



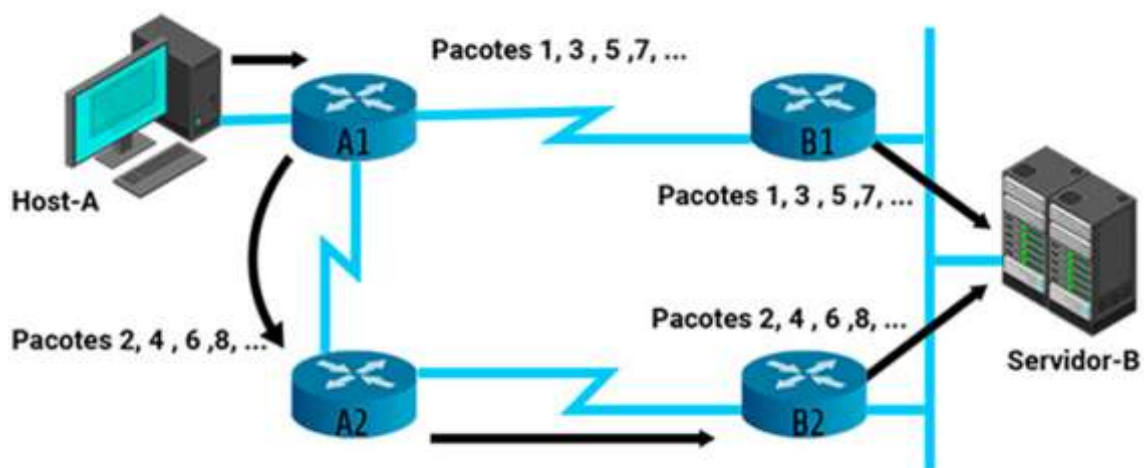
ALGORITMOS DE ROTEAMENTO

Encaminhando os pacotes na rede

Roteamento

A principal função da camada de rede é **rotear pacotes do hospedeiro origem para o hospedeiro destino da melhor forma possível**. Na maioria dos casos, os pacotes necessitarão passar por vários roteadores para chegar ao destino.

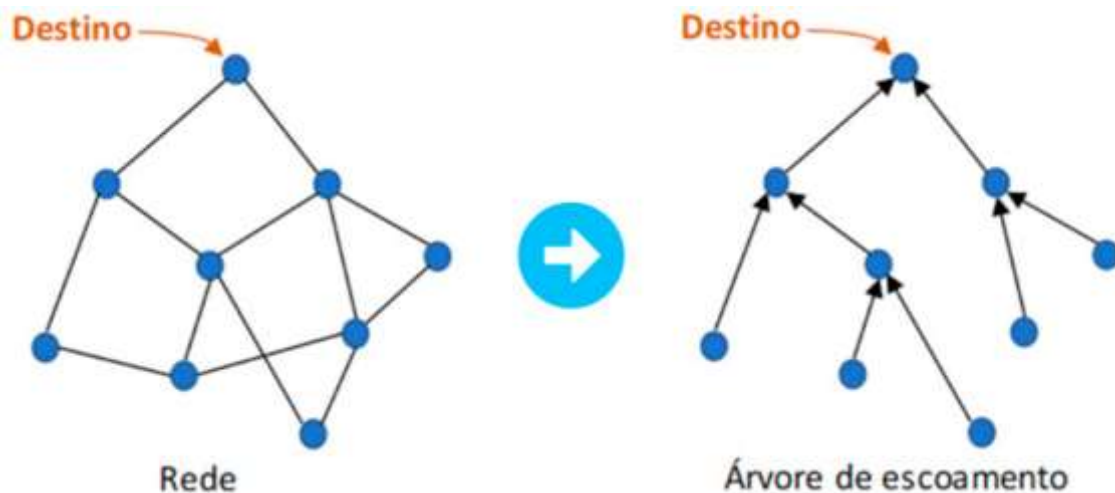
O algoritmo de roteamento é a parte do software da camada de rede responsável pela decisão sobre a linha de saída a ser usada na transmissão do pacote que entra. Caso a sub-rede utilize datagramas, essa decisão deverá ser tomada para todos os pacotes de dados recebidos e se utilizar circuitos virtuais internamente, as decisões de roteamento serão tomadas somente quando o circuito virtual estiver sendo estabelecido.



Princípio da otimização

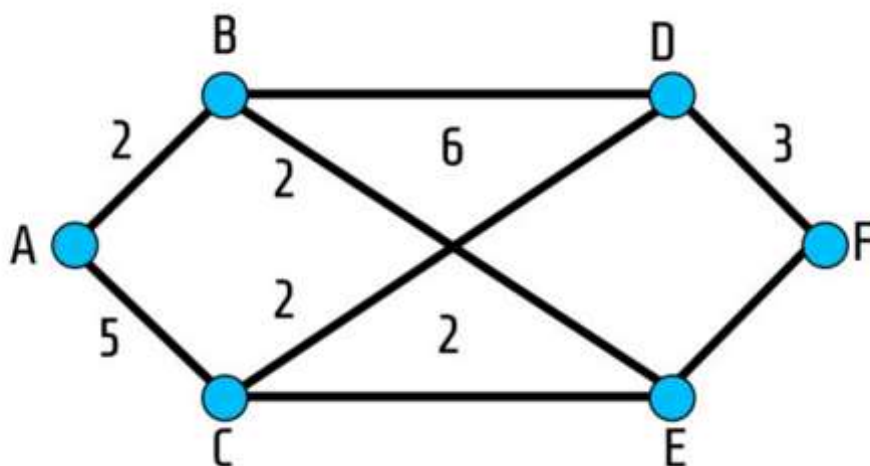
O princípio da otimização de Bellman (1957) estabelece que se o roteador B estiver no caminho ótimo entre o roteador A e o roteador C, o caminho ótimo de B a C também estará na mesma rota.

Esse princípio nos diz que o conjunto de rotas ótimas de todas as origens para um determinado destino é uma árvore conhecida como **árvore de escoamento** (*sink tree*, em inglês), cuja raiz é o hospedeiro destino. Os algoritmos de roteamento precisam descobrir a árvore de escoamento a partir da topologia da rede.



Roteamento pelo caminho mais curto

O **protocolo de roteamento pelo caminho mais curto** é um **protocolo estático** cuja ideia é criar um grafo da sub-rede, com cada nó do grafo representando um roteador e cada arco indicando um enlace. Para escolher uma rota entre um determinado par de roteadores, o algoritmo simplesmente encontra o caminho mais curto entre eles no grafo.



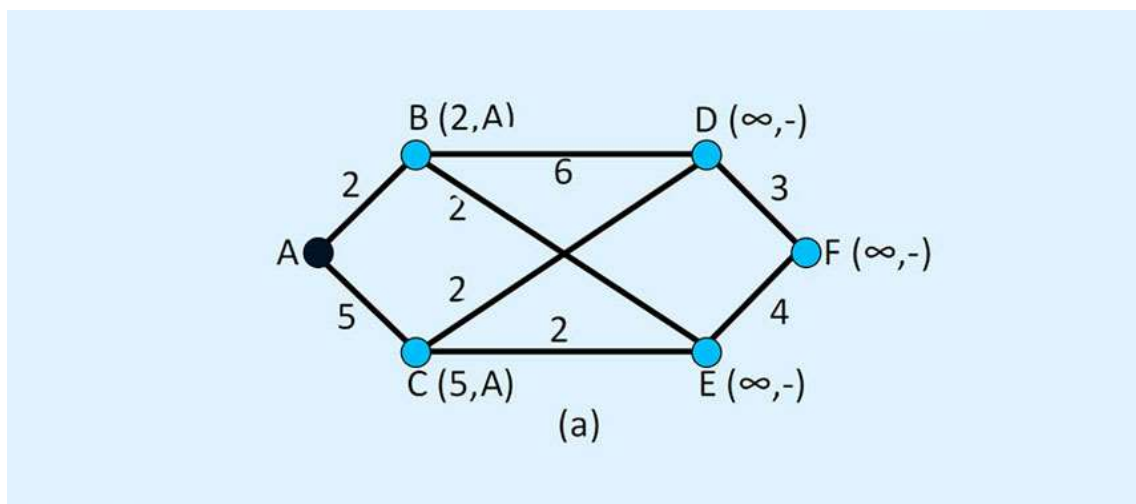
Uma forma de medir o comprimento do caminho é em **número de saltos (quantidade de enlaces que devem ser utilizados)**. Utilizando essa unidade métrica na figura anterior, os caminhos ABD e ABE são igualmente longos. Outra unidade métrica é a **distância geográfica**, caso em que ABD é mais longo que ABE.

Há também a unidade métrica baseada no tráfego entre os enlaces. Nesse grafo, o caminho mais curto é o caminho mais rápido, e não o caminho com menos arcos ou com menor distância.

Os valores dos arcos podem ser calculados como uma função da distância, da largura de banda, do tráfego médio, do custo de comunicação, do comprimento médio de fila, do retardo detectado ou de outros fatores. Alterando a função de atribuição de pesos, o algoritmo calcularia o caminho mais curto medido de acordo com determinados critérios que podem ser ou não combinados.

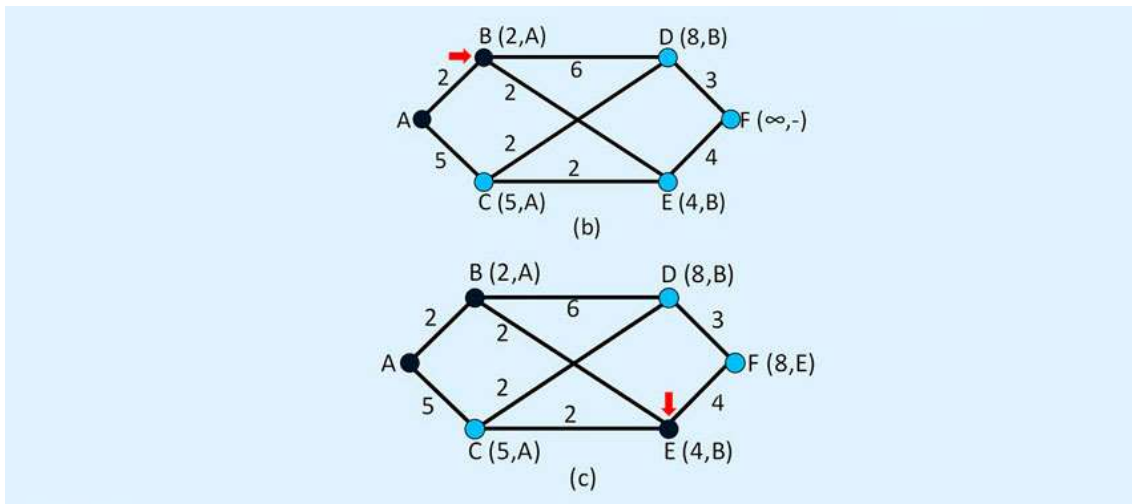
Existem diversos algoritmos para o cálculo do caminho mais curto. O mais conhecido deles foi desenvolvido por Dijkstra, em 1959. Nele, cada roteador armazena sua menor distância até a origem e o caminho a ser seguido. Na inicialização do algoritmo não existe caminho conhecido, assim, a distância é marcada como infinito. Conforme o algoritmo progride, os caminhos e seus custos são encontrados.

Roteamento pelo caminho mais curto (Dijkstra)

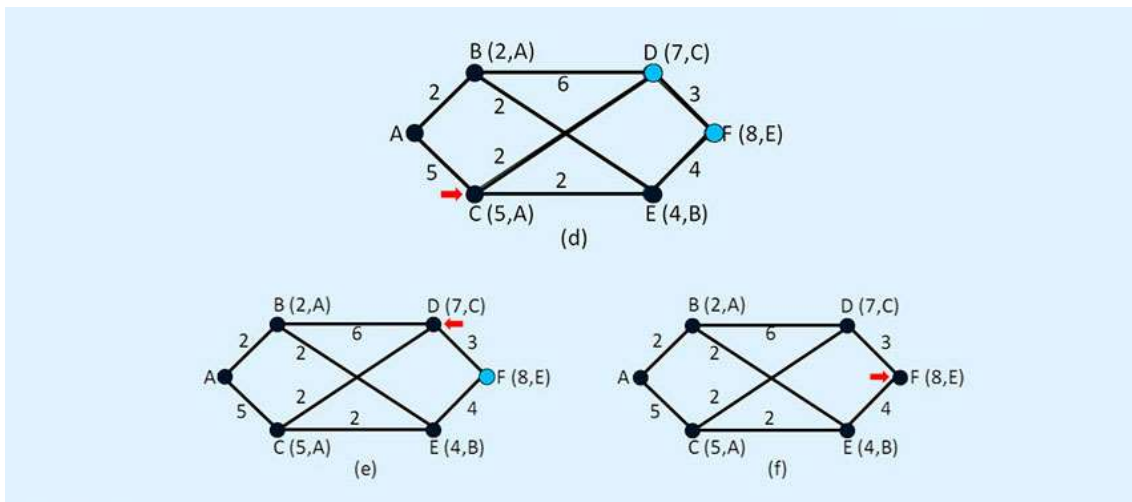


Para encontrar a árvore de escoamento de A, na figura, marca-se o nó A como permanente, o que é indicado por um círculo preenchido (a). Depois, é examinado, um a um, cada nó adjacente a A alterando o rótulo de cada um deles para a distância até A.

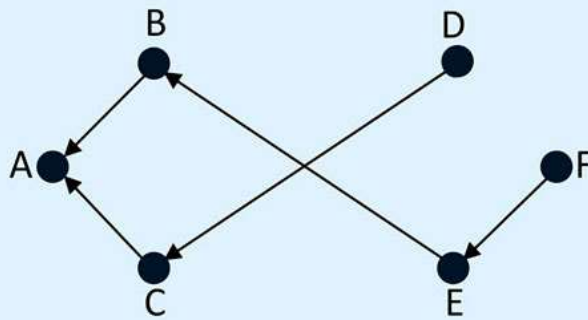
Sempre que um nó é rotulado novamente, ele também é rotulado com o nó a partir do qual o teste foi feito, assim, pode-se reconstruir o caminho final posteriormente. Após examinar cada nó adjacente a A, verifica-se todos os nós provisoriamente rotulados no grafo tornando permanente o de menor rótulo, que passa a ser o novo nó ativo.



No segundo passo (b), verifica-se que o nó B é o que possui o menor valor entre os nós rotulados não permanentes. Logo, ele é marcado como permanente e os nós adjacentes são rotulados. Como B tem custo 2 e o enlace entre B e D possui custo 6, o custo de D passando por B será $6 + 2 = 8$. Da mesma forma, o custo de E passando por B será rotulado como 4. E assim o algoritmo prossegue, até que todos os nós sejam marcados como permanentes.



Uma situação interessante acontece no quarto passo (d). Quando o nó C é marcado como permanente, verifica-se que D anteriormente foi rotulado com valor 8 passando por B, porém, como vizinho de C, seu custo será 7. Então, o nó D é rotulado novamente com o valor 7.



Ao final, para obter a árvore de escoamento até A, basta seguir o caminho marcado por cada roteador até seu vizinho de menor custo.

Algoritmos de roteamento

Conforme estudamos anteriormente, **os algoritmos de roteamento são executados pelos roteadores de uma sub-rede** para que sejam criadas as tabelas de repasse dos roteadores. Para isso, eles precisam trocar informações sobre o estado da rede e concordarem com a árvore de escoamento a ser montada para cada destino.

Os algoritmos de roteamento podem ser agrupados em:

Não adaptativos

Não baseiam suas decisões de roteamento em medidas ou estimativas do tráfego e da topologia atuais. A escolha da rota a ser utilizada é previamente calculada e transferida para os roteadores quando a rede é inicializada. Tal procedimento também é conhecido como roteamento estático.

Adaptativos

Mudam suas decisões de roteamento para refletir mudanças na topologia e/ou no tráfego. Tal procedimento também é conhecido como roteamento dinâmico.

Ainda, os algoritmos de roteamento podem ser classificados como algoritmos de roteamento globais ou descentralizados.

Global

Calcula o melhor caminho com base no conhecimento completo da rede. Para esse fim, o roteador deve obter informações sobre o estado de todos os roteadores e enlaces que compõem a sub-rede. Um exemplo desse tipo de roteamento é o roteamento de estado de enlace.

Descentralizado

Neste algoritmo, nenhum roteador possui informação completa sobre o estado da rede. As rotas são calculadas com base em informações obtidas com roteadores vizinhos. Um exemplo desse tipo de roteamento é o roteamento de vetor de distância.

Roteamento de vetor de distâncias

O roteamento de vetor de distância é um algoritmo dinâmico que opera fazendo com que cada roteador mantenha uma tabela que fornece a melhor distância conhecida a cada destino e determina qual linha deve ser utilizada para se chegar lá. Essas tabelas são atualizadas por meio da troca de informações com os vizinhos.

Trata-se do algoritmo de roteamento original da ARPANET que também foi utilizado na internet com o nome rip. alguns roteadores utilizam protocolos de vetor de distância mais aperfeiçoados.

Comentário

Cada roteador mantém uma tabela de roteamento indexada por cada roteador da sub-rede. Cada entrada contém duas partes: a linha de saída preferencial a ser utilizada para esse destino e uma estimativa do tempo ou distância até o destino. A unidade métrica utilizada pode ser o número de hops, o retardo de tempo, o número total de pacotes enfileirados no caminho ou algo semelhante.

Presume-se que o roteador conheça a distância até cada um de seus vizinhos. Se a unidade métrica for contagem de saltos, a distância será de apenas um salto. Se a unidade métrica for o comprimento da fila, o roteador examinará cada uma das filas. Se a unidade métrica for o retardo, o roteador poderá medi-lo com pacotes "ICMP ECHO".

E como saber qual o caminho que possui melhor distância apenas obtendo informações dos vizinhos?

Seja $dv(y)$ o custo do caminho de menor custo do roteador v para o roteador y , e seja $c(x,v)$ o custo da ligação entre os roteadores x e v . Seja ainda v um vizinho do roteador x .

Então, o custo do caminho para ir de x a y passando pelo vizinho v será $\{c(x,v) + dv(y)\}$.

Logo, pelo princípio da otimização, o menor custo para ir de x a y será:

$$dx(y) = \min_v \{c(x,v) + dv(y)\},$$

onde \min_v retornará o menor valor calculado para todos os vizinhos.

Conforme os vizinhos vão trocando informações sobre o estado da rede, o algoritmo converge para um **estado de equilíbrio** onde são obtidas as árvores de escoamento para cada destino.

Sempre que houver alguma modificação na rede, o roteador comunicará essa mudança a seus vizinhos, que atualizarão suas tabelas e repassarão essas alterações a seus próprios vizinhos. Assim, a alteração se propagará entre todos os roteadores e rapidamente a rede convergirá para um novo estado de equilíbrio.

Na Internet, o roteamento de vetor de distância é implementado pelo **RIP (Routing Information Protocol)**. A versão 1 do RIP é definida pela RFC 1058, enquanto sua versão 2 é definida pela RFC 2453.

Roteamento de estado de enlace

O roteamento de vetor de distância era utilizado na ARPANET até 1979, quando então foi substituído pelo **roteamento de estado de enlace**. Essa substituição foi basicamente motivada por dois problemas:

- A unidade métrica de retardo era o comprimento de fila, não levando em conta a largura de banda.
- O algoritmo geralmente levava muito tempo para convergir.

A ideia do roteamento de estado de enlace é simples. Cada roteador deve:

Descobrir seus vizinhos e aprender seus endereços de rede

Determinação dos vizinhos: quando um roteador é inicializado, ele envia um pacote HELLO em cada linha. O roteador da outra extremidade envia de volta uma resposta identificando-se.

Medir o retardo ou o custo para cada um de seus vizinhos

Medição do custo da linha: a forma mais simples de determinar o retardo é enviar um pacote ECHO pela linha. Medindo o tempo de ida e volta, o roteador pode obter uma estimativa razoável do retardo. Para obter resultados melhores, pode-se fazer o teste mais de uma vez e usar a média.

Criar um pacote que diga tudo o que acaba de ser aprendido

Criação de pacotes por estado de enlace: o pacote começa com a identidade do transmissor, seguida do número de sequência, da idade, e de uma lista de vizinhos. É fornecido o custo referente a cada vizinho. Um exemplo de sub-rede é ilustrado na figura a seguir, sendo os custos mostrados por linhas. Os pacotes de estado de enlace correspondentes a todos os seis roteadores são mostrados na figura seguinte.

Enviar esse pacote a todos os outros roteadores

Distribuição dos pacotes de estado de enlace: se estamos querendo criar as rotas para que os datagramas possam ser enviados pela rede, como distribuir as informações? A ideia é usar a inundação de pacotes, na qual os pacotes que são recebidos por uma interface são replicados para as outras interfaces, fazendo, assim, com que a informação se espalhe por toda a rede.

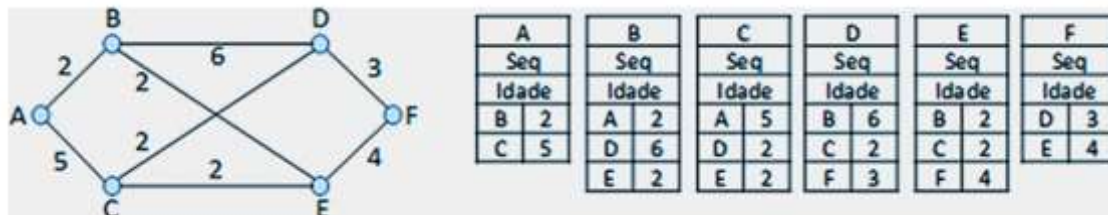
Para controlar a inundação, cada pacote contém um número de sequência que é incrementado para cada pacote enviado. Quando é recebido, o novo pacote de estado de enlace é conferido na lista de pacotes já verificados. Se for novo, o pacote será encaminhado a todas as linhas, menos para aquela em que chegou. Se for uma cópia, o pacote será descartado. Se for recebido um pacote com número de sequência inferior ao mais alto detectado até o momento, ele será rejeitado.

Mas se um roteador apresentar falha, ele perderá o controle de seu número de sequência. A solução é incluir a idade de cada pacote após o número de sequência e decrementá-la uma vez por segundo. Quando a idade atingir zero, as informações desse roteador serão descartadas. O campo de idade é também decrementado por cada roteador durante o processo inicial de inundação para garantir que nenhum pacote viverá por um período indefinido.

Calcular o melhor caminho para cada um dos outros roteadores

Cálculo das novas rotas: uma vez que um roteador tenha acumulado um conjunto completo de pacotes de estado de enlace, poderá criar o grafo completo da sub-rede. Nesse momento, o algoritmo de Dijkstra pode ser executado localmente para calcular o melhor caminho.

Na Internet, o roteamento de estado de enlace é implementado pelo protocolo OSPF (*Open Shortest Path First*), definido pela RFC 2328.



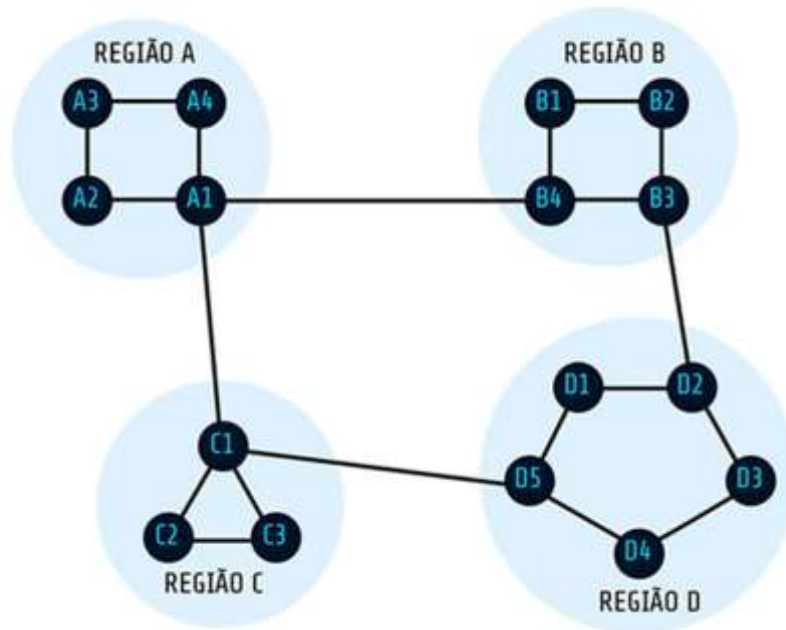
Assim, a topologia completa e todos os retardos são medidos e distribuídos para cada roteador. Em seguida, o algoritmo de Dijkstra pode ser usado para encontrar o caminho mais curto.

Roteamento hierárquico

Conforme as redes aumentam, as tabelas de roteamento crescem proporcionalmente até que chega um momento em que **não é mais possível que os roteadores conheçam todos os caminhos da rede**. Assim, é necessário pensar em uma nova forma de realizar o roteamento. Foi pensando nessa situação que foi criado o **roteamento hierárquico**.

Quando o roteamento hierárquico é utilizado, **os roteadores são divididos em regiões**, com cada roteador conhecendo todos os detalhes sobre como rotear pacotes para destinos dentro de sua própria região, mas sem conhecer a estrutura interna de outras. No caso de redes muito grandes, uma hierarquia de dois níveis pode ser insuficiente, sendo necessário agrupar as regiões em clusters, os clusters em zonas, as zonas em grupos etc.

A imagem abaixo mostra um **exemplo quantitativo do roteamento em uma hierarquia de dois níveis com quatro regiões**. A tabela de roteamento completa do roteador A1 tem 16 entradas, mas quando o roteamento é feito hierarquicamente são necessárias apenas 7 entradas. Porém, esse tipo de roteamento pode implicar um provável aumento do caminho para alguns hospedeiros.



Entenda cada região nas tabelas abaixo:

Tabela completa		
Destino	Linha	Saltos
A1	--	--
A2	A2	1
A3	A2	2
A4	A4	1
B1	B4	2
B2	B4	3
B3	B4	2

Tabela completa		
B4	B4	1
C1	C1	1
C2	C1	2
C3	C1	2
D1	C1	3
D2	B4	3
D3	B4	4
D4	C1	3
D5	C1	2

Tabela completa.

Fábio Contarini Carneiro.

Tabela hierárquica		
Destino	Linha	Saltos
A1	--	--
A2	A2	1
A3	A2	2

Tabela hierárquica		
A4	A4	1
B	B4	1
C	C1	1
D	C1	2

Tabela hierárquica

Fábio Contarini Carneiro.

Na Internet, o roteamento hierárquico é implementado por intermédio do protocolo BGP (*Border Gateway Protocol*), definido pela RFC 4271.