

## Protocolos de Roteamento

Em um sistema de troca de mensagens (comutação de mensagens), roteamento é o processo de escolha do caminho pelo qual iremos enviar os datagramas. Pode ser dividido em:

**Roteamento Direto:** comunicação entre dois *Hosts* alocados em uma mesma rede física. Possuem mesmo NETID.

**Roteamento Indireto:** conexão entre dois *Hosts* alocados em redes distintas. Possuem NETID diferentes. É necessário o uso de *Gateways* para efetuar o encaminhamento dos datagramas à rede destino.

Existem duas atividades que são básicas a um roteador:

### 1. A determinação das melhores rotas

Determinar a melhor rota é definir por qual enlace uma determinada mensagem deve ser enviada para chegar ao seu destino de forma segura e eficiente.

Para realizar esta função, o roteador utiliza dois conceitos importantes:

- o conceito de métrica e
- o conceito de tabelas de roteamento.

### 2. O transporte dos pacotes

Transportar os pacotes pela rede é uma função relativamente simples realizada pelos roteadores. Consiste em:

- verificar o endereço de rede para quem a mensagem está destinada,
- determinar se conhece este endereço,
- traduzir para um novo endereço físico e
- enviar o pacote.

## **Definição de Métrica**

- Métrica é o padrão de medida que é usado pelos algoritmos de roteamento para determinar o melhor caminho para um destino.
- Pode-se utilizar apenas um parâmetro ou vários parâmetros.
- A utilização de vários parâmetros permite uma melhor modelagem da métrica e uma decisão mais eficiente de qual é o melhor caminho.
- Alguns parâmetros utilizados:
  - Tamanho do caminho
  - Confiabilidade
  - Atraso
  - Largura de banda
  - Carga
  - Custo da comunicação

## **Definição de Tabela de Roteamento**

- Os roteadores constróem tabelas de roteamento para realizarem as suas tarefas.
- Estas tabelas de roteamento contêm entradas que relacionam um determinado destino com um enlace e uma métrica.
- Dependendo das implementações, as tabelas de roteamento podem apresentar mais dados, entretanto o destino, o enlace e a métrica são os dados essenciais.

## Exemplo de Tabelas de Roteamento

O algoritmo de roteamento IP utiliza tabelas de roteamento que contêm endereços de possíveis destinos e a maneira de alcançá-los, alocadas em *Hosts* e *Gateways*.

Temos então uma rede de comutação por mensagens com inteligência de roteamento descentralizada.

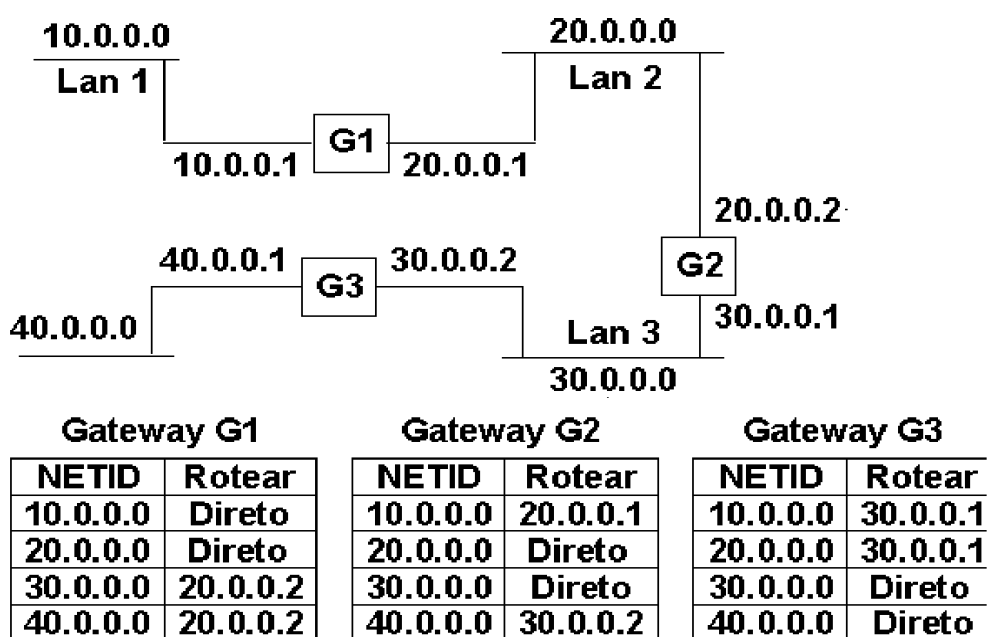
Uma tabela de roteamento IP contém dois campos:

- Campo N: especifica o NETID do endereçamento IP.
- Campo G: especifica o endereço IP completo do próximo *gateway* interconectado à rede descrita no campo N.

Os *Gateways* armazenam informações parciais em suas tabelas de roteamento devido a dois motivos:

- Necessidade de *buffers* muito grandes e alto tráfego para atualização de tabelas completas.
- Dificuldade de se ter disponível uma máquina rápida o suficiente para rotear todos os datagramas sem degradar o *throughput* e gerar congestionamento.

A Figura abaixo apresenta um exemplo de Tabela de Roteamento.



Exemplo de Tabela de Roteamento.

Com o crescimento da Internet, percebeu-se também a necessidade de adotar um processo automático de atualização de tabelas de roteamento.

Foi implementada então uma estrutura formada por *Core Gateways* e *Noncore Gateways*.

**Core Gateways** ou **Gateways Centrais** são roteadores de alta capacidade de processamento e armazenamento de dados, contendo tabelas com todos os destinos e rotas possíveis. Os *Core Gateways* estão ligados ao "backbone" principal da Internet. Não utilizam rotas *default*.

**Noncore Gateways** ou **Gateways Periféricos** são responsáveis pela manutenção das tabelas de rotas locais, possuindo rotas *default* para os *Gateways* Centrais. São responsáveis pelo roteamento de datagramas pertinentes a uma LAN de determinado "site".

## **Requisitos de um Roteador**

Para um roteador funcionar de forma adequada é necessário que desempenhe algumas tarefas.

- O roteador deve conhecer a topologia da sub-rede e escolher os caminhos adequados dentro da mesma.
- O roteador deve cuidar para que algumas rotas não sejam sobrecarregadas, enquanto outras fiquem sem uso.
- O roteador deve resolver os problemas que ocorrem quando a origem e o destino estão em redes diferentes.

## **Definição de Algoritmo de Roteamento**

- O algoritmo de roteamento é a parte do programa de nível de rede responsável por decidir para qual linha um pacote deve ser enviado a fim de chegar ao seu destino.
- Todos os roteadores executam um algoritmo de roteamento.
- Características desejadas em um algoritmo de roteamento:
  - Correção
  - Simplicidade
  - Robustez
  - Estabilidade
  - Consideração com o usuário
  - Eficiência global

## 1. Correção

- O algoritmo de roteamento tem de calcular rotas corretas para todos os destinos, não pode falhar para nenhum e não pode indicar uma rota inexistente.
- Esta é uma característica evidente que deve ser, ainda, complementada pela derivação da melhor rota.
- Não basta que o algoritmo descubra uma rota para um destino, é necessário que ele descubra a melhor rota possível.

## 2. Simplicidade

- O algoritmo de roteamento tem de ser eficiente sem sobrecarregar a máquina.
- administrador da rede precisa entender como o algoritmo é executado.

## 3. Estabilidade

- O algoritmo de roteamento tem de convergir rapidamente, e convergir para um estado correto.
- Por exemplo, quando acontece alguma modificação na topologia da rede, as tabelas de roteamento de alguns roteadores apresentarão uma informação errada. As tabelas deverão ser rapidamente corrigidas pelo algoritmo, por isso a necessidade de rápida convergência.
- No momento em que todos os roteadores da rede estiverem com suas tabelas certas, diz-se que o algoritmo convergiu.
- Quanto mais rápido for este processo, melhor.

## 4. Robustez

- Uma vez que a rede entre em operação, deve permanecer assim durante anos, sem que ocorram falhas de todo o sistema.
- Durante este período, ocorrerão falhas isoladas de *hardware* e *software* e a topologia da rede irá ser modificada diversas vezes.
- O algoritmo de roteamento deve ser capaz de resolver estas modificações sem requerer uma reinicialização.

## 5. Consideração com o usuário e eficiência global

- Estes dois requisitos são, de certa forma, contraditórios e precisa existir um compromisso entre eles.
- Às vezes, para melhorar o fluxo de dados na rede toda, seria necessário impedir o fluxo de dados entre duas máquinas específicas.
- Evidentemente, isto prejudicaria os usuários destas duas máquinas.
- Desta forma a melhora da eficiência global somente seria alcançada a partir da desconsideração de alguns usuários.
- Um algoritmo de roteamento deve melhorar a eficiência da rede sem deixar de levar em conta os diversos usuários.

## Dinâmica de um Algoritmo de Roteamento

- Extrai o endereço IP destino do datagrama (ID).
- Verifica se o NETID (IN) do endereço IP destino é igual ao endereço IP da rede diretamente conectada (Campo NETID).
- Se IN é igual ao endereço da rede conectada, efetua roteamento direto, encapsulando o datagrama no *frame* da rede física e enviando o mesmo ao destino. Se ID contém rota específica, roteia de acordo com a especificação da opção.
- Se IN não é igual ao endereço da rede, consulta tabela de roteamento e envia o datagrama ao *Gateway* mapeado.
- Se IN não consta da tabela do roteamento, verifica se existe a descrição de um *Gateway default* e envia o datagrama a este.

## Tipos de Algoritmo de Roteamento

- **Estáticos ou dinâmicos**
- **Estrutura plana ou hierárquica**
- **Intra-domínio ou inter-domínio**
- **Dois algoritmos são os mais comumente utilizados por protocolos de roteamento:**
  - **Vetor de distância**
  - **Estado do *link***



### **Algoritmos Estáticos**

- Um algoritmo de roteamento do tipo estático não baseia as suas decisões de roteamento em medidas ou estimativas de tráfego e em topologias correntes.
- As rotas são definidas anteriormente e carregadas no roteador no momento da inicialização da rede.

### **Algoritmos Dinâmicos**

- Um algoritmo de roteamento dinâmico procura mudar as suas decisões de roteamento de acordo com as mudanças de tráfego e de topologia.
- A tabela de roteamento vai sendo modificada com o passar do tempo.
- Algoritmos dinâmicos para roteamento apresentam uma flexibilidade e uma eficiência em condições adversas muito maiores do que os estáticos.

### **Algoritmos de Estrutura Plana**

- Neste tipo de algoritmo, todos os roteadores estão em um mesmo nível.
- As informações não são organizadas e distribuídas hierarquicamente.

### **Algoritmos de Estrutura Hierárquica**

- Neste tipo de algoritmo as informações de roteamento são organizadas hierarquicamente.
- Dependendo da hierarquia do roteador, a sua tabela de roteamento e a sua comunicação com outros roteadores serão diferentes.

### **Algoritmos Intra-domínio**

- Estes são algoritmos que são executados por roteadores internos a um determinado Sistema Autônomo (*Autonomous System*).
- Permitem que sejam definidas as rotas internas a uma determinada rede (pertencente a uma determinada corporação, por exemplo).

### **Algoritmos Inter-domínios**

- Estes são algoritmos que são executados por roteadores que estão nos limites dos domínios.
- Permitem a definição das rotas que são utilizadas para a comunicação com equipamentos de fora de um determinado Sistema Autônomo.

## Algoritmos

### Vetor de Distância e Estado do *Link*

Os protocolos de roteamento, de acordo com os algoritmos de roteamento, são divididos em dois tipos diferentes:

1. Os protocolos mais antigos, chamados de **protocolos de roteamento pelo vetor da distância**, enviam os pacotes na direção genérica de seu destino.
2. Os protocolos mais novos, chamados de **protocolos de roteamento pelo estado do *link***, mantêm uma tabela completa contendo os caminhos (vias) de cada roteador, e são capazes de fornecer instruções específicas de roteamento.

Algumas empresas se referem aos

- **protocolos de roteamento pelo vetor de distância como protocolos de "primeira geração"**
- e aos
- **protocolos de roteamento pelo estado do *link* como protocolos de "segunda geração".**

## **Algoritmo de Vetor de Distância** ***Distance Vector Algorithm***

- É bastante simples, baseia-se na distância entre dois pontos.
- Esta distância refere-se ao número de *Gateways* (ou número de roteadores) existentes na rota utilizada, sendo medida em *Hops*.
- Denominamos *Hop* a passagem de um datagrama por cada *gateway* (ou roteador).
- O roteador apresenta em sua tabela a rota para os roteadores vizinhos.
- No início, a tabela de roteamento de um *gateway* apresenta apenas os endereços das redes diretamente conectadas a ele, sendo a distância igual a zero.
- Em intervalos de tempo regulares o roteador envia cópias de toda a sua tabela de rotas para, e somente para, os seus vizinhos (os *gateways* que são alcançados diretamente), sendo feito assim a atualização das tabelas.
- Após algum tempo os diversos roteadores da rede convergem (ficam com as suas tabelas completas e atualizadas).
- As tabelas apresentam o endereço destino, a métrica, e o próximo roteador para onde a mensagem deve ser enviada.
- Exige menos recursos de memória e processamento do que o algoritmo de Estado do Enlace.
- Apresenta convergência mais lenta e alguns problemas enquanto o algoritmo não se estabilizou.
- A grande desvantagem deste algoritmo está no fato de não ser adaptado para redes extensas, pois quando cresce o número de sub-redes, cresce conseqüentemente o tamanho das tabelas de roteamento, aumentando o tráfego gerado exclusivamente para a manutenção das próprias tabelas.

## Exemplos de Tabelas de Roteamentos Algoritmo Vetor de Distância

### TR do Gateway G1

Destino	Distância	Rota
Net 1	0	Direta
Net 2	3	G7
Net 5	6	G3
Net 17	9	G13
Net 48	7	G5

### TR do Gateway G3

Destino	Distância
Net 1	1
Net 2	1
Net 6	4
Net 17	13
Net 48	11

### TR do Gateway G1 Atualizada

Destino	Distância	Rota
Net 1	0	Direta
Net 2	2	G3
Net 5	6	G3
Net 6	5	G3
Net 17	9	G13
Net 48	7	G5

## **Algoritmo de Estado do Enlace** ***Link State Algorithm***

- Com este algoritmo, ao invés de uma tabela, cada *Gateway* mantém um mapa da topologia da rede.
- A tarefa de um *gateway* é:
  - testar inicialmente a possibilidade de comunicação com os *gateways* diretamente conectados a ele (descobrir quem são os vizinhos e qual o estado do enlace dos vizinhos (*up* ou *down*)),
  - medir os custos associados aos diversos enlaces que possui e
  - divulgar as informações para todos os outros *gateways* (roteadores) da rede.
- Desta forma, o algoritmo constrói um mapa completo da rede.
- O próximo passo é constrói o melhor caminho para cada roteador da rede, utilizando o algoritmo de Dijkstra.
- A grande vantagem do algoritmo reside no fato das mensagens enviadas serem pequenas, o que evita um volume de tráfego desnecessário na rede.

## **Protocolos de Roteamento**

A função dos protocolos de roteamento é construir as tabelas de roteamento completas nos diversos roteadores de uma rede através da troca de mensagens entre eles.

### **Tipos de Protocolos de Roteamento**

- *igp (interior gateway protocol)* - São utilizados para realizar o roteamento dentro de um Sistema Autônomo.
- *egp (exterior gateway protocol)* - São utilizados para realizar o roteamento entre Sistemas Autônomos diferentes.

#### **Protocolos do tipo igp (*interior gateway protocol*)**

- *RIP (Routing Information Protocol)*
- *IGRP (Interior Gateway Routing Protocol)*
- *Enhanced IGRP*
- *OSPF (Open Shortest Path First)*
- *IS-IS (Intermediate System-to-Intermediate System)*

#### **Protocolos do tipo egp (*exterior gateway protocol*)**

- *EGP (Exterior Gateway Protocol)*
- *BGP (Border Gateway Protocol)*

## **Exemplos de Protocolos Vetor de Distância:**

### **1. RIP (*Routing Information Protocol*)**

#### **Características básicas do protocolo RIP:**

- Projetado como um protocolo intra-domínio (igp).
- Utiliza um algoritmo do tipo Vetor de Distância.
- A métrica utilizada é a distância da origem até o destino em número de enlaces que devem ser percorridos.
- Não permite o balanceamento do tráfego.
- A rota inatingível apresenta uma métrica igual a 16.
- Realiza atualizações a cada 30 segundos.

#### **Informações guardadas na tabela de roteamento do RIP:**

- Endereço de destino
- Endereço do próximo roteador
- Interface do *host* a ser utilizada
- Métrica da rota
- *Flags* e *timers* que controlam tempos de atualização

#### **Dados transmitidos nas mensagens de atualização do RIP:**

- Comando (*Request* ou *Response*)
- Identificador da família de endereços
- Endereço destino
- Métrica



## Processamento do RIP

- **Atualização da tabela de roteamento a cada chegada de um *Response***

As atualizações sempre chegam por mensagens designadas como *Response*.

Cada vez que chega uma atualização, o roteador busca na tabela a entrada correspondente e modifica, se as seguintes condições forem satisfeitas:

1. Se a rota não existe: acrescenta 1 à métrica recebida e coloca a rota na tabela.
2. Se a rota já existe na tabela e apresenta métrica maior: substitui a rota atual pela que chegou com métrica menor.

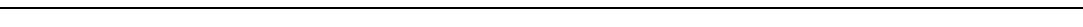
- **Características de estabilidade**

Para que o algoritmo de Vetor de Distância utilizado no RIP funcione de forma eficiente, os mecanismos de estabilidade relacionados abaixo são utilizados no RIP:

1. *Hop-count limit*
2. *Hold-down*
3. *Split horizon*
4. *Poison reverse updates*
5. *Triggered updates*

### ● Características gerais

1. O endereço *default* é referenciado pela destino 0.0.0.0.
2. A mensagem de *Request* é utilizada para que um roteador solicite a tabela de roteamento de um vizinho, ou apenas uma rota para um determinado destino.
3. Apresenta implementação simples, uma vez que utiliza um algoritmo simples e apenas duas mensagens.
4. Apresenta uma convergência lenta.
5. Os estados intermediários, isto é, até os roteadores convergirem, podem apresentar laços.



Um dos primeiros e mais difundidos protocolos IGP de roteamento, ainda encontrado em muitos sistemas de interligação de redes.

É também conhecido pelo nome do programa que o implementa: *Routed*.

Utiliza a técnica "*Vector Distance*" para atualização de tabelas de roteamento.

O roteamento pelo vetor da distância usa um algoritmo denominado Bellman-Ford que divide o *link* entre as redes em áreas lógicas.

Quando um roteador recebe um quadro, ele lê o endereço contido no pacote dentro do quadro e envia o pacote em direção à área lógica do destino baseando-se no menor número de pontos intermediários, ou *hops*.

O RIP, desenvolvido no início da década de 1980, foi definido originalmente como parte dos protocolos *Xerox Network Services* (XNS) e está definido como parte do conjunto de protocolos TCP/IP, identificado como RFC 1058. Também são encontrados pacotes transportando informações subordinadas a esse padrão como parte dos protocolos IPX da Novell.

O RIP usa algoritmos de vetor de distância que informam o caminho mais curto entre dois pontos de uma rede em termos do número de pontos intermediários, ou *hops*, entre esses pontos. Assim, quanto menor o n<sup>o</sup> de pontos intermediários, mais eficiente o caminho.

Embora esse esquema funcione com redes menores, o RIP se torna menos prático em sistemas complexos de redes.

Os *hosts*, assim que recebem mensagens RIP, efetuam a atualização de suas tabelas.

O protocolo RIP difere dos demais que utilizam o algoritmo "*Vector Distance*" na sua contagem de *hops*.

Para o RIP, uma rede ligada diretamente a um *gateway* está à distância de "1" *hop* do mesmo e não à distância "0" como é adotada nos demais protocolos.

Um *gateway* só deve efetuar uma alteração de rota quando aparecer uma rota de custo menor.

O RIP não permite que o número de pontos intermediários seja maior do que 16 devido à morosidade da divulgação de rotas, portanto, os roteadores não poderão transferir pacotes para outros segmentos da rede que fiquem a mais de 16 roteadores de distância. Quando uma rota é divulgada com valor de 16 *hops*, esta é considerada inatingível.

O RIP não é adaptativo e, embora seja distribuído, os roteadores que o utilizam transferem as tabelas de endereços inteiras entre si a cada 30 segundos, mencionando as redes e suas respectivas distâncias (em *hops*), o que gera intenso *overhead*.

Essas atualizações freqüentes podem, por si só, causar problemas na rede, pois se um ou mais roteadores não captarem a mensagem de atualização, suas tabelas de roteamento poderão se tornar diferentes, prejudicando a eficiência do sistema.

A perda de eficiência resulta em mais atualizações perdidas, o que, por sua vez, agrava ainda mais o problema.

A capacidade de retomar o tráfego normal da rede rapidamente, mesmo depois de uma perturbação grave no funcionamento dos circuitos da rede, é conhecida como "convergência".

O RIP é considerado um protocolo de pouca convergência e que pode gerar *loops*.

Em razão de todos esses fatores, o uso do RIP não é aconselhado para redes complexas, porém poderá ser utilizado em redes menores, onde as suas características indesejáveis não se manifestam.

## Técnicas utilizadas para contornar o problema dos *loops* ou da convergência lenta do RIP:

### ***Split Horizon:***

Um *gateway* registra a interface sobre a qual ele recebe a informação sobre uma dada rota e não propaga a sua informação sobre esta rota de volta na mesma interface.

As rotas que apontam para o vizinho do *gateway* são omitidas.

### ***Hold Down:***

Quando uma dada rota é removida, nenhuma outra rota nova deve ser aceita para o mesmo destino durante um período de tempo (60 s).

Desvantagem: durante este período, não são utilizadas rotas nem caminhos alternativos.

### ***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*.

### ***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.

## Formato das Mensagens RIP

<b>0</b>	<b>8</b>	<b>16</b>	<b>31</b>
<b>Command</b>	<b>Version (1)</b>	<b>Must be zero</b>	
<b>Family of Net 1</b>		<b>Must be zero</b>	
<b>IP Address of Net 1</b>			
<b>Must be zero</b>			
<b>Must be zero</b>			
<b>Distance to Net 1</b>			
<b>Family of Net 1</b>		<b>Must be zero</b>	
<b>IP Address of Net 1</b>			
<b>Must be zero</b>			
<b>Must be zero</b>			
<b>Distance to Net 1</b>			
...			

Utilizada para divulgar ou solicitar divulgação de rotas, os campos assumem os seguintes significados:

**COMMAND:** Especifica a finalidade da mensagem. Pode ser:  
 Request - requisição para informação de roteamento total ou parcial.

Response - resposta contendo os pares rede-distância para o emissor.

Traceon - obsoleto.

Traceoff - obsoleto.

Reserved - reservado.

**VERSION:** número da versão atual (1).

**FAMILY OF NET X:** identifica o protocolo (ou família de protocolos) que está sendo utilizado. Para o protocolo IP = 2.

**IP ADDRESS OF NET X:** endereço IP da rede X.

**DISTANCE TO NET X:** as distâncias assumem valores de 1 a 15. Para rede inatingível, o valor é 16.

## **Exemplos de Protocolos Vetor de Distância:**

### **2. GGP - Gateway to Gateway Protocol**

É o protocolo utilizado pelos primeiros *Core Gateways* para propagação de rotas.

Utiliza o algoritmo "*Vector Distance*", sendo encapsulado num datagrama IP.

Seu uso é restrito aos *Core Gateways*, e nenhum outro *gateway* interfere nesta comunicação.

São utilizados basicamente três formatos de mensagens GGP:

- Formato 1 GGP
- Formato 2 GGP
- Formato 3 - Teste de Comunicação.

## 1. Formato 1 GGP

0	8	16	31
Type(12)	Unused	Unused	
Sequence Number		Unused	
Update	Num. Dist.	Unused	
Distance D <sub>1</sub>	Num. Nt. at D <sub>1</sub>	Unused	
First Net at Distance D <sub>1</sub>			
Second Net at Distance D <sub>1</sub>			
...			
Last Net at Distance D <sub>1</sub>			
Distance D <sub>2</sub>	Num. Nt. at D <sub>2</sub>	Unused	
First Net at Distance D <sub>2</sub>			
Second Net at Distance D <sub>2</sub>			
...			

- TYPE: o valor 12 identifica um envio de Tabelas de Roteamento para atualização.
- SEQUENCE NUMBER: utilizado para validação e controle de fluxo das mensagens GGP. Cada mensagem carrega seu respectivo número de seqüência, personalizando assim o segmento.
- UPDATE: utilizado para que o originador solicite ou não uma mensagem de atualização do receptor.
- NUM DISTANCES: utilizado para identificar o número de distâncias existentes no segmento. (Este campo identifica quantas distâncias são utilizadas, visto que o segmento GGP envia as atualizações em pares NETID IP + distância correspondente).
- DISTANCE D<sub>1</sub>: identifica qual a distância (em *hops*) deste grupo de endereços.
- NUM. NETS AT D<sub>1</sub>: utilizado junto a determinado número de distância, especifica quantos endereços são mencionados para a distância em questão.
- FIRST NET AT DISTANCE D<sub>1</sub>: campos reservados para a inclusão dos endereços IP (NETID) mais a referida distância.



## 2. Formato 2 GGP

Enviada imediatamente após um *gateway* receber uma mensagem de atualização GGP.

Pode ser positiva ou negativa.

	<b>8</b>	<b>16</b>	<b>31</b>
<b>Type(2 or 10)</b>	<b>Unused</b>	<b>Sequence Number</b>	

- TYPE com valor "2" indica recepção positiva, e o campo SEQUENCE NUMBER carrega o número de seqüência do referido segmento de atualização.
- TYPE com valor "10" indica que ocorreu algum erro na recepção do segmento de atualização, e no campo SEQUENCE NUMBER vai o número da seqüência do último segmento recebido sem erros.

## 3. Formato 3 - Teste de Comunicação

Utilizada para que um *gateway* verifique se um determinado "parceiro" está respondendo.

<b>0</b>	<b>8</b>	<b>31</b>
<b>Type(0 or 8)</b>	<b>Unused</b>	

- TYPE com valor "8" identifica uma "ECHO REQUEST".
- TYPE com valor "0" identifica uma "ECHO REPLY".

## Exemplos de Protocolos Vetor de Distância:

### 3. HELLO

O algoritmo *Vector Distance* tem um emprego diferente neste protocolo, que mede as distâncias entre redes por tempo e não por saltos.

Possui duas funções básicas: manter o sincronismo dos relógios de todos os *gateways* envolvidos e divulgar alcance por tempo referente a cada rede.

#### Fomato das Mensagens

<b>0</b>	<b>16</b>	<b>24</b>	<b>31</b>
<b>Checksum</b>		<b>Date</b>	
<b>Time</b>			
<b>Timestamp</b>		<b>Local Entry</b>	<b>#Hosts</b>
<b>Delay 1</b>		<b>Offset 1</b>	
<b>Delay 2</b>		<b>Offset 2</b>	
- - -			
<b>Delay n</b>		<b>Offset n</b>	

Os campos do protocolo HELLO são os seguintes:

CHECKSUM: verificador de erros da mensagem.

DATE: data do originador da mensagem.

TIME: horário do host originador no momento de envio.

TIMESTAMP: tempo base para processamento (data e hora).

LOCAL ENTRY: especificação da rede local do originador.

#HOSTS: quantidade de hosts da mensagem.

DELAY: atraso para alcançar determinados hosts, especificado pelo destino.

OFFSET: tempo estimado pelo originador para alcançar o destino.

O HELLO pode ser utilizado juntamente com outros protocolos, como por exemplo RIP.

## Exemplos de Protocolos Estado do *Link* :

### 1. *Open Shortest Path First (OSPF)*

Os protocolos que pertencem à categoria de "roteamento pelo estado do *link*" são protocolos IGP que fornecem mais informações do que o RIP.

Os protocolos de estado do *link* exigem o uso de um processador mais poderoso para analisar as variáveis e tomar decisões, porém isso não chega a ser um inconveniente já que os microprocessadores modernos oferecem a capacidade necessária de tratamento de dados a preços razoáveis.

Os protocolos de estado do *link* se baseiam num conjunto de regras lógicas denominadas **algoritmo de Dijkstra** para identificar o caminho mais curto entre a origem e o destino.

O protocolo de estado do *link* mais usado nas redes TCP/IP - que substitui o RIP nas redes mais complexas - é conhecido como *Open Shortest Path First (OSPF)*.

O OSPF, desenvolvido em conjunto pela Proteon e por instituições acadêmicas, é definido pelo RFC 1131 como parte do conjunto de protocolos TCP/IP. Várias empresas fornecem produtos com o protocolo OSPF.

Os roteadores que utilizam o OSPF conseguem tomar decisões com base nos seguintes parâmetros:

- carga de tráfego,
- *throughput*,
- custo do circuito,
- prioridade de serviço atribuída aos pacotes que se originam ou se destinam a um ponto específico.

Os roteadores só transferem as tabelas de endereços quando necessário, e fazem a transferência com mais eficácia do que os roteadores que utilizam o RIP, tendo, portanto, uma excelente convergência.

A divisão de um sistema de interligação de redes em áreas de roteamento menores sob o protocolo OSPF não apenas reduz a quantidade de tráfego no *backbone* como também melhora o nível de segurança.

Uma vez que a topologia da rede inteira não é transmitida a todos os roteadores, certos segmentos da rede podem permanecer privados a menos que sejam endereçados especificamente. Isso os torna mais resistentes a tentativas eventuais de intrusão ou sabotagem.

O particionamento de grandes redes em áreas de roteamento também melhora a *performance* global, reduzindo o tráfego entre seus nós.

Um recurso do OSPF denominado roteamento pelo tipo de serviço define oito tipos de serviço. Cada tipo pode corresponder a um caminho de roteamento diferente.

É um padrão aberto, largamente utilizado e de alta performance.

Todas as mensagens OSPF são inicializadas com um cabeçalho padrão:

O Cabeçalho Padrão é formado pelos seguintes campos:

VERSION: utilizado para especificação da versão atual do OSPF.

TYPE: especifica o tipo da mensagem.

MESSAGE LENGHT: identifica o tamanho total da mensagem OSPF corrente.

SOURCE GATEWAY IP ADDRESS: utilizado para identificação do *gateway* gerador da mensagem OSPF, onde é introduzido o seu respectivo 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: utilizado para colocação de uma senha de validação.

## OSPF - Formato das Mensagens

### Formato OSPF Type2 (*Database Description*)

Os *gateways* OSPF trocam mensagens sobre a base de dados da topologia da rede em questão.

O *gateway* mestre envia mensagem contendo informações da base de dados e o escravo confirma.

0	16	29	31
<b>OSPF Header with Type = 2</b>			
<b>Must be zero</b>			<b>I   M   S</b>
<b>Database Sequence Number</b>			
<b>Link Type</b>			
<b>Link ID</b>			
<b>Advertising Gateway</b>			
<b>Link Sequence Number</b>			
<b>Link Checksum</b>		<b>Link Age</b>	
...			

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.

0	16	29	31
<b>OSPF Header with Type = 2</b>			
Must be zero			I M S
Database Sequence Number			
Link Type			
Link ID			
Advertising Gateway			
Link Sequence Number			
Link Checksum		Link Age	
...			

**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 *database* 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.

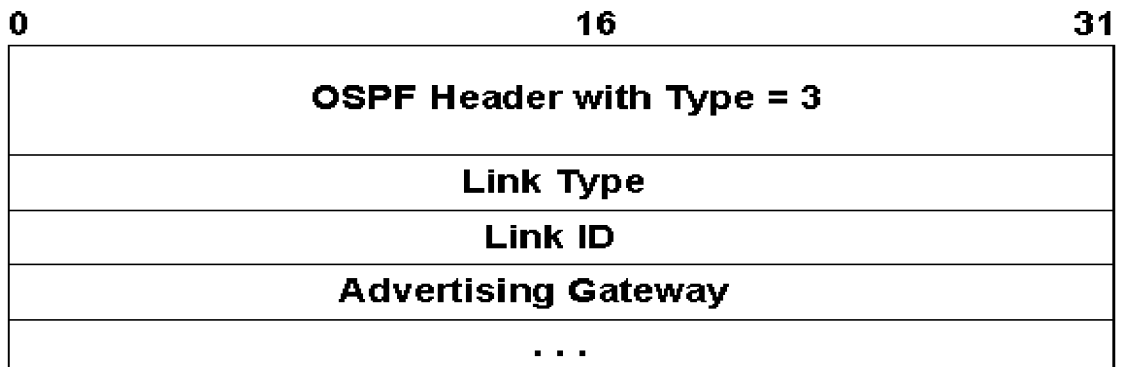
## OSPF - Formato das Mensagens

### Formato OSPF Type3 (*Link Status Request*)

Após a troca de mensagens tipo 2, os *gateways* podem descobrir se existem partes de seus bancos de dados desatualizadas.

Basta que um *gateway* envie 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 atualizada possível.





## OSPF - Formato das Mensagens

### Formato OSPF Type4 (*Link Status Update*)

Um *gateway* recebe uma mensagem "*Link Status Request*", envia uma resposta "*Link Status Update*", informando ao receptor todos os *links* que estão diretamente conectados a ele.

<b>0</b>	<b>16</b>	<b>31</b>
<b>OSPF Header with Type = 4</b>		
<b>Number of Link Status Advertisements</b>		
<b>Link Status Advertisement<sub>1</sub></b>		
. . .		
<b>Link Status Advertisement<sub>n</sub></b>		

Campo Link Status Advertisement:

<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>	

O campo LINK TYPE especificará ao receptor quais os *links* que estão dentro do mesmo "*site*" e quais são extremos.

## OSPF - Formato das Mensagens

### Formato OSPF Type5 (*Link Status Acknowledgement*)

Todo *link* divulgado tem uma respectiva confirmação.

<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>		

...

O número de seqüência garante o sincronismo entre mensagens enviadas e as respectivas confirmações.

## Exemplos de Protocolos Estado do *Link* :

### 2. PROTOCOLO IS-IS

O protocolo de roteamento OSPF faz parte do conjunto de protocolos TCP/IP.

Este conjunto de protocolos criado especificamente segundo o modelo ISO/OSI contém outro protocolo IGP de roteamento de estado do *link* denominado **IS-IS (*Intermediate-System-to-Intermediate-System*)** ou Sistema Intermediário para Sistema Intermediário), definido no documento 10589 da ISO.

O **Dual IS-IS**, uma versão especial do IS-IS que transporta pacotes compatíveis com os padrões TCP/IP e OSI, é definido pelo RFC 1195 e pelo documento 9542 da ISO.

Os roteadores IS-IS usam os mesmos tipos de critérios do OSPF na avaliação de pacotes e no seu envio entre os circuitos da rede.

O IS-IS oferece as mesmas vantagens do OSPF mas é projetado para permitir maior interoperabilidade entre equipamentos de fabricantes diferentes, por estar subordinado ao modelo *Open Systems Interconnection* (OSI).

A *Digital Equipment Corp.* incorporou o IS-IS a todos os roteadores do sistema de redes *DECnet Phase V* com a finalidade de fazer roteamento de pacotes OSI, IP e DECnet. Outras empresas, inclusive a Cisco Systems e a 3Com, oferecem roteadores IS-IS com recursos e graus de compatibilidade variados.

## **Exemplo de Protocolo de Roteamento EGP : *Exterior Gateway Protocol (RFC 904)***

O protocolo EGP possui três características principais:

- Suporta mecanismo de aquisição de vizinho.
- Faz testes contínuos para ver se os vizinhos estão respondendo.
- Divulga informação entre vizinhos utilizando mensagens de atualização de rotas.

Com estas três funções básicas, o protocolo EGP trabalha com os seguintes tipos de mensagens:

- 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: informação de roteamento.
- ERROR: resposta a mensagens incorretas.

## Cabeçalho Padrão do EGP

<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>			

- ◆ **VERSION:** utilizado para identificar a versão corrente do *software*.
- ◆ **TYPE:** utilizado junto ao campo **CODE**, indica o tipo da mensagem.
- ◆ **CODE:** identifica o código da mensagem (subtipo).
- ◆ **STATUS:** contém informações adicionais de estado dependente das mensagens.
- ◆ **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 comunicação com um "Vizinho Exterior" e incrementado a cada mensagem enviada.

## Mensagem de Aquisição de Vizinho

Um "Gateway Exterior" envia mensagens de "Neighbor Acquisition" para estabelecer comunicação com um "Vizinho Exterior".

<b>0</b>	<b>8</b>	<b>16</b>	<b>31</b>
<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>	

- Os campos CODE e STATUS podem assumir os valores:

Valor	CODE	STATUS
0	comando de requisição	não especificado
1	resposta de confirmação	modo ativo
2	resposta de rejeição	modo passivo
3	comando de terminação	recursos insuficientes
4	resposta de encerramento	proibido administrativamente
5	-	desativando
6	-	problema de parâmetro
7	-	violação de protocolo

- HELLO INTERVAL: utilizado para implementar um *timer* a ser utilizado para o envio de mensagens periódicas com o intuito de testar a resposta do vizinho.
- POLL INTERVAL: utilizado para estipular um *timer* para o envio de atualizações de rotas.

## Teste Contínuo de Funcionamento de Vizinho

0	8	16	31
<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>	

Os campos CODE e STATUS assumem os valores:

Valor	CODE	STATUS
0	comando HELLO	indeterminado
1	resposta I-Heard-You	estado UP
2	-	estado DOWN

## Mensagem POLL REQUEST

Quando um "Gateway Exterior" efetua aquisição de um determinado "vizinho", significa que este deseja trocar informações sobre roteamento.

As informações de roteamento divulgadas são pertinentes apenas ao Sistema Autônomo a que pertence o referido gateway.

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

O campo **IP SOURCE NETWORK** contém o número IP da rede sobre a qual estão sendo solicitadas as informações de alcançabilidade.

## Mensagem de Atualização de Rotas

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
...			
Last Net at Last Distance for Gateway N			Unused

- #INT. GWYS: identifica o número total de gateways interiores.
- #EXT. GWYS: identifica o número total de gateways exteriores.
- IP SOURCE NETWORK: número IP da rede sobre a qual estão sendo solicitadas as informações de alcançabilidade.
- GATEWAY "I" IP ADDRESS: contém o endereço do gateway destino.
- #DISTANCES: utilizado para identificar o número de distâncias existentes no segmento.
- 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) e a respectiva distância.