

Aula 18

Camada de Rede

Roteamento intradomínio: estado do enlace

Igor Monteiro Moraes
Redes de Computadores

ATENÇÃO!

- Esta apresentação contém partes baseadas nos seguintes trabalhos
 - Notas de aula do Prof. Luís Henrique M. K. Costa, disponíveis em <http://www.gta.ufrj.br/ensino/CPE825/cpe825.html>
 - Notas de aula do Prof. José Augusto Suruagy Monteiro, disponíveis em <http://www.nuperc.unifacs.br/Members/jose.suruagy/cursos>
 - Material complementar do livro Computer Networking: A Top Down Approach, 5th edition, Jim Kurose and Keith Ross, Addison-Wesley, abril de 2009

Vetor de Distância

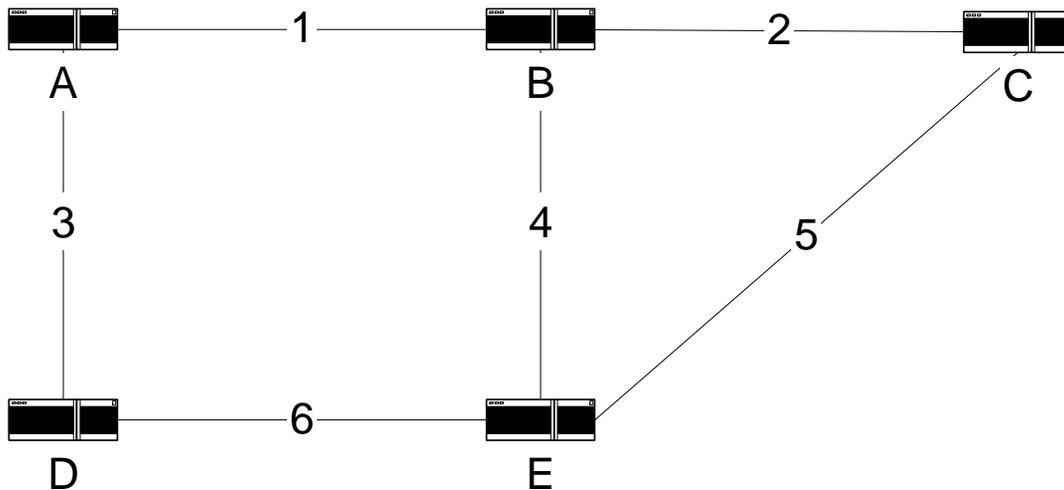
- Um nó conhece
 - Seus **vizinhos** diretos
 - **Custos** dos enlaces para os vizinhos
- } **Informação local**
- Cálculo de rotas
 - Baseado na **troca** de vetores de distância entre os **vizinhos**
 - Periodicamente, cada nó envia o seu próprio vetor de distâncias para os vizinhos
 - Quando um nó recebe um novo vetor de distância de um vizinho, ele atualiza a sua tabela de roteamento
 - Algoritmo de Bellman-Ford
 - Um nó conhece apenas o próximo salto para o destino

Estado do Enlace

- Um nó conhece
 - A **topologia completa** da rede
 - **Custos** de **todos** os enlaces até eles
- Informação
global**
- Cálculo de rotas
 - Baseado na **inundação da rede** com estados de enlaces
 - Constroem um **mapa distribuído** da topologia da rede
 - O mapa deve ser atualizado a cada mudança na topologia
 - Cada nó é capaz de calcular a melhor rota entre quaisquer dois nós da rede, a partir do mapa local
 - Algoritmo de Dijkstra

Mapa da Topologia

- Base de dados de estados do enlace

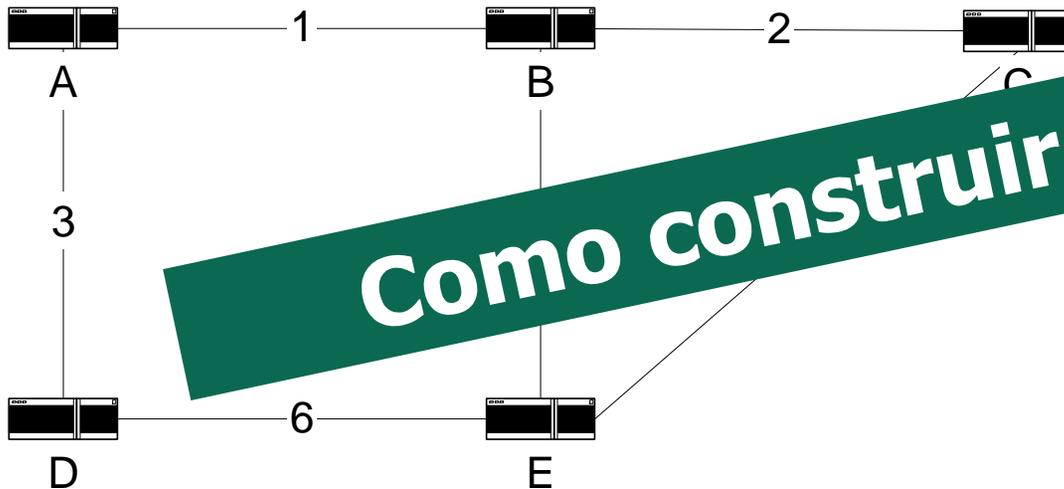


Todos os enlaces tem custo 1

De	Para	Enlace	Dist.
A	B	1	1
A	D	3	1
B	A	1	1
B	C	2	1
B	E	4	1
C	B	2	1
C	E	5	1
D	A	3	1
D	E	6	1
E	B	4	1
E	C	5	1
E	D	6	1

Mapa da Topologia

- Base de dados de estados do enlace



Como construir o mapa?

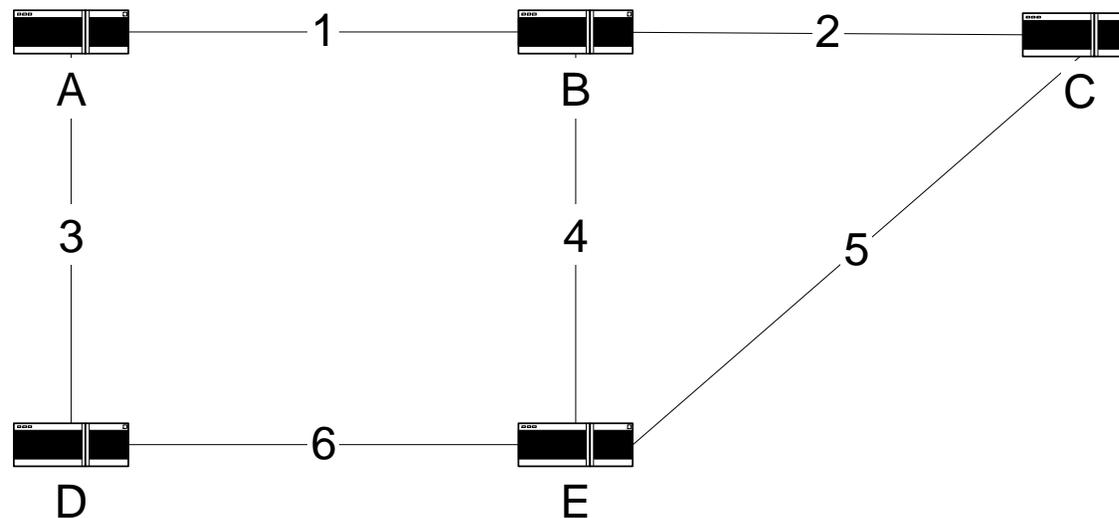
Todos os enlaces tem custo 1

De	Para	Enlace	Dist.
A	B	1	1
A	D	3	1
B	C	2	1
B	E	4	1
C	B	2	1
C	E	5	1
D	A	3	1
D	E	6	1
E	B	4	1
E	C	5	1
E	D	6	1

Mapa da Topologia

- Construído através da inundação de **estados do enlace**

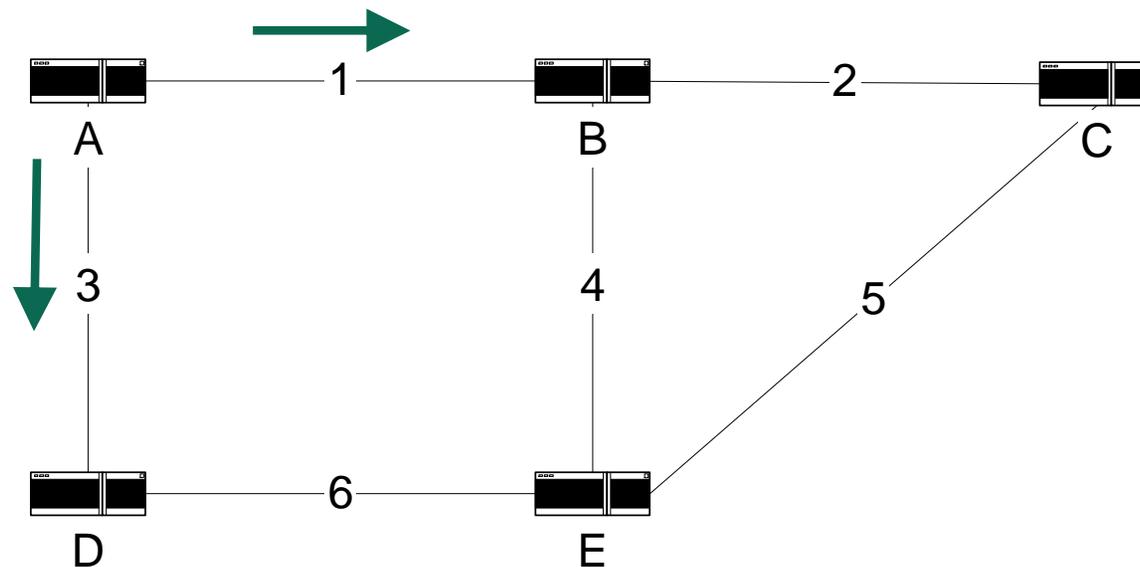
De **A**, para **B**, enlace **1**, dist. = **1**, no = **1**



Mapa da Topologia

- Construído através da inundação de **estados do enlace**

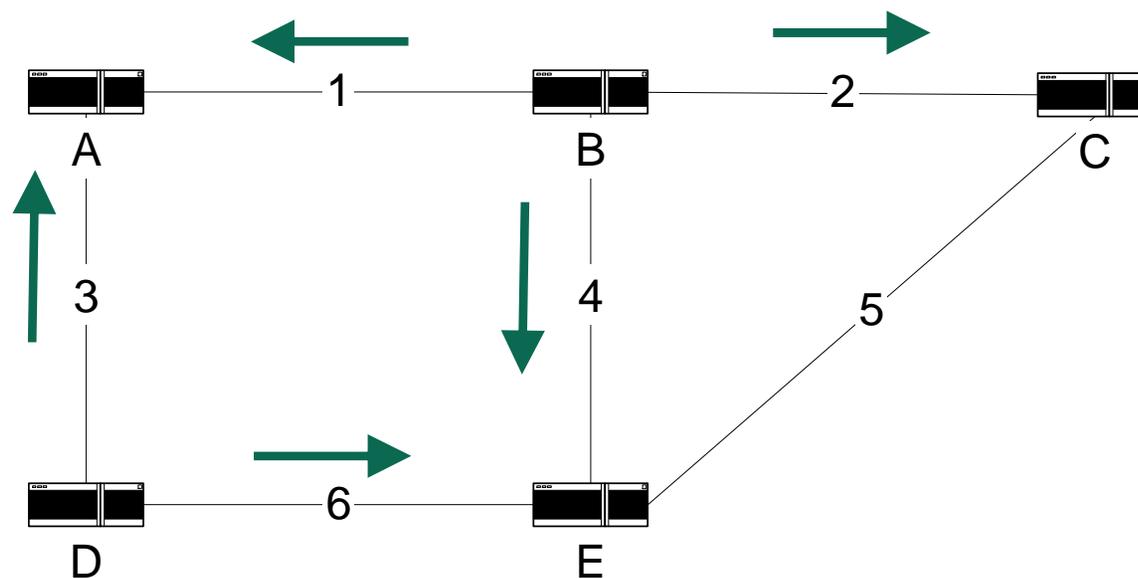
De **A**, para **B**, enlace **1**, dist. = **1**, no = **1**



Mapa da Topologia

- Construído através da inundação de **estados do enlace**

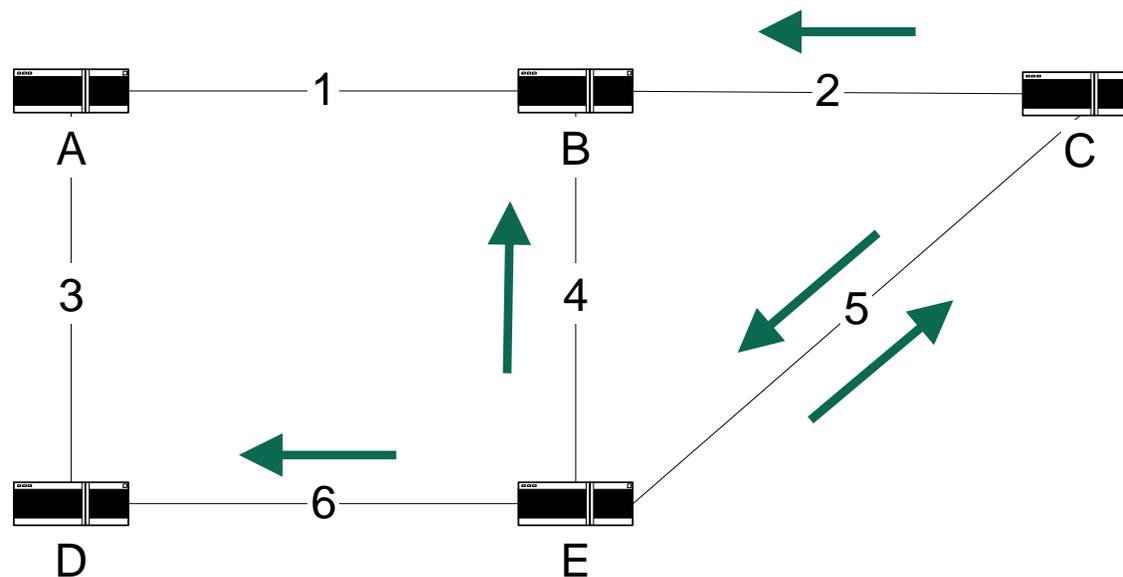
De **A**, para **B**, enlace **1**, dist. = **1**, no = **1**



Mapa da Topologia

- Construído através da inundação de **estados do enlace**

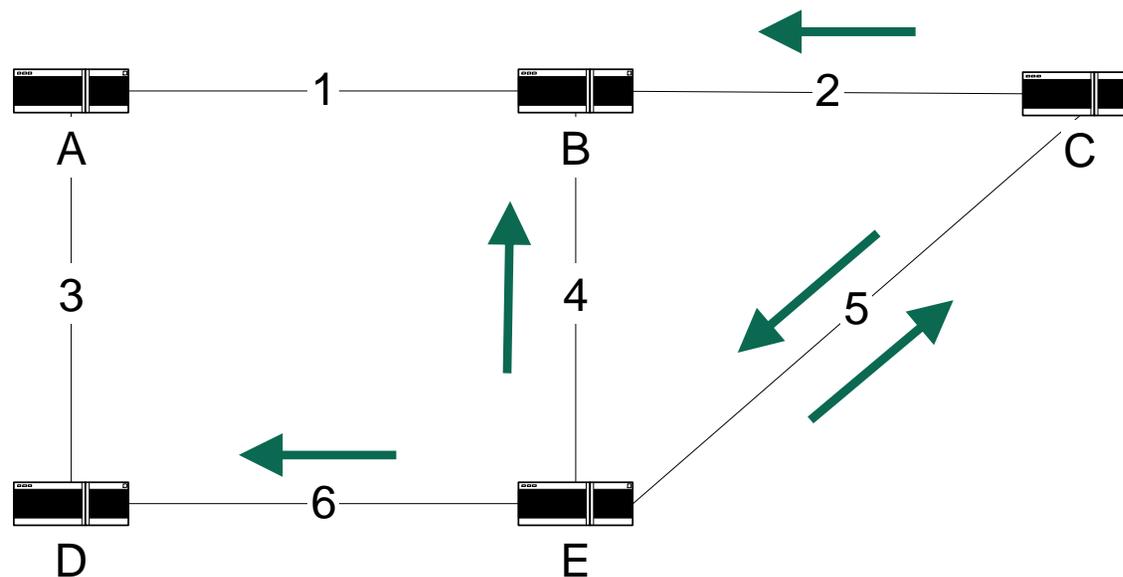
De **A**, para **B**, enlace **1**, dist. = **1**, no = **1**



Mapa da Topologia

- Construído através da inundação de **estados do enlace**

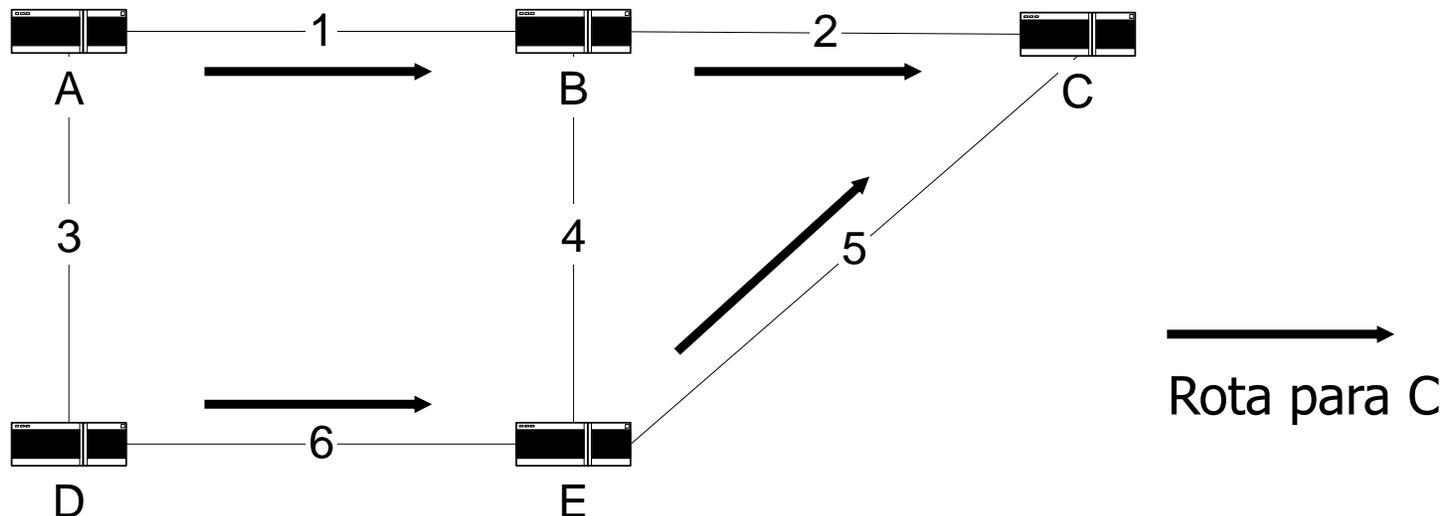
De **A**, para **B**, enlace **1**, dist. = **1**, no = **1**



Todos os nós possuem as mesmas informações

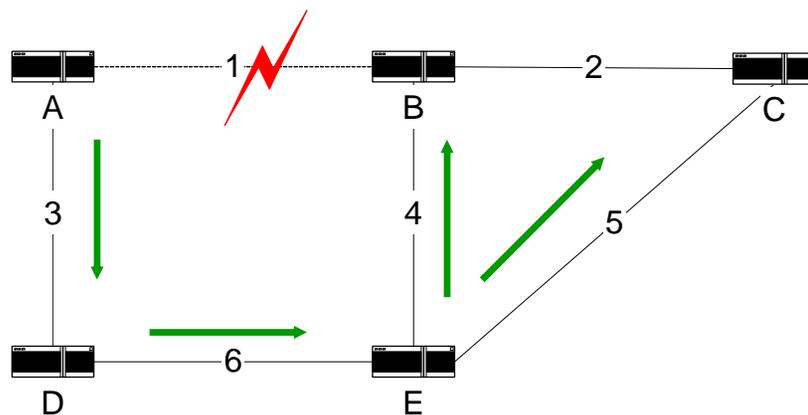
Estados do Enlace

- Cada nó pode calcular o caminho mais curto para todos os outros nós
- Não ocorrem *loops*
 - Todos os nós possuem a mesma base de dados
 - Bases devem ser consistentes



Protocolo de Inundação

- Detecção de falhas deve ser rápida e confiável
 - A solução é inundar a rede



De **A**, para **B**, enlace **1**, dist. = **inf.**

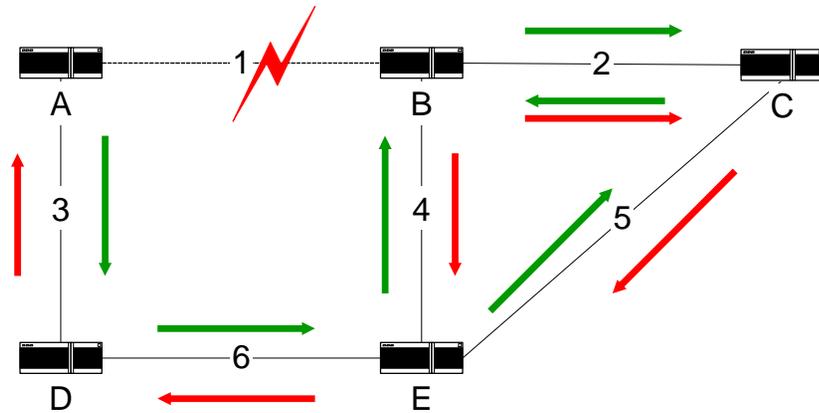
Protocolo de Inundação

- Mensagens “antigas” não devem contaminar as bases de dados
 - Mensagens devem ser identificadas (datadas)
 - Estampilha de tempo
 - Número de mensagem

Protocolo de Inundação

- Algoritmo
 1. Recepção da mensagem. Busca do registro na base de dados.
 2. **Se o registro não existia**, adicioná-lo à base. Enviar a mensagem em *broadcast*.
 3. Senão, **se o número na base de dados é menor** que o número recebido na mensagem, substituir o registro pelo novo valor. Enviar a mensagem em *broadcast*.
 4. Senão, **se o número na base de dados é maior**, transmitir o registro da base de dados em uma nova mensagem, na interface de entrada.
 5. Senão, **se os números são iguais**, não fazer nada.

Detecção de uma Falha



De	Para	Enl.	Dist.	No.
A	B	1	inf.	2
A	D	3	1	1
B	A	1	inf.	2
B	C	2	1	1
B	E	4	1	1
C	B	2	1	1
C	E	5	1	1
D	A	3	1	1
D	E	6	1	1
E	B	4	1	1
E	C	5	1	1
E	D	6	1	1



De A, para B, enlace 1, dist. = inf., n = 2



De B, para A, enlace 1, dist. = inf., n = 2

Números de Seqüência

- A rede deve funcionar por tempo indeterminado
 - Número de seqüência **circular**
- Como decidir se $X < Y$?
 - Supõe-se que os números de seqüência são incrementados lentamente
 - Devido a mudanças de estado de enlaces, ou estouro de temporizadores (suficientemente longos)
 - Se $X + dx = Y$, onde dx é “pequeno”, $X < Y$
 - Definição de dx deve ser precisa
 - Coerência das bases de dados em todos os nós

Desconexão da Rede

- Enlace 6 falha

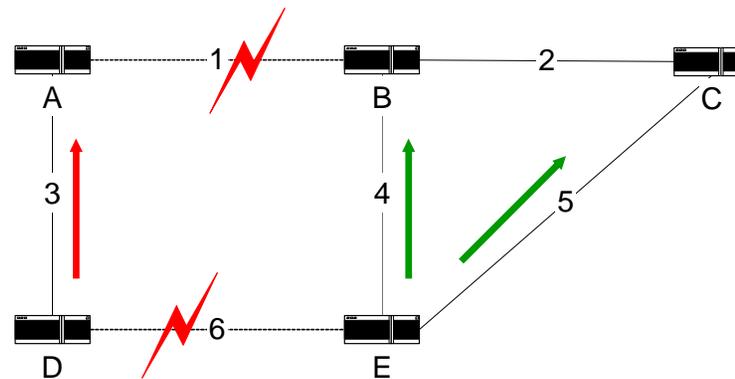


De **D**, para **E**, enlace **6**, dist. = **inf.**, n = **2**



De **E**, para **D**, enlace **6**, dist. = **inf.**, n = **2**

De	Para	Enl.	Dist.	No.
A	B	1	inf.	2
A	D	3	1	1
B	A	1	inf.	2
B	C	2	1	1
B	E	4	1	1
C	B	2	1	1
C	E	5	1	1
D	A	3	1	1
D	E	6	inf.	2
E	B	4	1	1
E	C	5	1	1
E	D	6	1	1



De	Para	Enl.	Dist.	No.
A	B	1	inf.	2
A	D	3	1	1
B	A	1	inf.	2
B	C	2	1	1
B	E	4	1	1
C	B	2	1	1
C	E	5	1	1
D	A	3	1	1
D	E	6	1	1
E	B	4	1	1
E	C	5	1	1
E	D	6	inf.	2

Mapas da topologia diferentes

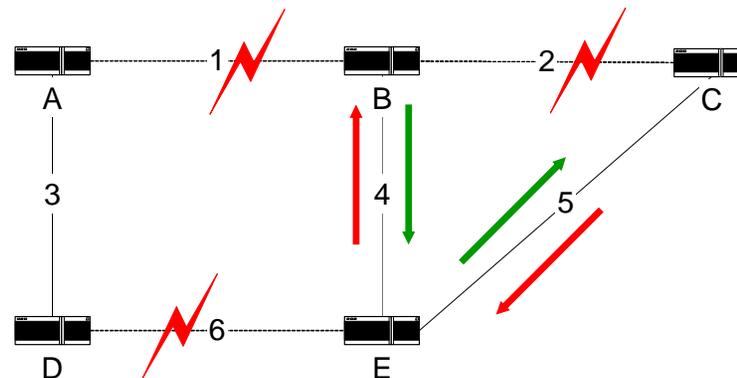
Não há problema, pois destinos na outra parte da rede estão inalcançáveis

Mapas evoluem separadamente

- Enlace 2 falha

De B, para C, enlace 2, dist. = inf., n = 2 De C, para B, enlace 2, dist. = inf., n = 2

De	Para	Enl.	Dist.	No.
A	B	1	inf.	2
A	D	3	1	1
B	A	1	inf.	2
B	C	2	1	1
B	E	4	1	1
C	B	2	1	1
C	E	5	1	1
D	A	3	1	1
D	E	6	inf.	2
E	B	4	1	1
E	C	5	1	1
E	D	6	1	1



De	Para	Enl.	Dist.	No.
A	B	1	inf.	2
A	D	3	1	1
B	A	1	inf.	2
B	C	2	inf.	2
B	E	4	1	1
C	B	2	inf.	2
C	E	5	1	1
D	A	3	1	1
D	E	6	1	1
E	B	4	1	1
E	C	5	1	1
E	D	6	inf.	2

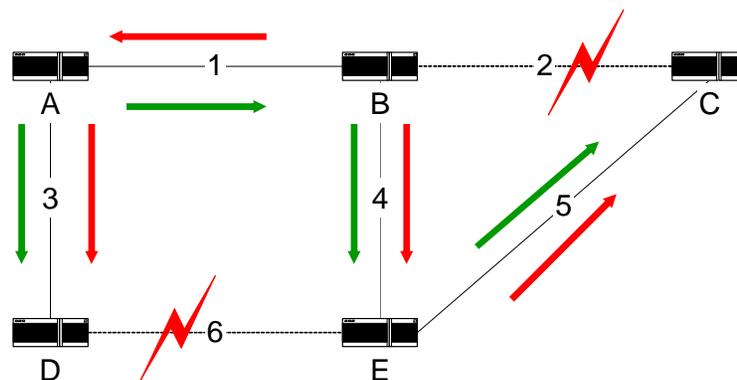
Recuperando Adjacências

- Enlace 1 é religado

De **A**, para **B**, enlace **1**, dist. = **1**, n = **3**

De **B**, para **A**, enlace **1**, dist. = **1**, n = **3**

De	Para	Enl.	Dist.	No.
A	B	1	1	3
A	D	3	1	1
B	A	1	1	3
B	C	2	1	1
B	E	4	1	1
C	B	2	1	1
C	E	5	1	1
D	A	3	1	1
D	E	6	inf.	2
E	B	4	1	1
E	C	5	1	1
E	D	6	1	1



De	Para	Enl.	Dist.	No.
A	B	1	1	3
A	D	3	1	1
B	A	1	1	3
B	C	2	inf.	2
B	E	4	1	1
C	B	2	inf.	2
C	E	5	1	1
D	A	3	1	1
D	E	6	1	1
E	B	4	1	1
E	C	5	1	1
E	D	6	inf.	2

O envio de apenas uma atualização de registro (enlace 1) não é suficiente

Recuperando Adjacências

- Alinhamento dos mapas
 - Identificadores de enlace + Números de versão
 - Para cada registro deve ser armazenada a versão mais recente
 - Maior número de versão
- Solução “ingênua”
 - Cada nó envia sua base de dados completa
 - Desperdício de recursos
 - Muitos registros podem ter versões em comum

Recuperando Adjacências

- OSPF
 - **Pacotes de descrição da base de dados**
 - Contêm apenas os identificadores de enlace e números de versão
 - São enviados numa primeira fase
 - Na segunda fase, apenas registros interessantes são pedidos ao vizinho
 - Registros novos ou com número de versão maior

Recuperando Adjacências

Após a o envio dos descritores...



De **D**, para **E**, enlace **6**, dist. = **inf.**, n = **2**

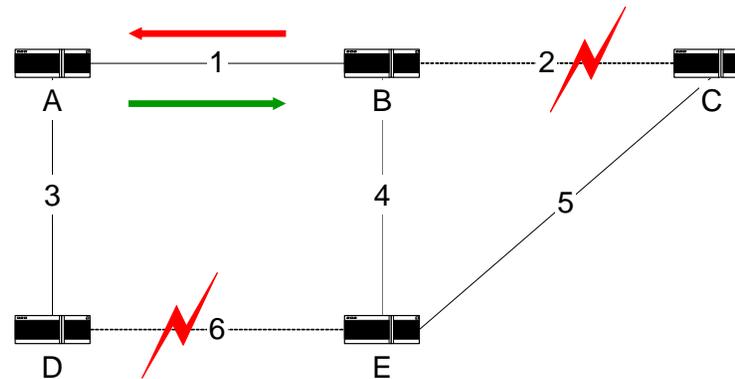


De **B**, para **C**, enlace **2**, dist. = **inf.**, n = **2**

De **C**, para **B**, enlace **2**, dist. = **inf.**, n = **2**

De **E**, para **D**, enlace **2**, dist. = **inf.**, n = **2**

De	Para	Enl.	Dist.	No.
A	B	1	1	3
A	D	3	1	1
B	A	1	1	3
B	C	2	1	1
B	E	4	1	1
C	B	2	1	1
C	E	5	1	1
D	A	3	1	1
D	E	6	inf.	2
E	B	4	1	1
E	C	5	1	1
E	D	6	1	1



De	Para	Enl.	Dist.	No.
A	B	1	1	3
A	D	3	1	1
B	A	1	1	3
B	C	2	inf.	2
B	E	4	1	1
C	B	2	inf.	2
C	E	5	1	1
D	A	3	1	1
D	E	6	1	1
E	B	4	1	1
E	C	5	1	1
E	D	6	inf.	2

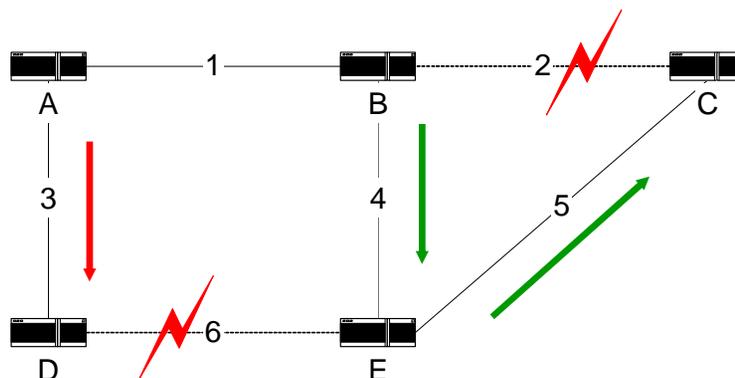
Recuperando Adjacências

Atualizações são propagadas

De **B**, para **C**, enlace **2**, dist. = **inf.**, n = **2**
 De **C**, para **B**, enlace **2**, dist. = **inf.**, n = **2**
 De **E**, para **D**, enlace **2**, dist. = **inf.**, n = **2**

De **D**, para **E**, enlace **6**, dist. = **inf.**, n = **2**

De	Para	Enl.	Dist.	No.
A	B	1	1	3
A	D	3	1	1
B	A	1	1	3
B	C	2	inf.	2
B	E	4	1	1
C	B	2	inf.	2
C	E	5	1	1
D	A	3	1	1
D	E	6	inf.	2
E	B	4	1	1
E	C	5	1	1
E	D	6	inf.	2



De	Para	Enl.	Dist.	No.
A	B	1	1	3
A	D	3	1	1
B	A	1	1	3
B	C	2	inf.	2
B	E	4	1	1
C	B	2	inf.	2
C	E	5	1	1
D	A	3	1	1
D	E	6	inf.	2
E	B	4	1	1
E	C	5	1	1
E	D	6	inf.	2

Mapas sincronizados

Proteção dos Mapas

- Sincronia das bases de dados
 - Fundamental para roteamento coerente
- Porém, acidentes podem ocorrer
 - Falhas na inundaç o ou sincronizaç o
 - Registros desatualizados
 - Erros de mem ria
 - Introduç o volunt ria de informaç o err nea

Proteção dos Mapas no OSPF

- A inundação inclui reconhecimentos salto-a-salto
- Os pacotes de descrição são transmitidos de maneira segura
- Cada registro de estado do enlace é associado a um temporizador e é retirado se não for devidamente atualizado
- Registros são protegidos por um *checksum*
- As mensagens podem ser autenticadas
 - Ex. senhas

Problema após falha de um nó

- Nó X desliga e religa após um curto intervalo
- X envia estado do enlace com no. de seq. = 1
- X recebe estados de enlace de nós vizinhos com no. de seq. maior
 - Mas na verdade, menos atuais
- Solução
 - X deve re-enviar os “seus” registros, com no. de seq. igual ao no. de seq. recebido + 1

Algoritmo de Dijkstra

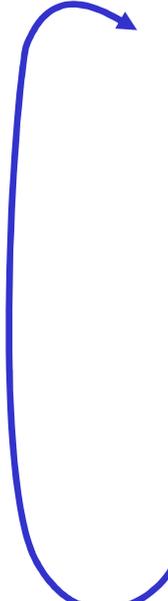
- Também chamado de *Shortest Path First* (SPF)
- Cálculo do caminho mais curto entre um nó e todos os outros nós da rede
 - Constrói a tabela de rota para um dado nó
- É iterativo
 - Depois de k iterações, sabe-se o menor custo p/ k destinos

Algoritmo de Dijkstra

- Notação
 - $c(x,y)$: custo do enlace do nó x ao nó y
 - Custo é infinito se não forem vizinhos diretos
 - $D(v)$: valor corrente do custo do caminho da origem ao destino v
 - $p(v)$: nó antecessor no caminho da origem ao nó v , imediatamente antes de v
 - N' : conjunto de nós cujo caminho de menor custo já foi determinado

Algoritmo de Dijkstra

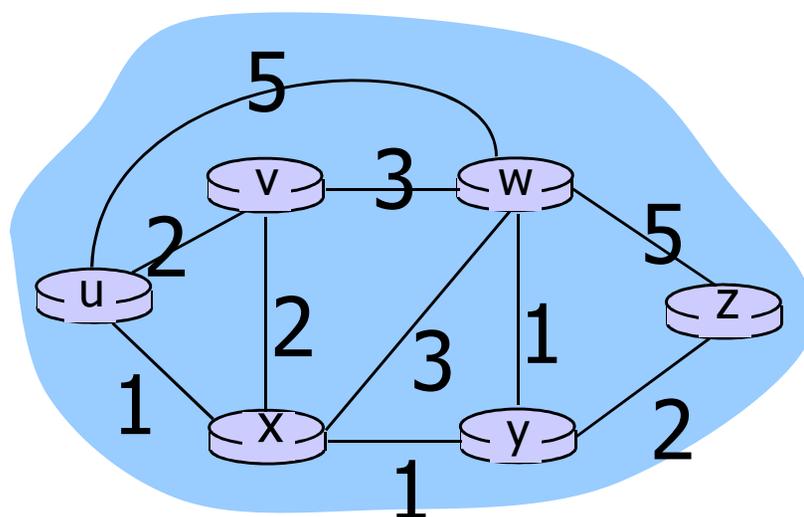
- 1 *Inicialização:*
- 2 $N' = \{u\}$
- 3 para todos os nós v
- 4 se v for um vizinho do nó u
- 5 então $D(v) = c(u,v)$
- 6 senão $D(v) = \infty$
- 7
- 8 *Loop*
- 9 encontre w não contido em N' tal que $D(w)$ é um mínimo
- 10 adicione w ao conjunto N'
- 11 atualize $D(v)$ para cada vizinho v de w e ainda não em N' :
- 12 $D(v) = \min(D(v), D(w) + c(w,v))$
- 13 /* o novo custo ao nó v é o custo velho a v ou o custo do
- 14 menor caminho conhecido para w , mais o custo de w a v */
- 15 *até que todos os nós estejam em N'*



Algoritmo de Dijkstra: Exemplo

Etapa	N'	$D(v), p(v)$	$D(x), p(x)$	$D(w), p(w)$	$D(y), p(y)$	$D(z), p(z)$
0	u	2, u	1, u	5, u	inf.	inf.

Nó fonte: u

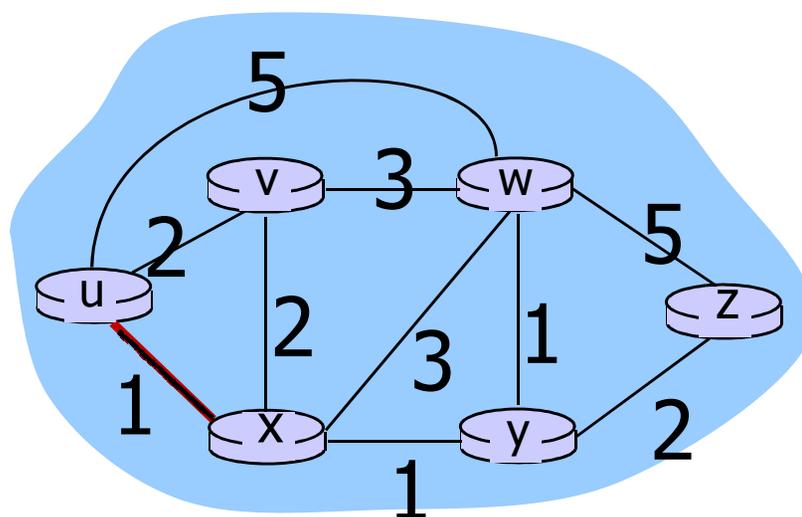


Algoritmo de Dijkstra: Exemplo

Etapa	N'	$D(v), p(v)$	$D(x), p(x)$	$D(w), p(w)$	$D(y), p(y)$	$D(z), p(z)$
0	u	2,u	1,u	5,u	inf.	inf.

Adicionar x ao conjunto N'

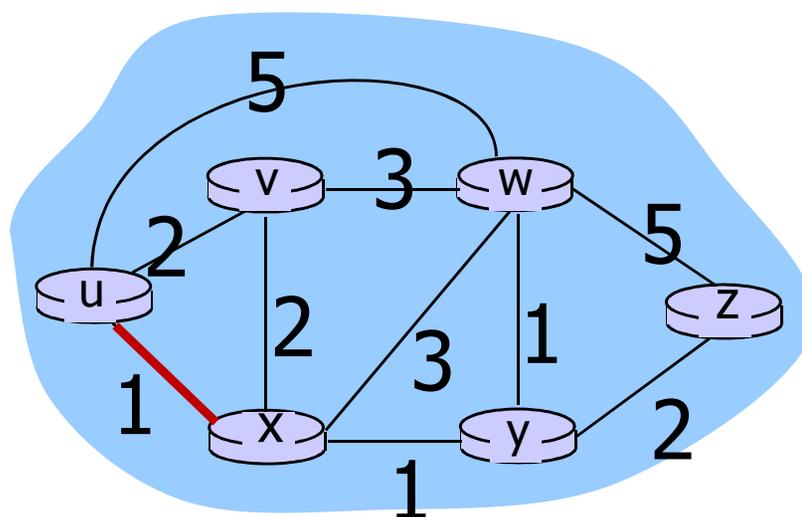
Nó fonte: u



Algoritmo de Dijkstra: Exemplo

Etapa	N'	$D(v), p(v)$	$D(x), p(x)$	$D(w), p(w)$	$D(y), p(y)$	$D(z), p(z)$
0	u	2,u	1,u	5,u	inf.	inf.
1	u,x	2,u		4,x	2,x	inf.

Nó fonte: u

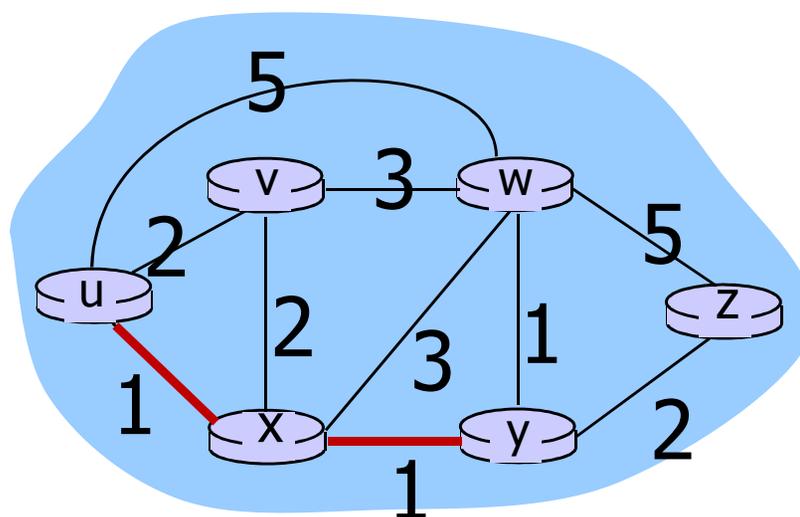


Algoritmo de Dijkstra: Exemplo

Etapa	N'	D(v),p(v)	D(x),p(x)	D(w),p(w)	D(y), p(y)	D(z),p(z)
0	u	2,u	1,u	5,u	inf.	inf.
1	u,x	2,u		4,x	2,x	inf.

Adicionar y ao conjunto N'

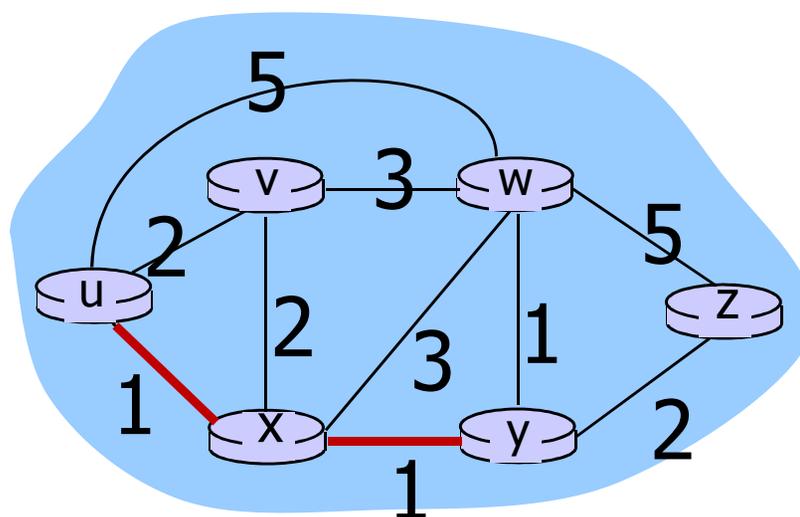
Nó fonte: u



Algoritmo de Dijkstra: Exemplo

Etapa	N'	$D(v), p(v)$	$D(x), p(x)$	$D(w), p(w)$	$D(y), p(y)$	$D(z), p(z)$
0	u	2,u	1,u	5,u	inf.	inf.
1	u,x	2,u		4,x	2,x	inf.
2	u,x,y	2,u		3,y		4,y

Nó fonte: u

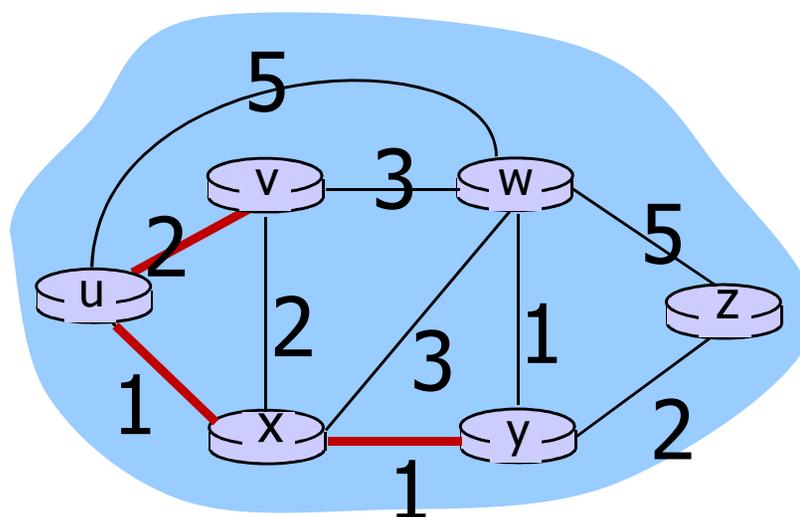


Algoritmo de Dijkstra: Exemplo

Etapa	N'	$D(v), p(v)$	$D(x), p(x)$	$D(w), p(w)$	$D(y), p(y)$	$D(z), p(z)$
0	u	2,u	1,u	5,u	inf.	inf.
1	u,x	2,u		4,x	2,x	inf.
2	u,x,y	2,u		3,y		4,y

Adicionar v ao conjunto N'

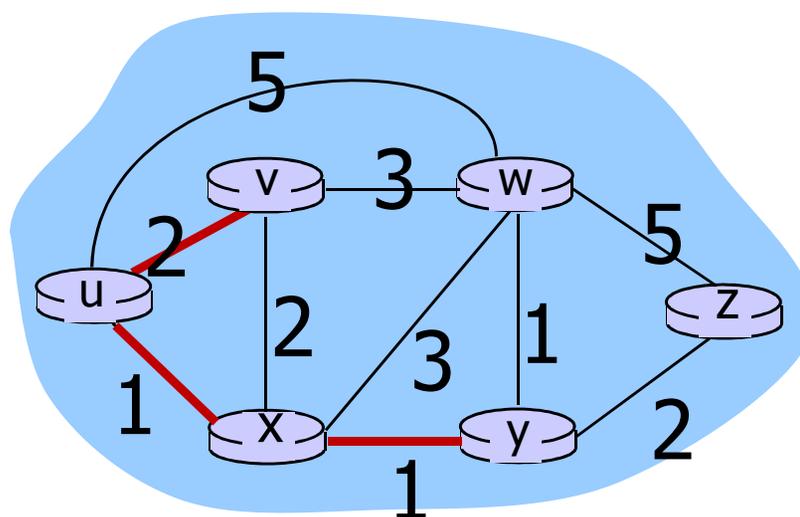
Nó fonte: u



Algoritmo de Dijkstra: Exemplo

Etapa	N'	$D(v), p(v)$	$D(x), p(x)$	$D(w), p(w)$	$D(y), p(y)$	$D(z), p(z)$
0	u	2,u	1,u	5,u	inf.	inf.
1	u,x	2,u		4,x	2,x	inf.
2	u,x,y	2,u		3,y		4,y
3	u,x,y,v			3,y		4,y

Nó fonte: u

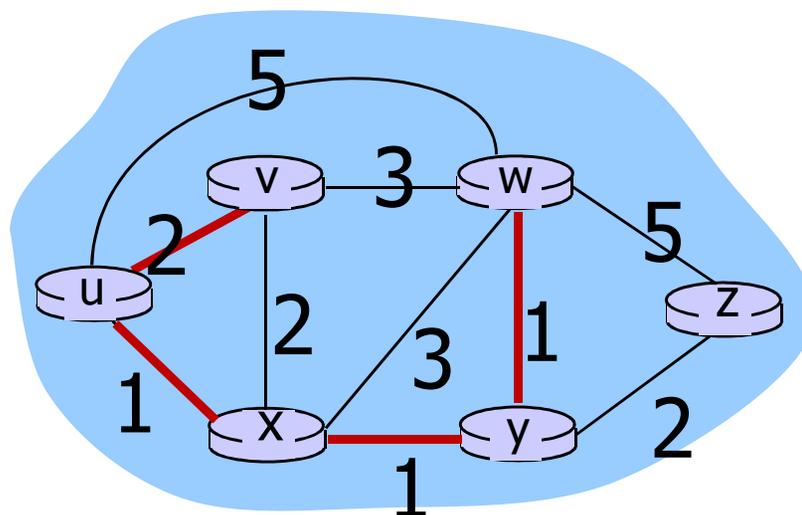


Algoritmo de Dijkstra: Exemplo

Etapa	N'	$D(v), p(v)$	$D(x), p(x)$	$D(w), p(w)$	$D(y), p(y)$	$D(z), p(z)$
0	u	2,u	1,u	5,u	inf.	inf.
1	u,x	2,u		4,x	2,x	inf.
2	u,x,y	2,u		3,y		4,y
3	u,x,y,v			3,y		4,y

Adicionar w ao conjunto N'

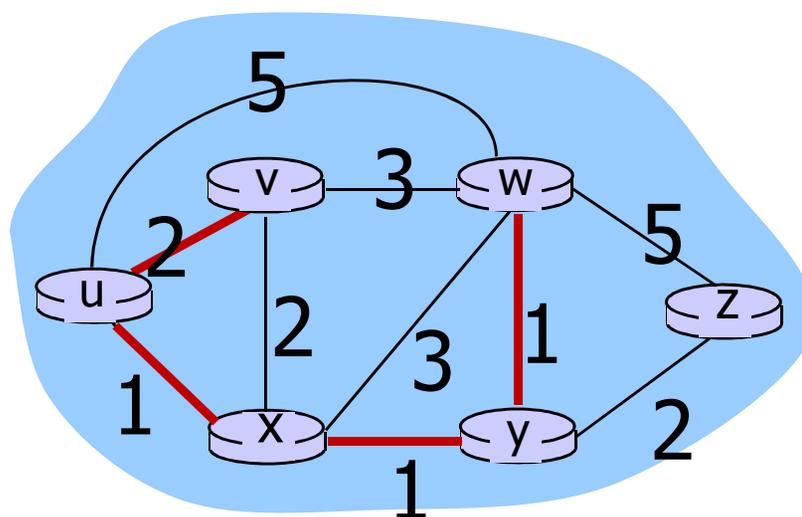
Nó fonte: u



Algoritmo de Dijkstra: Exemplo

Etapa	N'	D(v),p(v)	D(x),p(x)	D(w),p(w)	D(y), p(y)	D(z),p(z)
0	u	2,u	1,u	5,u	inf.	inf.
1	u,x	2,u		4,x	2,x	inf.
2	u,x,y	2,u		3,y		4,y
3	u,x,y,v			3,y		4,y
4	u,x,y,v,w					4,y

Nó fonte: u

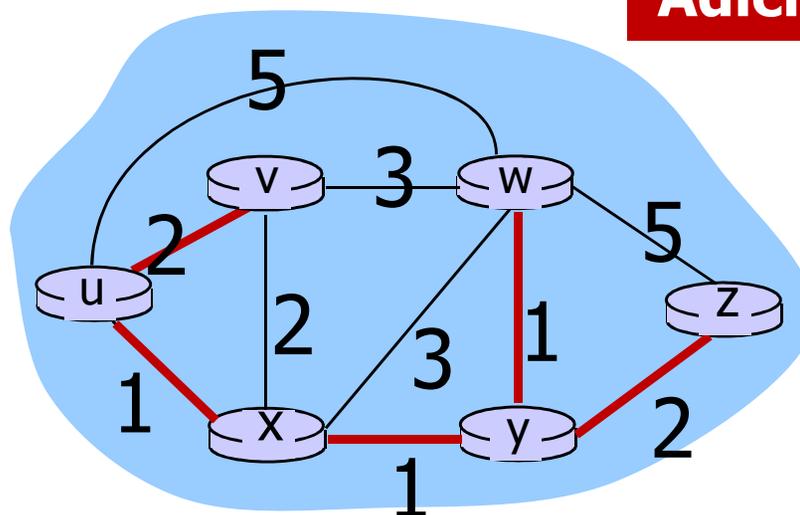


Algoritmo de Dijkstra: Exemplo

Etapa	N'	$D(v), p(v)$	$D(x), p(x)$	$D(w), p(w)$	$D(y), p(y)$	$D(z), p(z)$
0	u	2,u	1,u	5,u	inf.	inf.
1	u,x	2,u		4,x	2,x	inf.
2	u,x,y	2,u		3,y		4,y
3	u,x,y,v			3,y		4,y
4	u,x,y,v,w					4,y

Adicionar z ao conjunto N'

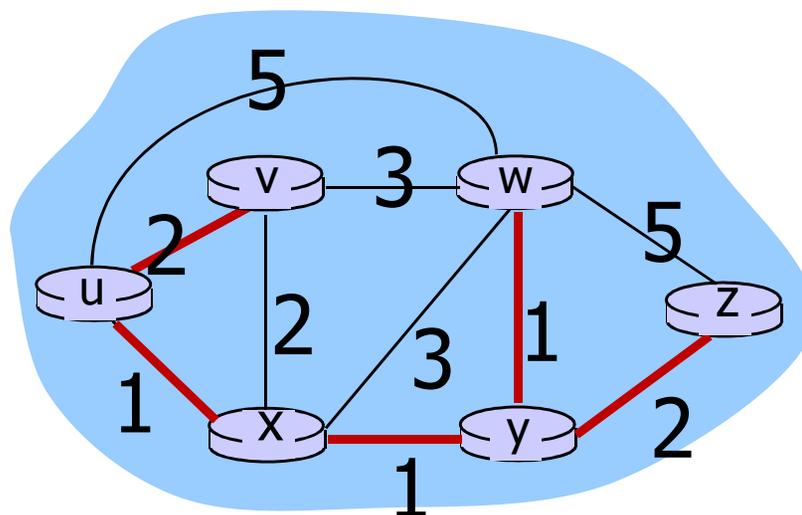
Nó fonte: u



Algoritmo de Dijkstra: Exemplo

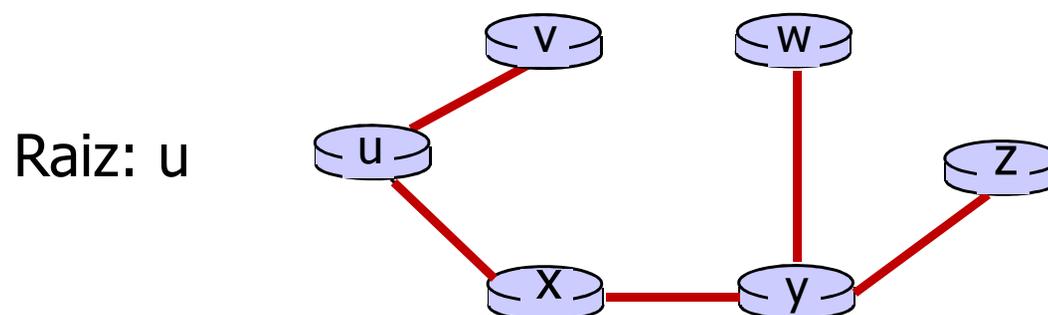
Etapa	N'	D(v),p(v)	D(x),p(x)	D(w),p(w)	D(y), p(y)	D(z),p(z)
0	u	2,u	1,u	5,u	inf.	inf.
1	u,x	2,u		4,x	2,x	inf.
2	u,x,y	2,u		3,y		4,y
3	u,x,y,v			3,y		4,y
4	u,x,y,v,w					4,y
5	u,v,x,w,y,z					

Nó fonte: u



Algoritmo de Dijkstra: Exemplo

- Árvore de caminhos mais curtos



- Tabela de encaminhamento: u para os demais nós

Destino	Enlace de saída
v	u,v
x	u,x
w	u,x
y	u,x
z	u,x

Complexidade do Algoritmo

- Se um rede possui n nós
- A cada iteração
 - É preciso checar todos os nós que não estão em N'
- Resulta em $n*(n+1)/2$ comparações $\rightarrow O(n^2)$
- Implementações mais eficientes
 - Limite é $O(n \log n)$
- Lembrete
 - O número de passos para convergir do Algoritmo de Bellman-Ford é $O(nm)$, onde m é o número de arestas

Vantagens do Estado do Enlace

- Convergência mais rápida e livre de *loops*
- Suporte de métricas precisas ou múltiplas métricas
- Suporte de múltiplos caminhos
- Representação separada de rotas externas

Convergência Rápida

- Vetor de distância (DV)
 - Número de passos proporcional ao número de nós
 - Igual ao número de saltos do caminho mais longo, no pior caso
- Estado do enlace (LS)
 - Transmissão da atualização através de inundação
 - Cálculo de rotas realizado localmente

Convergência Rápida

- LS mais rápido
 - Atualizações disparadas (DV) ~ inundação (LS)
 - No melhor caso, em que 1 DV de atualização é suficiente
 - Intervalo entre atualizações disparadas
 - 1 a 5 s no RIP
 - Cálculo de rotas no LS
 - 200 ms para rede com ~ 200 nós
 - Falha de enlace → vários destinos são “atingidos”
 - DV: tamanho de mensagem ~ número de destinos
- **Livre de *loops***
 - Logo após inundação e cálculo de rotas

Métricas “Precisas”

- Com o estado do enlace
 - Conhecimento **completo** da topologia
 - Cálculo de rotas é local com base no mapa da topologia
 - Métricas podem ser arbitrariamente precisas
 - Divulgadas pelos estados dos enlaces

- Com o vetor de distância

Granularidade fina
+
grande diferença entre a menor
e a maior métrica dos enlaces

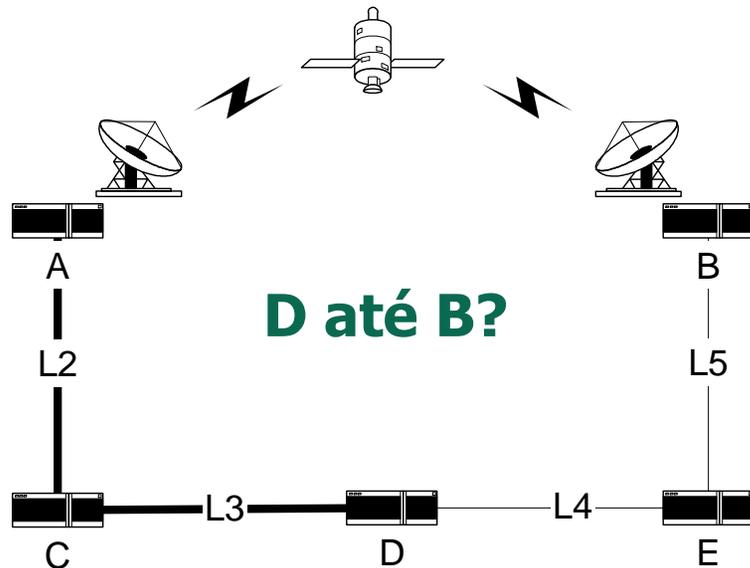
=

risco de convergência
muito lenta

Múltiplas Métricas

- Com o estado do enlace
 - Múltiplos estados do enlace
 - Um para cada métrica ou um informando várias métricas
 - Cálculo de múltiplas tabelas de roteamento
 - Uma para cada métrica
 - Indicação de que rota um determinado pacote deve seguir

Exemplo de Múltiplas Métricas



—————
Enlace T1 terrestre – 1,5 Mb/s, 10 ms

—————
Enlace terrestre – 64 kb/s, 10ms

Satélite

Enlace T1, 1,5 Mb/s, 275 ms

- $D \rightarrow C \rightarrow A \rightarrow B$
 - Vazão de 1,5 Mb/s, atraso de 295 ms (escolhido pela métrica vazão)
- $D \rightarrow E \rightarrow B$
 - Vazão de 64 kb/s, atraso de 20 ms (escolhido pela métrica atraso)
- Escolha da métrica deve ser coerente
 - Risco de formação de *loops*

Suporte a Múltiplos Caminhos

- O objetivo é o balanceamento de carga
 - Diminui o atraso médio dos pacotes
 - A capacidade de transmissão é maior
 - Após a falha de um caminho
 - Apenas uma parte do tráfego é re-roteada
 - Todo o tráfego é desviado se apenas uma rota é utilizada
 - Algoritmo de Dijkstra pode ser modificado
 - Múltiplos caminhos de mesmo custo
- Em contrapartida
 - Pode haver desordem de pacotes

- Cenário: rota de saída da rede é **única**
 - Anúncio de rota *default* (DV e LS)
- Cenário: **múltiplas** rotas de saída
 - Anúncio de rota *default*
 - DV: caminho mais curto para a saída é o utilizado
 - Anúncio de rotas específicas
 - DV: mais entradas nos vetores de distância
 - LS: estados de enlace especiais

- Comparação
 - Métricas
 - LS: *gateway* anuncia a métrica que convier ao enlace de saída
 - DV: métrica limitada pelo valor de infinito
 - Número de rotas externas
 - LS: tamanho da base de dados
 - DV: tamanho dos vetores de distância
 - Porém, custo SPF = $O(N \cdot \log N)$, custo Bellman-Ford = $O(N^2)$

- *Open Shortest Path First*
 - Toda a funcionalidade de protocolos de estado do enlace (LS)
 - Base de dados distribuída
 - Procedimento de inundação
 - Descoberta de adjacências
 - Registros especiais para rotas externas
 - Mais
 - Separação entre estações e roteadores
 - Suporte de redes *broadcast* (Ethernet, Token Ring, FDDI)
 - Suporte de redes não-*broadcast* (X.25, ATM)
 - Divisão de redes muito grandes em áreas (roteamento hierárquico)

OSPF x RIP

- RIP
 - Vetores de distância
 - 2 mensagens de controle
 - DVs enviados a cada 30s
 - Tabela de roteamento
- OSPF
 - Estados do enlace
 - 5 mensagens de controle e 3 “sub-protocolos”
 - LSAs com reconhecimentos
 - Tabela de roteamento + tabela de estados do enlace
- OSPF é mais complexo, porém mais eficiente que o RIP
 - “O OSPF calcula melhores rotas com menos mensagens”

OSPF x RIP

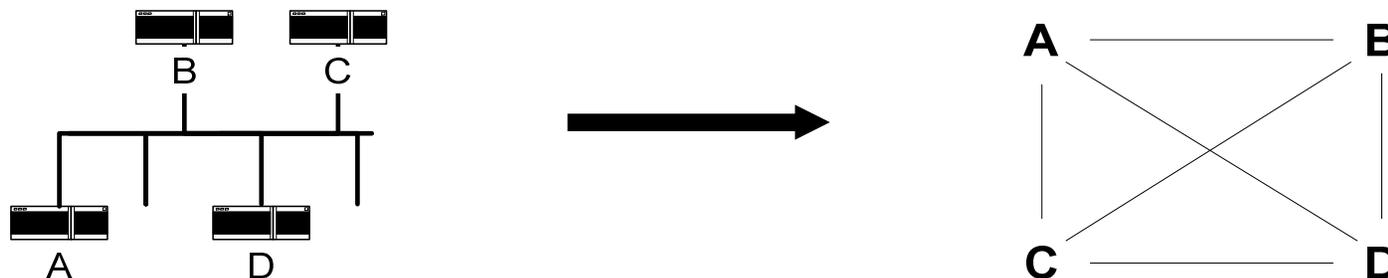
- Mais diferenças
 - Segurança
 - Todas mensagens OSPF autenticadas
 - Múltiplos caminhos de mesmo custo
 - O RIP permite e usa apenas uma rota
 - Suporte integrado para roteamento *unicast* e *multicast*
 - OSPF multicast (MOSPF) usa mesma base de dados de topologia usado pelo OSPF
 - OSPF é hierárquico
 - Pode ser usado em domínios grandes

Separação de Estações e Roteadores

- Em uma rede local, um estado de enlace por estação
 - Não é escalável
- OSPF
 - Estados de enlace
 - Enlace de roteador (*router link*)
 - Conexão entre roteadores
 - Endereço IP do vizinho
 - Enlace para rede stub (*link to a stub network*)
 - Conexão a uma rede local
 - Número de rede ou de sub-rede

Redes *Broadcast*

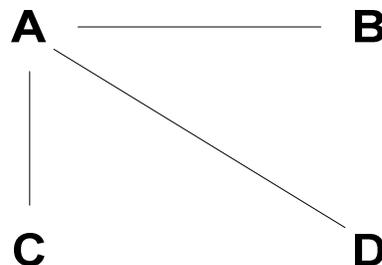
- Conectividade total
 - Todas as estações podem se falar diretamente
- Capacidade de difusão (nativa)
 - broadcast – todas as estações recebem
 - multicast – um grupo de estações recebe
- Problema: **adjacências**



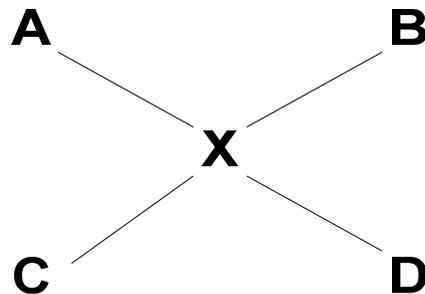
- Dados N roteadores, $N.(N-1) / 2$ adjacências

Redes *Broadcast*

- Cada roteador anuncia
 - N-1 estados de enlace (enlaces para os outros roteadores)
 - 1 estado para as estações na rede (*stub network link*)
- Total = N^2 mensagens
- Redução do número de adjacências
 - Um roteador é escolhido como roteador “designado”
 - Outros roteadores na rede local estabelecem adjacências apenas com este roteador

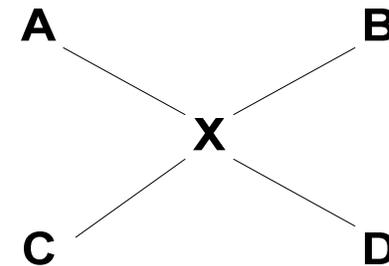


- Funcionamento
 - Primeiro passo: eleição do roteador designado
 - Segundo passo: outros roteadores recuperam adjacência com o roteador designado (ou se sincronizam com ele)
 - Se todos se sincronizam com **A**, todos estão sincronizados
 - N adjacências em vez de $N.(N-1) / 2$
- Além disso, o número de estados do enlace é reduzido
 - Roteador virtual (**X**) representa a rede broadcast



Redes *Broadcast*

- Base de dados
 - 2 enlaces por roteador
 - ***para*** e ***do*** nó virtual, X



- Anúncios
 - roteador para X – pelo *próprio roteador*
 - X para roteador – *pelo roteador designado*
 - Endereço IP do roteador designado ***na rede broadcast***
 - “Enlaces de rede” (*network links*)
 - Métrica nula para não causar problemas de cálculo de rotas
- Procedimento de inundação

- Inundação
 - Envio de LSA (*link state advertisement*)
 - Apenas para o roteador designado
 - “all-designated-routers” – 224.0.0.6
 - Se o LSA é novo, o roteador designado
 - Re-envia em todas suas interfaces
 - E na rede broadcast
 - “all-OSPF-routers” – 224.0.0.5

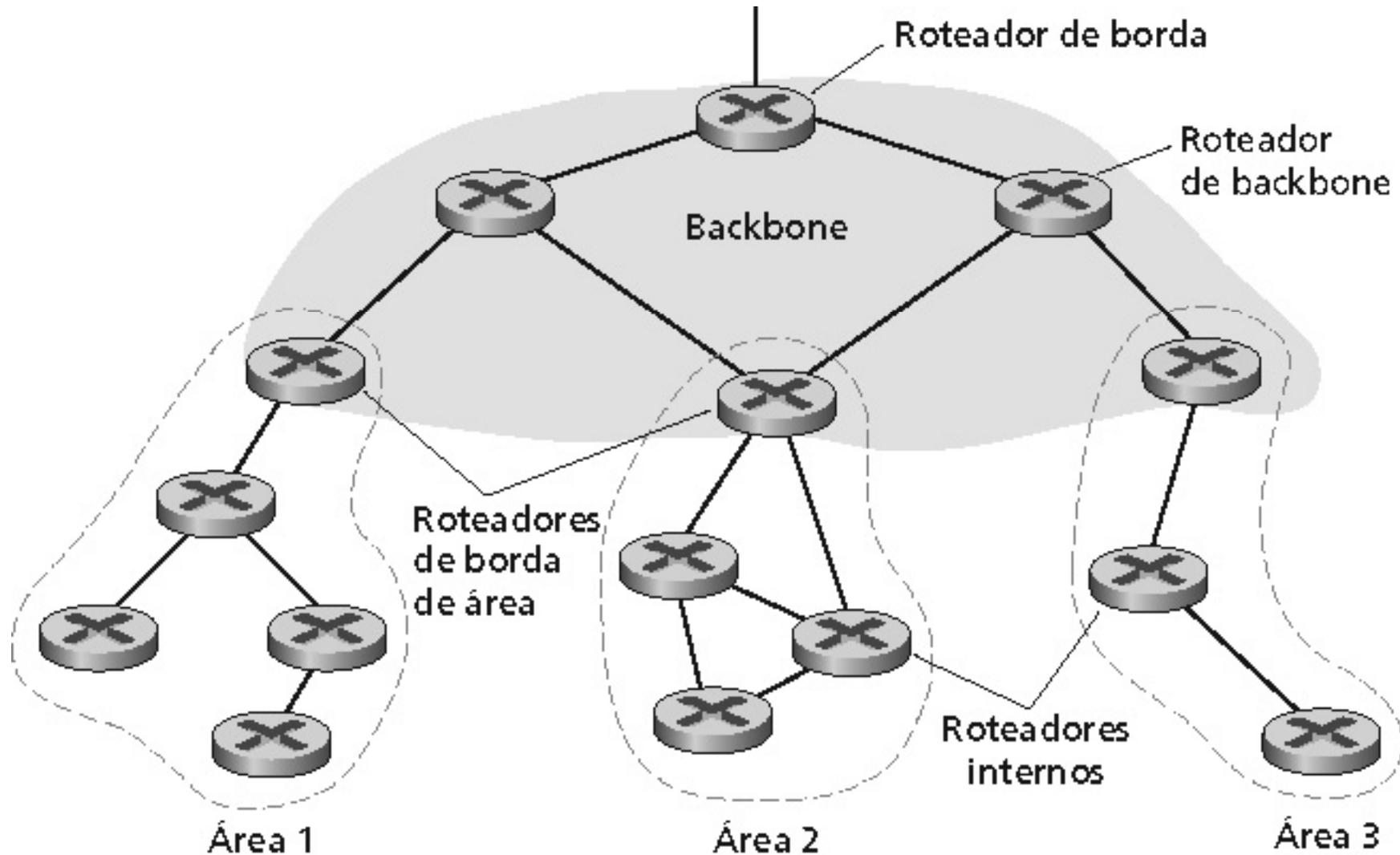
- Problema
 - Roteador designado – ponto de falha
- Roteador designado de *backup*
 - Roteadores mantêm adjacências com o designado e o backup
 - O backup escuta os anúncios, silenciosamente
 - Falha do roteador designado detectada pelo protocolo *Hello*

Áreas Múltiplas

- Roteamento hierárquico
 - Divisão da rede em diferentes partes conectadas por um *backbone*
 - OSPF: áreas conectadas através da área *backbone*
- Cada área se comporta como uma rede independente
 - Base de dados de estados do enlace
 - Inundação termina nas fronteiras da área
 - Roteadores calculam rotas dentro da área

Custo proporcional ao tamanho da área, não da rede!

Áreas Múltiplas



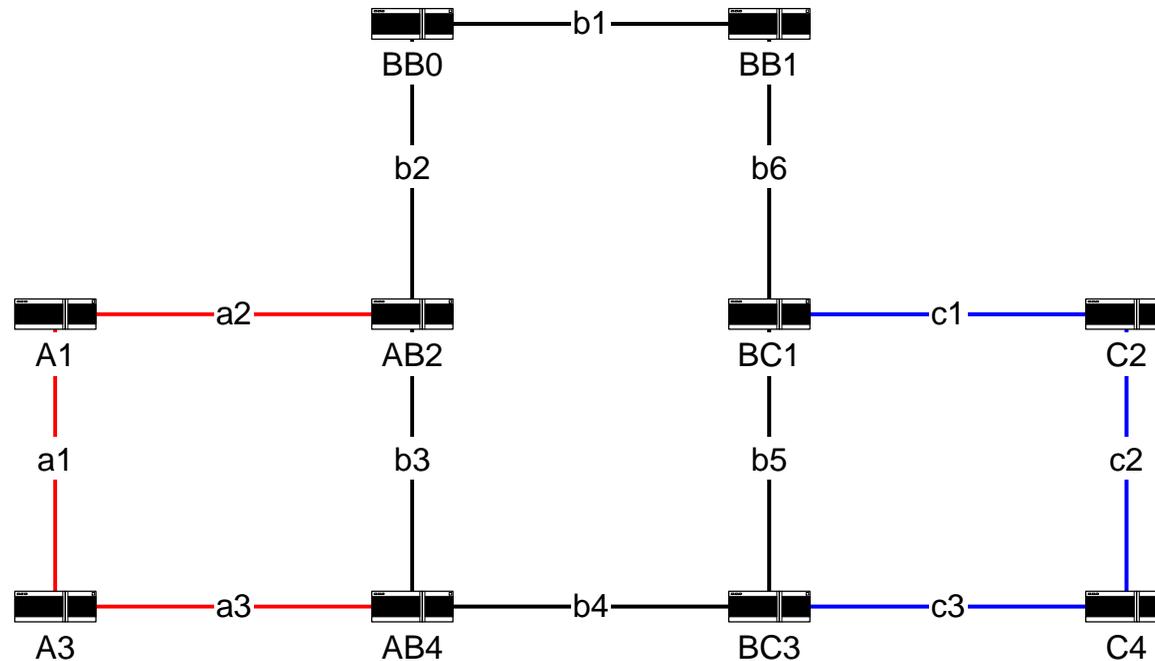
Áreas Múltiplas

- Hierarquia de dois níveis: área local e *backbone*
 - Anúncios de LS disseminados apenas na mesma área
 - Cada nó possui topologia detalhada da área
 - Apenas sabe o caminho mais curto para redes em outras áreas
- Roteador de borda de área
 - Resume distâncias às redes na sua própria área, anuncia a outros roteadores de fronteira de área
- Roteadores de *backbone*
 - Realizam roteamento OSPF limitado ao *backbone*
- Roteadores de borda
 - Ligação entre ASes

Conexão das Áreas Múltiplas

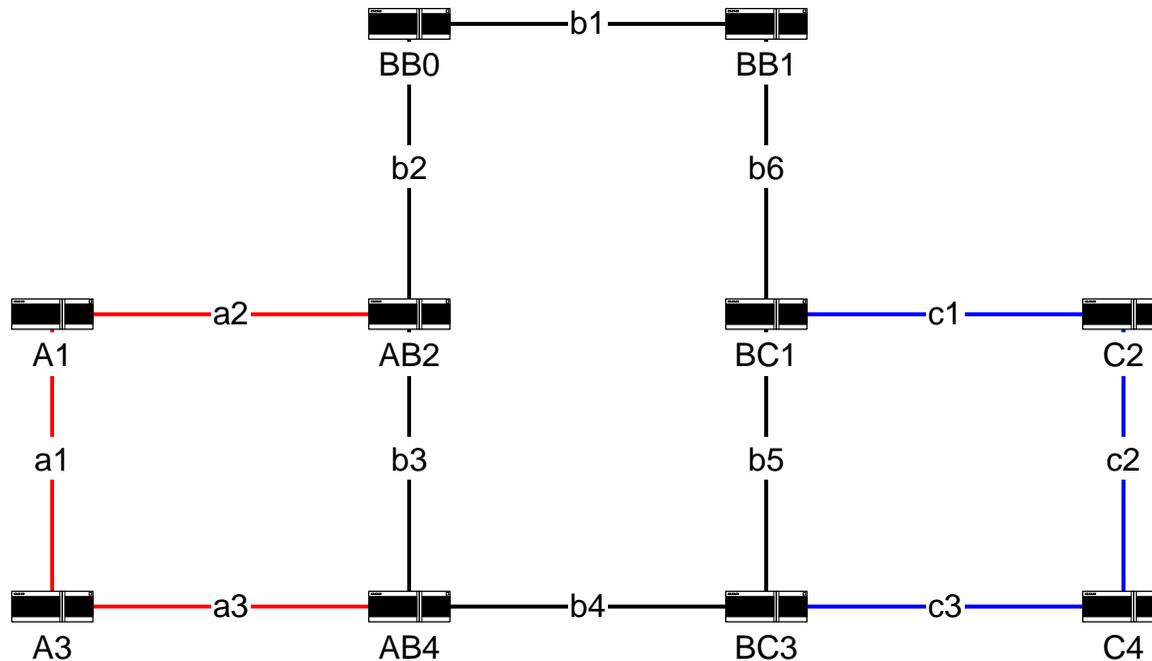
- Roteadores de borda de área (*area-border routers*)
 - Pertencem a várias áreas
 - Tipicamente, área de baixo nível + área *backbone*
 - Mantêm várias bases de dados, uma para cada área à qual pertencem
 - Cada área deve ter pelo menos um roteador de borda de área, conectando-a ao *backbone*
 - Anunciam “enlaces de sumário” (*summary links*)
 - Descrevem rotas internas
- Roteadores de borda (*border routers*)
 - Anunciam “enlaces externos” (*external links*)
 - Descrevem rotas externas

Exemplo de Áreas Múltiplas



- **Área A:** roteadores A1, AB2, AB4 e A3; enlaces a1, a2 e a3
- **Área C:** roteadores C2, C4, BC3 e BC1; enlaces c1, c2 e c3
- **Área B:** (*backbone*): roteadores de borda BB0 e BB1, roteadores de borda de área AB2, AB4, BC3 e BC1; enlaces b1, b2, b3, b4, b5 e b6

Exemplo de Áreas Múltiplas

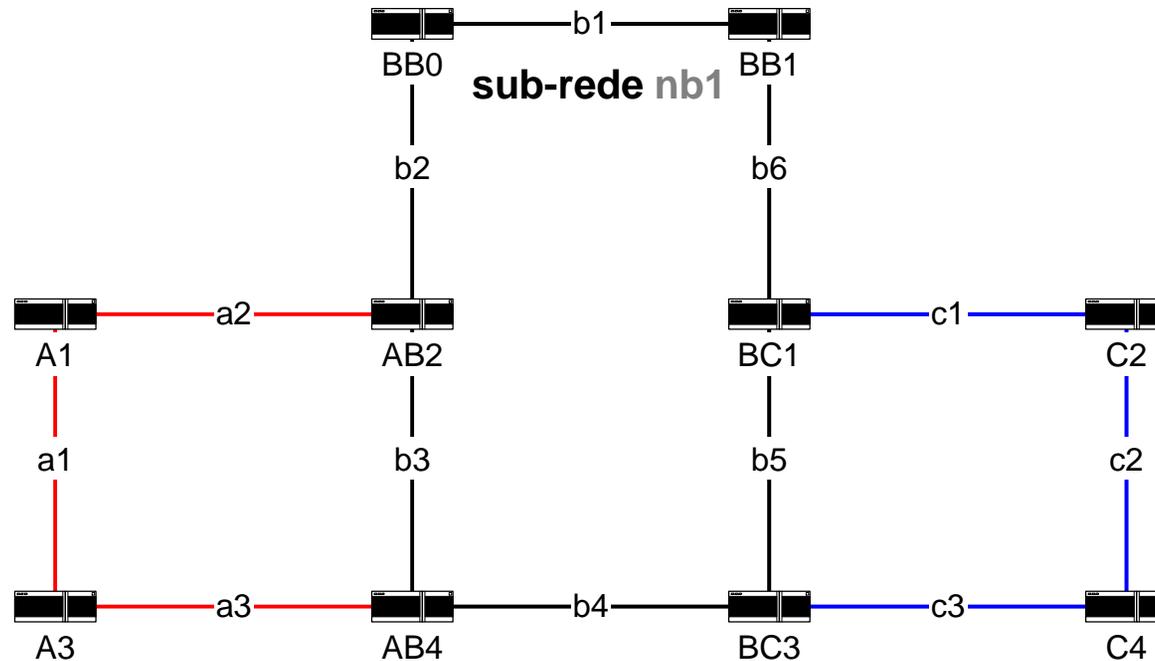


- Base de dados **Área A**:
 - Estados de enlace para a1, a2 e a3, enviados por A1, AB2, AB4 e A3
 - Registros **sumário** emitidos por AB2 e AB4, descrevendo redes e sub-redes da área *backbone* e da **área C**
 - Registros **externos** emitidos por BB0 e BB1, e retransmitidos por AB2 e AB4

Registros Sumário

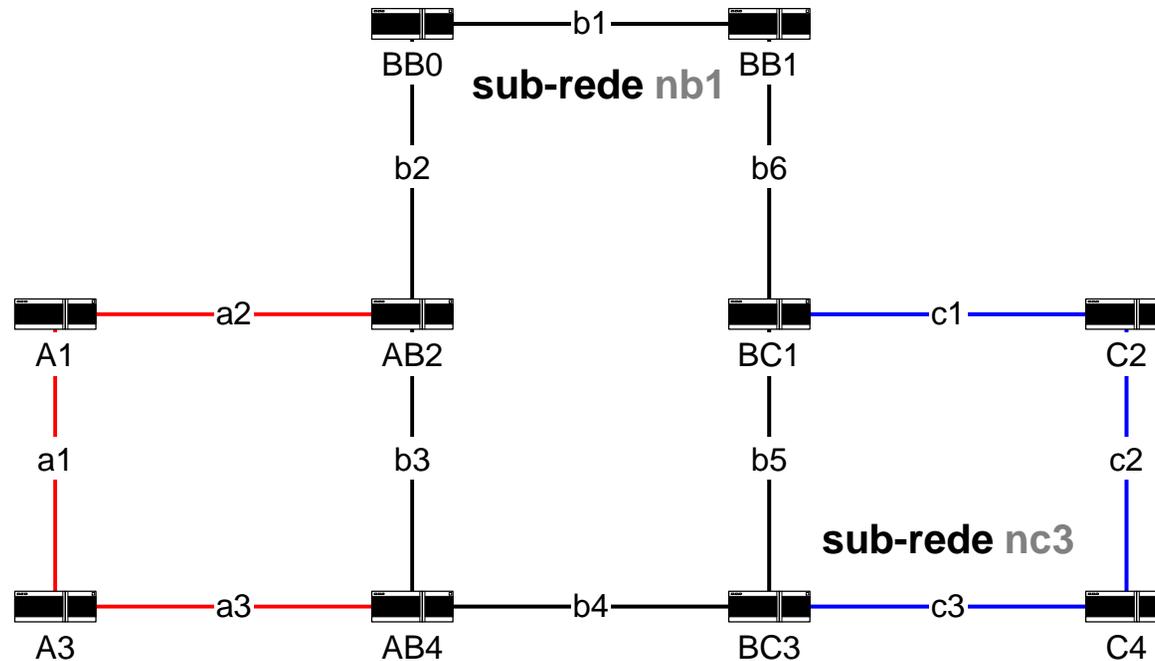
- Representam “enlaces” entre um roteador de borda de área e uma rede na área *backbone*, ou outra área
- Métrica igual à distância entre o roteador e a rede
- Propagação semelhante a vetores distância, mas sem riscos de loops, devido à hierarquia estrita
 - Áreas são conectadas apenas através da área *backbone*
- Exemplo
 - **b1**: rede Ethernet identificada pelo número de sub-rede nb1

Exemplo de Áreas Múltiplas



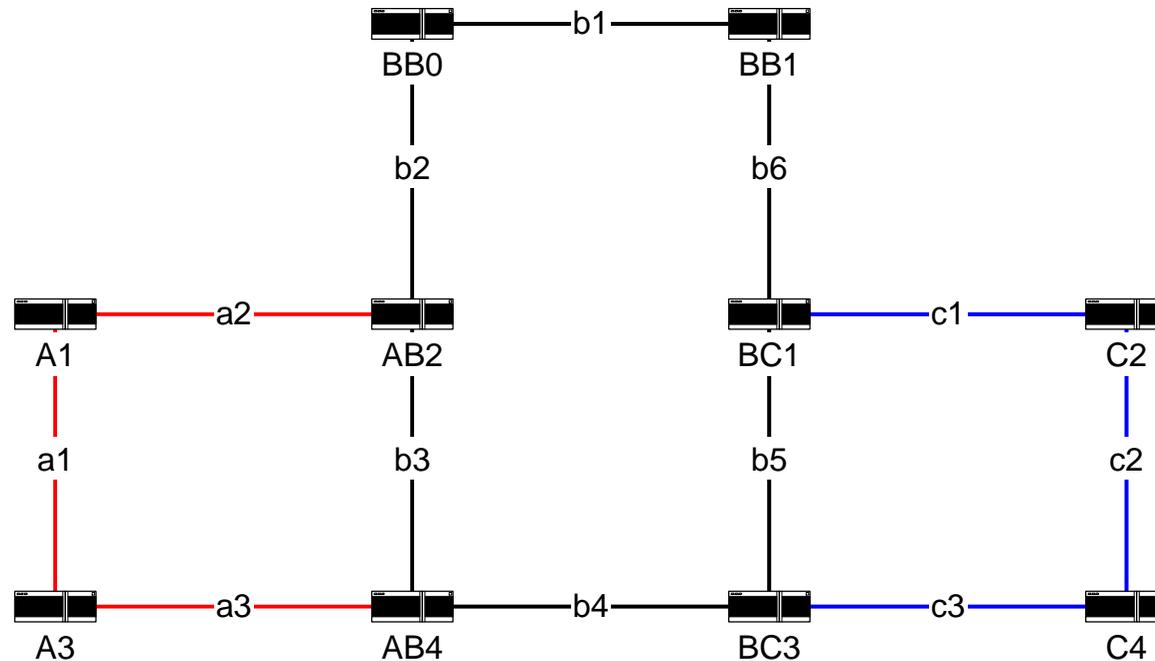
- AB2 anuncia enlace sumário para nb1 com métrica = $b1 + b2$
- AB4 anuncia enlace sumário para nb1 com métrica = $b1 + b2 + b3$

Exemplo de Áreas Múltiplas



- BC3 anuncia enlace sumário para nc3 com métrica = $c3$
- BC1 anuncia enlace sumário para nc3 com métrica = $c1 + c2 + c3$
- AB4 e AB2 calculam o caminho mais curto para nc3 através de BC3
- AB4 anuncia na área A enlace sumário para nc3 com métrica = $c3 + b4$
- AB2 anuncia na área A enlace sumário para nc3 com m. = $c3 + b4 + b3$

Rotas Externas



- Registros externos, gerados por BB0 e BB1, são copiados inalterados na base de dados da **área A** (e **C**)
- Registros de sumário enviados por AB2 e AB4 dizem como chegar a cada roteador na área de *backbone*
 - Portanto, rotas externas passando por BB0 e BB1 podem ser calculadas precisamente

- Geralmente, os registros mais numerosos na base de dados de estados de enlace correspondem às rotas externas
 - ~60.000 em 1999
- Algumas *pequenas* áreas são conectadas por apenas um roteador de borda de área ao *backbone*
- OSPFv2
 - Área *stub* (*stub area*)
 - Todas as rotas externas são resumidas por uma rota *default*

Base de Dados de Estados do Enlace

- Composta de *registros de estado do enlace*
 - *Link State (LS) records*
- Tipos de LS
 - Enlace de roteador (tipo 1)
 - Enlace de rede (tipo 2)
 - Enlace sumário (de rede IP) (tipo 3)
 - Enlace sumário (para um roteador de borda) (tipo 4)
 - Enlace externo (tipo 5)

Cabeçalho de Estado do Enlace

- Comum a todos tipos de registro de estado do enlace

0	1	2	3
0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9	0 1
LS age	options	LS type	
Link State ID			
Advertising router			
LS sequence number			
LS checksum		length	

- Advertising router – um dos endereços IP do roteador (identificador OSPF)
- Age – tempo em segundos desde o primeiro anúncio deste LS
- Option
 - E – enlace externo (utilizado pelo prot. Hello)
 - T – indica se o roteador suporta roteamento por TOS

Cabeçalho de Estado do Enlace

- Link State ID – identificação do enlace
 - Geralmente, endereço IP (depende do tipo de enlace)
 - (link state ID, advertising router, LS type) devem identificar unicamente o registro
- Checksum – calculado como no cabeçalho IP
 - Sobre cabeçalho + conteúdo
- Length – comprimento total do registro
- LS sequence number
 - $N = 2^{31}$ (usados negativos)
 - Números variam entre $1 - N$ e $N - 2$; $-N$ e $N - 1$ não utilizados

Cabeçalho de Estado do Enlace

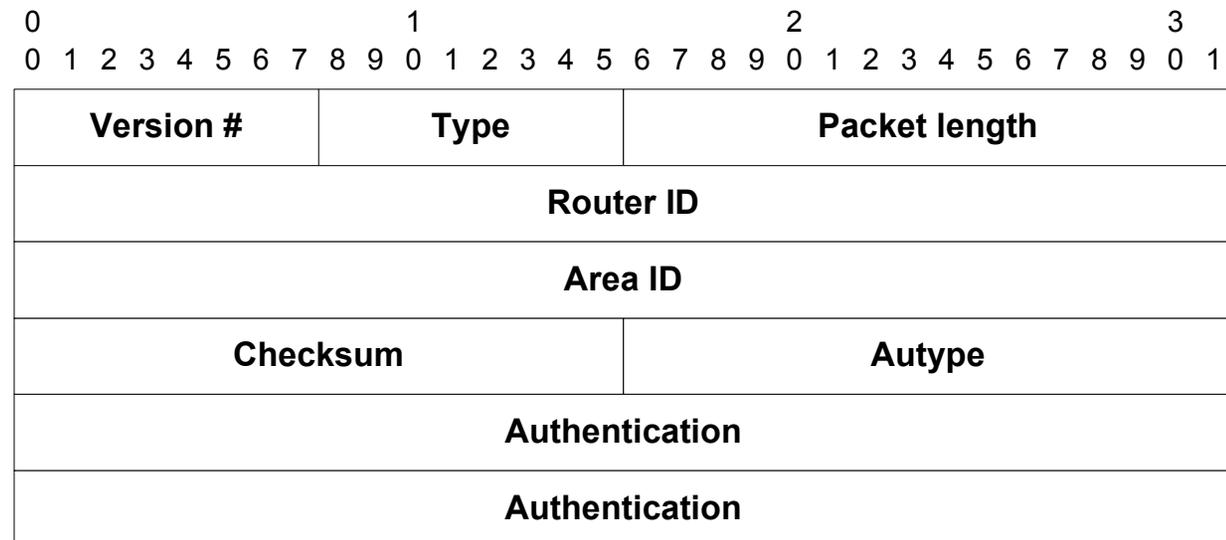
- Roteador é ligado
 - Começa de 1 – N e incrementa o número
 - Em N - 2, próximo número = 0 (N – 1 não usado)
 - Números agora em seqüência cíclica na parte positiva
- Comparação
 - Um número negativo – comparação direta
 - Ambos positivo ou nulo – comparação cíclica
 - **a** e **b**, com **a** menor que **b**
 - Se **(b - a) < (N - 1) / 2**
 - **b** mais recente que **a**
 - Senão, **a** mais recente que **b**
 - Problema se um roteador reinicia várias vezes em seqüência

Protocolos dentro do OSPF

- OSPF
 - Executado **diretamente** sobre o IP
 - “Sem” protocolo de transporte
 - Tipo de protocolo = 89
 - Sub-protocolos
 - **Protocolo Hello**
 - Alcançabilidade de vizinhos
 - Eleição do roteador designado
 - **Protocolo de Troca (*Exchange*)**
 - Sincronização inicial da base de dados de estados do enlace
 - **Protocolo de Inundação (*Flooding*)**
 - Atualização da base de dados de estados do enlace
 - Sincronização da idade dos registros

Cabeçalho Comum dos Pacotes OSPF

- Version # - 2
- Type – tipo de pacote OSPF
- Length – comprimento
- Router ID
 - identificador OSPF
- Area ID
 - 0 – área backbone
 - Utiliza-se na prática o número de rede como Area ID
- Checksum
 - Como no IP, sobre todo o pacote *menos* os 8 bytes de autenticação
- Autype
 - 0 – sem autenticação
 - 1 – autenticação simples
 - 2 – autenticação criptográfica

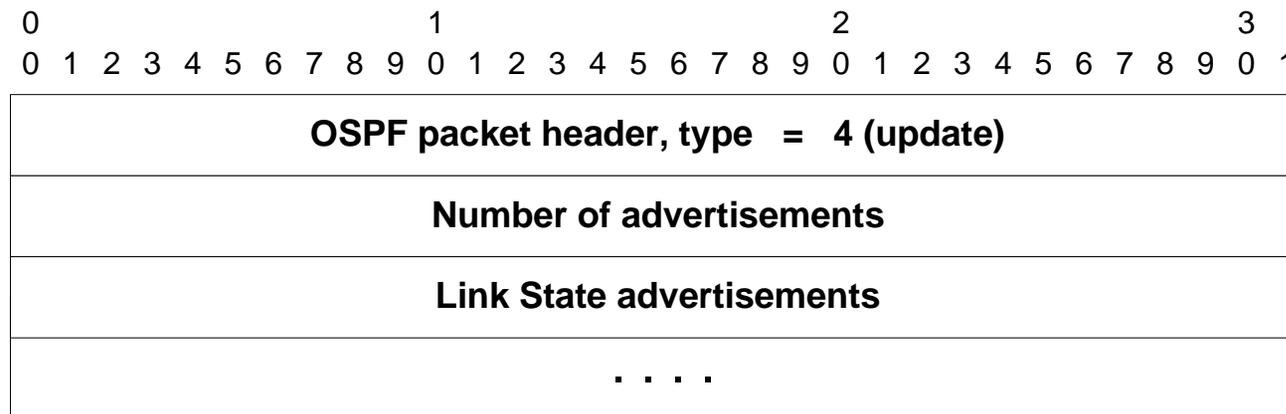


Protocolo de Troca (*Exchange*)

- Objetivo
 - Sincronização inicial da base de dados
 - Após estabelecimento de conectividade
 - Com vizinho no enlace ponto-a-ponto
 - Com roteador designado senão
 - Após a sincronização inicial, protocolo de inundação
- Protocolo Assimétrico
 - Passo 1: escolha do mestre e escravo
 - Passo 2: troca da descrição das bases de dados
 - Passo 3: listagem dos registros a serem pedidos
- Utiliza pacotes de descrição da base de dados OSPF

Protocolo de Inundação

- Atualizações de Estado do Enlace



- Cabeçalho OSPF + núm. de anúncios + anúncios LS
- Recepção
 - Se “nova” atualização, reenvio nas outras interfaces de saída
 - Em todo caso, envio de pacote de reconhecimento

Controle da Idade dos Registros

- Remoção de LSs desatualizados deve ser sincronizada
- Mecanismo de envelhecimento (*aging*)
 - Idade = 0 no envio do registro
 - Idade++ a cada reenvio
 - Idade++ a cada segundo, daí por diante
 - Se Idade = IdadeMax (1 hora)
 - Registro não utilizado para cálculo de rotas
 - Remoção condicionada ao aviso aos vizinhos

Controle da Idade dos Registros

- Recepção de anúncio duplicado
 - Se idade recebida = IdadeMax
 - Aceito (e será eventualmente apagado) e Re-enviado
 - Se pequena diferença (maioria dos casos)
 - Ignorado e Não re-enviado
 - (conseqüência do envio por diferentes caminhos)
 - Grande diferença (> 15min., casos raros)
 - Registro mais recente é guardado
 - Pode acontecer com um roteador que rebootou e utilizou o mesmo número de seqüência
- Registros têm a idade re-iniciada periodicamente (ao serem re-enviados)
 - ~30 min., pelo menos

Aula 18

Camada de Rede

Roteamento intradomínio: estado do enlace

Igor Monteiro Moraes
Redes de Computadores