Roteamento em redes de computadores

Autor: Valter Roesler
Universidades: UNISINOS e UFRGS

Data: maio de 2001

SUMÁRIO

1 Sistemas Autônomos 3
2 Algoritmos de roteamento 4
2.1 Roteamento por Vetor de Distância 4
2.1.1 Exemplo de funcionamento – cidades 4
2.1.2 Exemplo de funcionamento – real 5
2.1.3 Problemas da contagem ao infinito 6
2.1.4 Soluções para a contagem ao infinito 7
2.2 Estado de Enlace 8
2.2.1 Descobrir seus vizinhos e aprender seus endereços de rede 9
2.2.2 Medir o retardo para cada um dos vizinhos 9
2.2.3 O pacote LSP (*Link State Packet*) 9
2.2.4 Enviar esse pacote a todos outros roteadores (*Flooding*) 10
2.2.5 Calcular o caminho mais curto para cada um dos roteadores 11
2.3 Estado de Enlace x Vetor de Distância 11
3 RIP V1 (Routing Information Protocol Version 1) 12
4 RIP V2 (Routing Information Protocol Version 2) 13
4.1 Autenticação 14
4.2 Route Tag 14
4.3 Subnet Mask 14
4.4 Next Hop 14
4.5 Multicasting 15
4.6 RIP com IPv6 15
5 OSPF (Open Shortest Path First) 15
5.1 O banco de dados OSPF 15
5.1.1 O custo das rotas (métricas) 17
5.2 Os protocolos OSPF 18
5.2.1 O protocolo Hello 18
5.2.2 O protocolo Exchange 19
5.2.3 O protocolo Flooding 19
5.3 Múltiplas áreas no OSPF 19
5.4 OSPF para IPv6 20
6 Comparação RIP-1, RIP-2 e OSPF 20
7 BGP-4 (Border Gateway Protocol 4) 20
7.1 Sessão BGP 21
7.2 Mensagens BGP 21
7.2.1 Mensagens de cabeçalho 22
7.2.2 Mensagens de “Open” 22
7.2.3 Mensagem de KeepAlive 23
7.2.4 Mensagens de “Notification” 23
7.2.5 Mensagens de “Update” 24
7.3 ATRIBUTOS BGP 25
7.3.1 ORIGIN 26
7.3.2 AS_PATH 26
7.3.3 NEXT_HOP 27
7.3.4 MULTI_EXIT_DISC 27
7.3.5 LOCAL_PREF 27
7.3.6 AGGREGATOR 27
8 Constraint Based Routing 27

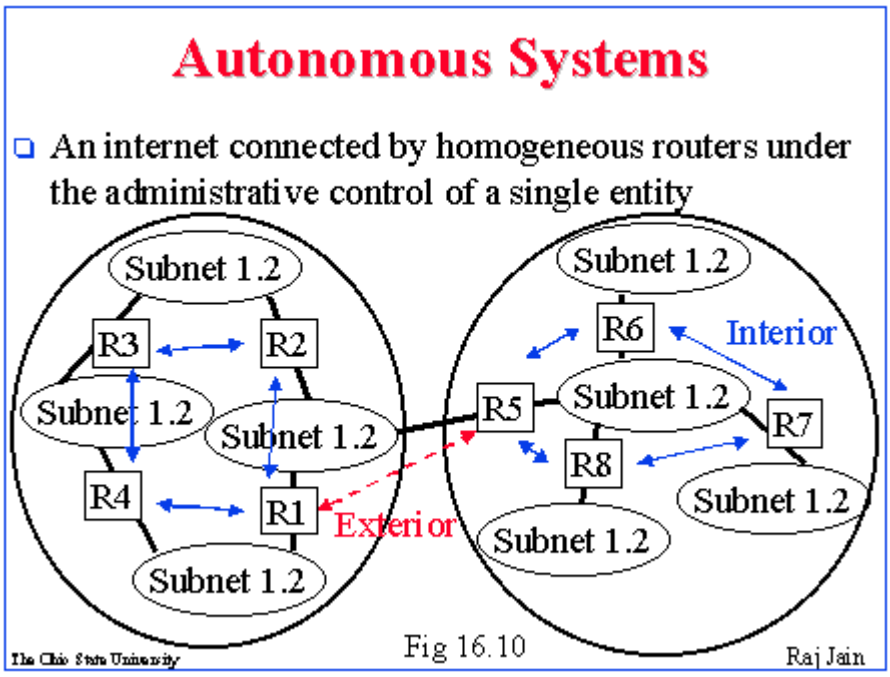
9 Bibliografia 27

1 Sistemas Autônomos

Internet é um conjunto de sistemas autônomos interligados, ou seja, uma interligação de vários domínios. Domínio é a rede interna de uma organização, ou seja, é uma coleção de estações e roteadores administrados por uma única entidade.

Cada domínio ou área possui um número de identificação AS (*Autonomous System*), que identifica todos os roteadores de uma área. O espaço de números de AS é finito, sendo definido atualmente como um inteiro de 16 bits (65536 números). O IANA reservou o bloco de **64512 a 65535 para uso privado** (não podem ser anunciados na Internet) [RFC 1930].

Existem duas classes principais de protocolos de roteamento, os IGP (*Interior Gateway Protocols*), e os EGP (*Exterior Gateway Protocols*), como mostra a figura a seguir.



3

Os protocolos IGP mantêm a comunicação entre os roteadores limitada ao domínio, ou seja, o flooding e a comunicação entre vizinhos ficam contidos na área. Exemplos de protocolos IGP são o **RIP** (*Routing Information Protocol*) e o **OSPF** (*Open Shortest Path First*), que utilizam, respectivamente, algoritmos **de vetor de distância e estado de enlace** para descobrir a topologia da rede.

Os protocolos EGP são necessários para que exista comunicação entre domínios, ou seja, deve haver um roteador de borda entre as áreas. Exemplos de protocolos EGP são o BGP (*Border Gateway Protocol*) e IDRP (*Interdomain Routing Protocol*).

É necessária uma diferenciação entre protocolos IGP e EGP, pois os objetivos não são os mesmos. Um protocolo IGP se preocupa com a melhor rota até a rede de destino, sem pensar em política. Já um protocolo EGP assume que empresas diferentes, organizações diferentes e até países diferentes precisam se comunicar, e existem certas políticas que devem ser respeitadas, até mesmo por questões econômicas e de segurança, como, por exemplo [TAN 96, pg 489].

- Não utilizar determinado AS;
- Nunca colocar o Iraque numa rota que começa no pentágono;
- Tráfego que começa ou termina numa empresa não deve passar pela concorrente.

2 Algoritmos de roteamento

Existem duas formas de um roteador ser configurado: através de **roteamento estático** e **roteamento dinâmico**. No roteamento estático, um administrador configura manualmente a rota. Uma delas configurada sempre é a rota *default*. Como provocam um trabalho considerável, o roteamento estático é adequado para rotas que não mudam com muita freqüência.

O roteamento dinâmico depende de algoritmos para descobrir o melhor caminho. Basicamente, existem dois tipos de algoritmos para roteamento dinâmico em redes IGP: o roteamento por **Vetor de Distância** e o roteamento por **Estado de Enlace**.

Estes algoritmos aprendem como alcançar outras redes através da troca de informações com outros roteadores, sabendo assim para onde encaminhar pacotes de uma rede para outra.

2.1 Roteamento por Vetor de Distância

No roteamento com vetor de distância, cada roteador possui uma tabela com a melhor distância conhecida aos diversos destinos alcançáveis, bem como a saída a ser usada para chegar lá. Essa tabela é atualizada a partir de troca de informações com os vizinhos, que informam sobre seus vizinhos, e assim por diante.

Esse algoritmo também é conhecido pelo nome de seus desenvolvedores, ou **Bellman-Ford** (1957) e **Ford-Fulkerson** (1962), e foi usado originalmente na Internet com o nome de RIP [TAN 96, pg 406].

A métrica permitida pelo algoritmo pode ser número de hops (usado no RIP), retardo de tempo em milisegundos, número de pacotes na fila ou algo semelhante. A seguir será feito um exemplo ilustrativo de funcionamento do protocolo e em seguida um exemplo mais real.

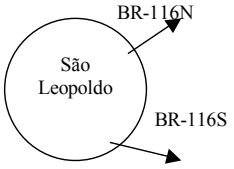
2.1.1 Exemplo de funcionamento – cidades

Imagine pessoas em cidades diferentes com a responsabilidade de saber como chegar em outras cidades e qual o melhor caminho a seguir. Para isso, supondo que a métrica para chegar ao destino seja a distância mais curta (poderia ser a melhor estrada, por exemplo), eles têm que saber a distância em quilômetros até cada um dos destinos.

Imagine também que existe um sistema de comunicação **somente entre cidades vizinhas**, ou seja, cada uma dessas pessoas **só consegue falar com seus vizinhos**, e é através deles que vai circular a informação de como chegar a todas as cidades existentes.

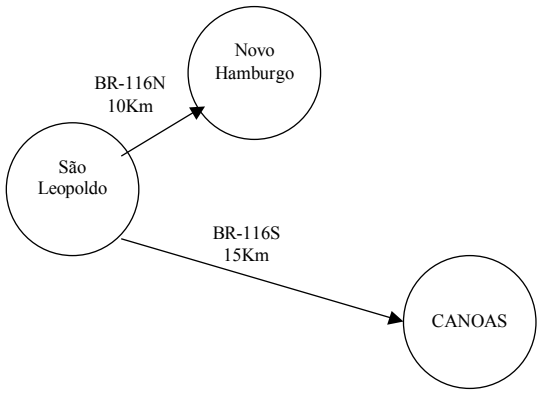
No exemplo a seguir, tudo que a pessoa sabe inicialmente é que ela se encontra em São Leopoldo (custo 0 –local) e a cidade possui duas saídas, uma é a BR-116 e outra é a RS-x. Assim, constrói uma tabela com essa informação.

Tabela SL		
Destino	**Custo**	**Rota**
SL	0	-



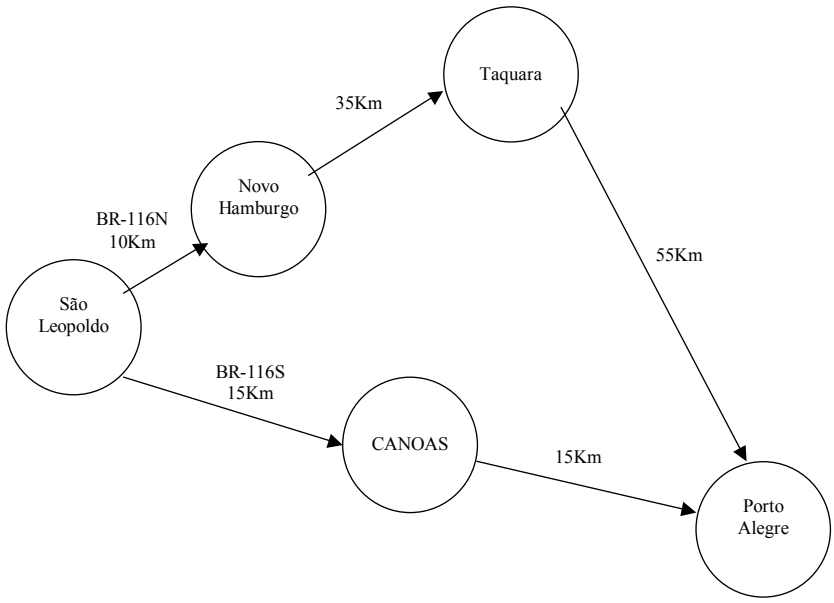
Em seguida, as duas cidades vizinhas enviam suas tabelas. Suponha que seja possível medir a distância entre os vizinhos através de mensagens especiais. Assim, a tabela de São Leopoldo é incrementada com Novo Hamburgo, a 10 km e Canoas, a 15 km.

Tabela SL		
Destino	**Custo**	**Rota**
SL	0	-
NH	10	BR-116N
CAN	15	BR-116S



Conforme o tempo passa, as outras cidades vão aprendendo rotas novas através da troca de informações com os vizinhos. Por exemplo, São Leopoldo agora informa a Novo Hamburgo que pode chegar em Canoas. Na figura a seguir, SL recebeu de NH distâncias para chegar em Porto Alegre e Taquara, e também recebeu de Canoas as distâncias para chegar em Porto Alegre. Como a distância mais curta para Porto Alegre é via Canoas, a tabela fica de acordo com a rota mais curta.

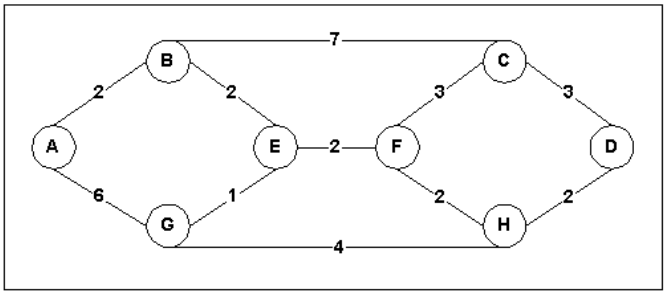
Tabela SL		
Destino	**Custo**	**Rota**
SL	0	-
NH	10	BR-116N
CAN	15	BR-116S
TAQ	45	BR-116N
POA	30	BR-116S



Cada roteador tem um ID e conhece o custo com seus vizinhos. De tempos em tempos, recebe de todos os vizinhos tudo que os vizinhos conhecem. Com essas informações, ele atualiza sua tabela com a menor distância e a rota de saída para chegar num destino.

2.1.2 Exemplo de funcionamento – real

Imagine a topologia de rede ilustrada na figura a seguir, onde a métrica é o retardo, e cada roteador mediu previamente o retardo através de pacotes de Echo, por exemplo.



O roteador A, quando é iniciado, monta uma tabela das rotas conhecidas. Esta tabela é enviada para todos os links do roteador.

Roteador A	
Destino	**Custo**
A	0
B	2
G	6

Cada roteador troca suas tabelas com os roteadores vizinhos de tempos em tempos. As novas rotas recebidas são calculadas e adicionadas. No exemplo abaixo, o roteador A recebe a tabela de G, gerando a nova tabela de A. Note que as rotas E e H foram adicionadas e um custo de 6 foi adicionado a cada rota recebida por causa do custo do link entre A e G.

Roteador G	
Destino	**Custo**
G	0
A	6
E	1
H	4

Roteador A		
Destino	**Custo**	**Rota**
A	0	-
B	2	B
G	6	G

Nova tabela de A com o que aprendeu de G

Roteador A		
Destino	**Custo**	**Rota**
A	0	-
B	2	B
G	6	G
E	6+1 = 7	G
H	6+4 = 10	G

Quando o roteador A receber a atualização de B, sua tabela fica da seguinte forma:

Roteador B	
Destino	**Custo**
A	2
B	0
C	7
E	2

Roteador A		
Destino	**Custo**	**Rota**
A	0	-
B	2	B
E	7	G
G	6	G
H	10	G

Nova tabela de A com o que aprendeu de G e B

Roteador A		
Destino	**Custo**	**Rota**
A	0	-
B	2	B
C	2+7=9	B
E	2+2=4	B
G	6	G
H	10	G

A nova rota C foi encontrada. Note que a rota para E também foi modificada, pois existe um caminho com custo menor por B.

2.1.3 Problemas da contagem ao infinito

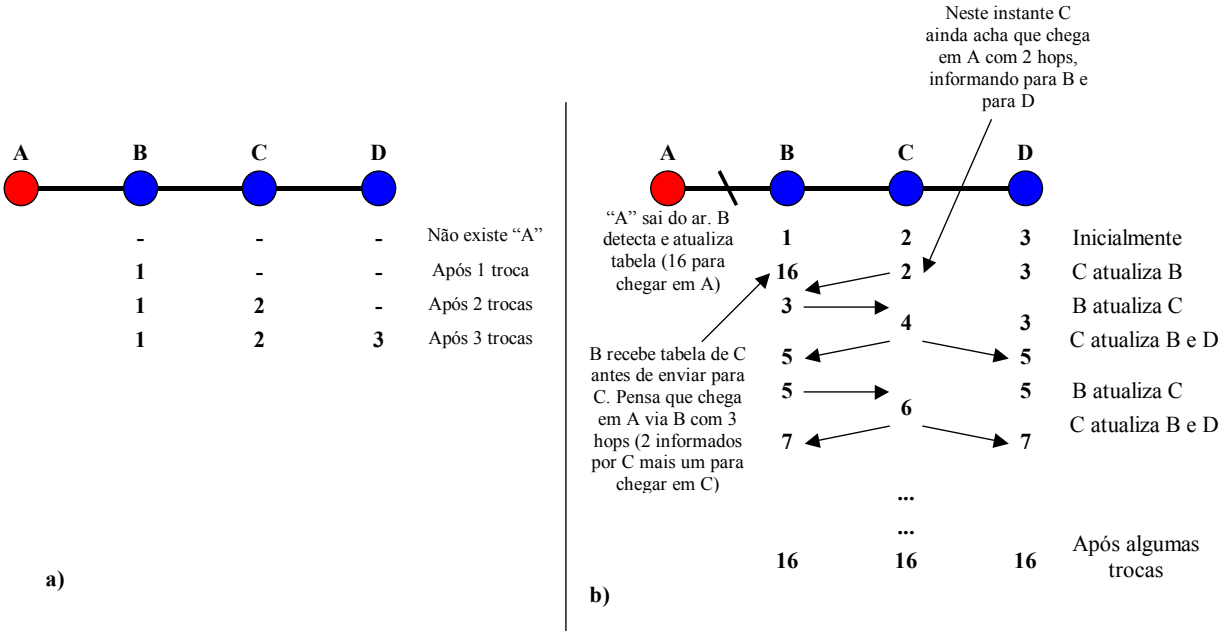
O algoritmo de vetor de distância tem um problema na velocidade de convergência quando ocorrem problemas na rede. Pode-se dizer que as boas notícias convergem rápido, mas as más notícias demoram a convergir. Para exemplificar, a figura a seguir (item “a”)

mostra as distâncias de todos os roteadores "recém ligados" para A. Pode-se ver que, em 3 atualizações da tabela, todos ficaram sabendo da sua existência.

Por outro lado, na figura a seguir item "b", pode-se ver que falhas na rede demoram a ser eliminadas, podendo ocasionar loops, que são pacotes que ficam sendo repassados de um roteador para outro durante certo tempo. Para o pacote não ficar em loop infinito em uma rede, todo o pacote possui um contador TTL.

O contador TTL (*Time to Live*) de cada pacote começa com um valor padrão de 15 e vai sendo diminuído à medida que passa por um roteador. Assim, se uma rede estive em loop, o TTL chegará 0 e o roteador irá descartar o pacote.

No exemplo a seguir (item "b"), quando A sai do ar, B atualiza a distância para "16", entretanto, recebe em seguida a atualização de C dizendo que sabe chegar em A com 2 hops. Aí B atualiza sua tabela para "2+1" = "3". Quando B atualizar C, C vai passar sua distância até A para "3+1"="4", e assim por diante. Vai levar em torno de 7 ciclos de atualizações para que todos saibam que A está fora, e isso dá mais de 3 minutos (considerando 30 segundos para cada ciclo de atualização).



2.1.4 Soluções para a contagem ao infinito

- ***Hold Down*** – Se um link "falhar", o roteador ignora todas atualizações para aquela rede por um tempo de, por exemplo, 180s. Assim, todos roteadores esquecem o caminho até "C", e um link quebrado não será propagado. Este tempo pode ser um problema para os pacotes sendo transmitidos para aquela rede **caso exista** outra rota alternativa.
- ***Poison Reverse and Triggered Updates*** – Se um link "falhar", modifica a informação da rota para uma distância infinita (16) e propaga **imediatamente** essa informação adiante (não dá tempo de receber a atualização de outro nó). Assim, se houver alguma outra rota para C, essa rota será utilizada, pois será melhor que uma distância infinita.

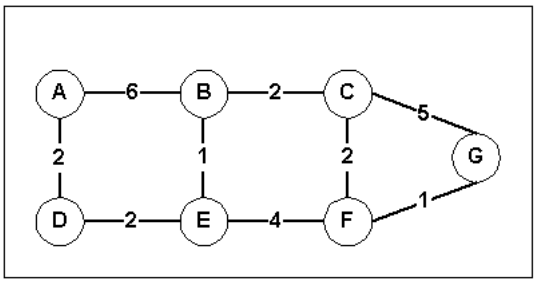
- **Dividir o Horizonte** – O roteador sempre propaga todas as rotas conhecidas, menos as rotas que foram recebidas pela mesma porta. Baseando-se na figura anterior, quando **C** anunciar as suas rotas, **ele não anunciará para B o que ele recebeu de B**. Veja o exemplo baseado no problema do loop.
  a) B nota uma falha em A, atualizando sua tabela para "16".
  b) Quando C envia sua tabela, não informa a rota de A, pois a havia recebido de B. Assim, não vai criar falsas expectativas de rotas, e quando B enviar sua atualização, C automaticamente passa para "16" o custo para chegar em A.

2.2 Estado de Enlace

O algoritmo de estado de enlace elimina alguns problemas críticos do vetor de distância, como a lenta convergência no caso de problemas. A idéia básica do algoritmo de estado de enlace é que cada roteador faça o seguinte [TAN 96]:

1. Descobrir seus vizinhos e aprender seus endereços de rede
2. Medir o retardo para cada um dos vizinhos
3. Criar um pacote que diga tudo o que acaba de ser aprendido. Cada roteador constrói um pacote chamado de "*Link State Packet*" ou LSP, que contém o seu nome, o nome de seus vizinhos e o custo necessário para chegar até eles.
4. Enviar esse pacote a todos outros roteadores
5. Calcular o caminho mais curto para cada um dos roteadores.

Estes itens serão analisados com mais detalhes a seguir, com base na seguinte topologia:



O objetivo é que cada roteador envolvido tenha um banco de dados completo de toda a topologia da rede, para conseguir traçar o caminho mais curto através de um algoritmo, como o de Dijkstra, por exemplo. Assim, para a topologia acima, cada roteador deve ter um banco de dados semelhante ao mostrado na tabela a seguir [HUI 99, pg 120]. Este banco de dados deve ser o mesmo em todos os roteadores, a fim de que todos tomem as mesmas decisões.

Origem	Destino	Custo
A	B	6
A	D	2
B	A	6
B	C	2
B	E	1
C	B	2
C	F	2
C	G	5
D	A	2
D	E	2
E	B	1
E	D	2
E	F	4
F	C	2
F	E	4
F	G	1
G	C	5
G	F	1

2.2.1 Descobrir seus vizinhos e aprender seus endereços de rede

Para o roteador saber quem são seus vizinhos, **pacotes Hello** (multicast para 224.0.0.5 = "todos roteadores SPF") são enviados para as portas de tempos em tempos. Se um roteador recebe um pacote Hello ele responde com outro pacote contendo seu nome. Os nomes dos roteadores não podem ser duplicados [TAN 96, pg 411]. Os pacotes Hello também são utilizados para saber se um link está operacional [HUI 99, pg. 145]. Maiores detalhes no estudo do OSPF.

2.2.2 Medir o retardo para cada um dos vizinhos

A forma mais simples de medir o retardo é enviar pacotes de ECHO para o vizinho e esperar resposta. A média de vários tempos de resposta dividida por dois é uma estimativa do retardo [TAN 96, pg 411]. O tamanho da fila e a carga na rede também podem ser levados em consideração.

No algoritmo de estado de enlace, outras métricas podem ser medidas e levadas em consideração, como a velocidade das linhas, sua segurança, e assim por diante. No item de comparação entre os algoritmos de estado de enlace e vetor de distância isso será mais detalhado.

2.2.3 O pacote LSP (*Link State Packet*)

Os roteadores utilizam o pacote LSP (*Link State Packet*) para trocar as informações de rotas com os outros roteadores. Seu formato é mostrado na tabela a seguir (para roteador B).

Identidade do Transmissor	
Número de seqüência	
Idade (Tempo)	
Destino	**Custo**
A	6
C	2
E	1

A identidade do transmissor deve identificar o mesmo univocamente na área em que ele se encontra, para evitar conflitos de informações (e.g. número IP). O número de seqüência e a

idade ficarão mais claros no próximo item, que trata da problemática de enviar este pacote a todos os outros roteadores da rede.

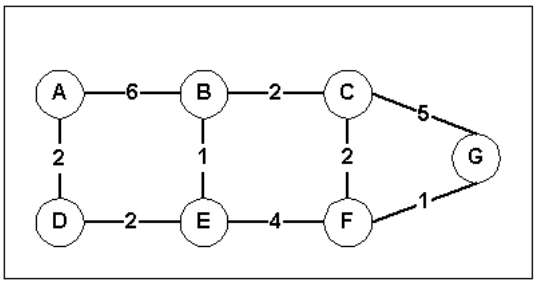
2.2.4 Enviar esse pacote a todos outros roteadores (*Flooding*)

Um pacote é mandado quando um roteador **descobre um novo vizinho**, **o custo de um link muda**, **um link cai** ou **passa determinado tempo** (30minutos no caso do OSPF).

Como cada LSP **deve ser enviado a todos os outros roteadores na rede**, utiliza-se flooding (inundação), onde cada pacote recebido é mandado para todas as portas, exceto a porta em que veio.

Para o flooding não se propagar ao infinito, gerando uma explosão de pacotes, pode ser usado um contador TTL com um limite de hops que é decrementado a cada roteador, e quando chega a zero, é descartado. O ideal é que o TTL seja inicializado com o comprimento do caminho da origem ao destino [TAN 96, pg. 400]. **No caso do OSPF, utiliza-se um número de seqüência ao invés do TTL, pois é mais eficiente, conforme descrito a seguir**.

Um pacote flooding pode ser recebido mais de uma vez em um roteador. Uma vez que cada pacote LSP enviado possui um número de seqüência diferente do anterior, o roteador sabe se o mesmo pacote foi recebido mais de uma vez, então, ele não repassa os pacotes repetidos. Caso receba um pacote **mais antigo** que o que têm no banco de dados, envia o pacote mais atual (do banco de dados) de volta para a porta pela qual recebeu, a fim de atualizar o roteador que está propagando rotas mais antigas. Caso receba um pacote com um número de seqüência maior do que o existente no banco de dados, **atualiza o banco de dados** com o pacote mais novo **e envia o pacote para todas suas portas**, menos pela qual o pacote foi recebido [HUI 99, pg 121].



Por exemplo, suponha que o link entre A e B tenha caído. A deve enviar essa mudança a todos os roteadores, e B deve fazer o mesmo. Supondo que o número de seqüência inicial tenha sido "1", A vai enviar a seguinte informação para D (única porta ativa).

Origem	Destino	Custo	Seqüência
A	B	infinito	2

O roteador D vai enviar esta informação para todas as portas menos A, ou seja, para E, que vai enviar para B e F. B envia a informação para C (única porta que sobrou em B que não é a que o pacote chegou). F envia a informação para G e C (que recebe duas vezes a mesma informação, descartando uma). Dependendo por onde C recebe primeiro o pacote, ele envia para as outras portas. Supondo que ele receba primeiro via B, ele envia a informação para F e G. F também pode receber o pacote por portas diferentes, enviando para as outras portas da primeira vez e descartando quando receber duplicado. G envia a informação para a porta pela qual o pacote não chegou.

Caso o link entre D e E caia também, o mesmo acontece, mas existirão duas redes separadas se atualizando independentemente. Caso um link volte ao ar, haverá a necessidade

de uma troca completa de informações, pois as tabelas estarão inconsistentes. No OSPF, isso é chamado "*bringing up adjacencies*" [HUI 99, pg 122].

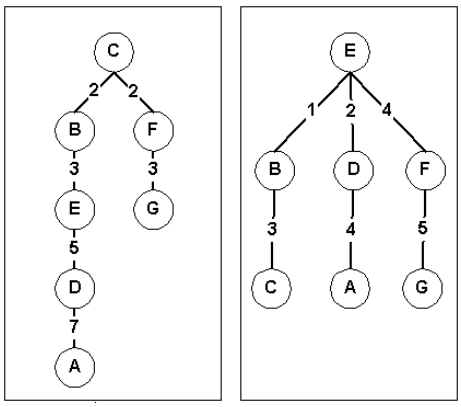
Além de manter a consistência, o banco de dados deve ser protegido contra corrupção acidental. Por exemplo, caso ocorra um erro de memória ou de transmissão, levando o número de seqüência para um número elevado, todas as atualizações de rota serão descartadas, pois serão sempre mais antigas que o número corrupto. Para evitar isso, o algoritmo do OSPF utiliza alguns mecanismos de proteção, como os seguintes [HUI 99, pg 124]:

- O procedimento de flooding inclui um reconhecimento (ACK) de cada mensagem enviada;
- Cada LSP é protegido por um timer, e é removido do banco de dados caso este timer expire;
- Todas transmissões são protegidas por checksum;
- As mensagens podem ser autenticadas por senhas.

2.2.5 Calcular o caminho mais curto para cada um dos roteadores.

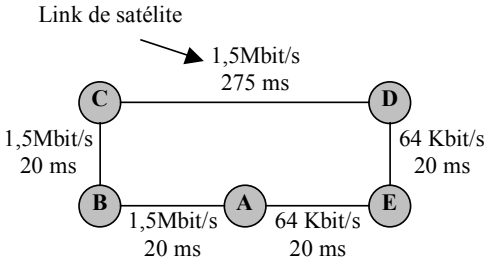
Após as informações serem distribuídas por flooding, o algoritmo de Dijkstra pode ser usado para encontrar o caminho mais curto para cada um dos outros roteadores.

Cada roteador, tendo posse dos LSP´s de todos os outros roteadores de sua área, deve construir uma árvore lógica para chegar a qualquer outro roteador pelo caminho com menor custo. A figura a seguir mostra os caminhos mais curtos (onde a métrica é a distância) para os roteadores C e E.



2.3 Estado de Enlace x Vetor de Distância

- **Suporte a múltiplos caminhos** – É possível no Estado de Enlace, e o roteador pode utilizar mais de um caminho simultaneamente (o que pode ocasionar entrega de pacotes fora de ordem, mas isso é consertado no nível TCP), deixando o fluxo mais rápido. O vetor de distância suporta apenas um caminho, o de menor custo;
- **Suporte a múltiplas métricas** – Como cada roteador tem consciência de toda a rede, basta a medição de outras métricas para que o algoritmo consiga decidir pelo menor custo dependendo da necessidade da aplicação. Por exemplo, suponha que na figura a seguir exista uma aplicação de voz e outra de dados partindo de A com destino a D. O melhor caminho para a aplicação de voz seria o de menor atraso (A-E-D), enquanto que para a de dados seria o de maior vazão (A-B-C-D). O OSPF versão 2 suporta os 4 mecanismos definidos no campo IP TOS (*Type Of Service*), que são: maior vazão, menor atraso, menor custo e maior segurança [HUI 99, pg 127];



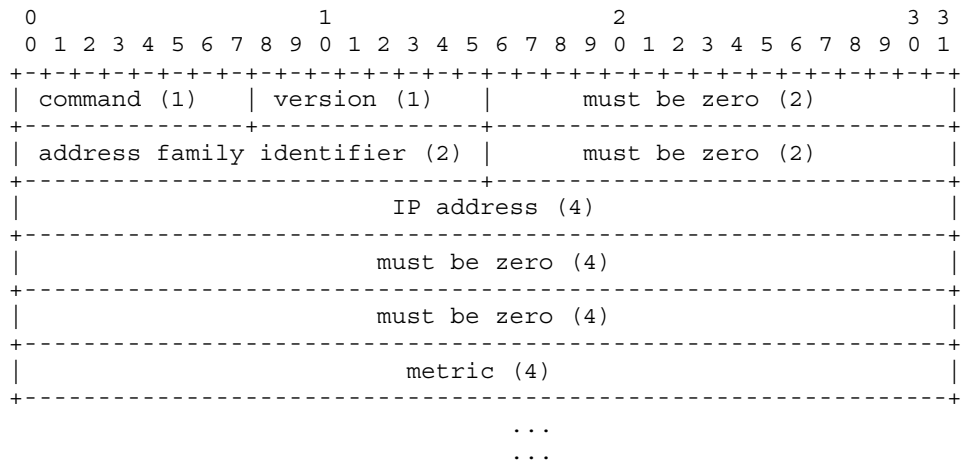
- **Banda Consumida** – O algoritmo de Estado de Enlace é melhor, pois as atualizações ocorrem somente quando algo muda na rede, e não a cada 30 segundos, por exemplo. Além disso, o Estado de Enlace necessita enviar somente as informações dos vizinhos, e não toda sua tabela;
- **CPU** – O Estado de Enlace gasta mais, pois deve calcular a árvore de menor custo;
- **Velocidade de Convergência** – O Estado de Enlace é melhor, pois a cada alteração na rede, essa informação se propaga imediatamente a todos os nós. Isso dá uma vantagem adicional, ou seja, **evita loops na rede**.

3 RIP V1 (Routing Information Protocol Version 1)

O RIP é um protocolo do tipo IGP destinado a redes de tamanho limitado. Foi criado inicialmente pela Xerox e baseado no algoritmo de vetor de distância visto anteriormente. Está definido na RFC 1058 e implementado no Unix Berkeley como "routed". Quando implementado em hosts, eles utilizam o modo passivo (somente ouvem as rotas e não enviam atualizações). Suas características são:

- Utiliza o número de hops como métrica;
- Comunica-se com seus vizinhos a cada 30s. Se uma rota não é re-anunciada em 180s, ela é considerada como infinita e removida da tabela;
- Transmite toda tabela de rotas para vizinhos sempre (e não só alterações da tabela), o que gera bastante tráfego;
- Distância máxima de 15 hops. Acima disso o destino é considerado inalcançável;
- Utiliza protocolo UDP (porta 520) com pacotes de 576 bytes, o que demanda múltiplos pacotes para enviar a tabela inteira. Por exemplo, uma tabela com 300 entradas necessita de 12 pacotes;
- Utiliza sempre o caminho mais curto, impossibilitando engenharia de tráfego ou roteamento baseado em QoS;
- Possui uma convergência lenta, pois uma nova rota vai sendo enviada a cada 30 segundos de vizinho em vizinho, demorando a atingir um destino distante;

O formato do pacote RIPv1 pode ser visto na figura a seguir. A parte do cabeçalho do identificador da família do endereço até a métrica pode se repetir até 25 vezes [RFC 1058].



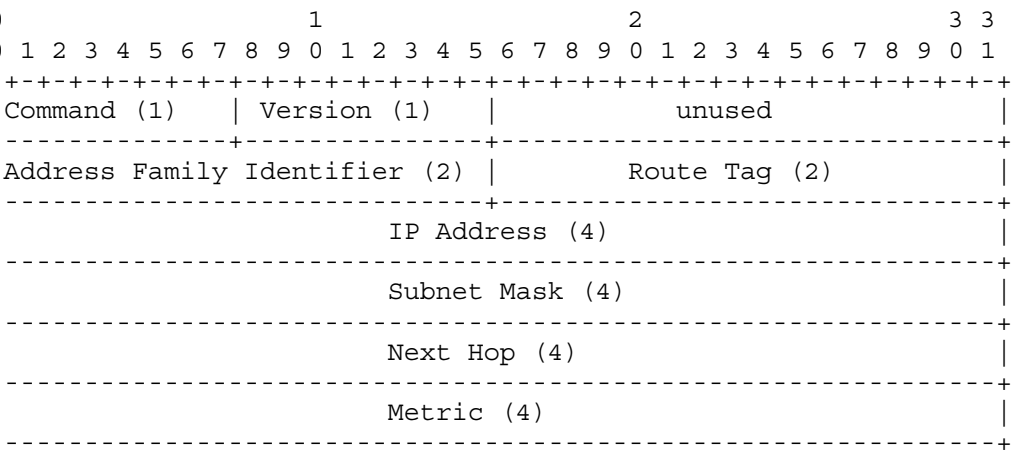
Os comandos mais comuns são **Comando 1**: requisição de tabela de roteamento e **Comando 2**: resposta contendo tabela. A família da rede é o número do protocolo, pois o RIP não foi feito somente para IP (A família do IP é a de número 2). O endereço informando a rede pode ter até 14 bytes (IP usa apenas 4). A métrica da rede pode conter números de 1 a 16, sendo que 16 é igual a infinito (destino inalcançável). Os campos reservados devem ser zero.

4 RIP V2 (Routing Information Protocol Version 2)

O RIP versão 2 está definido na RFC 1723, e possui compatibilidade com o RIP v1, porém, adiciona uma série de melhorias, como as descritas a seguir:

- **Autenticação**: Se uma família de rede é 0xFFFF, a primeira rota é uma senha;
- **Máscara de subrede**: útil para *classless*;
- **Rótulo de rota**: permite rotas aprendidas externamente, como, por exemplo, com BGP;
- **Next Hop**: endereço do próximo roteador para cada entrada de rota. Útil para redes locais com múltiplos roteadores;
- **Multicast**: RIPv2 usa o IP multicast 224.0.0.9, que equivale ao MAC 01-00-5E-00-00-09. O RIP v1 utiliza broadcast.

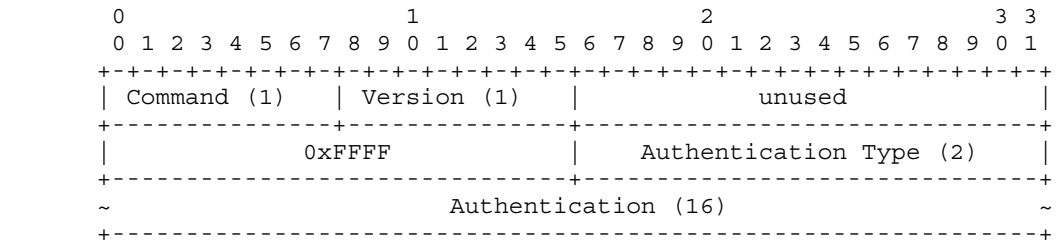
O formato do pacote RIPv2 pode ser visto na figura a seguir:



Os campos "*command*", "*Address Family Identifier*", "IP *address*" e "*metric*" possuem o mesmo significado da RFC 1058. O campo versão deve ser "2" nas mensagens que usam autenticação ou informações sobre os campos novos.

4.1 Autenticação

Para que o cabeçalho seja de autenticação, o campo "*Address Family Identifier*" deve ser 0xFFFF, e o restante do cabeçalho contém a autenticação, conforme mostrado na figura a seguir.



O único tipo de autenticação definida na RFC 1723 é a senha simples (tipo 2). Os 16 bytes seguintes contêm a senha em texto aberto.

4.2 Route Tag

O atributo *Route Tag* (RT) tem como objetivo separar rotas **internas** ao RIP de rotas **externas** ao RIP (anunciadas de outro IGP ou EGP) [RFC 1723].

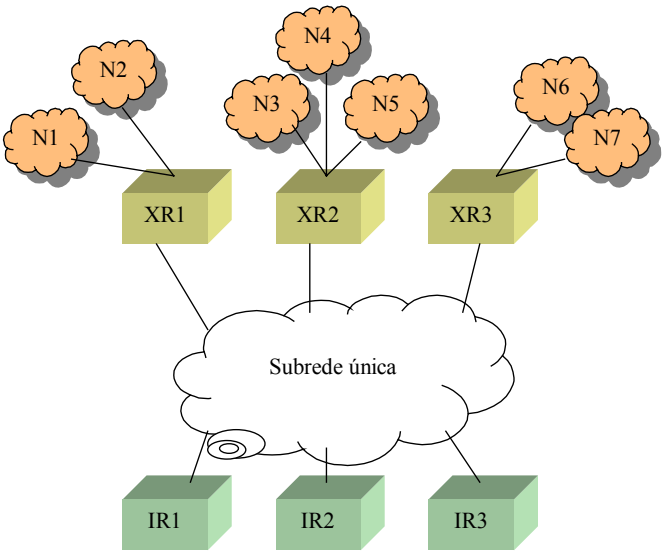
4.3 Subnet Mask

A máscara de subrede permite adaptar o esquema *classfull* de endereços IP para o uso de *classless*, ou seja, baseado em máscara. Se for zero, não existe máscara de subrede associada a este endereço IP.

4.4 Next Hop

O campo "*Next Hop*" indica o próximo endereço IP para o qual determinado pacote deve ser redirecionado para chegar no destino informado no cabeçalho. Os endereços do "*next hop*" devem, obrigatoriamente, ser alcançados diretamente através da subrede pela qual o anúncio é feito.

No exemplo a seguir, IR1 até IR3 são roteadores internos. XR1 até XR3 são roteadores de outra área, com outra administração. Configurando corretamente, o roteador XR3 seria o "next hop" para chegar nas redes N6 e N7, XR2 para as redes N3, N4 e N5, e XR1 para as redes N1 e N2. Caso um dos roteadores externos não enviasse o campo "next hop", seriam necessários pacotes adicionais com IR1, IR2 e IR3 informando que ele seria o responsável para chegar em N1 e N2 [RFC 1723, apêndice A].



4.5 Multicasting

A fim de reduzir carga desnecessária de CPU para hosts que não querem receber mensagens de roteamento, foi definido o grupo multicast 224.0.0.9 (roteadores RIP2 [RFC 1700]). Observe que IGMP não é necessário, pois essas mensagens não são redirecionadas, ficando na rede local.

4.6 RIP com IPv6

/**/ [HUI 99] pg 109.

5 OSPF (Open Shortest Path First)

O OSPF utiliza como base o algoritmo de estado de enlace, está definido na RFC 2178, e possui as seguintes características:

- Usa outras métricas além da contagem de hops;
- Permite máscara de subrede;
- Permite balanceamento de carga em caminhos de igual custo;
- Suporta Type of Service do IPv4;
- Autentica troca de rotas;
- Aprende de rotas externas (vindas de outros sistemas autônomos);
- Possui rápida convergência (que é um dos problemas do RIP);
- Roda sobre TCP. Todos LSAs enviados (via flooding) possuem reconhecimento com ACK. Se não veio ACK, o LSA é enviado novamente (via flooding).

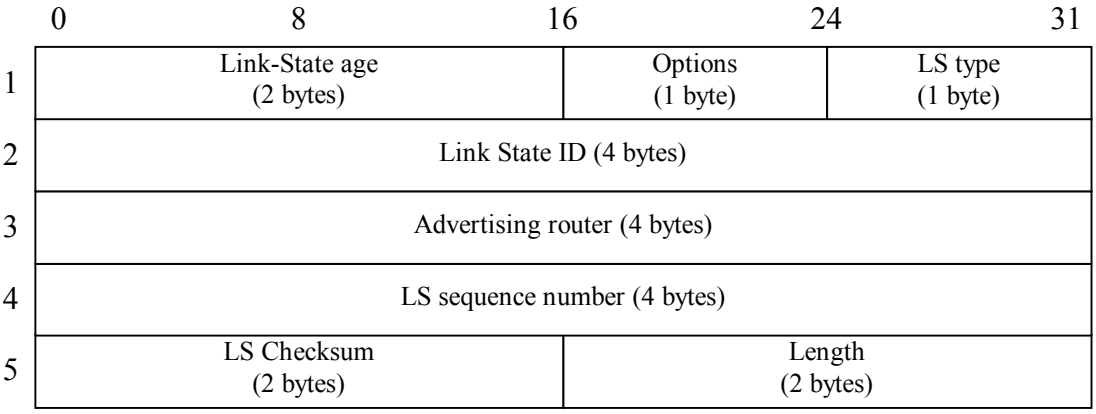
A informação é enviada novamente (*flooding*) toda vez que um roteador **descobre um novo vizinho**, **o custo de um link muda**, **um link cai** ou **passa determinado tempo** (30 minutos no caso do OSPF).

5.1 O banco de dados OSPF

Cada roteador OSPF de mesma área utiliza um banco de dados representando a topologia da rede (de acordo com o algoritmo de estado de enlace), e esses dados são

utilizados para encontrar o melhor caminho dependendo da métrica necessária para a aplicação [HUI 99, pg 138].

A troca de informações entre os roteadores para a atualização da base de dados se dá através da troca de pacotes de estado de enlace, ou LSPs (*Link-State Packets*). O cabeçalho é comum a todos, e mostrado na figura a seguir.



Os campos vistos na figura têm o seguinte significado [HUI 99, pg138]:

- **Link-State age**: serve para eliminar rotas "velhas" da base de dados, e indica o tempo em segundos desde que aquela rota foi anunciada pela primeira vez. O roteador envia a rota com idade zero. Durante o flooding, cada roteador por onde a mensagem passa incrementa de um esse valor, e depois vai incrementando de segundo em segundo. Quando ele atinge "MaxAge" (uma hora), o registro é considerado obsoleto e descartado. Entretanto, como todos os roteadores devem ter a mesma cópia do banco de dados, esse processo deve ser síncrono, portanto, é anunciado via flooding [HUI 99, pg 151]. Para garantir que os registros não fiquem velhos, os roteadores devem reenvia-los a intervalos regulares, no mínimo a cada 30 minutos. Este novo registro deve possuir o mesmo conteúdo, mas com o número de seqüência incrementado e a idade em zero.
- **Options**: a versão 2 do OSPF define somente dois bits deste campo:
  - **T – bit "0":** indica que o roteador suporta o uso do campo "type of service" definido na RFC 1349. Isso será detalhado a seguir.
  - **E – bit "1"**: indica que o roteador é capaz de enviar ou receber rotas externas, ou seja, este é um roteador de borda.
- **LS type**: determina o tipo do registro que está sendo trocado entre roteadores. Pode assumir um dos 5 valores a seguir:
  1. **router link**: enviados pelos roteadores a fim de definir os custos dos links que começam no próprio roteador (é o tipo mais comum);
  2. **network link**: são enviados pelos roteadores designados com destino a roteadores de redes trânsito;
  3. **summary link (rede IP)**: anunciado pelos roteadores de borda destinado aos roteadores de dentro de uma área, e a diferença para o *router link* é que envia um anúncio separado para cada destino;
  4. **summary link (border router)**: o mesmo que acima só que para o roteador de borda da área;
  5. **external link**: registros enviados por roteadores de borda e destinados a roteadores externos ao AS.

- **Link-State ID**: é a identificação **do link** escolhida pelo roteador (normalmente um número IP). Pela definição, os campos *Link-State ID* mais o *Advertising router* mais o *LS type* devem identificar univocamente o registro.
- **Advertising router**: é a identificação do roteador, ou seja, seu número IP.
- **LS sequence number**: é o número de seqüência necessário para que o roteador saiba se o pacote que está chegando é mais atual ou mais antigo que o existente no banco de dados. A utilidade deste campo foi explicada em maiores detalhes anteriormente, na explicação do flooding do algoritmo de estado de enlace.
- **LS Checksum**: é usado para proteger os dados e o cabeçalho (menos o campo LS Age, a fim de que o checksum não tenha que ser modificado toda vez que o registro passa por um roteador).
- **Length**: é o tamanho total do registro em bytes, incluindo os cabeçalhos.

5.1.1 O custo das rotas (métricas)

O caminho mais curto do OSPF pode ser baseado no tipo de serviço (campo TOS da RFC 1349). Caso o roteador tenha suporte a TOS, ele deve obter uma tabela de custo separada para cada métrica, e dependendo da aplicação, utilizar a métrica correspondente. A definição do tipo de métrica é feita de acordo com a tabela a seguir [HUI 99, pg 143]:

Código OSPF	Código RFC 1349	Definição
0	0000	Serviço normal
2	0001	Minimiza custo
4	0010	Maximiza segurança
8	0100	Maximiza vazão
16	1000	Minimiza atraso

Assim, caso um pacote IP possua o campo TOS de 1000, a métrica a ser utilizada pelo roteador deve ser a de menor atraso, ou de número 16. Caso o roteador não tenha suporte a TOS, deve zerar o bit "0" do campo "options". Esses campos são codificados no anúncio das rotas, em tabelas definidas através do tipo do registro (LS Type). Maiores detalhes na RFC 2178 pg 176 ou em [HUI 99, pg 140].

Os valores de métrica devem ser inteiros, sendo que os valores menores correspondem às melhores rotas.

A métrica de maximizar a vazão utiliza a fórmula 10^8/ifspeed [RFC 1850, pg 39]. A tabela a seguir especifica alguns valores para a métrica 8.

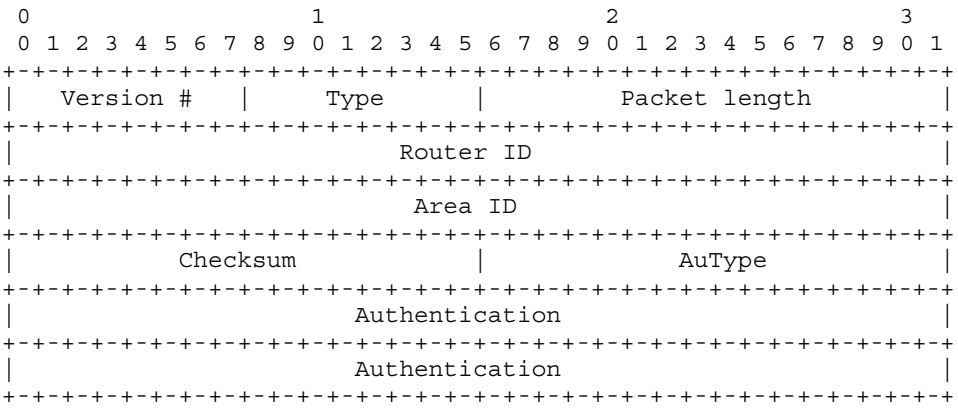
Velocidade	Métrica
9,6 Kbit/s	10.416
56 Kbit/s	1.785
2.048 Mbit/s	48
10 Mbit/s	10
100 Mbit/s e acima	1

Isso cria um problema para redes de alta velocidade, pois não tem como diferenciar um link de 100 Mbit/s de outro de 2,4 Gbit/s.

A métrica para atraso pode ser definida como o número de unidades de 10μs necessário para encaminhar uma mensagem de 1000 bits [HUI 99, pg 143]. Assim, um link de satélite com atraso de 275ms iria gerar um valor de 27.500 para a métrica de número 16.

5.2 Os protocolos OSPF

O OSPF é composto de três subprotocolos básicos: Hello, Exchange e Flooding. Todos eles utilizam um formato único de cabeçalho, visto na figura a seguir [RFC 2178, pg 164].



- ***Version***: número da versão. A RFC 2178 especifica o número 2;
- ***Type***: O tipo do pacote. Os seguintes estão definidos:
    1. **Hello**: Serve para descobrir quem são seus vizinhos;
    2. **Database Description**: Fornece o número de seqüência de todas as entradas de estado de enlace mantidas atualmente pelo transmissor. Comparando com seus próprios valores, o receptor sabe quem tem os valores mais atuais. Essas mensagens são usadas quando uma linha é interrompida (*bringing up adjacencies*);
    3. **Link State Request**: Solicitação de atualização por parte de um roteador a seu vizinho;
    4. **Link State Update**: Mensagens emitidas periodicamente para seus vizinhos, informando seu estado e custo de acesso aos outros vizinhos. Cada mensagem incrementa o número de seqüência;
    5. **Link State Acknowledgment**: Todas as mensagens são reconhecidas para aumentar a segurança.
- ***Packet Length***: número de bytes do pacote;
- ***Router ID***: número IP definido para identificar o roteador;
- ***Area ID***: Identificação da área. A de número 0 (0.0.0.0) é reservada para o backbone. É usual a escolha de um número IP para identificar uma área [HUI 99, pg 144];
- ***Checksum***: é feito para todo pacote OSPF, menos o campo de autenticação;
- ***AuType***: define o tipo de algoritmo de autenticação a ser utilizado. As definições são:
    0. Sem autenticação;
    1. Autenticação simples (senha aberta);
    2. Criptografia (usa checksum MD5).

5.2.1 O protocolo Hello

O protocolo Hello é trocado entre vizinhos para descobrir os vizinhos, ver se eles estão operacionais e eleger o roteador designado (para representar a LAN, por exemplo). Os pacotes devem ser trocados a cada "*hello-interval*" segundos (o default é de 10 segundos) [RFC 1850, pg 44]. Caso eles não cheguem em "*dead-interval*", o link é considerado falho (o default é 60 segundos) [RFC 1850, pg 44].

5.2.2 O protocolo Exchange

Quando dois roteadores estabeleceram uma conexão, eles devem sincronizar seus bancos de dados. Para isso, eles utilizam o protocolo Exchange. A primeira sincronização é feita através deste protocolo, e as seguintes através de flooding.

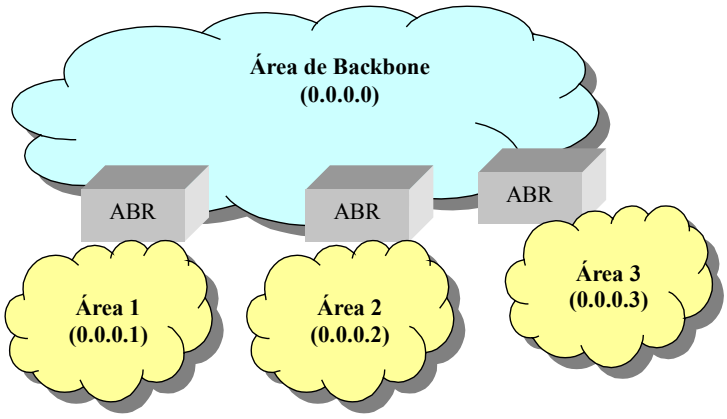
5.2.3 O protocolo Flooding

Cada mudança de estado do roteador (ou após 30 minutos), ele é responsável por enviar essa informação aos outros roteadores da mesma área, através de flooding, com o número de seqüência incrementado em relação ao anterior e a idade (LS Age) igual a zero.

Cada roteador que recebe um pacote de flooding (tipo = 4, update), deve retornar um pacote de reconhecimento (tipo = 5, ack). Isso serve como garantia que o banco de dados fique consistente entre todos os roteadores da área.

5.3 Múltiplas áreas no OSPF

Muitos Sistemas Autônomos na Internet são grandes e difíceis de gerenciar. O OSPF permite que eles sejam divididos em áreas, a fim de diminuir o tráfego de roteamento. Grandes áreas são divididas em áreas menores, e cada área tem uma identificação de 32 bits. A área de backbone é a área 0 (0.0.0.0), e as outras podem ser "0.0.0.1", "0.0.0.2", e assim por diante. As áreas se comunicam entre si via backbone, ou seja, todas as áreas devem estar ligadas ao backbone central, conforme mostra a figura a seguir.



Os roteadores de borda de área (ABR – *Area Border Routers*) sumarizam a topologia e a transmitem para a área de backbone que, por sua vez, retransmitem essa informação para as outras áreas. Esses roteadores rodam algoritmos OSPF para as duas áreas a que pertencem.

Durante a operação normal, podem ser necessários três tipos de rotas: na mesma área (usam algoritmo completo entre eles), entre áreas (usam o ABR para ir e voltar ao backbone) e entre sistemas autônomos (necessitam de protocolo de roteamento externo). Para suportar isso, o OSPF define quatro classes de roteadores [TAN 96, pg 487]:

- **Roteadores internos**: ficam inteiramente dentro de uma área;
- **Roteadores de borda de área (ABRs)**: conectam uma área com o backbone;
- **Roteadores de backbone**: redirecionam pacotes dentro do backbone;
- **Roteadores de fronteira do AS**: interagem com roteadores de outros sistemas autônomos.

5.4 OSPF para IPv6

/**/ [HUI 99], pg. 152

6 Comparação RIP-1, RIP-2 e OSPF

A tabela a seguir resume as diferenças entre RIP-1, RIP-2 e OSPF.

	RIP-1	RIP-2	OSPF
RFC	**RFC 1058**	**RFC 1723**	**RFC 2178**
Algoritmo	**Vetor de distância**	**Vetor de distância**	**Estado de enlace**
Suporte a múltiplas métricas	**Não (hops)**	**Não (hops)**	**Sim**
Suporte a CIDR	**Não**	**Sim**	**Sim**
Suporte a múltiplos caminhos	**Não (o mais curto)**	**Não (o mais curto)**	**Sim**
Máximo diâmetro da rede	**15 Hops**	**15 Hops**	**65.535**
Intervalo de Atualização	**30 segundos**	**30 segundos**	**A cada mudança ou 30min**
Atualizações	**Toda a tabela de rotas**	**Toda a tabela de rotas**	**Somente vizinhos (LSP)**
Intervalo para eliminação	**180s**	**180s**	**?**
Autenticação	**Não**	**Sim (senha aberta)[1]**	**Sim (senha MD5)**
Velocidade de Convergência	**Devagar**	**Devagar ??/**/**	**Rápido**
Suporta AS#	**Não**	**Sim ?? /**/**	**Sim**

1. A RFC 2082 define um esquema para encriptar a senha via MD5 para evitar a transmissão de senhas abertas [HUI 99, pg 106].

7 BGP-4 (Border Gateway Protocol 4)

O BGP é um protocolo entre sistemas autônomos, definido na RFC 1761 e usado desde 1989, entretanto, seu uso cresceu mesmo nos últimos anos. Sua função principal é trocar informações de acessibilidade com outros sistemas BGP com o objetivo de traçar um grafo da conectividade do AS, visando eliminar problemas de laços e reforçar decisões de políticas do sistema (como, por exemplo, a política de anunciar para os AS vizinhos somente as rotas que ele usa, refletindo o paradigma "hop-by-hop" usado na Internet).

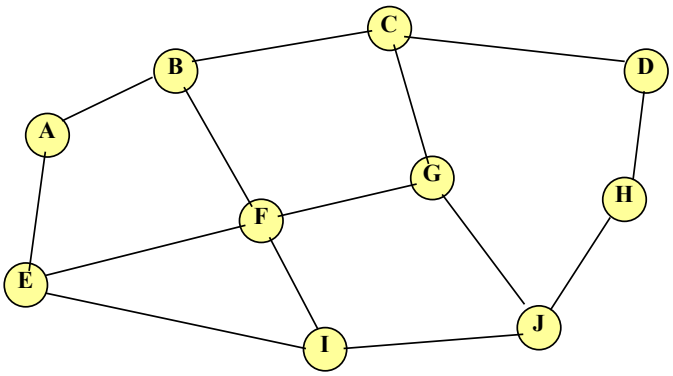
A versão 4 do BGP possui um novo conjunto de mecanismos para suportar CIDR (*Classless Interdomain Routing*), como suporte para anunciar um prefixo IP (*extended network prefix* – e.g. /24) e eliminar o conceito de "classes". Além disso, inclui mecanismos para agregação de rotas [RFC 1761].

O BGP é um dos protocolos destinados a efetuar as políticas tratadas no item de Sistemas Autônomos. Ele é fundamentalmente um protocolo de vetor de distância, porém, ao contrário do RIP, **cada roteador mantém o controle total de cada caminho** (da origem ao destino) utilizado por cada roteador envolvido na rede [TAN 96, pg 490].

Para exemplificar, considere os roteadores mostrados na figura a seguir [TAN 96, pg 491]. Considere o roteador F. Suponha que ele esteja utilizando o caminho FGCD para chegar a D. Os seus vizinhos devem informar a rota completa, como mostra a figura (B, G, I e E).

Depois de receber todas as rotas dos vizinhos, F procura o melhor. Ele descarta as rotas de I e F, pois usam ele mesmo como passagem. Assim, sobra analisar as distâncias para B e

G. Qualquer rota que passe por alguma rede politicamente restrita deve ser evitada (distância marcada como infinito). Após essas eliminações, o roteador usa a rota com distância mais curta.



Informação sobre D que F recebe dos vizinhos

De B: "rota BCD"
De G: "rota GCD"
De I: "rota IFGCD"
De E: "rota EFGCD"

O problema da contagem ao infinito não existe no BGP. Suponha que G falhe ou que a linha FG seja desativada. Rapidamente ele descobre que o único caminho é via FBCD [TAN 96, pg 491].

7.1 Sessão BGP

O BGP roda sobre TCP, porta 179, sendo, portanto, confiável (a aplicação não precisa se preocupar com a confiabilidade). Dois sistemas BGP "vizinhos" abrem uma sessão através de uma conexão de transporte entre eles, trocando mensagens para receber e confirmar os parâmetros da sessão (e.g. tempo máximo de espera de mensagens – *hold time*). Não havendo concordância nos parâmetros, a sessão não é aberta.

O primeiro fluxo enviado é a tabela de roteamento completa, e depois disso, somente atualizações incrementais (mensagens de update) são feitas à medida que as tabelas mudam (e.g. uma rota se tornou inválida ou surgiu uma nova). Isso elimina a necessidade de atualizações periódicas na tabela de roteamento inteira, diminuindo a carga na rede. Portanto, um roteador BGP mantém a versão atual da tabela de todos os seus vizinhos durante toda a duração da conexão.

Cada atualização das tabelas no BGP (mensagens update) provoca um incremento no número de versão, permitindo verificar inconsistências nas informações de roteamento. O exemplo a seguir mostra a verificação da versão da tabela de roteamento [MOU 99].

```
roteador> sh ip bgp
BGP table version is 72076, local router ID is 192.168.4.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
Network Next Hop Metric LocPrf Weight Path
(...)
```

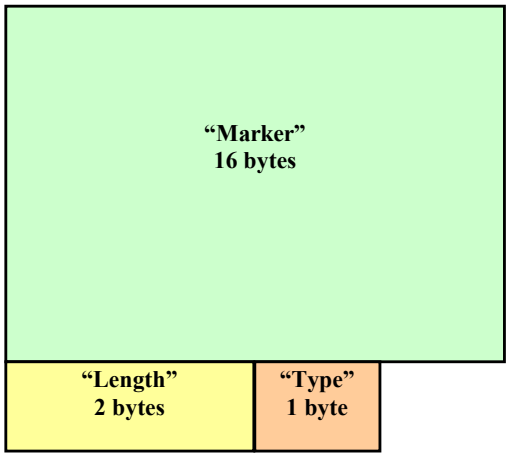
7.2 Mensagens BGP

Existem diversas mensagens padronizadas no BGP. Cada mensagem é composta de uma mensagem de cabeçalho mais a mensagem desejada. Mensagens de "KeepAlive" são enviadas periodicamente para assegurar que a conexão está ativa. Mensagens de notificação são enviadas em resposta a erros ou condições especiais. Se der erro numa conexão, uma mensagem de notificação é enviada e a conexão é fechada. As rotas são atualizadas através de

mensagens do tipo “update”. Os itens a seguir descrevem algumas dessas mensagens. Maiores detalhes na RFC 1761.

7.2.1 Mensagens de cabeçalho

As mensagens de cabeçalho são enviadas sempre no início de qualquer outra mensagem, ou seja, estas mensagens servem como cabeçalho para as outras. O formato é mostrado na figura a seguir [RFC 1761, pg 7].

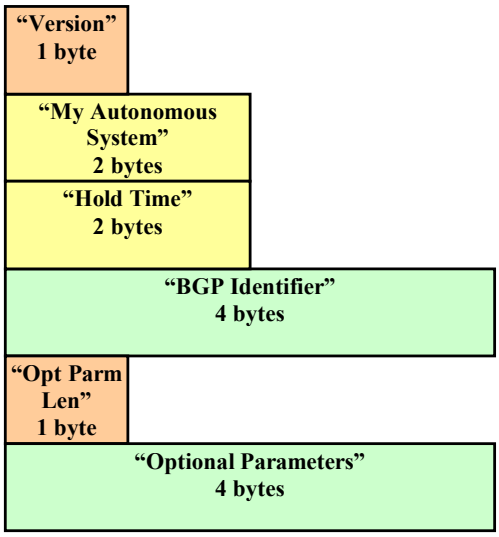


- **“Marker”**: pode ser usado para detectar perda de sincronização entre um par BGP ou para autenticar mensagens BGP entrantes, possuindo dois formatos: Caso o campo “*Type*” informe que a mensagem é de “*Open*” então todos os bits deste campo devem ser preenchidos com o valor “1”. Caso contrário, o valor deste campo pode ser previsto como parte de informação de autenticação.
- **“Length”**: indica o tamanho total da mensagem (em bytes), incluindo o cabeçalho. A maior mensagem existente é 4096 bytes, e a menor é de 19 bytes (*KeepAlive*).
- **“Type”**: indica o código da mensagem. Os códigos definidos são:
  - OPEN
  - UPDATE
  - NOTIFICATION
  - KEEPALIVE

7.2.2 Mensagens de “Open”

Após a conexão TCP estabelecida, a primeira mensagem enviada é de “Open”. Para confirmar, é enviado uma mensagem de “KeepAlive” como confirmação, e a partir deste momento mensagens de “Notification” podem ser trocadas.

Além do cabeçalho, as mensagens de “Open” possuem o seguinte formato [RFC 1761, pg 8]:



- **"Version"**: atualmente, é versão 4 (BGP-4).
- **"My Autonomous System"**: informa o AS do transmissor.
- **"Hold Time"**: indica o número máximo de segundos entre sucessivas mensagens "KeepAlive". Quando recebe uma mensagem de "Open", o receptor deve usar como "hold time" o menor valor entre o seu valor configurado e o proposto na mensagem. Este tempo deve ser zero ou, no mínimo, 3 segundos. Caso seja zero, não são enviadas mensagens de "KeepAlive", caso contrário, elas devem ser enviadas a aproximadamente 1/3 do valor do "hold time".
- **BGP Identifier**: número IP do transmissor da mensagem. Este número é determinado no início e é o mesmo para toda interface local e todo para BGP.
- Optional Parameters Length: indica o comprimento total (em bytes) do campo de parâmetros. Se for zero, não existem parâmetros opcionais.
- Optional Parameters: cada parâmetro é codificado através da tripla <*Parameter Type, Parameter Length, Parameter Value*>. Os parâmetros definidos na RFC 1761 são: Informação de autenticação; Código de autenticação; Dados de autenticação.

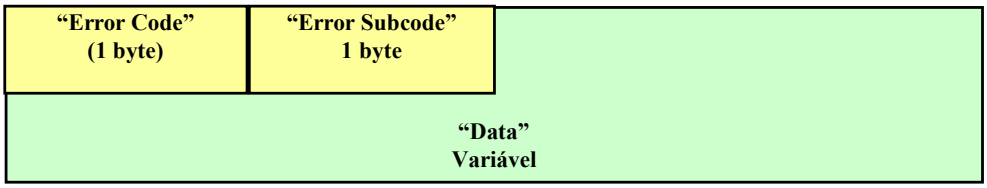
7.2.3 Mensagem de KeepAlive

As mensagens de KeepAlive são enviadas para manter a conexão viva entre os pares BGP. Elas têm que ser enviadas numa freqüência tal que não excedam o tempo "hold time", configurado durante a mensagem de Open. O valor sugerido é enviar uma mensagem KeepAlive num tempo de 1/3 do tempo de "hold time".

Caso o tempo de "hold time" tenha sido configurado como zero, a conexão é considerada sempre ativa, e não é necessário enviar mensagens de KeepAlive.

7.2.4 Mensagens de "Notification"

Uma mensagem de "Notification" é enviada quando uma condição de erro é detectada. Como conseqüência, a conexão BGP é encerrada logo após seu envio. Uma mensagem "Notification" consiste dos seguintes campos:



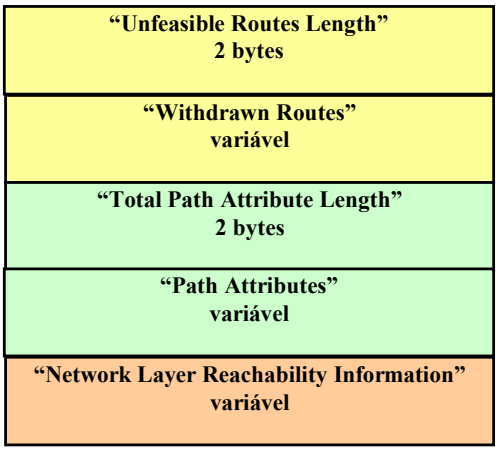
Os bytes de "Error Code" e "Error subcode" representam o código e subcódigo da mensagem de erro. Os seguintes erros foram definidos [MOU 99], [RFC 1761]:

Código de erro	Sub códigos de erro
1. Erro no cabeçalho da mensagem	1. Conexão não sincronizada 2. Comprimento da mensagem inválido 3. Tipo de mensagem inválido
2. Erro na mensagem "Open"	1. Número de versão não suportado 2. Número de AS vizinho inválido 3. Identificador BGP inválido 4. Parâmetro opcional não suportado 5. Falha na autenticação 6. Tempo de espera inaceitável
3. Erro na mensagem "Update"	1. Lista de atributos mal formada 2. Atributo *Well-known* desconhecido 3. Atributo *Well-known* faltando 4. Erro nas flags de atributos 5. Erro no comprimento do atributo 6. Atributo de origem inválido 7. Loop de roteamento em AS 8. Atributo *next-hop* inválido 9. Erro no atributo opcional 10. Campo de rede inválido 11. *AS-path* mal formado
4. Hold Time expirou	
5. Erro na máquina de estados finitos	Qualquer erro detectado na máquina de estados do BGP gera esta notificação de erro.
6. Cease – encerra conexão	Na ausência de erros a conexão pode ser desfeita por qualquer parte, e este tipo não deve ser considerado erro, mas encerramento de conexão.

7.2.5 Mensagens de "Update"

As mensagens de "Update" são usadas para transferir informações de rotas entre pares BGP, e essas informações podem ser usadas para construir um grafo com os relacionamentos dos vários AS envolvidos. Aplicando certas regras, loops e outras anomalias podem ser detectadas e removidas do roteamento.

Uma mensagem de "Update" é usada para anunciar uma rota com o par BGP ou para eliminar rotas inviáveis. Este tipo de mensagem sempre contém um cabeçalho e outros campos, conforme mostra a figura a seguir [RFC 1761, pg. 11], [MOU 99].



- **Unfeasible Routes Length**: indica o tamanho total do campo "*Withdrawn Routes*". Um valor de zero indica que nenhuma rota está sendo removida e o campo seguinte não está presente na mensagem.
- **Withdrawn Routes (Rotas Removidas)**: contém uma lista de números IP para as rotas que estão sendo eliminadas do serviço. Cada endereço é codificado na tupla <length, prefix>. Maiores detalhes na RFC 1761.
- **Total Path Attribute Length**: indica o tamanho total do campo "*Path Attributes*" e permite saber o comprimento do campo "*Network Layer Reachability*". Um valor de zero indica que não existe campo de "*Network Layer Reachability*" nesta mensagem.
- **Path Attributes**: este campo variável **está presente em toda mensagem de Update** e representa um conjunto de atributos associados a uma determinada rota que influenciam no **processo de decisão para o BGP escolher a melhor rota**. Tais atributos são compostos pela tripla <*attribute type, attribute length, attribute value*>.
- **Network Layer Reachability Information (NLRI)**: contém uma lista de prefixos IP no formato <length, prefix> com informações de alcance da rede. Por exemplo, uma entrada do tipo <20, 200.248.168.0> indica uma rota para a subrede 200.248.168.0 com máscara 255.255.240.0, ou 200.248.168.0/20 (na notação CIDR).

7.3 ATRIBUTOS BGP

Como visto acima, as mensagens de "Update" incluem o campo "Path Attributes", que possui informações dos atributos BGP para as rotas anunciadas no campo NLRI. As categorias de atributos são:

- ***Well Known Mandatory (WKM)***: atributos conhecidos (por todas versões de BGP) e obrigatórios (devem estar presentes em todas mensagens BGP)
- ***Well Known Discretionary (WKD)***: atributos conhecidos (por todas versões de BGP) mas não precisam estar presentes em todas mensagens
- ***Optional Transitive (OT)***: atributos opcionais (não precisam ser conhecidos por todas versões BGP) que são retransmitidos para os vizinhos caso a flag "*transitive*" esteja ativada.
- ***Optional NonTransitive (ONT)***: atributos opcionais (não precisam ser conhecidos por todas versões BGP) que são ignorados caso a implementação BGP não o reconheça.

A tabela a seguir resume rapidamente as categorias de atributos BGP. Maiores detalhes em [MOU 99] e na RFC 1761. O campo "código de atributo" possui valores padronizados

pelo IANA, e podem ser encontrados em http://www.isi.edu/in-notes/iana/assignments/bgp-parameters.

Tipo	Atributo	Referência:	Categoria
1	ORIGIN	RFC 1771	WKM
2	AS_PATH	RFC 1771	WKM
3	NEXT_HOP	RFC 1771	WKM
4	MULTI_EXIT_DISC	RFC 1771	ONT
5	LOCAL_PREF	RFC 1771	WKD
6	ATOMIC_AGGREGATE	RFC 1771	WKD
7	AGGREGATOR	RFC 1771	OT
…			
255	Reservado para desenvolvimento	-	-

7.3.1 ORIGIN

O atributo ORIGIN indica a origem do anúncio da rota. Pode ser originado de um IGP (de dentro do mesmo AS – identificado por "i" na tabela de rotas), um EGP (origem é um AS externo – identificado por "e") ou incompleto (origem desconhecida ou configurada por rota estática – identificado por "?"). O exemplo a seguir mostra o resultado de uma consulta (observe que os identificadores i, e e ? aparecem na direita da tabela de rotas do BGP [MOU 99].

```
roteador>show ip bgp
BGP table version is 117080, local router ID is 200.180.1.142
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
Network Next Hop Metric LocPrf Weight Path
* i0.0.0.0 204.189.152.169 100 0 3561 i
* i 204.189.152.153 100 0 3561 i
*> 204.189.152.141 0 3561 i
*> 12.128.70.48/28 200.255.125.4 0 4230 8031 i
* i 200.230.6.4 100 0 4230 8031 i
*> 32.96.204.0/24 200.255.125.4 0 4230 2685 ?
* i 200.230.6.4 100 0 4230 2685
```

7.3.2 AS_PATH

O AS_PATH representa a seqüência de ASN (AS Numbers) que uma rota cruza para alcançar determinada rede destino. À medida que a mensagem passa pelos ASs, cada um acrescenta seu ASN no início da seqüência de ASNs, repassando a informação para seus pares (que irão repetir o processo). No final, a lista conterá todos os ASs que a mensagem passou naquele caminho (*AS sequence*). Caso um AS receba uma lista com seu próprio AS nela, é sinal que **pode haver loop de roteamento**, e, para evitar isso, este anúncio será rejeitado e descartado.

Caso o AS_PATH seja anunciado para um vizinho do mesmo AS, a informação contida no AS_PATH não é alterada.

A informação contida no AS_PATH é uma das usadas no **processo de escolha da melhor rota para um destino**. Ao comparar duas rotas para o mesmo destino, o BGP vai preferir a de menor AS_PATH.

7.3.3 NEXT_HOP

O atributo NEXT_HOP possui o endereço IP da interface do próximo roteador para chegar a determinado destino. O exemplo a seguir mostra informações de NEXT_HOP aprendidas através de mensagens "Update" [MOU 99].

```
roteador> show ip bgp
BGP table version is 117080, local router ID is 204.189.152.142
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
Network Next Hop Metric LocPrf Weight Path
*> 139.82.0.0 200.159.255.253 20 0 2715 i
*>i143.54.0.0 200.132.0.16 1 100 0 i
```

7.3.4 MULTI_EXIT_DISC

O atributo MULTI_EXIT_DISCRIMINATOR (MED) informa aos vizinhos BGP externos qual o melhor caminho para uma determinada rota dentro do próprio AS, influenciando-os no caso do AS possuir diversos pontos de entrada.

Somente o AS origem pode fazer anúncios MED a outros ASs vizinhos, entretanto, os outros AS não podem repassar este valor a outros ASs, e somente usa este atributo para tomada de decisões internas.

7.3.5 LOCAL_PREF

O atributo LOCAL_PREF serve para anunciar o caminho preferencial de saída (de pacotes) para uma determinada rota, destinada a uma rede externa ao AS.

7.3.6 AGGREGATOR

O atributo AGGREGATOR pode ser incluído em mensagens "Update" que sejam formadas por agregação. Este atributo contém o ASN do último roteador que formou uma rota agregada, seguido do seu ASN e endereço IP.

8 Constraint Based Routing

/**/ ver apostila de QoS – está explicado superficialmente lá.

9 Bibliografia

[HUI 99] HUITEMA, Christian. ***Routing in the Internet***. Second edition. Ed. Prentice Hall, New Jersey, 1999. 384 p.

[MOU 99] MOURA, Alex Soares. **O protocolo BGP – partes 1, 2 e 3**. RNP: News Generation. 1999.

[RFC 1058] HEDRICK, C. ***Routing Information Protocol***. IETF: California. Jun 1988.

[RFC 1723] MALKIN, G. ***RIP Version 2. Carrying Additional Information***. IETF: California. Nov 1994.

[RFC 1850] BAKER, F. COLTUN, R. ***OSPF Version 2 Management Information Base***. IETF: California. Nov 1995.

[RFC 1930] HAWKINSON, J. BATES, T. ***Guidelines for creation, selection and resgistration of an Autonomous System*** **(AS)** (Best Current Practice). IETF: California. Mar 1996.

[RFC 1761] REKHTER Y. LI, T. ***A Border Gateway Protocol 4 (BGP-4)*** (Standards Track). IETF: California. Mar 1995.

[RFC 2178] MOY, J. ***OSPF Version 2***. IETF: California. Jul 1997.

[TAN 96] TANENBAUM, Andrew C. **Redes de Computadores**. 3a edição. Ed. Campus, Rio de Janeiro, 1996. 923p.