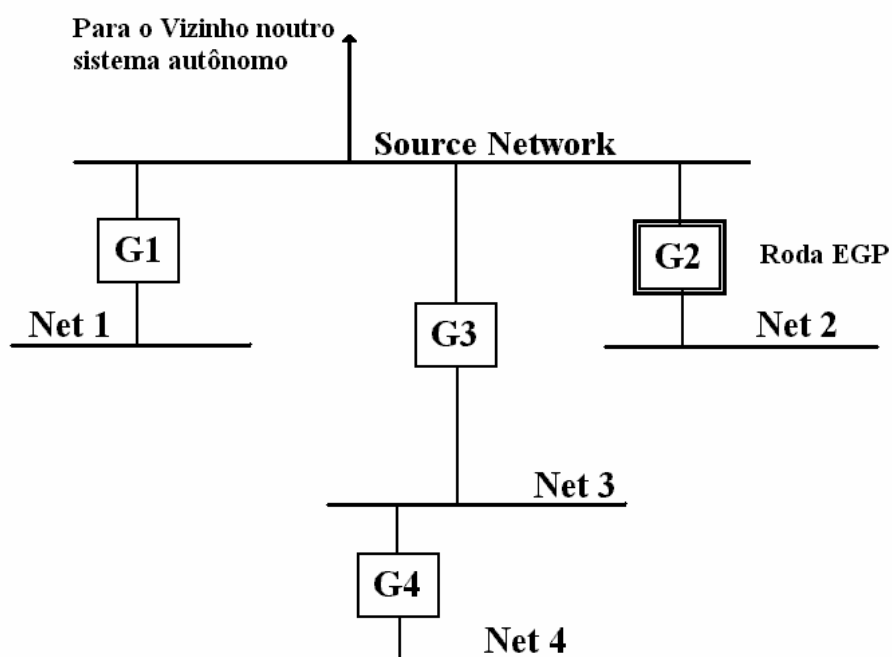
Os campos da Figura 8.6 são descritos na Tabela 8.5



<b>Campo</b>	<b>Descrição</b>
#INT. GWYS	Identifica o número total de <b>gateways interiores</b> que aparecem na mensagem
#EXT. GWYS	Identifica o número total de <b>gateways exteriores</b> que aparecem na mensagem
IP SOURCE NETWORK	Número IP da rede sobre a qual estão sendo solicitadas as informações de alcançabilidade, ou seja, a rede sobre a qual a alcançabilidade é medida
ROUTER 1 IP ADDRESS	Contém o endereço do roteador (gateway) destino
#DISTANCES	Utilizado para identificar o número de distâncias existentes no segmento relativo ao I-ésimo Gateway
DISTANCE $D_{ii}$	Identifica qual a distância (em hops) deste grupo de endereços
#NETS AT $D_{ii}$	Especifica quantos endereços são mencionados para a distância em questão, já que é utilizado juntamente com um determinado número de distância
NETWORK "I" AT DISTANCE $D_{ii}$	Reservado para a inclusão dos endereços IP (NETID) na respectiva distância

**Tabela 8.5 – Descrição dos campos da mensagem de atualização de rotas**

Para exemplificar o funcionamento do protocolo EGP observe a Figura 8.7



**Figura 8.7 – Mapa de redes com EGP**

Neste exemplo, o Gateway G2 foi designado a rodar EGP neste sistema autônomo. É este gateway que se comunicará com outros gateways exteriores. A obrigação de G2 é informar a alcançabilidade das redes 1 a 4 (Net1 à Net4).

G2 reportará então ao Gateway exterior que a Rede 1 (Net1) é alcançável pelo gateway G1, que as redes 3 e 4 (Net3 e Net4) são alcançáveis por G3 e a Rede 2 (Net2) é alcançável por G2 (ele mesmo).

Do ponto de vista de G2, Rede 2 está a uma distância zero. No entanto, ele reporta a Rede 2 com distância 1, a distância da SOURCE NETWORK.

## 9 DNS - Domain Name System

Do ponto de vista da Internet, as máquinas são muito bem identificadas pelo endereço IP. São estes valores (endereços IP) que realmente viajam no pacote IP, conforme já visto anteriormente. No entanto, do ponto de vista do usuário, estes números são difíceis de serem memorizados. É mais fácil para o usuário guardar o nome **www.microsoft.com.br** do que o seu respectivo endereço IP **200.240.13.31**.

A fim de facilitar a vida do usuário foi criado o serviço de DNS (Domain Name System – Serviço de Nomes), que **associa nomes simbólicos a endereços IP**, tornando mais fácil ao usuário final a utilização da Internet como um todo.

Um nome é meramente um identificador que consiste de uma sequência de caracteres escolhidas num alfabeto. O endereço IP é considerado um nome de baixo nível, no entanto os usuários preferem endereços de alto nível.

Uma forma de estabelecer um sistema de nomes seria criar um único banco de dados que contivesse a relação *<nome, endereço IP>* de todas as máquinas da Internet. Este banco de dados estaria numa única máquina (talvez em duas por questão de segurança) e todas as outras máquinas consultariam esta para conseguir traduzir os nomes para endereços IP.

Qualquer máquina que fosse adicionada a esta Internet deveria ser cadastrada neste servidor de nomes central.

Para uma Internet pequena este sistema funcionaria bem. Porém com seu crescimento este sistema traria alguns inconvenientes.

- Toda inclusão de nomes teria que ser feita por uma única pessoa (ou grupo de pessoas)
- Haveria problema na escolha dos nomes, devido as poucas opções que se teria

Devido a estas limitações em um sistema único, foi escolhido construir esse *Banco de Dados* de forma distribuída. Hoje, a Internet possui em cada país um órgão responsável pela criação de domínios e distribuição de endereços IPs, entre outras atividades. No Brasil esta organização é a FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo) – **www.fapesp.br**.

A FAPESP é responsável pela criação e manutenção de domínios no Brasil, é ela que possui o banco de dados **Nome do Domínio x Endereço IP** para todos os domínios cadastrados no Brasil. Caso você faça uma solicitação de inclusão de um domínio é preciso verificar antes se ele já existe - **registro.fapesp.br** – caso já exista você terá que escolher um outro nome. Caso não exista você pode providenciar o seu registro sozinho ou com a ajuda de um provedor de acesso.

O *Domain Name System* (DNS) é responsável por diversas tarefas. Ele cria uma hierarquia de *domínios*, referências, ou grupos de computadores. Estabelece um *nome de referência* (também conhecido como *endereço* da Internet) para cada computador na rede. As referências principais tem a responsabilidade de manter listas e endereços de outras referências do nível imediatamente inferior em cada grupo. Este nível inferior de referências é o responsável pelo próximo nível e assim por diante até o usuário final, ou computador final. O DNS utiliza esta hierarquia para transformar um nome de computador, escrito por extenso, em um número denominado *endereço IP*. O protocolo TCP/IP precisa saber o endereço da máquina local e o endereço IP da máquina que se deseja conectar. Quando o usuário informa o nome de uma máquina e não o seu endereço IP é o serviço de DNS que se responsabiliza em transformar aquele nome de máquina em endereço IP, para que se possa estabelecer a comunicação. Em geral, este processo é totalmente transparente ao usuário final.

Apesar da FAPESP ser responsável pelos domínios, são os domínios que são responsáveis pelos nomes das suas máquinas. Por exemplo é de responsabilidade da FAPESP registrar e divulgar o domínio UFES.BR, mas é de responsabilidade da UFES definir quais serão o nome das suas máquinas e caso existam sub-domínios quais são os nomes das máquinas destes sub-domínios, por exemplo o sub-domínio INF.UFES.BR possui uma máquina que se chama CAIRO.INF.UFES.BR.

Todo serviço pode ser acessado pelo nome hierárquico ou pelo endereço IP correspondente. Logo digitarmos em um Browser **<http://www.microsoft.com.br>** ou **<http://200.240.13.31>** tem o mesmo significado, sendo que a primeira opção só funciona se o computador possui o serviço de DNS instalado, já a segunda opção funciona em todos os casos.

A descrição do serviço de DNS é feita na RFC 1034 – <ftp://ds.internic.net/rfc/rfc1034.txt> - escrita por Paul Mockapetris.

## 9.1 Nomes Hierárquicos

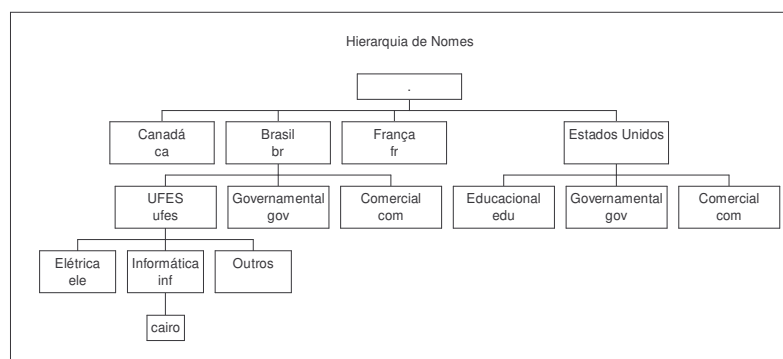
A Internet usa a analogia do sistema postal para definir o nome de um determinado domínio ou máquina. No sistema postal, uma casa pode ser localizada através de seu país, estado, cidade, rua e número, onde o número esta contido na rua, que por sua vez esta contida na cidade e assim por diante.

Na Internet o mecanismo é bastante semelhante, uma máquina é “endereçada” por um conjunto de informações, por exemplo a máquina “cairo” do Laboratório de Informática do CT-III da UFES (Universidade Federal do Espírito Santo) pode ser localizada da seguinte forma:

**cairo.inf.ufes.br**

Este “endereço” indica que a máquina “cairo” está contida no conjunto “inf” que por sua vez está contida no conjunto “ufes” que está no conjunto “br”. O que foi referido como conjuntos, no mundo da Internet recebe o nome de “domínios”. Desta forma, existe o

grande domínio “br”. Este contém vários sub-domínios dentre os quais o sub-domínio “ufes”, que por sua vez contém o sub-domínio “inf”, que possui a máquina “cairo”. A Figura 9.1 ilustra como essa divisão é feita na Internet.



**Figura 9.1 – Hierarquia de nomes na Internet**

A Figura 9.1 mostra uma hierarquia onde o domínio “.” é o nível mais alto, vindo abaixo dele a designação dos países, menos para os Estados Unidos, que não tem essa designação. Abaixo de cada país é feita uma estrutura de acordo com as necessidades daquele país. Observe entretanto, que não é somente uma máquina que responde pelo domínio “br”, nem tão pouco uma que responde pelo domínio “.”. Este é só um modelo esquemático da Hierarquia de nomes, na realidade, várias máquinas respondem por cada um dos domínios especificados, com objetivo de evitar falhas na resolução de nomes. Nem as próprias ligações físicas precisam existir necessariamente, podendo ser somente conceitual, estando ligadas de outra forma.

Com os conceitos de “domínios” já discutidos pode-se mostrar como a Internet está estruturada. A Internet é formada por um conjunto de grande domínios globais, divididos em países, como mostrado abaixo:

- br            Brasil
- ca            Canada
- uk            Reino Unido
- it            Itália

Existem ainda alguns domínios globais pertencentes aos Estados Unidos. Estes foram os domínios iniciais da Internet, antes das expansões para os outros países, e por isso não tem nenhuma extensão, como US por exemplo:

- mil           Militar
- gov           Governamental
- edu           Educacional
- com          Comercial
- net           Empresas/grupos preocupados com a Administração da Internet
- org           Outras organizações da Internet

Cada um destes domínios apresenta vários sub-domínios pelos quais são responsáveis. Por exemplo, o grande domínio global “br”, possui alguns sub-domínios:

- ufes.br                      UFES
- rnp.br                      Rede Nacional de Pesquisa
- usp.br                      USP

Existe ainda os sub-domínios de um domínio. Por exemplo o domínio com.br que denota as empresas possui vários sub-domínios dentre deste domínio maior:

- uol.com.br                Universo Online
- zaz.com.br                Provedor ZAZ
- digital.com.br            Digital, empresa de Informática

## 9.2 Ferramenta Nslookup

O sistema operacional UNIX apresenta uma ferramenta bastante útil para ajudar o administrador de rede a encontrar problemas de DNS. Esta ferramenta chama-se *nslookup*. A seguir serão mostrados alguns exemplos ilustrativos do uso do *nslookup*:

Comando	Significado
vitoria.inf.ufes.br> <b>nslookup</b> Default Server: cairo.inf.ufes.br Address: 200.241.16.8 >	Chama o utilitário <i>nslookup</i> , a partir da máquina vitoria.inf.ufes.br. O <i>nslookup</i> informa que o servidor de DNS <i>default</i> é cairo.inf.ufes.br (200.241.16.8) O símbolo “>” indica que o utilitário está esperando por um comando
> dix.mines.colorado.edu. Server: cairo.inf.ufes.br Address: 200.241.16.8  Name: dix.mines.colorado.edu Address: 138.67.12.10	Para saber o endereço IP de <i>dix.mines.colorado.edu</i> para digitar o nome e o endereço será mostrado na linha seguinte.
> www.zaz.com.br Server: cairo.inf.ufes.br Address: 200.241.16.8  Name: zazpoa7.zaz.com.br Address: 200.248.149.68 Aliases: www.zaz.com.br	Para saber o endereço IP de <i>www.zaz.com.br</i> , basta digitar o endereço. Será mostrado então o verdadeiro nome da máquina: <i>zazpoa7.zaz.com.br</i> e o seu endereço IP 200.248.149.68, bem como o <i>Aliases</i> para <i>www.zaz.com.br</i>

**Tabela 9.1 – Comando Nslookup**

Ainda dentro do Nslookup é possível definir que tipo de servidor estamos procurando. Se desejarmos saber qual o servidor de *mail* que responde pelo domínio **microsoft.com** basta digitarmos *setquerytype=mx* para indicarmos que o tipo de resposta é para um servidor de e-mail e logo após digitar o nome do domínio. Observe que é necessário colocar o ponto (.)

no final do domínio. No exemplo abaixo, este domínio possui cinco servidor de mail com a mesma prioridade (10), e abaixo é mostrado o endereço IP de cada servidor de e-mail.

```
> setquerytype=mx
> microsoft.com.
Server:  cairo.inf.ufes.br
Address:  200.241.16.8

microsoft.com  preference = 10, mail exchanger = mail1.microsoft.com
microsoft.com  preference = 10, mail exchanger = mail2.microsoft.com
microsoft.com  preference = 10, mail exchanger = mail3.microsoft.com
microsoft.com  preference = 10, mail exchanger = mail4.microsoft.com
microsoft.com  preference = 10, mail exchanger = mail5.microsoft.com
mail1.microsoft.com      inet address = 131.107.3.125
mail2.microsoft.com      inet address = 131.107.3.124
mail3.microsoft.com      inet address = 131.107.3.123
mail4.microsoft.com      inet address = 131.107.3.122
mail5.microsoft.com      inet address = 131.107.3.121
```

É possível ainda mudar o nome do servidor de DNS, para tanto basta escrever o comando **server** e o “nome do servidor de DNS”, no caso **caracol.inf.ufrgs.br**. A partir deste momento o servidor que esta sendo consultado para as pesquisas de DNS é **caracol.inf.ufrgs.br** e não mais **cairo.inf.ufes.br**, como pode ser visto a seguir.

```
cairo.inf.ufes.br> nslookup
Default Server:  cairo.inf.ufes.br
Address:  200.241.16.8

> server caracol.inf.ufrgs.br
Default Server:  caracol.inf.ufrgs.br
Address:  143.54.11.7

> www.altavista.digital.com
Server:  caracol.inf.ufrgs.br
Address:  143.54.11.7

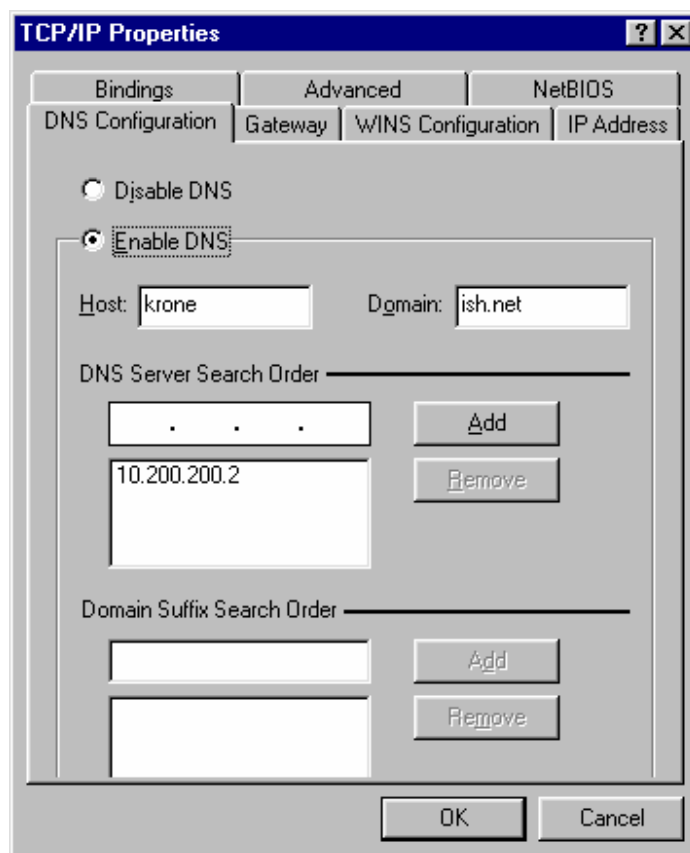
Non-authoritative answer:
Name:    altavista.digital.com
Addresses:  204.123.2.69, 204.123.2.75, 204.123.2.107, 204.74.103.37
          204.123.2.66
Aliases:  www.altavista.digital.com

>
```

Observe que a resposta do endereço IP de **www.altavista.digital.com** agora é dada por **caracol.inf.ufrgs.br** e não mais por **cairo.inf.ufes.br**. Esta opção é bastante útil para testar se um servidor esta respondendo corretamente, de forma remota.

### 9.3 Configuração do DNS

A configuração do serviço de DNS para o usuário final é bastante simples. Ele só precisa informar qual é o endereço IP do servidor de DNS. Vai ser este servidor entretanto que irá converter os nomes hierárquicos em endereços IP e retornar as respostas à aplicação. A Figura 9.2 mostra a configuração de um cliente de DNS em uma estação Windows 95.



**Figura 9.2 – Configuração do serviço de DNS em um cliente Windows 95**

Na Figura 9.2 é possível observar que existe a opção *Enable DNS* (habilitar o serviço de DNS) ou *Disable DNS* (Desabilitar o serviço de DNS). No campo *DNS Server Search Order* o usuário informa qual é o endereço IP que responderá à perguntas de DNS. Observe, que é possível colocar vários servidores de DNS, isto para que caso um servidor falhe o outro continue respondendo. Nesta mesma tela é possível definir o nome da máquina (*Host*), no caso **krone** e o nome do domínio (*Domain*), no caso **ish.net**.

A configuração do Servidor é feita basicamente editando um arquivo colocando o endereço IP da máquina e o seu respectivo nome. O servidor de DNS faz somente checar se o endereço informado consta na sua tabela de rotas. Caso a afirmativa seja verdadeira, ele retorna o endereço IP, caso contrário será informado uma mensagem de erro, informando que o endereço hierárquico não possui um endereço IP.

vitoria	IN	A	200.241.16.7
cairo	IN	A	200.241.16.8
berlim	IN	A	200.241.16.9
quebek	IN	A	200.241.16.21
otawa	IN	A	200.241.16.23
www	IN	CNAME	cairo



Estas são algumas máquinas do domínio **inf.ufes.br**. A máquina **otawa**, cujo endereço IP é 200.241.16.23 é conhecida dentro do domínio somente como *otawa* ou como *otawa.inf.ufes.br*, já para as redes externas, essa máquina é conhecida somente como *otawa.inf.ufes.br*. Da mesma forma a máquina *berlim.inf.ufes.br* possui o endereço 200.241.16.9 e assim sucessivamente.

É muito comum que o nome de máquinas de uma determinada sub-rede tenham alguma relação entre si, não sendo isto parte do padrão. É possível identificar as máquinas por qualquer nome, como XYZ, abc, X1Z2, etc. Observa-se entretanto que as pessoas tendem a colocar nomes dentro de um conjunto de valores, como por exemplo nomes de cidades, frutas, escolas de samba, nome de moedas, etc.

Observe que existe um nome WWW que aponta para a máquina **cairo** (200.241.16.8). Isto significa que quando o usuário digitar **www.inf.ufes.br** ele deverá fornecer o endereço IP de *cairo.inf.ufes.br*.

Vale ressaltar que o serviço de DNS só faz a tradução de um nome para um endereço IP, para que a máquina *cairo.inf.ufes.br* possa responder ao serviço de **www** é necessário que este serviço esteja instalado e funcionando nesta máquina.

Outro detalhe que vale a pena ressaltar é que o serviço de **www** não precisa estar em uma máquina com o nome na forma **www.** + “**nome do domínio**”, isto sendo somente uma convenção das pessoas utilizado amplamente.

## 9.4 Resolução de Nomes

Veremos nesta sessão como são trocadas as mensagens para resolução de nomes entre um cliente e um servidor de nome (DNS).

### 9.4.1 Formato da Mensagem

Assumiremos que um usuário invoca uma aplicação e fornece o nome da máquina com a qual a aplicação deve se comunicar. Antes de usar os protocolos UDP ou TCP para transmitir os dados pela rede, a aplicação deve encontrar o endereço IP da máquina destino (Lembre-se que o cabeçalho do nível de rede contém o endereço IP de destino e não o **nome** da máquina).

A aplicação passa o nome da máquina para o **RESOLVER** local. O *Resolver* verifica se o nome informado está na memória *cache*. Caso esteja, ele informa o endereço IP à aplicação, caso contrário ele monta uma mensagem, como a mostrada na Figura 9.3 e envia ao servidor de DNS, que é especificado na Figura 9.2.

As mensagens do *resolver* são encapsulados em datagramas UDP.

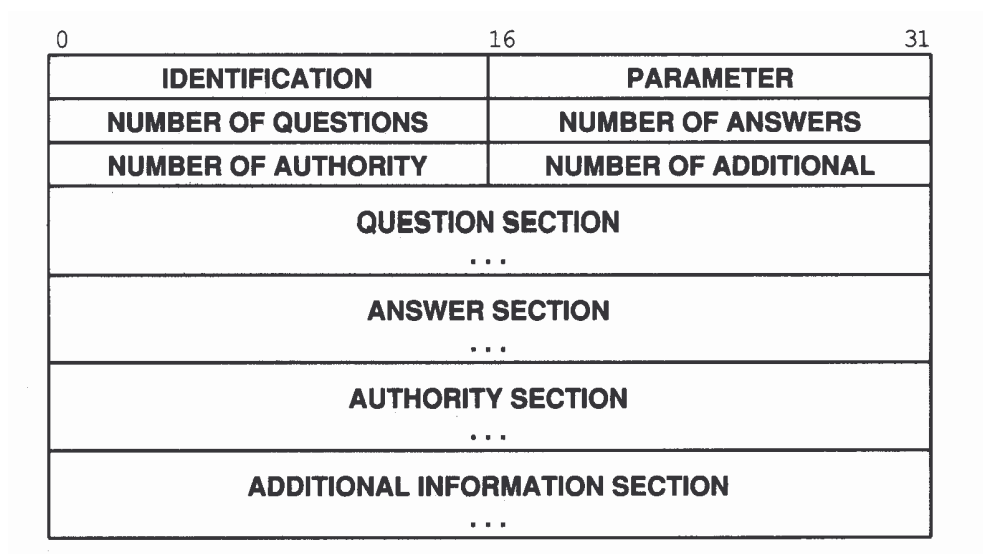


Figura 9.3 – Formato da mensagem DNS

O campo **IDENTIFICATION** é usado para identificar de forma única qual foi a pergunta feita com a sua respectiva resposta. O campo **PARAMETER** identifica qual a operação esta sendo solicitada. Os possíveis valores para este campo são descritos na Figura 9.4

Bit of <b>PARAMETER</b> field	Meaning
0	Operation: 0 Query 1 Response
1-4	Query Type: 0 Standard 1 Inverse 2 Completion 1 (now obsolete) 3 Completion 2 (now obsolete)
5	Set if answer authoritative
6	Set if message truncated
7	Set if recursion desired
8	Set if recursion available
9-11	Reserved
12-15	Response Type: 0 No error 1 Format error in query 2 Server failure 3 Name does not exist

Figura 9.4 - Opções do campo *Parameter*

Basicamente, o servidor pode responder a esta mensagem de três formas distintas.

No primeiro caso, ele responde com uma mensagem de erro informando por exemplo que o servidor não existe, ou o servidor falhou, ou há um mal formato na mensagem, etc.

Uma outra forma de resposta é o servidor de DNS responder alguns possíveis endereços IP de servidores DNS que podem resolver este nome. Daí o *Resolver* manda a mesma mensagem para este(s) novo(s) endereço. O processo continua até que uma mensagem informando o endereço IP ou uma mensagem de erro seja entregue.

Na outra forma, a mensagem é enviada ao primeiro servidor de DNS, caso este não saiba qual o endereço IP daquela máquina, ele próprio se encarrega de enviar uma ou mais mensagens para outro(s) servidor(es) de DNS a procura daquele nome. O processo continua até que chegue uma única resposta ao solicitante inicial, informando o endereço IP da máquina ou uma mensagem de erro.

## 10 Aplicações

As aplicações da arquitetura TCP/IP não possuem uma padronização comum. As aplicações ditas *padronizadas*, como o SNMP (e-mail), FTP, HTTP, etc possuem uma ou mais RFC's que as identificam. Logo, qualquer pessoa ou empresa pode escrever um programa de e-mail utilizando os padrões descritos nas RFC's sobre SNMP por exemplo.

Outras aplicações usadas largamente na Internet entretanto não possuem uma padronização, como é o caso por exemplo dos programas ICQ ([www.icq.com](http://www.icq.com)), RealAudio e RealVideo ([www.real.com](http://www.real.com)), Microsoft Netmeeting ([www.microsoft.com](http://www.microsoft.com)), entre outros. Este programas não possuem uma padronização a nível de RFC's. A especificação da aplicação fica a cargo da própria empresa que a confeccionou, cabendo a ela divulgar isto ou não.

Como visto anteriormente, toda aplicação precisa estar relacionada a uma porta e a um protocolo (TCP ou UDP). Para as aplicações padrão os números de porta estão reservados de 1 à 1.024. Já para as portas não padrão pode-se escolher entre 1.025 à 65.536. Cada fabricante escolhe um número de porta qualquer, podendo eventualmente ocorrer conflito de portas, se dois fabricantes resolverem fazer uma aplicação diferente utilizando a mesma porta. Pensando nisso a maioria dos sistemas permite configurar o número de porta que a aplicação servidor será utilizada, para evitar conflito com outras aplicações.

Veremos a seguir algumas aplicações utilizadas na Arquitetura TCP/IP.

### 10.1 Telnet – Terminal Remoto

O serviço de terminal remoto (telnet) permite que um usuário conecte-se a uma máquina e tenha acesso a todos os seus recursos (ou pelo menos aqueles que sua senha permite) como se estivesse trabalhando nesta máquina fisicamente.

É possível por exemplo a um usuário conectar-se a uma máquina através do telnet e imprimir um texto na impressora desta rede. Ou executar um programa utilizando os recursos de hardware da máquina remota. Esta opção é particularmente interessante quando uma pessoa deseja utilizar a velocidade de processamento de uma máquina que ele não possui. Neste caso ele pode usar o “poder” de processamento da máquina remota, em geral super-computadores. Esta opção foi amplamente utilizada no início da Internet onde somente algumas universidades americanas possuíam super-computadores, mas como todas as universidades estavam conectadas à Internet, todas elas podiam utilizar os recursos computacionais destas máquinas. Até hoje este serviço é bastante utilizado, inclusive no Brasil que possui três super-computadores que servem a várias universidades brasileiras.

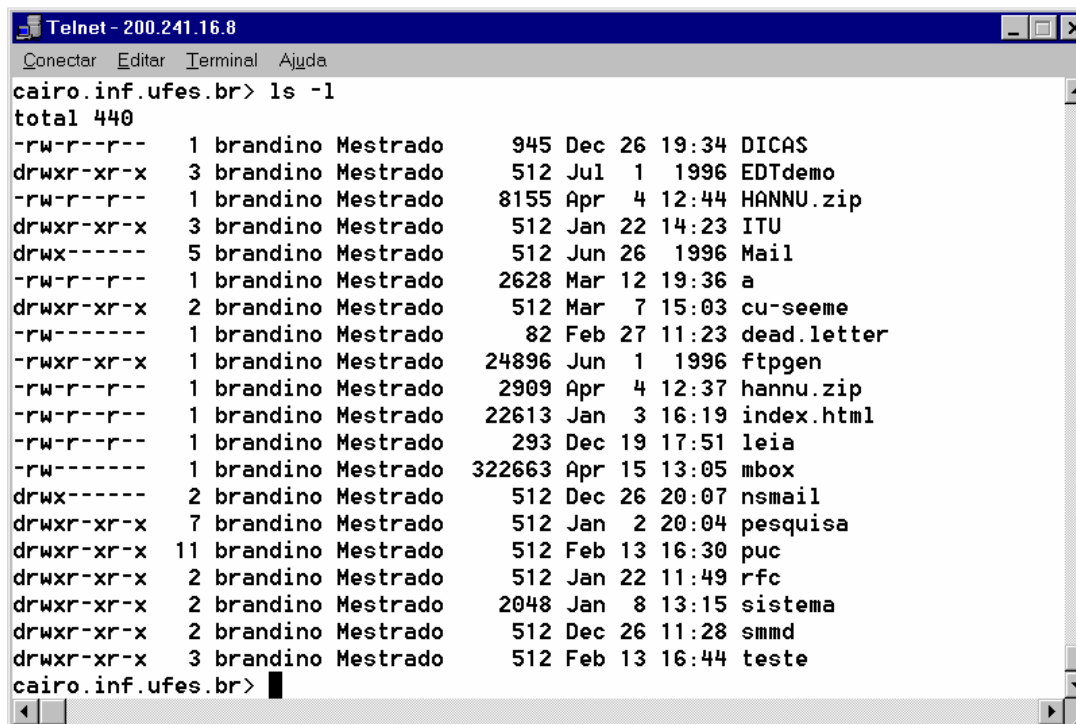
O serviço de telnet também é muito utilizado por administradores de sistemas ou redes, para concertarem configurações de programas sem precisarem irem ao local fisicamente.

Como toda aplicação client-server, o telnet também precisa de um servidor de telnet e um cliente telnet. Todas as máquinas possuem o servidor de telnet instalado, podendo as vezes

ser desabilitado por problemas de segurança, como acontece em vários provedores de acesso que tem medo de um *cracker* invadir o sistema utilizando o telnet. O sistema operacional Windows NT não possui o serviço de servidor telnet.

O Telnet é uma aplicação que utiliza o protocolo TCP para transportar os dados, pois necessita de certeza de entrega dos pacotes entre as máquinas envolvidas. O serviço de telnet esta na porta 23.

Qualquer máquina pode ter acesso ao serviço de telnet, desde que o usuário esteja cadastrado na máquina servidora e possua uma senha no sistema. Alguns sistemas permitem o usuário se logar como *guess* (convidado). A Figura 10.1 mostra a tela de um cliente telnet acessando um servidor telnet. Neste caso específico o servidor telnet é uma máquina UNIX (cairo.inf.ufes.br – 200.241.16.8) e o cliente telnet esta sendo executado em uma estação Windows 95/NT qualquer.



The image shows a Telnet window titled "Telnet - 200.241.16.8". The window has a menu bar with "Conectar", "Editar", "Terminal", and "Ajuda". The main text area shows a command prompt for "cairo.inf.ufes.br" where the user has entered "ls -l". The output is a long list of files and directories with their permissions, owner, group, size, date, time, and filename. The files listed include DICAS, EDTdemo, HANNU.zip, ITU, Mail, a, cu-seeme, dead.letter, ftpgen, hannu.zip, index.html, leia, mbox, nsmail, pesquisa, puc, rfc, sistema, smmd, and teste. The prompt "cairo.inf.ufes.br>" is visible at the bottom of the text area.

```

Telnet - 200.241.16.8
Conectar  Editar  Terminal  Ajuda
cairo.inf.ufes.br> ls -l
total 440
-rw-r--r--  1 brandino Mestrado   945 Dec 26 19:34 DICAS
drwxr-xr-x  3 brandino Mestrado   512 Jul  1 1996 EDTdemo
-rw-r--r--  1 brandino Mestrado  8155 Apr  4 12:44 HANNU.zip
drwxr-xr-x  3 brandino Mestrado   512 Jan 22 14:23 ITU
drwx----- 5 brandino Mestrado   512 Jun 26 1996 Mail
-rw-r--r--  1 brandino Mestrado  2628 Mar 12 19:36 a
drwxr-xr-x  2 brandino Mestrado   512 Mar  7 15:03 cu-seeme
-rw-----  1 brandino Mestrado    82 Feb 27 11:23 dead.letter
-rwxr-xr-x  1 brandino Mestrado 24896 Jun  1 1996 ftpgen
-rw-r--r--  1 brandino Mestrado  2909 Apr  4 12:37 hannu.zip
-rw-r--r--  1 brandino Mestrado 22613 Jan  3 16:19 index.html
-rw-r--r--  1 brandino Mestrado   293 Dec 19 17:51 leia
-rw-----  1 brandino Mestrado 322663 Apr 15 13:05 mbox
drwx----- 2 brandino Mestrado   512 Dec 26 20:07 nsmail
drwxr-xr-x  7 brandino Mestrado   512 Jan  2 20:04 pesquisa
drwxr-xr-x 11 brandino Mestrado   512 Feb 13 16:30 puc
drwxr-xr-x  2 brandino Mestrado   512 Jan 22 11:49 rfc
drwxr-xr-x  2 brandino Mestrado 2048 Jan  8 13:15 sistema
drwxr-xr-x  2 brandino Mestrado   512 Dec 26 11:28 smmd
drwxr-xr-x  3 brandino Mestrado   512 Feb 13 16:44 teste
cairo.inf.ufes.br>

```

Figura 10.1 – Usuário acessando o serviço de telnet

Observe que mesmo o usuário remoto estando em uma máquina Windows, quando ele se conecta ao servidor de telnet, que é uma máquina UNIX, ele tem que usar os comandos deste sistema operacional. O mesmo aconteceria se o sistema operacional fosse o VMS por exemplo.

É possível também uma máquina UNIX fazer uma conexão telnet com outra máquina UNIX, bastando para isso digitar na linha de comando:

**telnet** <nome da máquina ou endereço IP>

O telnet é um pouco ineficiente do ponto de vista de transporte de informações. Isto acontece porque todo caracter digitado é mandado para o servidor telnet na forma de um pacote separado dos demais. Diferente do serviço de FTP, onde o usuário digita toda uma linha, e só após o *enter* do usuário é que toda a linha é enviada. Isto não ocorre no telnet porque as vezes a aplicação que será executada pede ao usuário para digitar somente um único caracter (as vezes sem enter). Logo esta é a única forma de compatibilizar todas as aplicações.

## 10.2 FTP - Transferência de Arquivos

O FTP (File Transfer Protocol – Protocolo de Transferência de Arquivos), como o próprio nome indica é um serviço que possibilita a transferência de arquivos entre máquinas.

É possível transferir qualquer tipo de arquivo entre máquinas com diferentes sistemas operacionais. Basicamente você tem um servidor de FTP, que contem vários arquivos armazenados nele e um cliente FTP, que se conecta ao servidor de FTP para buscar um arquivo (download) ou colocar um arquivo no servidor (upload).

Existem basicamente duas classes de FTP. O FTP anônimo que é aquele onde o usuário não precisa se identificar. Neste caso ele só tem acesso a arquivos públicos e sem direito a transferir arquivos para o servidor. Só tendo direito de pegar arquivos do servidor para ele.

Já no FTP identificado, o usuário entra com o seu *login* e a sua senha, que foram previamente cadastradas e tem acesso a sua área de trabalho, onde pode fazer *downloads* ou *uploads* de arquivos. Dependendo basicamente da sua configuração junto ao servidor de FTP, que é definida pelo administrador da rede.

O FTP é um serviço utilizado por várias pessoas. Este serviço vem nativo nas máquinas UNIX tanto a versão servidor como a cliente, e nas máquinas Windows pode ser instalado a versão servidora, a cliente, no modo caracter já vindo nativa. Outras aplicações visuais entretanto transforma a tarefa de se transferir arquivos numa tarefa muito mais fácil para o usuário final.

No modo a caracter, tanto no UNIX quanto no Windows, o usuário digita basicamente:

**ftp** <nome da máquina>  
Ex: ftp ftp.microsoft.com

Após este comando e com estabelecimento da conexão, será requisitado o nome da conta:

name:

Logo após, a senha:

password:

O usuário informa os dados e a partir daí irá usar os comandos do FTP. Caso seja um FTP anônimo (o usuário não tem conta na máquina remota), basta ele digitar **anonymous** no campo *name* e o seu **e-mail** no campo *password*.

Os principais comandos do FTP são: **put** <nome do arquivo> para enviar um arquivo da máquina local para a máquina remota e **get** <nome do arquivo> para trazer um arquivo da máquina remota para a máquina local. Alguns outros comandos do FTP são mostrados na Tabela 10.1

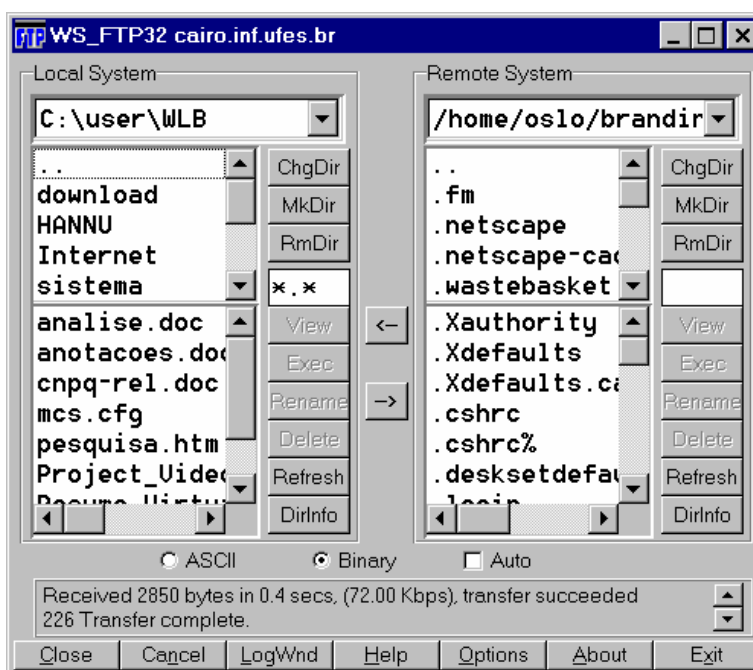
Comando	Descrição
dir ou ls	Mostra a listagem do diretório corrente
cd <caminho>	Muda o diretório remoto
lcd <caminho>	Muda o diretório local
Pwd	Informa o diretório remoto atual
! Ex: !del *.txt	Permite a execução de comandos na máquina local Deleta os arquivos *.txt na máquina local
get <arquivo> Ex: get scan.exe	Transfere um arquivo da máquina remota para a máquina local
Mget <arquivos> mget *.txt	Transfere múltiplos arquivos da máquina remota para a máquina local
put <arquivo> Ex: put agenda.dat	Transfere um arquivo da máquina local para a máquina remota
mput <arquivos> mput *.gif	Transfere múltiplos arquivos da máquina local para a máquina remota
hash	Imprime o símbolo “#”, a cada 1024 bytes transferidos
help ou ?	Obtém a lista dos comandos disponíveis
quit ou bye	Encerra uma conexão FTP
ascii	Para transferência de arquivos no forma ASCII Esta transferência, faz conversão de formatos, no caso de estarmos transferindo arquivos de máquinas com sistemas operacionais diferentes. <b>Aconselhável para transferência de arquivos textos</b>
bin	Para transferência no formato binário Esta transferência não altera nenhum bit do arquivo, transferindo-o de forma a deixa-lo igual na máquina local e remota. <b>Aconselhável para transferência de programas</b>

**Tabela 10.1 – Comandos do FTP**

Observe que os comandos bin e ascii tem que ser informados ao computador antes do início da transferência de um arquivo. Antes de transferir algum dado, verifique se o FTP esta no modo apropriado, porque se não você pode transferir dados e depois não conseguir ter acesso a eles.

Existem basicamente três tipos de arquivos: ASCII (American Standard Code for Information Interchange, ou Código Americano Padrão para Intercâmbio de Informações), EBCDIC (Extended Binary Coded Decimal Interchange Code, ou Código Estendido de Intercâmbio Decimal Codificado em Binário) e binário. ASCII e EBCDIC não são nada mais que protocolos, ou modos padronizados para organizar bits de dados em algo que os humanos possam entender. ASCII é o denominador comum para a computação baseada em caracteres. Os códigos ASCII em seu computador representam os caracteres que você vê na tela. O EBCDIC funciona como o ASCII, mas é usado somente entre certos tipos de computadores de grande porte e normalmente não se tem muito contato com ele. Já os códigos binários, normalmente representam programas executáveis que só rodam em um tipo de computador, um programa em Mac não roda no Windows.

Visando facilitar a transferência de arquivos, alguns programas foram desenvolvidos para tornar esta tarefa mais fácil e intuitiva. A Figura 10.2 mostra a tela de um sistema que permite transferir arquivos entre duas máquinas utilizando o protocolo FTP. Observe que todos os comandos visuais são traduzidos internamente em comandos do FTP e executados pelo protocolo.



**Figura 10.2 – Sistema de FTP no Windows**

É possível colocar todos os comandos que devem ser executados pelo FTP em um arquivo do tipo *script* e mandar executar este arquivo. Todos os comandos serão executados em sequência após o término de cada comando. É mostrado abaixo um exemplo de *script* trazados.



```

open ftp.embratel.net.br
user anonymous brandino@inf.ufes.br
verbose
cd /pub/windows/winsock
hash
ascii
mget *.txt
get readme.msg
bin
get twsk20b.exe
bye

```

Neste exemplo, é possível observar que primeiro será aberta uma conexão FTP com a máquina [ftp.embratel.net.br](http://ftp.embratel.net.br), após o estabelecimento da conexão será informado o login de anonymous e o e-mail do usuário. Só após estes comandos será trocado o diretório e os arquivos do tipo \*.txt serão transferidos, após a transferência de todos os arquivos, ele trará mais dois outros arquivos, um do tipo **msg** e o último arquivo é um programa **exe**. A conexão é encerrada através do comando **bye**.

Os sistemas UNIX permitem ainda que este script seja executado em *Background*, enquanto o usuário faz uma outra tarefa ou agendado para ser executado em um determinado horário. Uma muito utilizada em sistemas UNIX é o FTP por comandos em lote. Este serviço consiste basicamente de um *script* com comando FTP que serão executados.

Para executar o comando em *background* basta digitar:

```
$ ftp -n <trazdados> trazdados.out&
```

Este comando irá executar o FTP, que terá como parâmetros o arquivo trazdados, e colocará as saídas de erro ou de sucesso no arquivo trazdados.out

Como dito anteriormente, este comando pode ser executado em um determinado horário, para tanto, basta usar o comando **at** e a hora a ser executado.

```

$ at 1040
ftp -n <cesar.com> cesar.out&
<CTRL> + D
$

```

Isto fará com que o FTP seja executado as 10:40h, em Background.

## 10.3 E-mail - Mensagens Eletrônicas

O E-MAIL (*Eletronic Mail*) é um serviço de correio eletrônico, onde pode-se trocar correspondência de uma forma rápida e barata com outras pessoas, de forma análoga ao correio tradicional. Utilizando-se desta analogia, uma carta, quando enviada, deve conter o endereço do destinatário e do remetente. No correio eletrônico também usa-se endereços, denominados endereços eletrônicos.

Como já foi visto, cada máquina possui seu endereço. Visto que vários usuários utilizam-se de uma mesma máquina, faz-se necessário a identificação de cada usuário. Esta identificação é feita acrescentando a “identificação do sistema” ao endereço da máquina, unidos pelo símbolo “@” (arroba) ou “at”. Por exemplo, o usuário “brandino” que possui uma conta na máquina de endereço “cairo.inf.ufes.br” será identificado da seguinte forma:

brandino@cairo.inf.ufes.br

ou

brandino at cairo.inf.ufes.br

Entretanto, na maioria dos casos quando configura-se o serviço de DNS informamos qual máquina responderá pelo serviço de e-mail daquele domínio. Neste caso a máquina cairo.inf.ufes.br responde pelo domínio **inf.ufes.br**, logo podemos utilizar endereços da forma:

brandino@inf.ufes.br

ou

brandino at inf.ufes.br

Desta forma, pode-se enviar mensagens para qualquer usuário de qualquer máquina do mundo. Uma boa prática é utilizar como identificação do usuário, seu nome verdadeiro, porém nem sempre isso é possível, basta imaginar quantos nomes de Maria, João ou José temos, daí usa-se apelidos, sobrenomes, ou combinação destes. Veja alguns exemplos de endereços eletrônicos abaixo:

- zegonc      #Nome real José Gonçalves Pereira Filho
- brandino    #Nome real Wandreson Luiz **Brandino**
- rcesar      #Nome real **Roberto Cesar** Cordeiro

Caso você cometa algum erro ao escrever o endereço do destinatário, a sua mensagem não será entregue e retornará a você para que possa enviá-la à pessoa correta. Entretanto, se a combinação de nomes digitada existir na Internet, a sua carta será entregue a uma outra pessoa, e não retornará à você.

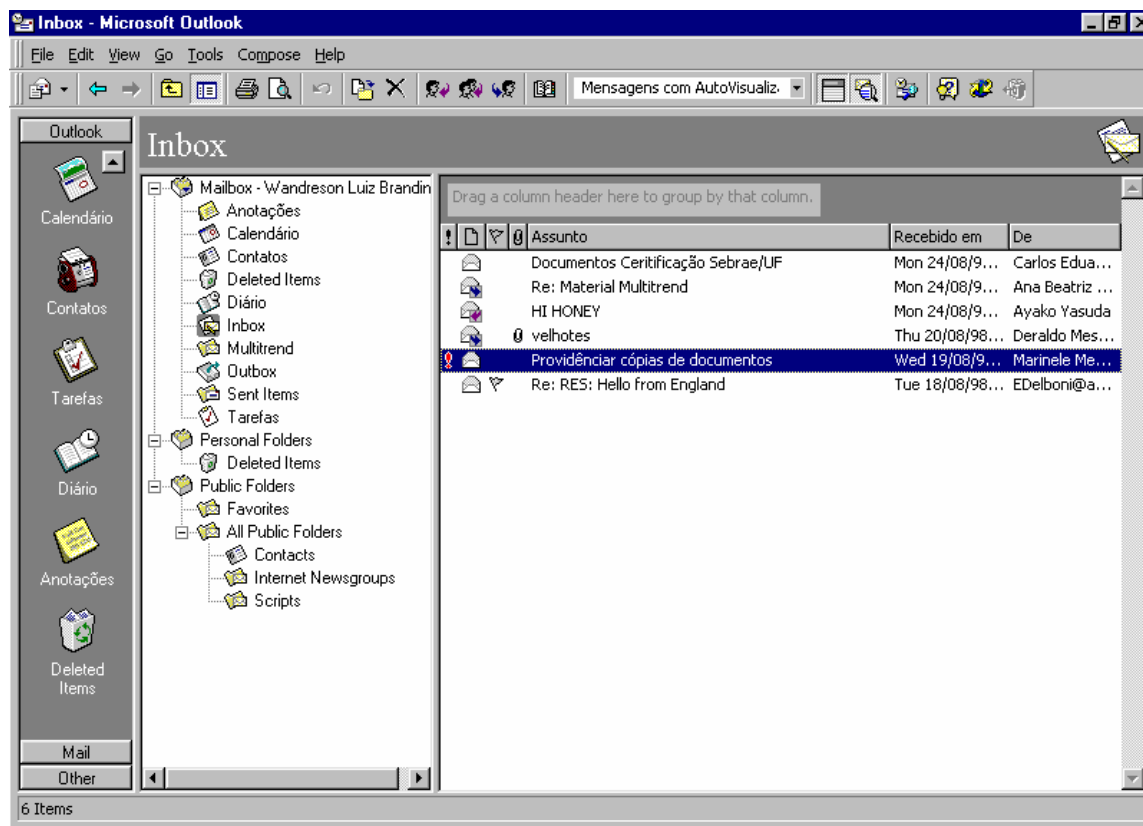
As mensagens de E-MAIL possuem alguns identificadores básicos que formam o cabeçalho da mensagem, a fim de identificar origem, destino, assunto, etc. Veja agora como é a estrutura interna de uma mensagem:

Message 9:	#(número da mensagem)
From rogerio@telemidia.puc-rio.br Fri Apr 11 10:40:35 1997	#(end. do remetente e data de envio)
Date: Fri, 11 Apr 1997 10:40:35 -0400	#(data de chegada)
From: rogerio@telemidia.puc-rio.br (Rogerio F. Rodrigues)	#(end. do remetente e seu nome real)
To: brandino@inf.ufes.br	#(end. do destinatário)
Subject: Congresso SBRC/97	#(Assunto da Mensagem)
	#(Corpo da Mensagem)
<p>Ola Wandreson,</p> <p>Como vao as coisas ai em Vitoria? Aqui na PUC esta aquela batalha de sempre, estou acabando a tese. Quando terminar a versao draft te mando uma copia. A proposito, voce vai no SBRC/97? Mande mensagem confirmando pra gente armar alguma coisa.</p> <p>Abracos Rogerio</p>	

Observe que a mensagem é enviada para brandino@inf.ufes.br e não para brandino@cairo.inf.ufes.br, isto porque pode-se omitir o nome da máquina, e deixar só o nome do domínio, desde que se configure na máquina responsável pelo domínio inf.ufes.br, que qualquer mensagem enviada a inf.ufes.br deve ser direcionada à máquina cairo.inf.ufes.br. Ou seja, uma mensagem enviada para brandino@inf.ufes.br ou para brandino@cairo.inf.ufes.br vão para o mesmo destinatário.

Alguns dados que vão com a mensagem não são descritos aqui, servem para identificar por onde a mensagem passou. Isto ajuda a confirmar se a mensagem veio de onde o usuário indica que ela veio realmente. É uma forma de validação da mensagem, utilizada em conjunto com a identificação da mensagem permite confirmar se o remetente é realmente quem ele diz que é.

Para mandar uma mensagem pela Internet, o usuário deve usar um programa de e-mail. Existem vários programas deste tipo. O UNIX traz o programa *mail* de forma nativa no sistema operacional, já os *Browsers* Netscape e Internet Explorer possuem também versões de e-mail, existindo ainda os programas como o Eudora e o Microsoft Outlook, entre outros. A Figura 10.3 mostra a tela principal do Microsoft Outlook, onde é possível ler, enviar e armazenar mensagens eletrônicas entre outras informações.



**Figura 10.3 – Tela principal do programa Microsoft Outlook**

Todos os aplicativos de e-mail utilizam o protocolo SMTP (Simple Mail Transfer Protocol) para a troca de mensagens entre diferentes servidores. Vários padrões de formato de mensagens também são suportados.

Quando o primeiro programa de e-mail foi criado ele só aceitava caracteres e sem acentos. Hoje já existem padrões para mandar figuras, áudio, vídeo, textos formatos e acentuados, além de arquivos e página HTML (Home Pages).

## 10.4 News Group

Usenet é o conjunto de máquinas que intercambia artigos identificados por um ou mais rótulos reconhecidos universalmente, chamados de 'newgroups' (ou apenas 'groups'). A UseNet reúne mensagens sobre um único tópico em um grupo de notícias. Os grupos de notícias são bancos de dados destas mensagens que estão frequentemente duplicadas em alguns computadores.

Este serviço permite a realização de fóruns de debates e acesso a notícias e assuntos variados, desde fabricação de computadores à culinária alemã. O usuário envia comandos para trazer artigos de seu interesse ou participa de debates mandando a sua opinião ou

fazendo uma pergunta que será passada aos outros usuários que participam da discussão. Após algum tempo o usuário começara a receber respostas a sua pergunta ou mais informações sobre o assunto em pauta.

Este serviço é suportado pelo UseNet, que é o protocolo que determina como os grupos de mensagens (as informações) serão transferidas entre os computadores ou mesmo armazenadas já que algumas instalações mantêm arquivos de discussões anteriores. O NewsGroup é acessado por um leitor de News, como os encontrados no Netscape e o Internet Explorer.

A diferença básica entre uma lista de discussão e o NewsGroup é que na lista a mensagem é enviada para o usuário, na sua caixa-postal (via e-mail), já no caso do NewsGroup o usuário tem que se conectar a um servidor de NewsGroup e ler a mensagem. Todos os *Browsers* dão suporte a News Groups. Alguns servidores ou provedores de acesso limitam o acesso as mensagens de News somente para os seus usuários, outros como o universo on-line (UOL) por exemplo permitem com que qualquer pessoa tenha acesso aos fóruns de discussão. O endereço do UOL é [news://news.uol.com.br](mailto:news://news.uol.com.br).

## 10.5 WWW (Word Wide Web)

O serviço de WWW é o serviço que mais cresce na Internet, perdendo somente para o serviço de e-mail, que é sem dúvida o mais utilizado pelos Internautas. Este serviço permite que os usuários naveguem através de páginas de página HTML (HyperText Markup Language – Linguagem de Marcação de Texto), conhecidas como *HomePages* tendo acesso a textos, figuras, sons, vídeos, imagens, etc e através de *Hiperlinks* que são pontos do texto ou figuras que ao se clicar o usuário é levado a uma outra página e assim sucessivamente.

O serviço de WWW foi o grande responsável pela popularização da Internet, pela sua forma simples e atrativa de apresentar informações. O WWW é suportado pelo protocolo HTTP (HyperText Transfer Protocol – Protocolo de Transferência de Hiper-Texto).

O serviço de WWW é acessado através dos *Browsers* de navegação. O Internet Explorer e o Netscape Navigator são os mais utilizados atualmente. A Figura 10.4 mostra um usuário navegando em uma máquina através do Internet Explorer.



Figura 10.4 – Página HTML no *Browser* Internet Explorer

O usuário precisa informar apenas o endereço da página, e a página começará a ser carregada em seu *Browser*. As páginas WWW dão acesso a uma infinidade de informações, como compras de livros, CD's, diversão (piadas, bate-papo, etc), informações de pesquisas, visitas a museus, entre outros.

Alguns exemplos de endereços são:

Serviço Oferecido	Endereço na Internet
Venda de Livros	<a href="http://www.books.com">http://www.books.com</a>
Venda de Livros	<a href="http://www.bookpool.com">http://www.bookpool.com</a>
Laboratório Informática da UFES	<a href="http://www.inf.ufes.br">http://www.inf.ufes.br</a>
ATM Forum	<a href="http://www.atmforum.com">http://www.atmforum.com</a>
Microsoft	<a href="http://www.microsoft.com">http://www.microsoft.com</a>
Revista PlayBoy	<a href="http://www.playboy.com">http://www.playboy.com</a>
AT&T	<a href="http://www.att.com">http://www.att.com</a>
Cerveja Antarctica	<a href="http://www.antarctica.com.br">http://www.antarctica.com.br</a>
Nasa	<a href="http://www.nasa.gov/">http://www.nasa.gov/</a>
Declaração Imposto de Renda	<a href="http://www.receita.fazenda.gov.br/">http://www.receita.fazenda.gov.br/</a>
Jornal do Brasil	<a href="http://www.jb.com.br">http://www.jb.com.br</a>
Sebrae - ES	<a href="http://www.sebes.com.br">http://www.sebes.com.br</a>



Serviço Oferecido	Endereço na Internet
Cerveja Skol	http://www.skol.com.br
Supermercados Roncetti	http://www.roncetti.com.br
Universo On-Line	http://www.uol.com.br
Piadas	http://www.humortadela.com
Embratel	http://www.embratel.net.com
RNP - Rede Nacional Pesquisas	http://www.rnp.br
FAPESP	http://www.fapesp.br
Flamengo	http://www.flamengo.com.br
Rede Globo	http://www.redeglobo.com.br/
Disney	http://www.disney.com/
Compra de Livros	http://www.bookpool.com
Compra de Livros	http://www.books.com
Compra de Livros	http://www.amazon.com
Universo On Line	http://www.uol.com.br
Volkswagen	http://www.volks.com.br
Brahma	http://www.brahma.com.br
SebraeSat	http://www.sebraesat.com.br
IBGE	http://www.ibge.gov.br
TecToy	http://www.tectoy.com.br
Microsoft Brasil	http://www.microsoft.com/brasil
Bradesco	http://www.bradesco.com.br
Memoria Ayrton Senna	http://www.africanet.com.br/senna
Serviço de Busca: Cadê?	http://www.cade.com.br
Serviço de Busca: Altavista	http://www.altavista.digital.com
Venda de Livros	http://www.booknet.com.br

## 10.6 Ping

O **Ping** é utilizado basicamente para verificar se uma máquina está “viva” e quanto tempo uma mensagem UDP gasta para sair da máquina local, ir à máquina remota e retornar. Ao final de sua execução é feita uma estatística da comunicação. Através desta estatística pode-se verificar se existe contato com a máquina remota e qual a *performance* da comunicação com a máquina. Veja o exemplo abaixo:

```
cairo.inf.ufes.br> ping microsoft.com
PING microsoft.com (207.68.137.53): 56 data bytes
64 bytes from 207.68.137.53: icmp_seq=0 ttl=54 time=1046 ms
64 bytes from 207.68.137.53: icmp_seq=1 ttl=54 time=960 ms
64 bytes from 207.68.137.53: icmp_seq=2 ttl=54 time=1023 ms
64 bytes from 207.68.137.53: icmp_seq=3 ttl=54 time=619 ms
64 bytes from 207.68.137.53: icmp_seq=4 ttl=54 time=516 ms
64 bytes from 207.68.137.53: icmp_seq=5 ttl=54 time=630 ms
64 bytes from 207.68.137.53: icmp_seq=6 ttl=54 time=778 ms
```

```
64 bytes from 207.68.137.53: icmp_seq=7 ttl=54 time=788 ms
```

```
-----microsoft.com PING Statistics-----
```

```
9 packets transmitted, 8 packets received, 11% packet loss
```

```
round-trip (ms) min/avg/max = 516/795/1046 ms
```

```
cairo.inf.ufes.br>
```

Observe que é informado um endereço hierárquico e este endereço é transformado primeiramente em endereço IP, através do serviço de DNS, logo após são enviadas mensagens para o serviço de **ping remoto**, e este as re-envia para a máquina que solicitou o serviço e calcula o tempo em milissegundos gasto para o pacote ir e voltar.

Analisando estes dados, é possível verificar se a conexão está muito lenta (tempo em milissegundos muito grande) e qual a taxa de erros de pacotes (packet loss), incidindo que os pacotes estão sendo perdidos ao longo da rede, geralmente por problemas de congestionamento ou de perda de pacotes entre os *gateways* da origem até o destino.

Todas as linhas são mostradas ao longo do tempo. Para acabar o ping, basta pressionar CTRL+C. Neste momento será mostrada a estatística da conexão. Mostrando o: número de pacotes transmitidos, recebidos, porcentagem de pacotes perdidos e tempo de resposta. Assim, tem-se uma avaliação da conexão com a máquina remota.

## 10.7 Finger

O Finger é um aplicativo Internet que permite obter informações sobre um usuário específico numa máquina específica. Por exemplo, ao executarmos o comando:

```
finger cecilio@salgueiro.telemidia.puc-rio.br
```

Obtemos o seguinte resultado:

```
cairo.inf.ufes.br> finger cecilio@salgueiro.telemidia.puc-rio.br
[salgueiro.telemidia.puc-rio.br]
Login   Name      TTY      Idle   When   Where
cecilio Edmundo Lopes Cecili 024    <Apr 15 12:47> 200.20.120.177
cairo.inf.ufes.br>
```

Além das informações básicas sobre a conta do usuário (Nome real, Departamento ou setor, diretório, shell) ainda diz se o usuário está *logado* ou não.

Quando se usa o comando finger sem nenhum argumento, ele dá a lista dos usuários que estão logados no sistema naquele momento.

```
finger @npd1.ufes.br
```



## 11 Evolução e Conclusão

Como podemos ver, a arquitetura TCP/IP traz grandes facilidades na interligação de redes heterogêneas, provendo vários serviços amplamente utilizados atualmente, como: e-mail, www, ftp, telnet, entre outros. A Internet é realmente um fenômeno tecnológico, que proporciona mudanças comportamentais e culturais nas pessoas e empresas que dela se utilizam.

É possível, por exemplo trabalhar em casa e se corresponder com a empresa somente por e-mail, ter acesso as mais diversas informações, como jornais, revistas, livros, etc, e ainda pedir pizza, alugar carro, fazer reserva em vôos ou hotéis, comprar comida em supermercados ou CDs tudo sem sair de casa ou do escritório.

Sem dúvida muito já foi feito, com uma arquitetura extremamente simples e por isso amplamente utilizada. Entretanto, muita coisa ainda precisa ser feita ou melhorada

Hoje podemos ouvir uma rádio na Internet ou conversar com uma pessoa, vendo-a inclusive, mas a qualidade é ainda muito baixa. Projetos como a **Internet 2**, que começou nos Estados Unidos, Europa e que agora já se encontra em fase de testes no Brasil, vão aumentar significativamente a velocidade e a qualidade das redes atuais impactando diretamente na qualidade das aplicações.

Hoje, no Brasil, a maior velocidade de uma canal de comunicação é de 2 Mbps. Caso o usuário queira maior velocidade terá que alugar vários canais de 2 Mbps. Entretanto muitas empresas estão ligadas à Internet por velocidades ainda menores como de 64, 128 ou 256 Kbps. Na Internet 2 a velocidade mínima é de 155 Mbps, utilizando as redes **ATM** (Assynchronous Transfer Mode) que provêm **Qualidade de Serviço**, garantindo ao usuário final, que por exemplo, uma música não será interrompida no meio do caminho como acontece hoje em dia na Internet tradicional. É diferente do conceito atual de *best-force* da Internet atual.

Para isso novos protocolos estão sendo desenvolvidos como é o caso do **IPv6** (Internet Protocol versão 6 ou IP New Generation). Este protocolo aumentará o número de bits de endereçamento, acabando com um problema sério hoje da escassez de números IPs. Provendo entre outras coisas suporte a *multicasting* (o envio de uma única mensagem para vários destinos), extremamente útil nas aplicações multimídia existentes e que ainda estão por vim.

Fatores como **segurança** também precisam ser melhorados. Quando o TCP/IP foi desenvolvido não se imagina a necessidade do uso de criptografia para garantir a segurança dos dados que trafegam em diferentes redes. Hoje em dia, este requisito é fundamental, haja visto a grande quantidade de aplicações de comércio eletrônico que existem na rede.

Há muito ainda por se desenvolver, mas parece-me que estamos no caminho certo...

## 12 Bibliografia

Comer, Douglas E., Internetworking with TCP/IP – Volume I – Principles, Protocols and Architecture – Third Edition, Prentice Hall, 1995

Cordeiro, Roberto Cesar, Apostila TCP/IP – Departamento de Informática Universidade Federal do Espírito Santo – UFES, 1996

Microsoft Press, Internetworking with TCP/IP on Windows NT 4.0, Microsoft Press, 1997

### **Sites Internet:**

- [www.dicas-l.unicamp.br](http://www.dicas-l.unicamp.br)
- [www.penta.ufrgs.br/hometcp.htm](http://www.penta.ufrgs.br/hometcp.htm)
- [www.geocities.com/SiliconValley/Horizon/4761/tcpip.html](http://www.geocities.com/SiliconValley/Horizon/4761/tcpip.html)