

Roteamento

**Roteamento estático, roteamento dinâmico e métricas, protocolos de roteamento
(Vetor de Distância, Estado de Enlace), protocolos RIP e OSPF**

**Versão 1.1
Novembro de 2018**

Prof. Jairo

jairo@uni9.pro.br
professor@jairo.pro.br

<http://www.jairo.pro.br/>

Sumário

1 – Tabela de rotas e rotas conectadas, estáticas e dinâmicas.....	3
1.1 – Tabela de rotas.....	3
1.2 – Rotas conectadas.....	4
1.3 – Rotas estáticas.....	5
1.4 – Rotas dinâmicas.....	6
2 – Roteamento Estático.....	7
3 – Roteamento Dinâmico.....	8
3.1 – Métricas.....	9
3.2 – Sistema Autônomo (<i>Autonomous System - AS</i>).....	9
3.3 – Protocolos de Roteamento.....	10
3.3.1 – Protocolo de Roteamento Interno.....	11
3.3.2 – Protocolo de Roteamento Externo.....	11
3.3.3 – Classificação dos Protocolos de Roteamento.....	12
3.3.4 – Algoritmos de Roteamento.....	13
3.3.5 – Heurísticas em que se baseiam os Algoritmos de Roteamento.....	14
3.4 – Distância Administrativa.....	15
3.5 – Vetor de Distância (<i>Distance Vector</i>).....	17
3.6 – Estado de Enlace (<i>Link State</i>).....	21
3.7 – Protocolos RIP e OSPF.....	22
3.7.1 – RIP.....	23
3.7.2 – OSPF.....	24

1 – Tabela de rotas e rotas conectadas, estáticas e dinâmicas

Nos roteadores, roteamento é o processo de encaminhamento de um pacote de dados entre diferentes redes.

A função primária de um roteador é encaminhar os dados (pacotes) para a rede de destino, que está no endereço IP de destino do pacote. Mas para isso, o roteador precisa fazer uso das informações armazenadas na sua **tabela de rotas**.

1.1 – Tabela de rotas

Tabela de Rotas, Tabela de Roteamento ou Base de Informação de Roteamento (*Routing Information Base – RIB*) é uma tabela de dados que lista as rotas para destinos particulares, incluindo as métricas associadas a estas rotas. A tabela de dados é armazenada no roteador e contém informações a respeito da topologia da rede nas redondezas do equipamento.

A tabela de rotas é um arquivo de dados na memória do roteador, e que é usada para guardar informações sobre redes tanto diretamente conectadas ao equipamento quanto redes remotas.

Toda vez que um nó na rede precisa enviar dados para outro nó, este precisa saber antes para onde enviar estes dados. Na maior parte dos casos, o nó que envia os dados não está diretamente conectado no nó que recebe, então este encaminhamento ocorre via outros nós, que encaminham os dados até o destinatário, isso ao longo de uma rota (um caminho) apropriada.

Deste modo, *hop-by-hop* (salto a salto, de nó para nó), os dados vão sendo encaminhados para o próximo dispositivo ao longo do caminho, desde a origem até o destinatário. *Hop-by-hop* é uma característica fundamental da camada 3.

Não são apenas os roteadores que têm uma tabela de rotas, mas de um modo geral todo nó na rede precisa ter. Por exemplo, num computador pessoal, a tabela é simples e contém algumas poucas rotas, como pode ser visto na figura 1, abaixo.

```
shell$ route -n
Tabela de Roteamento IP do Kernel
Destino      Roteador      MáscaraGen.  Opções  Métrica    Ref    Uso      Iface
0.0.0.0      192.168.0.1  0.0.0.0      UG      100        0      0       eth0
192.168.0.0  0.0.0.0      255.255.255.0 U       100        0      0       eth0
192.168.122.0 0.0.0.0      255.255.255.0 U       0          0      0       virbr0
```

Fig 1: exemplo de tabela de rota de computador pessoal

Na figura 1 acima, temos:

Destino: a rede ou máquina de destino;

Roteador: o endereço IP do roteador;

Máscara Genérica: máscara para a rede destino, onde 0.0.0.0 é a rota *default*;

Opções: U para *up* (rotas em equipamento ativos), G para indicar que a rota usa roteador;

Métrica: é a distância (custo) até o destino (geralmente contada em *hops*);

Ref: número de referências à rota;

Uso: contagem do número de procuras pela rota;

Interface: interface através da qual os pacotes serão enviados.

De um modo geral, em qualquer tabela de rotas precisa haver pelo menos três campos de informação:

- i) A rede de destino;
- ii) O custo (métrica) para o caminho através do qual o pacote possa ser enviado;
- iii) O próximo salto (*hop*) ou *gateway*, que é o endereço do próximo nó para o qual o pacote possa ser enviado no seu caminho para o endereço de destino.

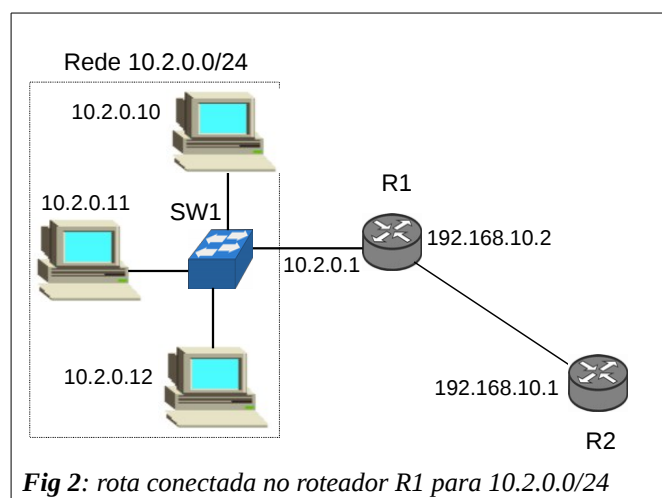
Porém, nas arquiteturas de roteadores modernas, as tabelas de roteamento não são usadas diretamente para o encaminhamento de pacotes, mas são usadas para gerar informações que vão para uma tabela menor, que é a **tabela de encaminhamento** (*forwarding table*). As tabelas de encaminhamento contém apenas as rotas as quais são consideradas preferidas pelo algoritmo de roteamento, são normalmente compactadas, otimizadas para armazenamento e facilidade na busca de rotas. Apenas as rotas consideradas “as melhores” da tabela de roteamento é que vão para a tabela de encaminhamento.

No roteador, as rotas nas tabelas de rotas podem ser rotas **conectadas diretamente**, rotas **estáticas** ou rotas **dinâmicas**.

1.2 – Rotas conectadas

As rotas são diretamente conectadas quando a rede está diretamente conectada a uma interface de rede do roteador. Para isto, nesta interface de rede é necessário haver um endereço IP e uma máscara de rede configurado na mesma rede para o qual vai rotear estes pacotes de dados, portanto esta interface do roteador se torna um nó (*host*) conectado a esta rede.

Na figura 2, ao lado, é mostrado um exemplo de topologia com o roteador **R1**



diretamente conectado à rede 10.2.0.0/24.

Neste caso, o roteador vai adicionar na tabela de roteamento uma rota **diretamente conectada** para esta rede, que inclui o tipo e a interface de rede em questão, e vai ser capaz de rotear todos os pacotes destinados aos *hosts* da rede 10.2.0.0/24.

Para poder adicionar a rota diretamente conectada é necessário que as duas interfaces de rede estejam ativas (*up*), e que seja adicionado na interface de rede do roteador o endereço IP e a máscara.

Na figura 3, abaixo, é mostrada esta rota no roteador **R1**.

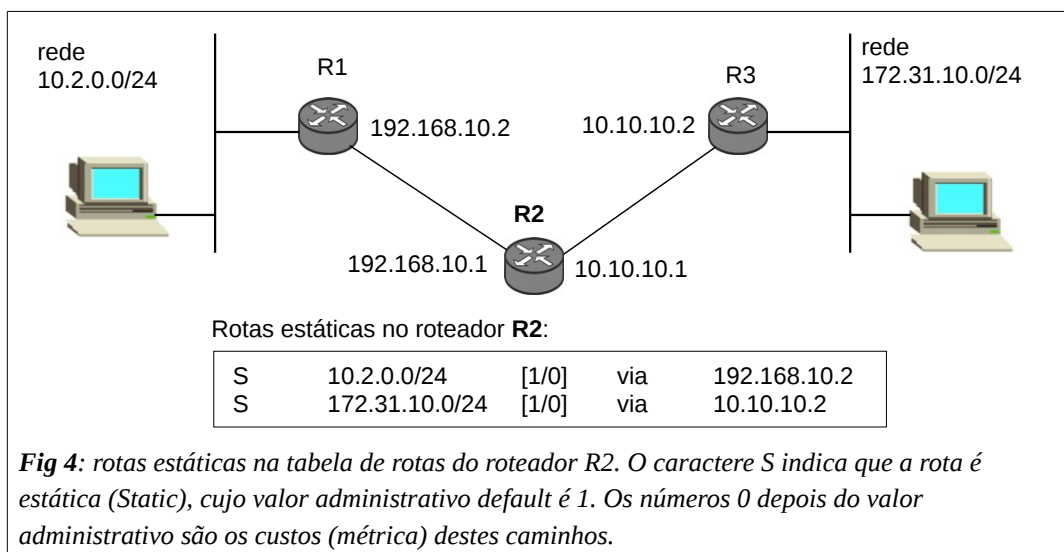
```
R1# show ip route connected
C    10.2.0.0/24 is directly connected, FastEthernet0/0
```

Fig 3: conteúdo da tabela de rotas, mostrando apenas a rota conectada no roteador R1, na interface de rede FastEthernet0/0. O caractere C indica que a rota é diretamente conectada.

1.3 – Rotas estáticas

As rotas são estáticas quando o administrador da rede manualmente adiciona estas rotas, que vão para a tabela de rotas do roteador.

As rotas estáticas definem explicitamente caminhos entre dois roteadores, e tem como principal vantagem consumir menos banda de dados que o roteamento dinâmico. Além disso, quando comparado com o roteamento dinâmico, não consome ciclos de CPU do roteador para calcular e analisar as atualizações de rotas.



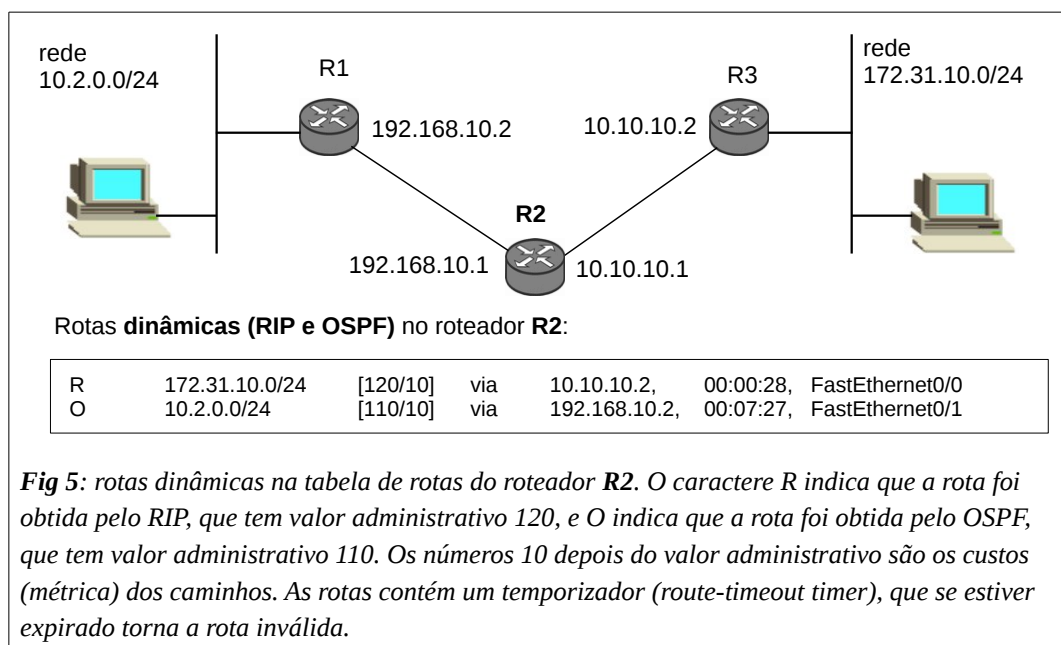
Rotas estáticas são usadas em ambientes onde o desenho da rede é simples e onde o tráfego de rede é previsível. Rotas estáticas não podem ser usadas em redes grandes, sujeitas a mudanças constantes, pois o roteamento estático não é capaz de acompanhar estas mudanças. Muitas redes usam roteamento dinâmico para comunicação entre roteadores.

As rotas estáticas também são úteis para especificar um *gateway* (porta de saída) como último recurso, que é um roteador *default* (padrão) para o qual serão enviados todos os pacotes que não encontrarem uma rota específica.

Quando necessário, as rotas estáticas podem complementar as rotas dinâmicas. Rotas estáticas podem ser redistribuídas pelos algoritmos de roteamento dinâmicos, mas não é possível redistribuir informações de rotas calculadas por algoritmos de roteamento para tabelas de roteamento estático.

1.4 – Rotas dinâmicas

As rotas são dinâmicas quando o roteador, pelo uso de protocolos de roteamento, pode aprender as rotas. Neste caso, é necessário o uso de protocolos de roteamento nos roteadores, para que eles troquem informações de roteamento entre si, e usem estas informações para construir dinamicamente as suas tabelas de rotas.



No roteador, é necessário apenas definir quais rotas precisam ser anunciadas, que os protocolos de roteamento vão se encarregar de construir a tabela de rotas. Para anunciar as rotas (*route advertisements*), basta habilitar a interface de rede do roteador e incluir um endereço IP e

máscara de rede, que então os protocolos de roteamento vão usar esta interface para comunicação com o outro roteador, e assim enviar e receber informações sobre rotas.

Nos roteadores, de um modo geral, o roteamento de pacotes IP é habilitado por padrão (*default*). Já os protocolos de roteamento estão desabilitados por *default*, então se for necessário usar os protocolos RIP ou OSPF (por exemplo), estes terão que ser habilitados no equipamento.

2 – Roteamento Estático

Os roteadores encaminham os pacotes de dados tanto pelo uso de informação de entradas fixas da tabela de rotas (**roteamento estático**), quanto pelas informações de rotas calculadas com o uso de algoritmos de roteamento dinâmico (**roteamento dinâmico**).

Toda vez que um roteador recebe um pacote de dados, ele inspeciona o endereço IP de destino no pacote e, através da sua tabela de rotas, tenta encontrar uma possível rota para o pacote. Caso a rede de destino não seja encontrada na sua tabela de rotas, o pacote é descartado.

O roteamento estático ocorre a partir de rotas fixas, normalmente adicionadas manualmente pelo administrador da rede. Estas rotas não podem ser atualizadas automaticamente, portanto precisam ser modificadas manualmente quando ocorrer alguma modificação na topologia da rede.

Rotas estáticas são entradas criadas por meios não automáticos numa tabela de rotas, as quais são fixas ao invés de serem resultado de algum procedimento de descoberta da topologia da rede.

Mas mesmo usando roteamento dinâmico, os roteadores podem ter uma ou duas rotas estáticas configuradas para casos especiais.

De um modo geral, o **roteamento estático** pode ser usado nos seguintes casos:

- Para definir uma rota *default* (rota padrão), usada para encaminhar para um roteador específico quando nenhuma outra rota for encontrada para o endereço destino do pacote de dados;
- Em pequenas redes com apenas um ou dois roteadores, caso em que é mais eficiente pois não desperdiça link de dados trafegando informações dinâmicas de roteamento;
- Como complemento ao roteamento dinâmico, para prover *backup* de segurança contra falhas no caso de uma rota dinâmica ficar indisponível. Isto é chamado de rota estática flutuante (*floating static route*);
- Para ajudar a transferir informações de roteamento de um protocolo de roteamento para o outro, que é a redistribuição de rotas. Por exemplo, para que mudanças na topologia de uma parte da rede seja reconhecida automaticamente por toda a rede.

Mas apesar de simples, o roteamento estático tem algumas desvantagens:

- **Erros humanos:** ao incluir as rotas estáticas, administradores de redes podem incluir erros nas tabelas de roteamento;
- **Tolerância à falhas:** não é tolerante a falhas, portanto o tráfego não será roteado caso haja mudança na topologia da rede ou mesmo falhas em equipamentos;
- **Distância administrativa:** tipicamente, as rotas estáticas têm precedência sobre as rotas dinâmicas definidas pelos protocolos de roteamento, e isso pode impedir os protocolos de roteamento de trabalhar conforme o esperado;
- **Sobrecarga administrativa:** as rotas estáticas precisam ser configuradas manualmente, que pode tomar muito tempo caso haja muitos roteadores na rede. Isto significa que a reconfiguração é lenta e ineficiente. Como comparação, o roteamento dinâmico automaticamente propaga as mudanças no roteamento.

3 – Roteamento Dinâmico

Para o roteamento dinâmico, ao invés do administrador da rede inserir manualmente as rotas, são usados protocolos de roteamento nos roteadores. Estes protocolos obtêm informações de rotas e, após convergência, automaticamente atualizam a tabela de rotas.

O protocolo de roteamento pode escolher dinamicamente uma rota diferente caso caia algum enlace, portanto este roteamento é tolerante a falhas. E, ao contrário do roteamento estático, não existe a necessidade de configurar manualmente rota por rota, que diminui substancialmente a carga de trabalho dos administradores de rede.

Em geral, as operações dos protocolos de roteamento dinâmico podem ser descritas como:

- i) O roteador envia e recebe mensagens de roteamento nas suas interfaces;
- ii) O roteador compartilha mensagens e informações de roteamento com os outros roteadores que estão usando o mesmo protocolo de roteamento;
- iii) Os roteadores trocam informações de roteamento para aprender a respeito de redes remotas;
- iv) Quando um roteador detecta uma mudança na topologia, o protocolo de roteamento notifica esta mudança aos demais roteadores na rede.

Uns poucos protocolos de roteamento chamados de Estado de Enlace (*Link State*) habilitam o roteador a perseguir e construir um mapa completo de todos os enlaces de rede numa região, enquanto outros, chamados de Vetor de Distância (*Distance Vector*), trabalham com menos informações sobre a área da rede.

3.1 – Métricas

O processo de roteamento exige a seleção do "melhor" caminho para o pacote de dados. Este caminho é função da métrica, que envolve, entre outros parâmetros:

- Quantidade de saltos (*hops*);
- Largura de banda;
- Latência (atraso);
- Carga na rede;
- Confiabilidade da rede;
- Custo da comunicação.

A métrica é o padrão de medida usado para determinar o melhor caminho para um determinado destino. Depois de determinada, é apresentada como um número inteiro positivo. Quanto menor este número, melhor é o caminho.

Mas se a rota for **estática**, o valor *default* (padrão) para a métrica é 1. Por exemplo, ao adicionar uma rota estática numa interface de rede com métrica de valor 30, se usar o *default* vai dar 31 de métrica (30 + 1). Porém, se for necessário, o valor da métrica da rota pode ser modificado manualmente.

No **roteamento dinâmico**, a determinação da métrica pode utilizar um único parâmetro ou vários, dependendo do protocolo de roteamento utilizado. Por exemplo, RIP usa o parâmetro quantidade (contagem) de saltos como métrica de custo. A utilização de vários parâmetros permite uma melhor modelagem da métrica, que leva a uma decisão mais eficiente sobre qual é o melhor caminho.

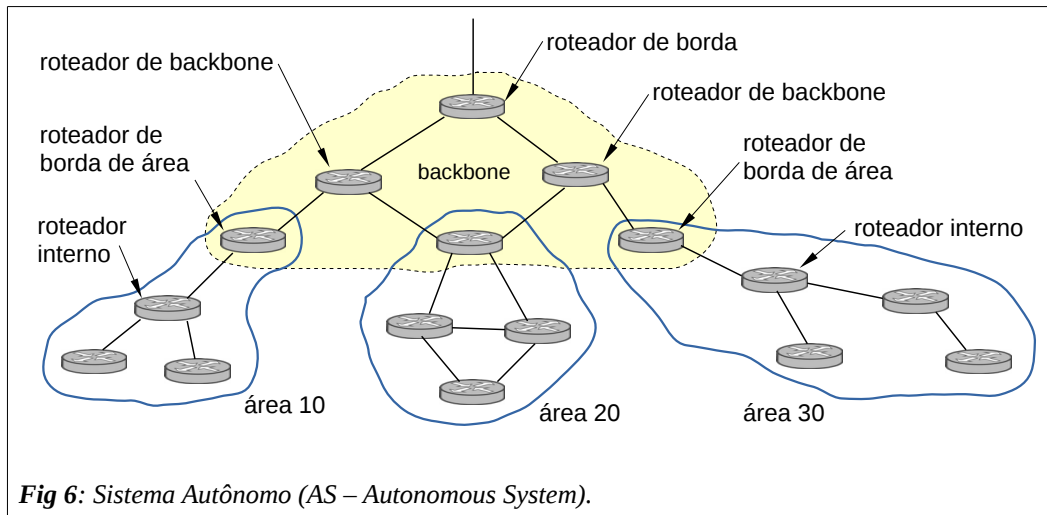
3.2 – Sistema Autônomo (*Autonomous System - AS*)

Um Sistema Autônomo (*Autonomous System – AS*) é um grupo de redes administrado pela mesma organização técnica. Por exemplo, a rede de um provedor de internet (*Internet Service Provider – ISP*) é um AS, assim como as redes internas de uma empresa. Um Sistema Autônomo é um grupo de redes IP, abaixo de uma única gerência técnica e que compartilha uma mesma política de roteamento.

No interior da internet, o AS é uma coleção de prefixos de roteamento IP (*Internet Protocol*) conectados sob o controle de um ou mais operadores de redes, em favor de uma única entidade administrativa ou domínio, e que apresenta uma política de roteamento para a internet, política esta comum e claramente definida para o AS.

O Número do Sistema Autônomo (*Autonomous System Number – ASN*) é uma identificação única e global de um AS, que foi inicialmente definido como um número inteiro de 16 bits (faixa de 0 a 65535), mas que posteriormente ampliou o espaço de endereçamento de 16 para 32 (faixa de 0 a

4294967295). A IANA¹ reservou os seguintes blocos de ASN para uso privado e para documentação, que não devem ser anunciados na internet: 64512-65535, 64496-64511 e 65536-65551.



3.3 – Protocolos de Roteamento

Os Protocolos de Roteamento são uma família de protocolos de rede que permite aos roteadores se comunicarem entre si, e assim decidirem a melhor maneira de encaminhar o tráfego de dados entre as suas respectivas redes.

O objetivo dos Protocolos de Roteamento é a construção dinâmica da tabela de rotas do roteador.

Quanto à vizinhança entre *gateways* que trocam informações de roteamento, os protocolos de roteamento dinâmico são separados em duas categorias, **IGP** (*Interior Gateway Protocol* – Protocolo de Roteamento Interno) e **EGP** (*Exterior Gateway Protocol* - Protocolo de Roteamento Externo).

1 IANA: *Internet Assigned Numbers Authority* (Autoridade para Atribuição de Números da Internet) é a organização mundial que supervisiona a atribuição global dos números na Internet, entre os quais estão os números das portas, os endereços IP, os números do sistemas autônomos (ASN), servidores-raiz (root-servers) de DNS, etc.

3.3.1 – Protocolo de Roteamento Interno

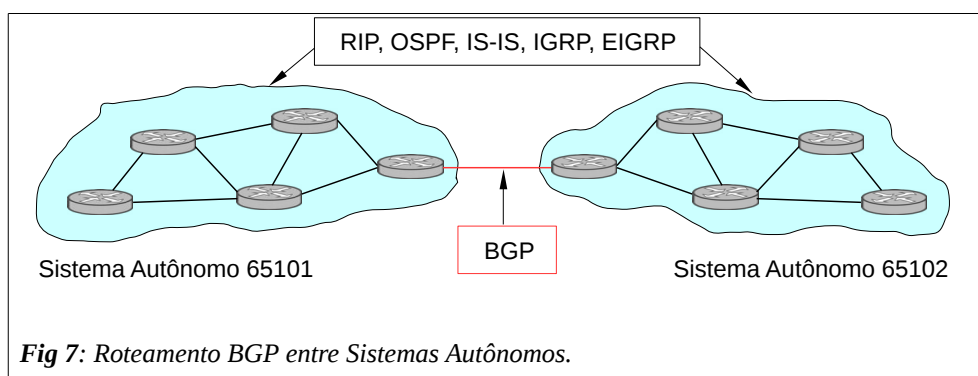
O **Protocolo de Roteamento Interno** (*Interior Gateway Protocol - IGP*) é usado quando as informações de roteamento são distribuídas internamente somente entre roteadores pertencentes ao mesmo Sistema Autônomo (*Autonomous System – AS*), isto é, o roteamento IGP ocorre apenas dentro de um único Sistema Autônomo. Alguns exemplos de protocolos IGP são:

- RIP: *Routing Information Protocol* (Protocolo de Informação de Roteamento);
- OSPF: *Open Shortest Path First* (Abrir Primeiro o Caminho mais Curto);
- IS-IS: *Intermediate System to Intermediate System* (Sistema Intermediário para Sistema Intermediário);
- IGRP: *Interior Gateway Routing Protocol* (Protocolo de Roteamento de *Gateway* Interior);
- EIGRP: *Enhanced Interior Gateway Routing Protocol* (Protocolo de Roteamento de *Gateway* Interior Aprimorado).

Normalmente, no interior de um Sistema Autônomo são usadas métricas comuns para determinar como rotear os pacotes. Mas mesmo que seja usado mais de um protocolo IGP e muitas métricas diferentes, ainda assim, visto pelos outros Sistemas Autônomos (externamente), vai aparentar que tem um único plano coerente de roteamento e uma visão consistente de quais destinos são alcançáveis através dele.

3.3.2 – Protocolo de Roteamento Externo

O **Protocolo de Roteamento Externo** (*Exterior Gateway Protocol - EGP*) é usado quando as informações de roteamento são distribuídas entre os limites dos Sistema Autônomos (AS), entre roteadores pertencentes a distintos Sistemas Autônomos. EGP trata do roteamento entre diferentes Sistemas Autônomos. Como exemplo de EGP temos o Protocolo de Roteamento de Borda (*Border Gateway Protocol – BGP*), que é usado para rotear o tráfego através do *backbone* da Internet, entre diferentes Sistemas Autônomos.



3.3.3 – Classificação dos Protocolos de Roteamento

Os protocolos de roteamento podem ser classificados em diferentes grupos, de acordo com as suas características. Especificamente, podem ser classificados quanto ao **propósito, operação e comportamento** (*behavior*):

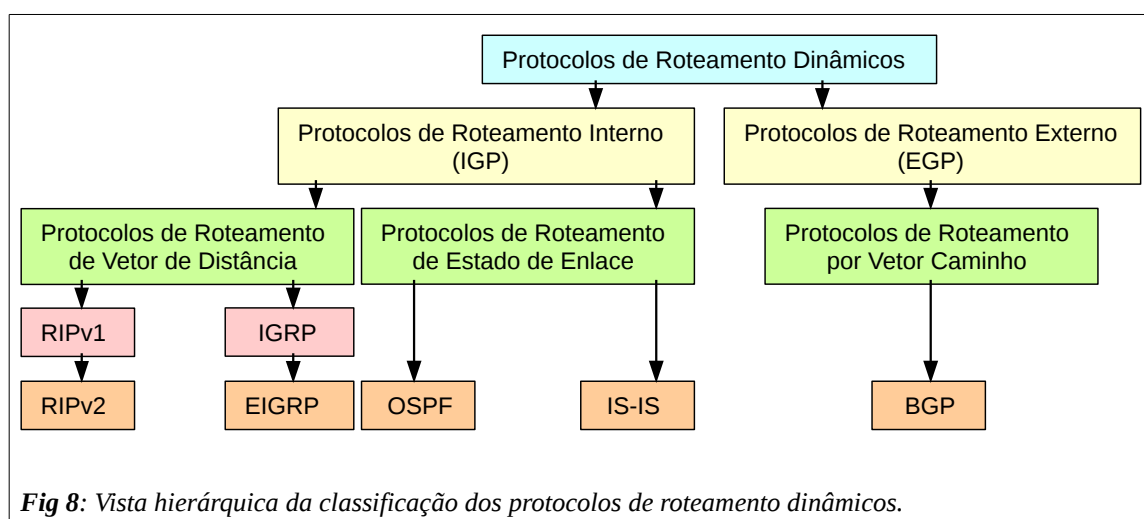
- Propósito:
 - Protocolo de Roteamento Interno (IGP);
 - Protocolo de Roteamento Externo (EGP);
- Operação:
 - Protocolo de Roteamento de Vetor de Distância (*Distance Vector Routing Protocol*);
 - Protocolo de Roteamento de Estado de Enlace (*Link-State Routing Protocol*);
 - Protocolo de Roteamento por Vetor Caminho (*Path-Vector Routing Protocol*);
- Comportamento:
 - Classful* (legado);
 - Classless* (CIDR).

Em acordo com esta classificação, na tabela 1, abaixo, são mostrados alguns exemplos de protocolos de roteamento ipv4.

Protocolo	Propósito	Operação	Comportamento
RIPv1 (legado)	IGP	Vetor de Distância	<i>Classful</i>
RIPv2	IGP	Vetor de Distância	<i>Classless</i>
OSPF	IGP	Estado de Enlace	<i>Classless</i>
IS-IS	IGP	Estado de Enlace	<i>Classless</i>
IGRP (legado)	IGP	Vetor de Distância	<i>Classful</i>
EIGRP	IGP	Vetor de Distância	<i>Classless</i>
BGP	EGP	Vetor Caminho	<i>Classless</i>

Tabela 1: Classificação dos Protocolos de Roteamento.

Na tabela acima, à exceção dos protocolos IGRP e EIGRP, que são proprietários da Cisco, os demais são padrões abertos.



Os protocolos de roteamento *classful* RIPv1 e IGRP são legados, e por isso só são usados em redes antigas. Respectivamente, eles evoluíram para os protocolos de roteamento *classless* RIPv2 e EIGRP. Os protocolos de roteamento de Estado de Enlace são *classless* pela natureza deles.

Internamente, os protocolos de roteamento usam **algoritmos de roteamento** para calcular o "melhor" caminho a ser escolhido para transportar os dados desde a origem até o destinatário. Cada protocolo de roteamento usa apenas um algoritmo específico. Por exemplo, RIP usa o algoritmo Vetor de Distância e OSPF usa Estado de Enlace.

3.3.4 – Algoritmos de Roteamento

Um algoritmo de roteamento é um software responsável pela decisão sobre o caminho para o qual deve ser encaminhado o pacote de dados. A finalidade de um algoritmo de roteamento é simples: dado um conjunto de roteadores conectados por enlaces, o algoritmo descobre o "melhor" caminho entre o roteador de origem e o roteador de destino. O "melhor" caminho é aquele que tem o menor custo.

Existem duas classes de algoritmos de roteamento, **não adaptativo** (estático) e **adaptativo** (dinâmico):

- Não adaptativo (estático): não baseiam as suas decisões de roteamento em medidas ou estimativas de tráfego e topologia.
Exemplos de algoritmos: Menor Caminho (*Shortest Path*) e Inundação (*Flooding*);
- Adaptativo (dinâmico): as decisões de roteamento são adaptadas e modificadas para refletir mudanças na topologia e no tráfego. Estes algoritmos diferem entre si nos seguintes aspectos:
 - forma como obtêm a informação, que pode ser local, a partir de roteadores adjacentes ou a partir de todos os roteadores;
 - intervalos de tempo em que mudam as rotas, que pode ser a cada intervalo de tempo (em

- segundos), quando a carga muda, ou quando a topologia da rede muda;
- métrica que é utilizada para otimização, que pode ser a distância, o número de saltos ou o tempo estimado de trânsito.
- Exemplos de algoritmos: Vetor de Distância (*Distance Vector*) e Estado de Enlace (*Link State*).

Se estiver sendo usado algoritmo de roteamento dinâmico e os dados estão sendo transmitidos como datagramas (protocolo de transporte UDP), a decisão sobre o roteamento deverá ser tomada para cada pacote de dados recebido, pois o "melhor" caminho pode ter sido alterado desde o último pacote.

A **convergência** é uma noção importante para um conjunto de roteadores empenhados no roteamento dinâmico. Todos os protocolos de roteamento de interior (IGP) dependem da convergência para funcionarem adequadamente. Ter convergido é o estado normal de um Sistema Autônomo operacional. Já o protocolo de roteamento externo (EGP) tipicamente nunca converge, pois a internet é demasiadamente grande para que as mudanças sejam comunicadas com suficiente rapidez.

RIP é um protocolo de roteamento que converge tão lentamente que mesmo uma rede com poucos roteadores precisa de alguns minutos para convergir. Caso um novo roteador seja anunciado, as atualizações que são disparadas podem apressar a convergência, mas para descarregar um roteador que previamente existia demora mais devido aos temporizadores em uso.

OSPF é um exemplo de protocolo de roteamento de convergência rápida. Uma rede com poucos roteadores pode convergir em questão de segundos.

3.3.5 – Heurísticas em que se baseiam os Algoritmos de Roteamento

Um algoritmo é um conjunto definido de instruções computacionais sequenciais, utilizado para obter um determinado resultado a partir de dados fornecidos para a resolução do problema.

Mas se o problema for demasiado complexo para ser resolvido explicitamente, então são usadas as heurísticas. Do ponto de vista de um algoritmo, um problema é demasiado complexo quando não existe uma regra computacional para encontrar a solução. Heurísticas também são algoritmos, mas que operam num contexto mais amplo, e portanto são consideradas diretivas ou "regras por alto" de como resolver um problema.

No caso do roteamento, a heurística é necessária devido à complexidade e a dinâmica da rede.

Alguns exemplos de heurísticas para algoritmos de roteamento, são:

- **Menor Caminho** (*Shortest Path*): o conceito empregado aqui é a construção de um grafo de sub-rede. Cada ponto do grafo representa um roteador, e cada linha um meio de comunicação. Para achar uma rota entre dois roteadores quaisquer, o algoritmo acha a rota

com menor custo. O custo da rota é obtido por diversas métricas, tais como número de saltos (*hops*) entre roteadores, estado e largura de banda de enlaces. O algoritmo Menor Caminho também é conhecido por algoritmo de **Dijkstra**;

- **Vetor de Distância** (*Distance Vector*): conceitualmente, opera de forma que cada roteador mantenha uma tabela (ou vetor), que fornece a menor distância conhecida até o destino, e também indica qual linha deve ser utilizada para a transmissão para cada destino através de uma linha específica. O vetor é mantido atualizado através do processo dinâmico de troca de informações com os roteadores vizinhos. Algumas informações do vetor são número de saltos (*hops*), tempo de retardo (em milissegundos) e o número total de pacotes enfileirados ao longo do caminho (*path*). O algoritmo de Vetor de Distância é interativo, assíncrono e distribuído. É distribuído porque cada nó recebe alguma informação com respeito a um ou mais vizinhos diretamente conectados, faz cálculos e, após, distribui os resultados de seus cálculos para seus vizinhos. O interativo vem da troca de dados constante, até que não seja mais possível realizar tal troca. É assíncrono porque não requer que todos os nós rodem simultaneamente. Este algoritmo também é conhecido como **Bellman-Ford**, e recebe este nome em homenagem aos pesquisadores, Bellman, 1957 e Ford em 1962. Os algoritmos dos protocolos baseados nesta abordagem são dinâmicos. É um dos algoritmos mais utilizados;
- **Estado de Enlace** (*Link State*): Este algoritmo substituiu o Vetor de Distância na rede ARPANET, pois a métrica da distância não levava em consideração a largura de banda dos enlaces, e o conceito de vetor demora muito na convergência. Este algoritmo se baseia em cinco pontos:
 - descobrir os vizinhos e seus endereços de rede;
 - calcular o retardo ou custo para cada um dos vizinhos;
 - construir um pacote contando tudo o que aprendeu;
 - propagar este pacote, que é conhecedor de todas as informações, para os demais roteadores da rede;
 - computar o menor caminho para todos os outros roteadores.

Os algoritmos dos protocolos baseados nesta abordagem são dinâmicos. É um dos mais populares empregados em redes modernas.

3.4 – Distância Administrativa

Num mesmo roteador, podem rodar simultaneamente múltiplos protocolos de roteamento. Se isto ocorrer, tornam-se críticas tanto as trocas de informações de rotas quanto a capacidade de selecionar o melhor caminho para o pacote de dados, pois a maioria dos protocolos de roteamento possui estruturas métricas e algoritmos que não são compatíveis entre si.

Então, quando dois diferentes protocolos de roteamento apresentarem duas ou mais rotas diferentes para o mesmo destino, precisará haver algum critério para a escolha do melhor caminho.

Isto torna necessário implementar, no roteador, um processo que garanta que não exista simultaneamente, na tabela de rotas, múltiplas rotas apontando para o mesmo destino.

A **distância administrativa** é que fornece o critério de seleção para o melhor caminho, pela escolha do menor valor de distância padrão. Desse modo, cada processo de protocolo de roteamento que roda no roteador anuncia seu valor de distância administrativa, que é usado para selecionar qual rota tem maior prioridade e por isto será a usada. Uma vez que uma determinada rota tenha sido selecionada, é atualizado o banco de dados de informações de roteamento. E se ainda assim duas rotas tiverem a mesma distância administrativa, o roteador usa um algoritmo específico do vendedor para determinar qual rota será a instalada.

É a distância administrativa que define a confiabilidade do protocolo de roteamento, pois atribui a cada protocolo uma prioridade que vai de mais confiável para menos confiável, isso pelo uso de um valor *default* (padrão) de distância administrativa.

O valor da distância administrativa é um número, em unidade arbitrária, atribuído a rotas dinâmicas, rotas estáticas e rotas diretamente conectadas.

A distância administrativa é um valor único associado a cada tipo de rota IP. É usada para avaliar múltiplos caminhos de igual custo para o mesmo destino, obtidos por diferentes fontes tais como RIP ou OSPF. Cada caminho tem uma distância administrativa, porém isso não é usado em balanceamento de carga.

O valor da distância administrativa é determinado pela fonte (origem) da rota, que é um valor único para cada fonte. Quando o software do roteador recebe caminhos de diferentes fontes para diferentes destinos, ele compara as distâncias administrativas e seleciona a de menor distância, e só depois desta decisão é que adiciona à tabela de rotas. Por exemplo, para determinado destino, se os caminhos que o roteador aprendeu vieram um do OSPF e o outro do RIP, somente o caminho com a menor distância administrativa é que vai para a tabela de rotas.

Protocolo de roteamento/Origem da rota	Valor da Distância Padrão
Interface diretamente conectada (valor não configurável)	0
Rotas estáticas	1
<i>External Border Gateway Protocol</i> (EBGP)	20
<i>Internal (Interior) IGRP</i>	90
IGRP	100
OSPF	110
<i>Intermediate System-to-Intermediate System</i> (IS-IS)	115
<i>Routing Information Protocol</i> (RIP)	120
<i>Internal (Interior) IBGP</i>	200
Desconhecido (por ser desconhecida, não é usada)	255

Tabela 2: Distância Administrativa.

Valores baixos para a distância administrativa são preferidos em detrimento de valores altos. Por exemplo, se para a mesma rede o equipamento receber rotas do OSPF (110) e do RIP (120), ele vai preferir a rota do OSPF por padrão.

Com base na distância administrativa do protocolo de roteamento, as rotas são escolhidas e criadas na tabela de roteamento. São instaladas na tabela de roteamento as rotas com as menores distâncias administrativas, conforme detectadas pelo protocolo de roteamento. Se um único protocolo de roteamento determinar mais de um caminho para o mesmo destino, então estes diferentes caminhos terão a mesma distância administrativa, e o melhor caminho será selecionado com base nas métricas.

No roteador, a distância administrativa também pode ser configurada manualmente para sobrepor o valor da distância padrão e dar prioridade a um protocolo específico em detrimento do padrão (*default*).

As rotas estáticas tem, por padrão, distância administrativa de 1. E por considerar as rotas de menor número como as mais curtas, o roteador dá preferência às rotas estáticas.

Mas se for necessário que uma rota dinâmica sobreponha uma rota estática, pode ser especificado um valor arbitrário de distância administrativa para a rota estática. Por exemplo, especificar um valor maior que 120 para uma rota estática para que ela seja sobreposta por duas rotas dinâmicas com valor de distância administrativa de 120.

Rota estática flutuante (*floating static route*) é uma rota estática que o roteador usa como substituta (*backup*) de rota dinâmica, que é usada caso a rota dinâmica seja perdida por algum motivo. Mas para que o roteador mantenha a preferência pela rota dinâmica, precisa que a rota estática seja configurada para um valor de distância administrativa mais alto que o da rota dinâmica.

3.5 – Vetor de Distância (*Distance Vector*)

Um protocolo de roteamento de Vetor de Distância pertence à classe de roteamento interno (IGP) que usa o algoritmo Bellman–Ford ou Ford–Fulkerson para calcular os caminhos.

O Vetor de Distância tem menos complexidade computacional e menor carga (*overhead*) de mensagens na rede quando comparado com os protocolos de Estado de Enlace, que requerem que o roteador informe periodicamente a todas as suas vizinhanças sobre as mudanças na topologia da rede.

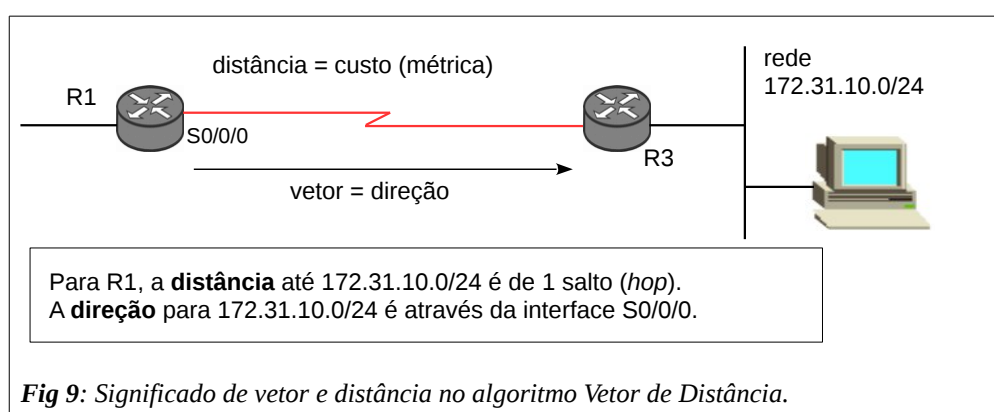
O termo Vetor de Distância se refere ao fato de este protocolo manipular vetores (*arrays*) de distâncias para os outros nós na rede. Foi o algoritmo de roteamento original da ARPANET, e ainda é usado atualmente nos protocolos RIPv1, RIPv2, IGRP e EIGRP.

Os roteadores que usam o protocolo Vetor de Distância não tem conhecimento do caminho inteiro para o destinatário, pois não possuem o conhecimento completo da topologia de rede, e as informações completas de roteamento precisam ser periodicamente atualizadas. A determinação das melhores rotas leva em consideração as informações obtidas dos roteadores vizinhos, como módulo e direção. Usam dois métodos para o roteamento:

- A direção na qual, ou para qual interface, os pacotes precisam ser encaminhados;
- A distância para a rede do destinatário.

Os protocolos de Vetor de Distância são baseados no cálculo da direção e distância para qualquer enlace na rede. Direção usualmente significa o próximo endereço para o salto (*hop*) ou a interface de saída. Distância é a medida do custo para atingir um certo nó. A rota de menor custo entre quaisquer dois nós é a rota com menor distância. Cada nó mantém um vetor (tabela) da distância mínima para cada nó. O custo para atingir um destinatário é calculado com o uso de várias métricas de rotas. RIP usa o contador de salto (*hop*) para o destinatário, enquanto IGRP leva em consideração outras informações tais como demora (*delay*) do nó e largura de banda disponível.

O custo é um vetor de distância, pois frequentemente o custo é equivalente ao número de saltos entre o roteador e a rede de destino.



Quanto à operação, alguns protocolos de roteamento que usam Vetor de Distância fazem o roteador periodicamente transmitir a tabela de roteamento inteira para cada um dos seus vizinhos. Este método é ineficiente, pois estas atualizações não só consomem banda de dados, como também consomem recursos de CPU do roteador para processar estas atualizações.

RIP envia as atualizações periodicamente em intervalos regulares de 30 segundos, e IGRP envia a cada 90 segundos. Mesmo que a topologia não tenha mudado por diversos dias, estas atualizações periódicas continuam sendo enviadas para todos os seus vizinhos.

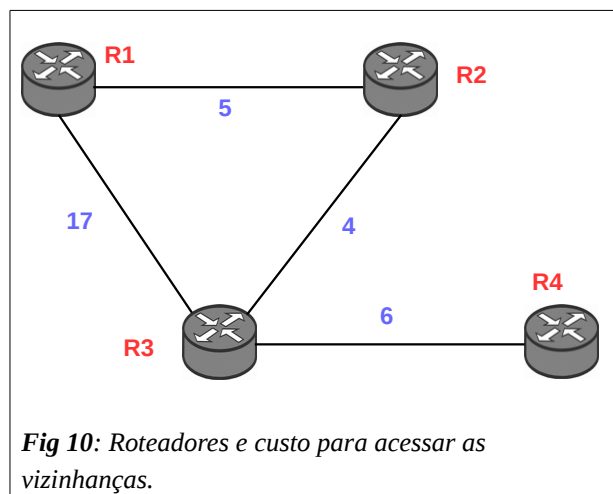
Um exemplo de rede com quatro roteadores, onde é mostrado também o custo (distância) para alcançar as vizinhanças, é mostrado na figura 10.

Neste exemplo, está sendo considerado o mesmo custo para ir e voltar. Por exemplo, tanto para ir de R1 para R2 quanto de R2 para R1 o custo é 5.

O algoritmo de Vetor de Distância opera criando vetores (ou matrizes) de distância para cada roteador em relação às suas vizinhanças imediatas. Periodicamente, cada roteador envia a sua própria lista de distâncias para os demais roteadores através dos seus vizinhos. À medida que os vizinhos vão recebendo esta lista, eles vão recalculando novamente os menores caminhos.

Por uma questão de simplicidade, neste exemplo será considerado apenas o processo de convergência do caminho de **R1 para R4**.

No instante **T=0** o roteador R1 conhece apenas os custos para os seus vizinhos imediatos, R2 e R3. Mas a cada iteração (**T**), o roteador R1 recebe a lista de custos dos seus vizinhos e recalcula as suas distâncias para obter os menores caminhos. R1 também envia a sua lista de custos para os seus vizinhos.



Origem: R1 (T=0)	Via: R2	Via: R3	Origem: R1 (T=1)	Via: R2	Via: R3
Destino: R2	5		Destino: R2	5	
Destino: R3		17	Destino: R3		17
Destino: R4			Destino: R4		23 (17+6)

Após a iteração **T=1**, R1 obtém a distância (custo) de R3 para R4, que é 6. Com esse valor, recalcula os custos para acessar R4 via R3 e envia a sua nova lista de custos para os seus vizinhos.

Na iteração **T=2**, R2 informa o custo para R3, que é 4, e R1 recalcula o custo e direção para R4.

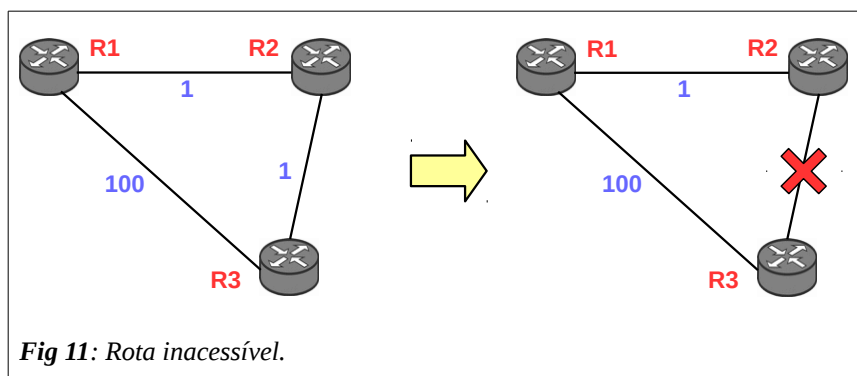
Origem: R1 (T=2)	Via: R2	Via: R3	Origem: R1 (T=3)	Via: R2	Via: R3
Destino: R2	5	21 (17+4)	Destino: R2	5	21
Destino: R3	9 (5+4)	17	Destino: R3	9	17
Destino: R4	15 (5+4+6)	23 (17+6)	Destino: R4	15	23

Após **T=3**, as trocas de informações de listas de custo não permitem a R1 reduzir os custos para os valores já calculados, então não é feita mais nenhuma mudança na tabela de roteamento. Agora, para R1, o "melhor" caminho para R4 é via R2, com custo 15. Para R3 também o melhor caminho é via R2, com custo 9.

Um problema com o algoritmo Bellman-Ford é não prevenir os *loops* de roteamento, quando ocorre o problema de **contagem para o infinito**. Um exemplo simples de ocorrência de contagem para o infinito segue abaixo.

Na figura 11, os três roteadores, R1, R2 e R3 já convergiram para o "melhor" caminho em $T=0$, no lado esquerdo da figura. Depois disso, R3 se torna inacessível para R2.

Em $T=1$, R2 já perdeu o acesso a R3. Mas após R3 se tornar inacessível, R2 "pensa" que pode redirecionar os pacotes para R3 via R1, que tem custo 2 para alcançar R3, então calcula 3 (1+2) para alcançar R3 via R1 e envia esta lista para R1. R1 que **não sabe** que R3 está inacessível via R2, recalcula o custo, obtém 4 para R3 e envia a lista para R2.



Origem: R1 ($T=0$)	Via: R2	Via: R3	Origem: R1 ($T=1$)	Via: R2	Via: R3
Destino: R2	1	101 (100+1)	Destino: R2	1	103 (100+3)
Destino: R3	2 (1+1)	100	Destino: R3	4 (3+1)	100

Desse modo, em $T=2$, R2 recalcula os custos para R3 com base na lista recebida por R1, e o custo de R2 para R3 agora é 5 (4+1). R2 envia esta lista para R1, que recalcula o custo para R3 via R2 e obtém 6 (5+1).

Origem: R1 ($T=2$)	Via: R2	Via: R3	Origem: R1 ($T=3$)	Via: R2	Via: R3
Destino: R2	1	105 (100+5)	Destino: R2	1	<i>loop</i>
Destino: R3	6 (5+1)	100	Destino: R3	<i>loop</i>	100

Em $T=3$ já fica claro que o custo para R3 via R2 vai incrementando a cada iteração. Mas quando a contagem passar de 100 vai cessar o *loop*, pois a partir de então o menor custo para R1 acessar R3 passa a ser diretamente e não mais via R2.

3.6 – Estado de Enlace (*Link State*)

Os Protocolos de Roteamento que fazem uso do Estado de Enlace tem o mesmo objetivo dos que usam Vetor de Distância, porém com aproximações diferentes. O roteador que usa Estado de Enlace informa a cada nó o que ele sabe a respeito das suas vizinhanças imediatas, ou seja, informa a todos os demais roteadores a respeito da topologia local.

As "melhores" rotas são determinadas baseado em algumas informações obtidas pelos roteadores presentes na rede, como estado e largura de banda de enlaces (*links*). Os roteadores possuem conhecimento completo da topologia de rede, e apenas quando é necessário é que são enviadas informações de roteamento. A determinação das melhores rotas é realizada de maneira independente para cada roteador na rede.

O Estado de Enlace determina o "melhor" caminho em duas fases:

- i) Inundação Confiável (*Reliable Flooding*): informa a todos os outros roteadores a respeito da topologia local. Cada roteador transmite um Pacote de Estado de Enlace (*Link State Packet - LSP*) para os seus próprios enlaces, e os roteadores vizinhos encaminham este pacote para todos os seus próprios enlaces exceto o de origem. Os roteadores que recebem o pacote mantém uma cópia localmente e não encaminham os que já foram previamente vistos. Os LSP são numerados sequencialmente para distinguir os novos dos velhos, de modo que os roteadores apenas aceitem e encaminhem os novos. A inundação ocorre tanto periodicamente (por exemplo, a cada 30 minutos) quanto pode ser disparada por uma mudança na topologia na rede;
- ii) Cálculo de Menor Caminho (*Path Calculation*): cada roteador calcula o melhor caminho para a alcançar a rede inteira. Para isso usa o algoritmo de Dijkstra.

Um exemplo de Cálculo de Menor Caminho segue na figura 12 abaixo, onde o roteador **R1** usa o algoritmo Dijkstra para tentar achar o caminho de menor custo, e uma vez encontrado, confirmar o nó como melhor caminho.

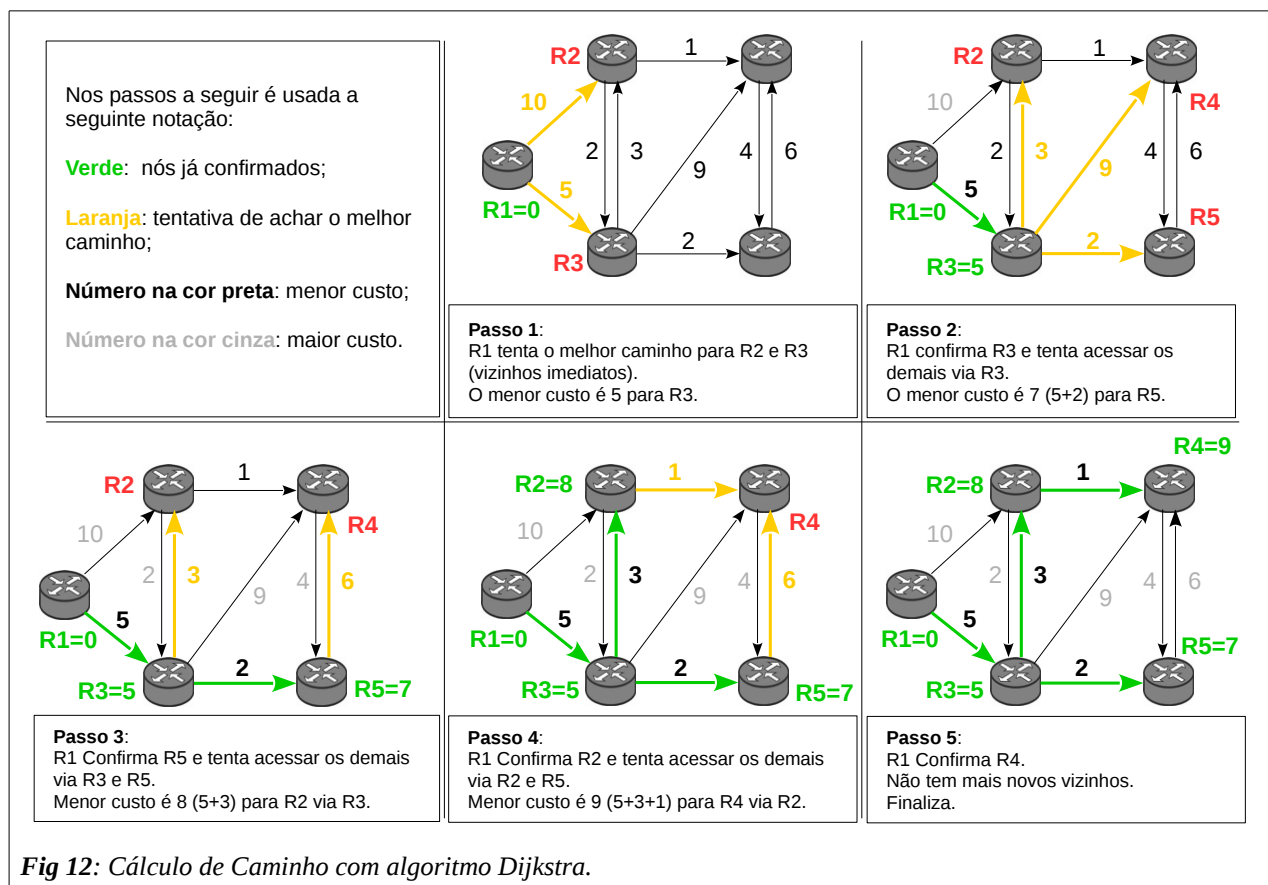


Fig 12: Cálculo de Caminho com algoritmo Dijkstra.

Embora obtenha convergência rápida, os Protocolos de Estado de Enlace têm algumas fraquezas, e as principais são:

- A inundação precisa ocorrer ao longo de toda a rede, que pode originar questões sobre a escalabilidade de tal solução;
- Cada roteador precisa manter em memória o estado de toda a topologia da rede, e se for muito grande sobrecarrega o equipamento.

3.7 – Protocolos RIP e OSPF

Atualmente, os dois principais protocolos internos (IGP) abertos (*open*) são RIP, por razões históricas, e OSPF pelo seu uso.

3.7.1 – RIP

O RIP foi desenvolvido nos anos 1980, e foi projetado para redes internas pequenas ou com tamanho moderado, que se conectavam à internet daqueles tempos. Não é destinado a uso em ambientes mais complexos. RIP somente é capaz de rotear pacotes através de uma rede com no máximo 15 saltos, e qualquer destino de custo maior que isso é considerado inalcançável. Embora limite o uso para redes maiores, a baixa contagem de máximos saltos previne *loops* de contagem para o infinito.

O RIP é um protocolo de Vetor de Distância que usa a contagem de saltos como métrica de custo, portanto cada enlace tem custo 1.

Tradicionalmente, RIP suportava somente redes ipv4, mas o novo padrão RIPng também suporta ipv6. Para esta comunicação, os roteadores RIP utilizam a porta UDP 520, e os roteadores RIPng usam a porta UDP 521. Por operar sobre o protocolo de transporte UDP, RIP é essencialmente um protocolo de camada de aplicação.

A principal diferença entre o RIP versão 1 e versão 2 é que o primeiro usa o modelo *classfull* e o outro, *classless*. Portanto, RIPv1 não pode ser usado em sub-redes, pois sem as máscaras, os roteadores vão classificar os endereços como classes de redes A, B e C.

Em intervalos agendados (geralmente a cada 30 segundos), o roteador envia a sua tabela de rotas para todos os vizinhos (*broadcast*), para que qualquer mudança possa ser propagada ao longo da rede. Isso causa sobrecarga (*overhead*) na rede. Quando o RIP altera a tabela de roteamento local, em seguida também propaga tal informação transmitindo anúncios aos seus vizinhos. Um roteador também pode solicitar informações com relação aos custos das rotas.

Se um roteador não “ouvir” nada de seu vizinho, ao menos uma vez a cada 180 segundos, esse vizinho será rotulado como inalcançável. Isto pode ocorrer caso o vizinho esteja inoperante ou o enlace ter algum problema.

Os roteadores com RIP habilitado descobrem a rede pelo envio de uma mensagem aos dispositivos vizinhos. Esta mensagem é uma requisição à qual os roteadores vizinhos respondem com o envio da tabela de roteamento inteira ao requisitante. A mensagem de resposta enviada pelos vizinhos contém, num mesmo pacote, uma lista de até 25 rotas de destino dentro do AS, bem como as distâncias entre o remetente a cada uma destas sub-redes. Mensagens de resposta também são conhecidas como anúncios RIP. Depois de receber as tabelas de roteamento, o requisitante procede com um algoritmo para juntar todas as atualizações na sua própria tabela de roteamento.

A tabela 3, abaixo, mostra o número de saltos desde o roteador **R1** da figura 13 até todas as suas sub-redes.

R1 para a rede de destino	Número de saltos
A	1
B	2
C	3
D	3
E	2
F	2

Tabela 3: Número de saltos para a rede de destino.

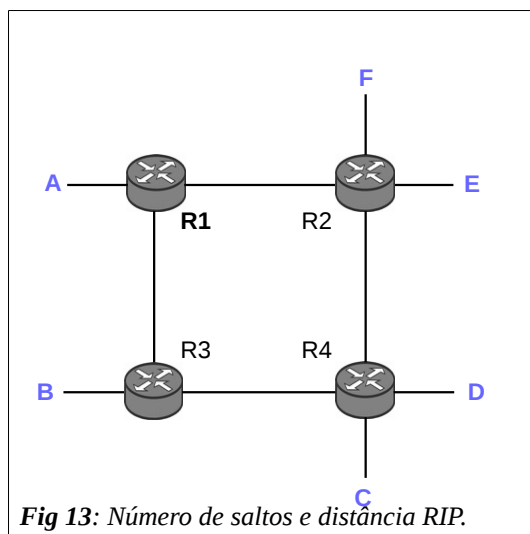


Fig 13: Número de saltos e distância RIP.

Apesar de medir o custo baseado em saltos, os roteadores RIP podem modificar o custo de uma rota pela adição de um viés (um número) à rota. Neste caso, o número real de saltos permanece o mesmo mas a rota adquire um custo administrativo maior, e então se torna menos provável de ser usada devido à existência de outras rotas de menor custo para a mesma rede de destino.

3.7.2 – OSPF

O protocolo OSPF (*Open Shortest Path First*) foi criado para superar algumas limitações do RIP, incluindo a restrição da contagem de 15 saltos e a geração de picos de tráfego na rede devido à necessidade dos roteadores reenviarem toda a sua tabela de roteamento em intervalos fixos de tempo. Também contribuiu a favor do OSPF a dificuldade em organizar as grandes redes internas numa hierarquia de roteamento, necessário para melhorar o desempenho e gerenciamento da rede.

OSPF é um padrão público aberto desde 1990, com ampla adoção entre muitos fabricantes de roteadores. Foi criado pela IETF (*The Internet Engineering Task Force* – Força Tarefa de Engenharia de Internet).

Os roteadores habilitados para OSPF descobrem a rede pelo envio de mensagens de identificação para todos os demais, seguido de mensagens para obter itens específicos de roteamento. Ao contrário do RIP, os equipamentos não enviam a tabela de roteamento inteira.

Ao contrário de RIP e BGP, OSPF não usa um protocolo de transporte, mas encapsula os dados diretamente em pacotes IP com protocolo número 89². Devido a isso, OSPF implementa as suas próprias funções de detecção e correção de erros de camada de transporte.

OSPF usa endereçamento *multicast*³ para distribuir informações de rota num domínio de

2 Número do Protocolo Internet é o número que vai no cabeçalho do Protocolo IP. Em ipv4, vai no campo Protocolo (*Protocol*), em ipv6 vai no campo Próximo Cabeçalho (*Next Header*). Por exemplo, o protocolo icmp (*Internet Control Message Protocol*) tem número 1, TCP é 6 e UDP é 17.

3 Endereçamento *Multicast* é definido no CIDR como a faixa de endereços 224.0.0.0/4 (classe D).

*broadcast*⁴, enquanto RIP usa *broadcasting*. Ou seja, RIP envia para todos os roteadores que estiverem conectados, e OSPF usa *multicasting*, que envia só para um grupo de roteadores para diminuir a utilização de largura de banda.

Atualmente o OSPF está na versão 2, com ampla utilização. A versão 3 também existe, para poder utilizar com ipv6.

OSPF é um protocolo de estado de enlace que usa *broadcasting (flooding)* de informação de estado de enlace e o algoritmo de menor custo Dijkstra.

Existem cinco tipos de pacotes OSPF, que são: ***Hello, Database Description, Link State Request, Link State Update e Link State Acknowledgement.***

Os pacotes ***Hello*** são utilizados para descobrir e manter relacionamentos de adjacência entre roteadores. Os pacotes ***Database Description*** são usados para formar adjacências. Os pacotes ***Link State Update*** e ***Link State Acknowledgement*** têm a função de atualizar as informações de roteamento. À exceção dos pacotes ***Hello***, todos os demais são enviados somente para as adjacências.

O OSPF fica constantemente enviando pacotes para manter o *status* da rede atualizado. A cada 10 segundos são enviados pacotes ***Hello***, então aparenta que este protocolo use muita largura de banda. No entanto, os pacotes ***Hello*** são muito reduzidos quando comparados aos do RIP, pois o OSPF só envia todas as informações se houver alguma alteração de topologia na rede, como queda ou adição de um novo enlace ou roteador.

As mensagens ***Database Description*** fornecem os números de sequência de todas as entradas de estado de enlace mantidas no momento pelo transmissor. Assim, é realizada uma comparação com seus próprios valores e do transmissor, em que o receptor pode determinar quem tem valores mais recentes. Esse tipo de pacote é enviado quando se deseja iniciar uma adjacência, tendo como conteúdo a descrição da base de dados topológica, descrição esta que pode necessitar de mais de um pacote. Também é usado quando o enlace é interrompido.

Cada roteador pode solicitar informações de estado de enlace um ao outro, usando mensagens ***Link State Request***. Assim, o roteador adjacente verifica quem tem dados mais recentes e as novas informações que estão sendo divulgadas. Este pacote é utilizado quando um roteador, tendo recebido as informações da base de dados topológica, envia uma mensagem requerendo informações por "perceber" a existência de informações obsoletas. Estes pacotes são enviados aos roteadores que têm uma base de dados topológica mais atualizada. Pode ser necessário o envio de pacotes deste tipo para mais de um roteador, para obter informações de fato atualizadas.

Durante o processo normal, por inundação, cada roteador emite periodicamente mensagens ***Link State Update*** para cada um de seus dispositivos adjacentes. Nestas mensagens estão contidas informações como estado e custo usados no banco de dados da topologia. As mensagens são confirmadas depois de recebidas, portanto possuem um serviço confiável, e também possuem número de sequência para o roteador saber se a mensagem recebida é antiga ou recente. Tais

4 Domínio de *Broadcast* (*Broadcast Domain*) é uma divisão lógica da rede, na qual os nós podem se comunicar em camada 2 (enlace). A separação entre dois domínios de *broadcast* é conseguida pelo uso de um roteador entre estas duas redes.

mensagens também são enviadas caso o enlace seja ativado ou desativado, ou quando os custos são alterados. Este tipo de pacote é utilizado no processo de atualização das bases de dados topológicas.

O **Link State Acknowledgement** é utilizado no processo de atualização das bases de dados topológicas, é enviado como reconhecimento do recebimento de um *Link State Update*.

As principais melhorias do OSPF em relação ao RIP são:

- Analisa mais métricas além do salto (*hop*);
- Dinâmico, converge rapidamente;
- Capacidade de rotear baseado no Tipo do Serviço (*Type of Service - ToS*). Isso possibilita rotear o tráfego de tempo real (ex: *stream* áudio) com QoS e os outros tráfegos de maneira distinta. ToS está definido num campo do cabeçalho IP;
- Balanceamento de carga, que consiste em dividir o tráfego por várias linhas (enlaces);
- Atender a redes muito grandes, onde nenhum roteador é capaz de conhecer a topologia inteira. Neste caso, o Sistema Autônomo (AS) pode ser dividido em grupos contíguos de redes chamadas Áreas OSPF.

Muitos Sistema Autônomos (AS) são grandes e difíceis de gerenciar, mas o OSPF permite que eles sejam divididos em áreas numeradas, onde uma área é uma rede ou um conjunto de redes contíguas. Em uma área, cada roteador transmite o seu estado de enlace a todos os outros roteadores da mesma área, e com isso os detalhes internos permanecem invisíveis para todos os roteadores externos.

O conceito de áreas traz benefícios, e o mais relevante é a escalabilidade facilitada pela forma hierárquica assim proporcionada. Em cada área, um ou mais roteadores de borda são encarregados do encaminhamento de pacotes para fora desta área, assim uma área do AS é configurada para ser a área de *backbone*. A principal tarefa da área de *backbone* é rotear o tráfego entre os outros AS.

Num Sistema Autônomo com protocolo OSPF, os roteadores podem ser classificados nas seguintes quatro categorias abaixo:

- **roteadores internos** (*internal routers*): são roteadores que fazem somente roteamento interno no Sistema Autônomo, e não estão no *backbone*. Estes roteadores têm todas as redes diretamente conectadas e pertencem à mesma área. Estes roteadores rodam uma única cópia do algoritmo básico de roteamento;
- **roteadores de borda de área** (*area border routers*): são roteadores que pertencem tanto a uma área quanto ao *backbone*. Estes roteadores rodam múltiplas cópias do algoritmo básico de roteamento, uma cópia para cada área ligada a ele e uma cópia adicional para o *backbone*. Cada roteador de borda de área condensa a informação topológica das áreas ligadas a ele, para distribuição para o *backbone*. E o *backbone* distribui esta informação para as outras áreas;
- **roteadores de backbone** (mas não de borda): são roteadores que fazem roteamento dentro do *backbone*, mas não no roteador de borda (*boundary routers*). Eles têm uma interface de rede no *backbone* e podem atender a uma ou mais áreas. A partir das informações de rotas

dos roteadores de *backbone* é que os roteadores internos aprendem sobre a existência de rotas para as outras áreas;

- **roteadores de borda** (*boundary routers*): o roteador de borda troca informações de rotas com roteadores pertencentes a outros Sistemas Autônomos. Este roteador tem rotas para AS externos, que são anunciadas através do Sistema Autônomo. O caminho para cada roteador de borda é conhecido por todos os roteadores no interior do Sistema Autônomo. O roteador de borda usa, por exemplo, protocolo BGP para fazer roteamento entre Sistemas Autônomos.

Um Roteador de Borda de Sistema Autônomo (*Autonomous System Boundary Router - ASBR*) é um roteador que roda múltiplos protocolos e serve como gateway para os roteadores do lado de fora do Sistema Autônomo. O ASBR é capaz de importar e traduzir rotas de diferentes protocolos para OSPF, através do processo de **redistribuição**.

Um determinado protocolo de roteamento redistribui rotas quando faz anúncio de rotas que foram obtidas por outros meios, tais como aprendidas de outro protocolo, rotas estáticas ou conectadas diretamente. O desejável é usar um único protocolo em toda a rede, porém em alguns casos como fusões de empresas, a convivência com multiprotocolos passa a fazer parte do dia a dia. Se no ambiente existem vários protocolos de roteamento, a redistribuição se torna uma necessidade.