

Universidade Estácio de Sá
Prof. Welsing Moreira Pereira
<http://www.professorwelsing.webnode.com>



O que sabemos é uma gota, o que ignoramos é um oceano.
Isaac Newton (1642-1727) – Físico e Matemático Inglês.

Estácio

- Princípios de roteamento
 - É tarefa da camada de rede **determinar o caminho ou rota** que os pacotes devem seguir, uma vez que uma origem deseja transferir pacotes para um destino.
 - Essa tarefa é independente do serviço oferecido:
 - Datagrama ou Circuito Virtual
 - Para isso, protocolos de roteamento fazem uso de algoritmos “**algoritmos de roteamento**” que determinam o trajeto que os pacotes devem percorrer.

- Princípios de roteamento
 - A finalidade do algoritmo de roteamento é simples:
 - Dada uma série de roteadores conectada com enlaces, o algoritmo de roteamento descobre um “bom” caminho entre a fonte e o destino.
 - Um bom caminho é aquele que oferece o “menor custo”.
 - Custo:
 - Valor unitário (salto - hop)
 - Distância
 - Capacidade na linha de transmissão
 - Custo da linha de comunicação

- Princípios de roteamento
 - Dada a abstração gráfica, o problema de descobrir o caminho de menor custo entre uma fonte e um destino requer a identificação de uma série de enlaces tal que:
 - O primeiro enlace no caminho esteja conectado à fonte;
 - O último enlace no caminho esteja conectado ao destino;
 - Para todo i , o enlace i e o $i - 1$ no caminho estejam conectados ao mesmo nó, e
 - Para o caminho de menor custo, a soma dos custos dos enlaces no caminho seja a mínima dentre todos os possíveis caminhos entre a fonte e o destino.

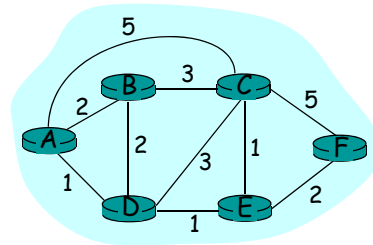
Princípios de roteamento

Algoritmos de roteamento são descritos por grafos:

- Nós do gráfico são roteadores
- Arestas do gráfico são enlaces
 - Custo do enlace: atraso, preço ou nível de congestão, salto.

“bons” caminhos:

- tipicamente corresponde aos caminhos de menor custo



Qual o caminho de menor custo entre os nós A a F?

- O que te levou a escolher esse caminho?
- Verificou todos os caminhos possíveis? São 17 no total.
- Para escolher esse caminho, vc se utilizou de um algoritmo de roteamento “centralizado”, em seu cérebro.

Algoritmo de roteamento global (Centralizado)

- Calcula o caminho de menor custo a partir do conhecimento completo da topologia da rede.
- Ou seja, para cálculo o algoritmo conhece a conectividade entre todos os nós e todos os custos dos enlaces.
- O cálculo em si, pode ser feito em um local, ou duplicado em múltiplos locais.
- Esses algoritmos são chamados de algoritmos **link-state**, já que estão a par dos custos de cada enlace na rede.

Algoritmo de roteamento descentralizado

- Nenhum nó tem a informação completa sobre os custos de todos os enlaces da rede.
- Cada nó começa tendo apenas conhecimento dos enlaces diretamente ligados a ele (vizinhos).
- Por meio de troca de informação com seus vizinhos, um nó gradualmente calcula o caminho de menor custo até um destino ou um conjunto de destinos.
- Um algoritmo descentralizado bem conhecido é o “**algoritmo distance-vector**”.

Estático ou Dinâmico

- Estático (não-adaptativos)
 - Nesses algoritmos as rotas mudam muito devagar com o tempo, muitas vezes como resultado da intervenção humana.
 - Problemas:
 - Probabilidade de má utilização dos meio de comunicação(desperdício do meio).
 - Dificuldade para manutenção e gerência.
- Dinâmico (adaptativos)
 - Mudam os caminhos de roteamento à medida que mudam as cargas de tráfego ou a topologia da rede.
 - Suscetíveis a problemas como loops de roteamento e oscilação das rotas.

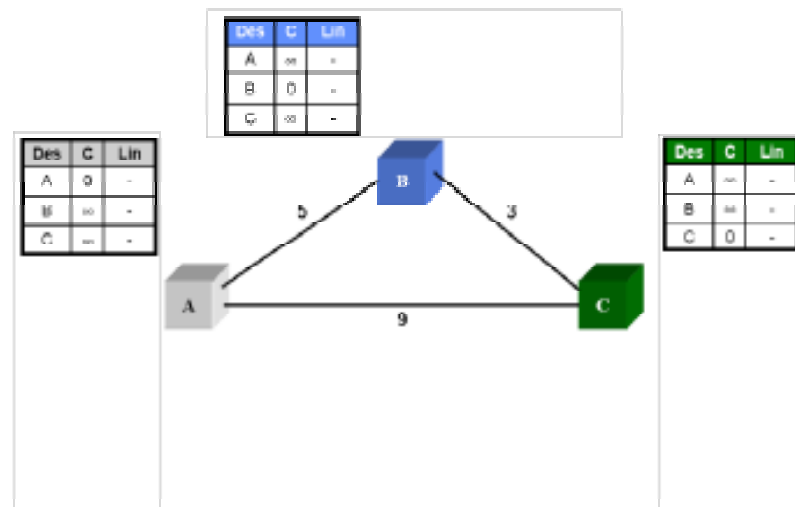
- Internet
 - Apenas dois algoritmos de roteamento são tipicamente usados na Internet:
- Um algoritmo dinâmico descentralizado
 - distance vector;
- Um algoritmo dinâmico global
 - link-state.

9

- Distance-vector
 - Cada nó possui:
 - Identificador único
 - Custo de cada enlace
 - O nó transmite o seu vetor de distâncias (com destino/custo) para cada um dos seus vizinhos sempre que o seu vetor se modifica (ou periodicamente)
 - Cada nó calcula o seu próprio vetor de distâncias, minimizando o custo para cada destino.
 - O vetor de distâncias é recalculado sempre que:
 - Um vizinho enviar um vetor de distâncias contendo informações diferentes das anteriores
 - Houver a queda em um enlace para um vizinho. Nesse caso, o vetor de distâncias desse vizinho é descartado para que o vetor de distâncias do nó seja recalculado.

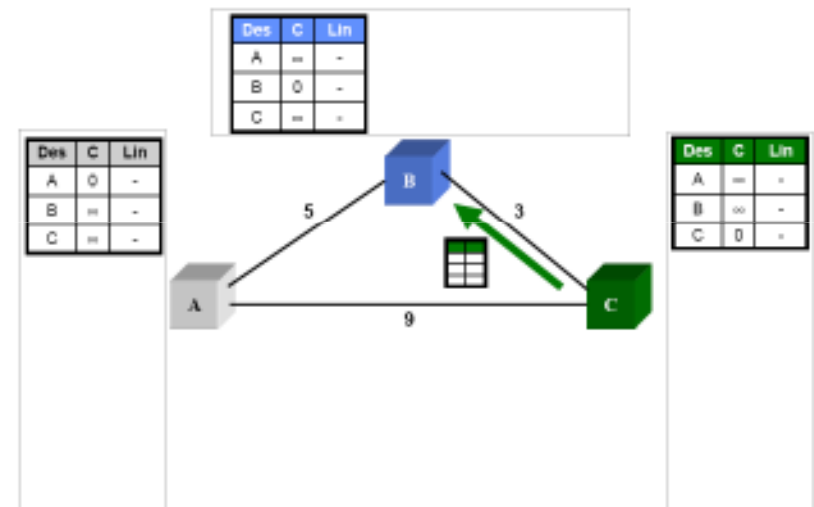
10

Distance Vector



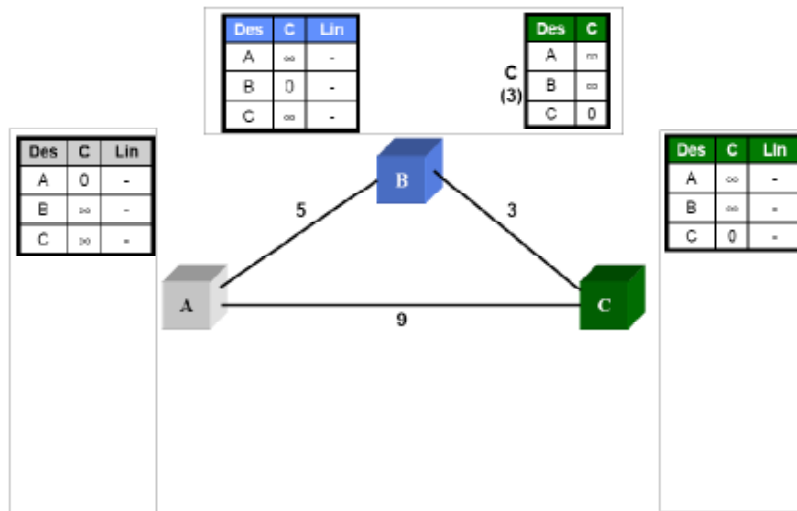
11

Distance Vector



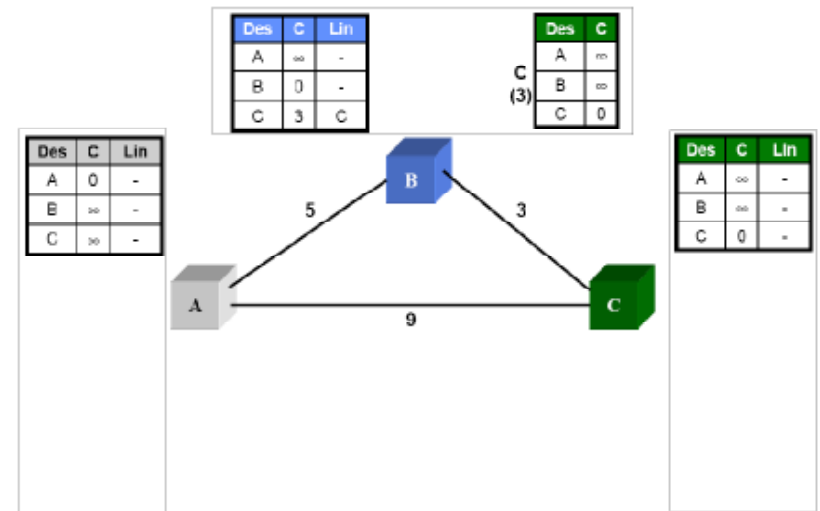
12

Distance Vector



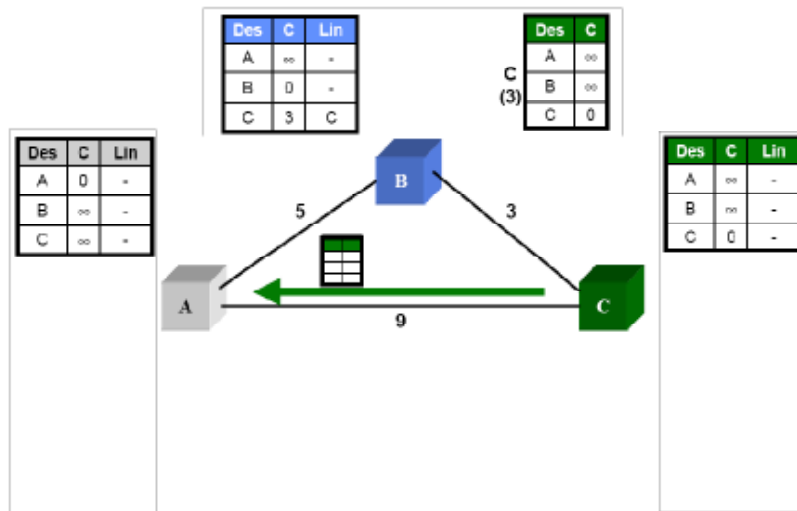
13

Distance Vector



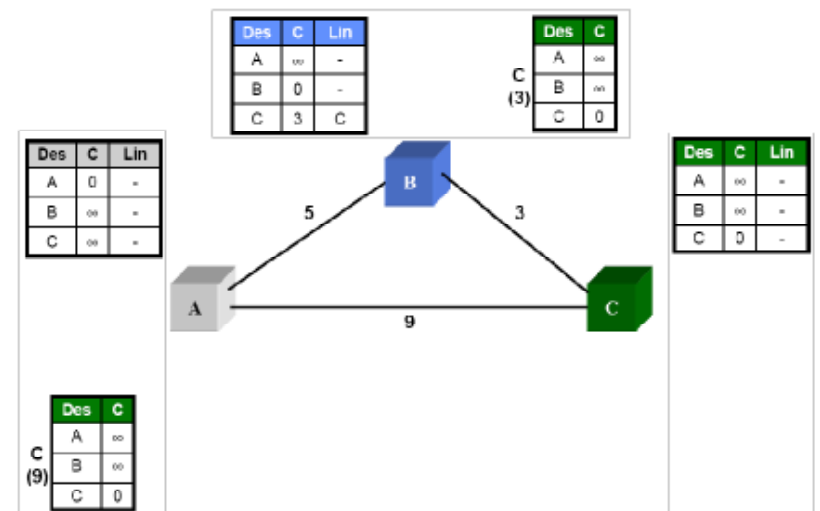
14

Distance Vector



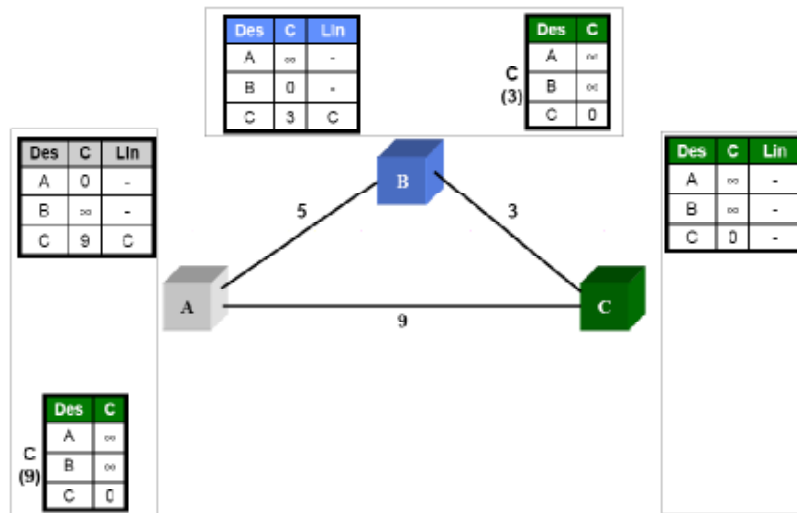
15

Distance Vector



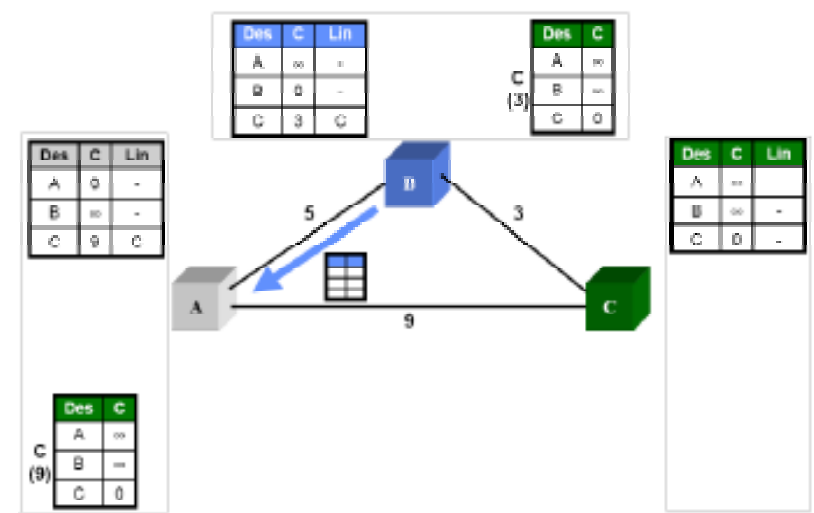
16

Distance Vector



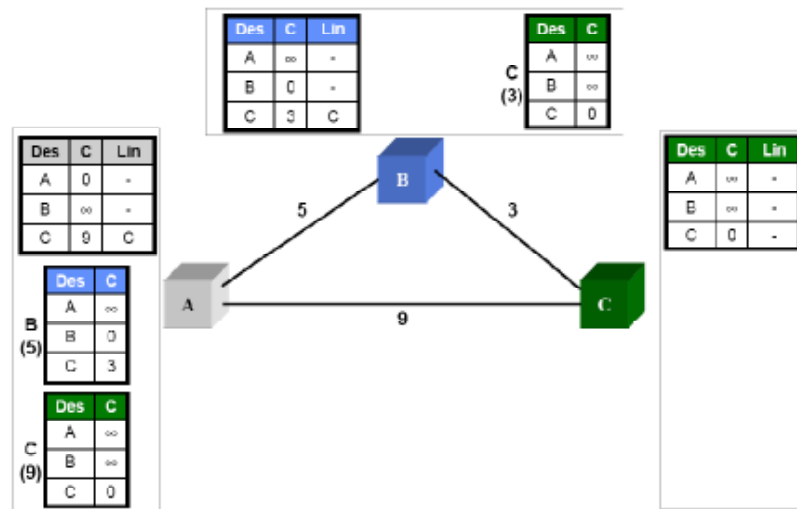
17

Distance Vector



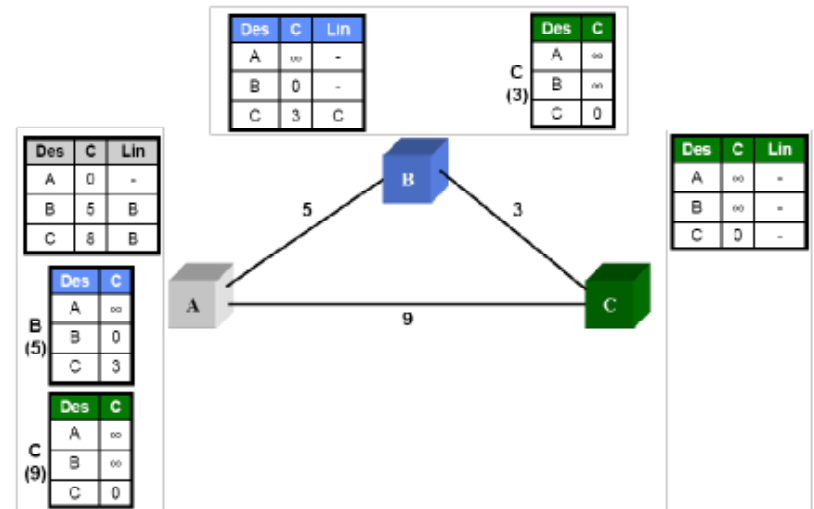
18

Distance Vector



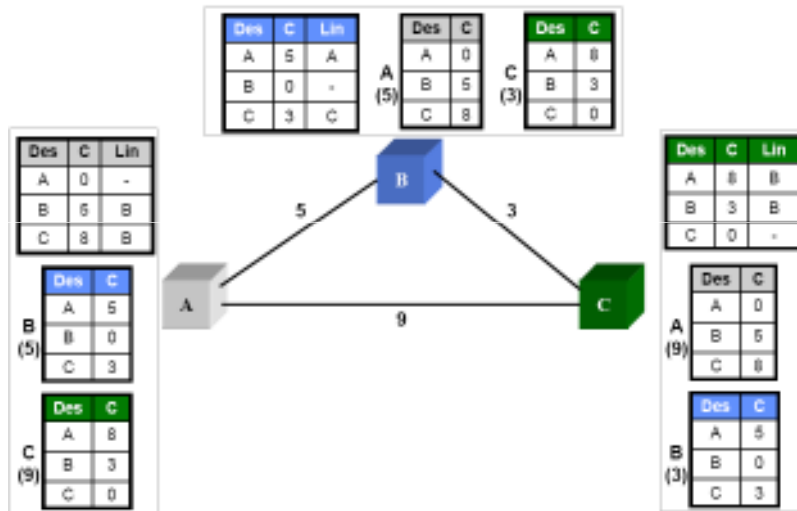
19

Distance Vector



20

Distance Vector



21

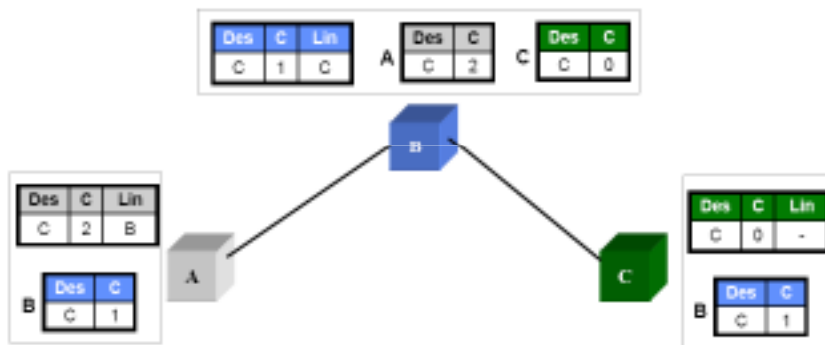
Distance Vector



- Vantagens
 - Simplicidade
 - Tempo de convergência baixo quando a rede opera bem.
- Principal desvantagem
 - Tempo de convergência muito alto quando ocorrem problemas na rede.
 - Problema da contagem para infinito (count-to-infinity problem)

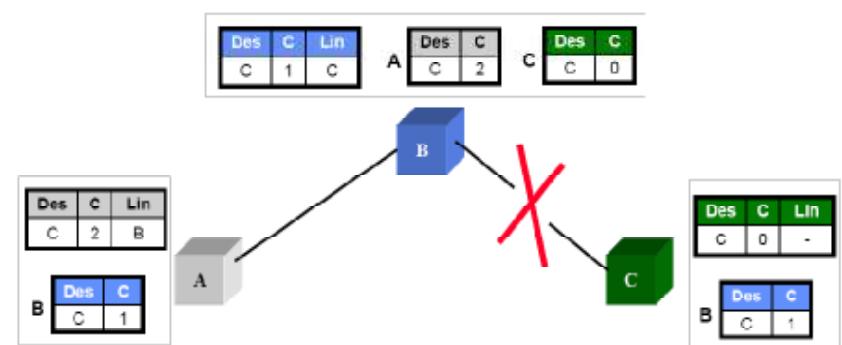
22

Distance Vector – Cont. p/ o infinito



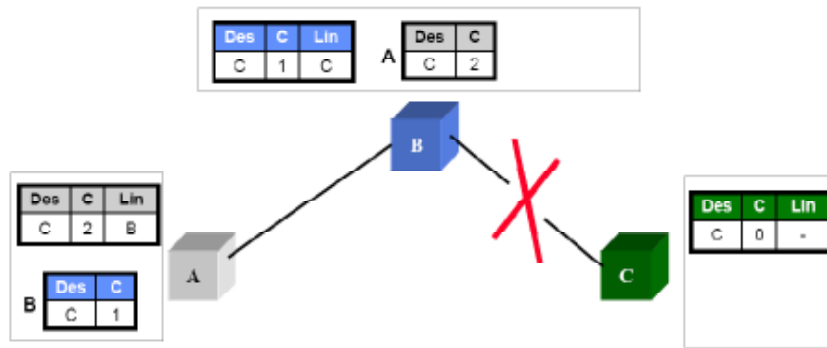
23

Distance Vector – Cont. p/ o infinito



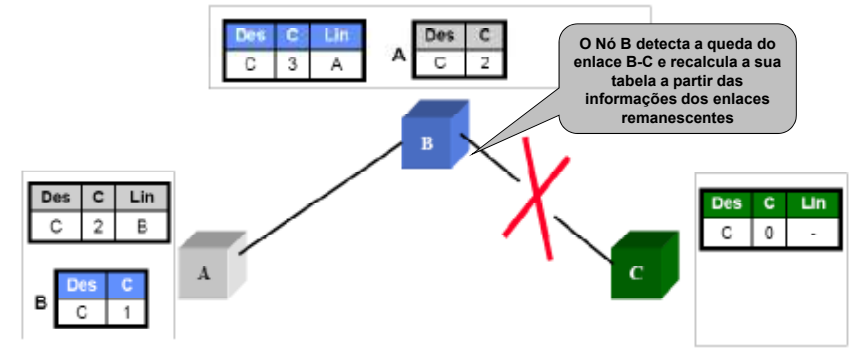
24

Distance Vector – Cont. p/ o infinito



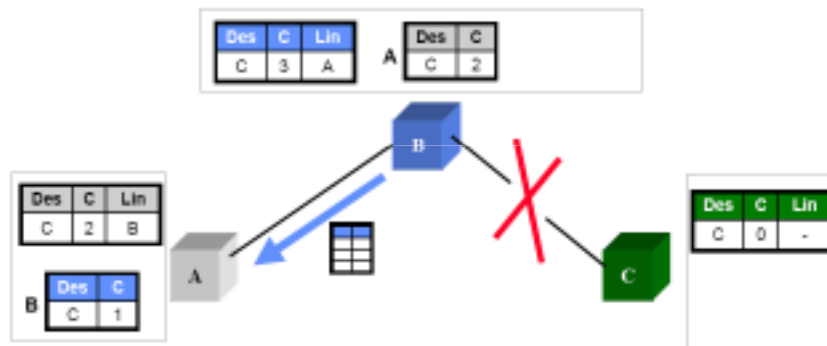
25

Distance Vector – Cont. p/ o infinito



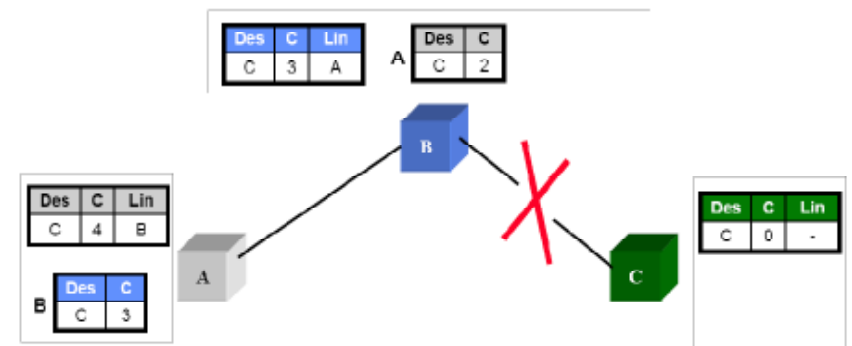
26

Distance Vector – Cont. p/ o infinito



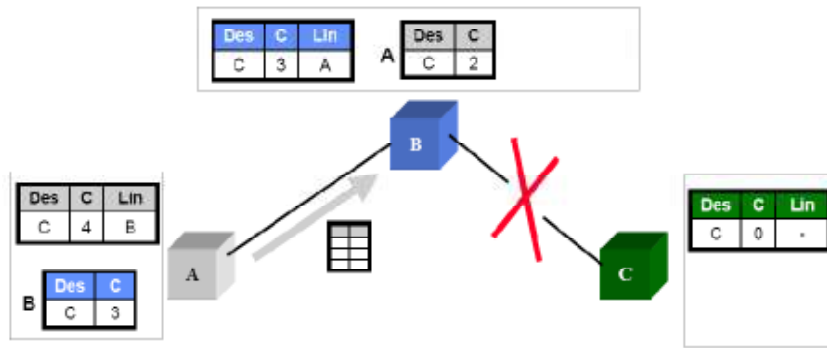
27

Distance Vector – Cont. p/ o infinito



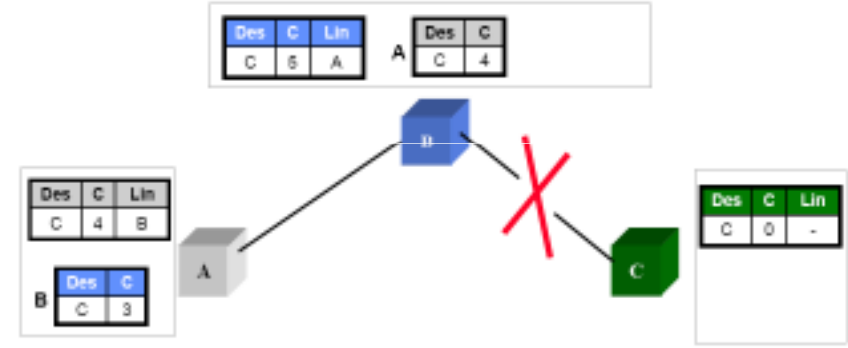
28

Distance Vector – Cont. p/ o infinito



29

Distance Vector – Cont. p/ o infinito



30

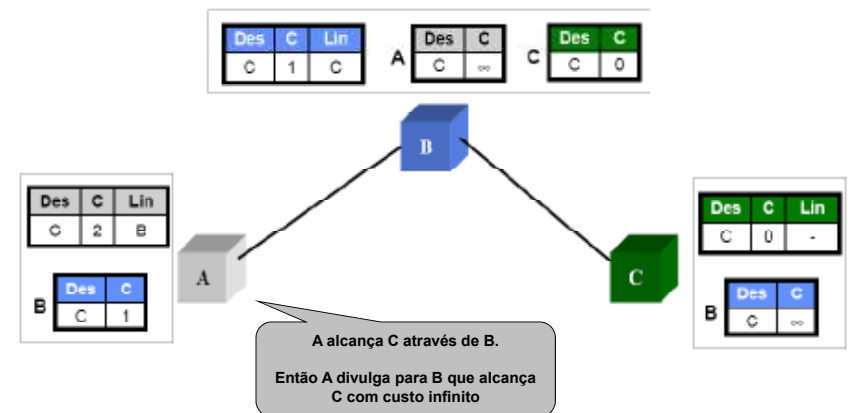
Distance Vector



- Proposta para melhorar a questão do tempo de convergência.
 - Split Horizon*: a distância ao nó X não é reportada na linha por onde pacotes com destino X são encaminhados
 - poison reverse*: a distância é reportada como sendo infinito

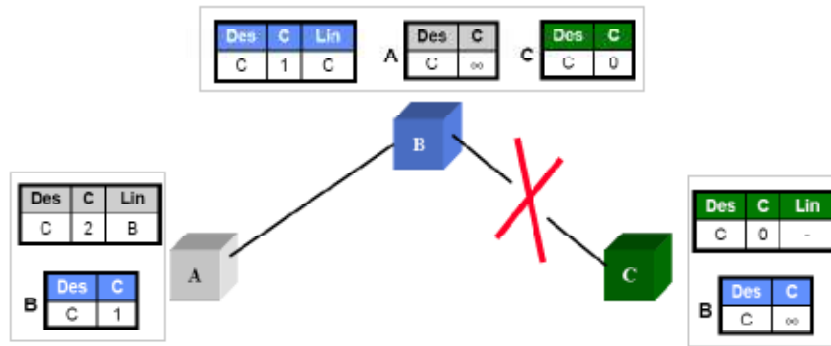
31

Distance Vector – Split horizon



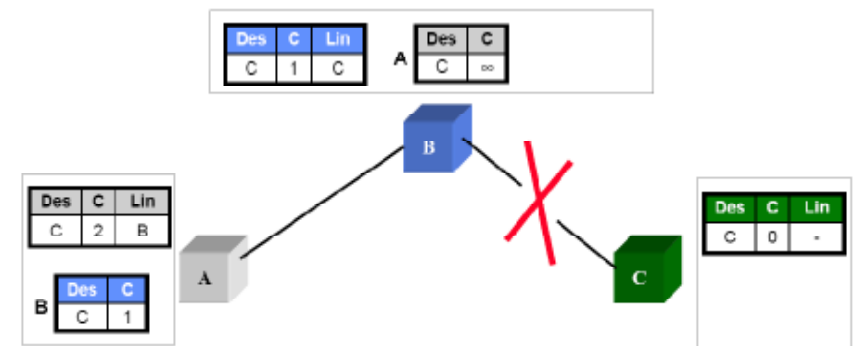
32

Distance Vector – Split horizon



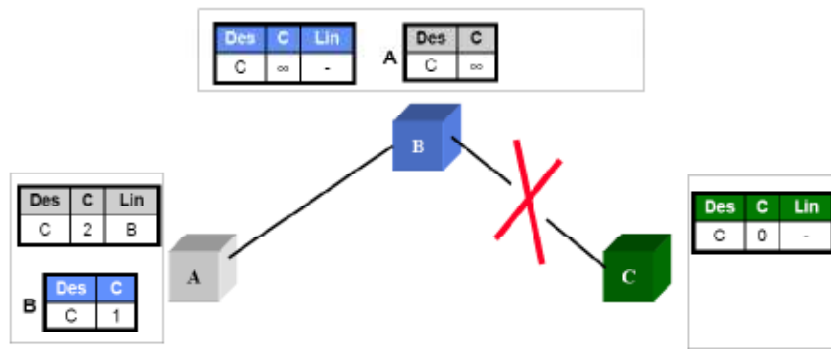
33

Distance Vector – Split horizon



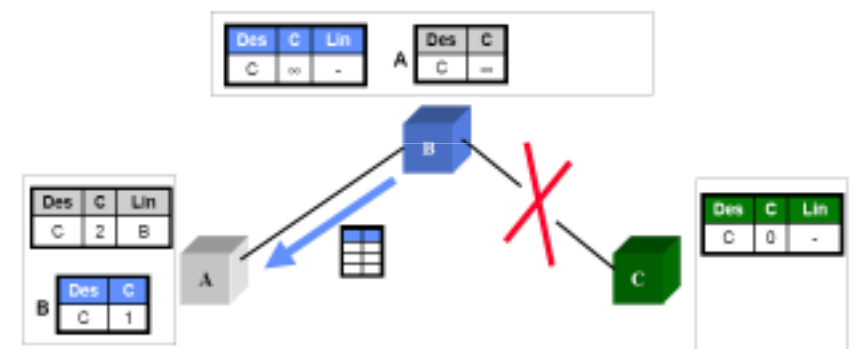
34

Distance Vector – Split horizon



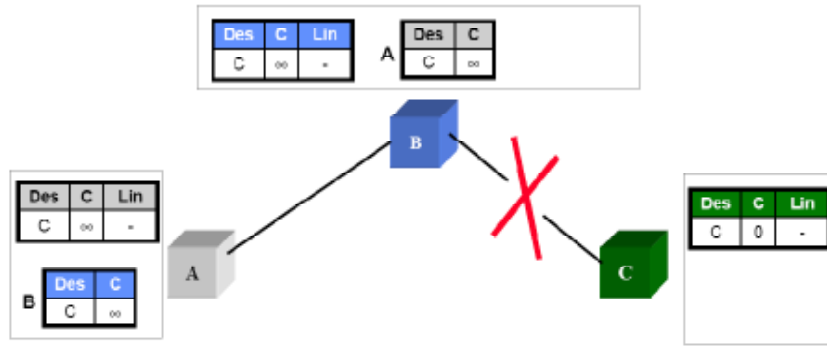
35

Distance Vector – Split horizon



36

Distance Vector – Split horizon



Exercícios

- Descreva o funcionamento do algoritmo de roteamento Distance Vector.
- Cite vantagens e desvantagens da utilização do algoritmo Distance Vector para a construção de tabelas de rotas.