

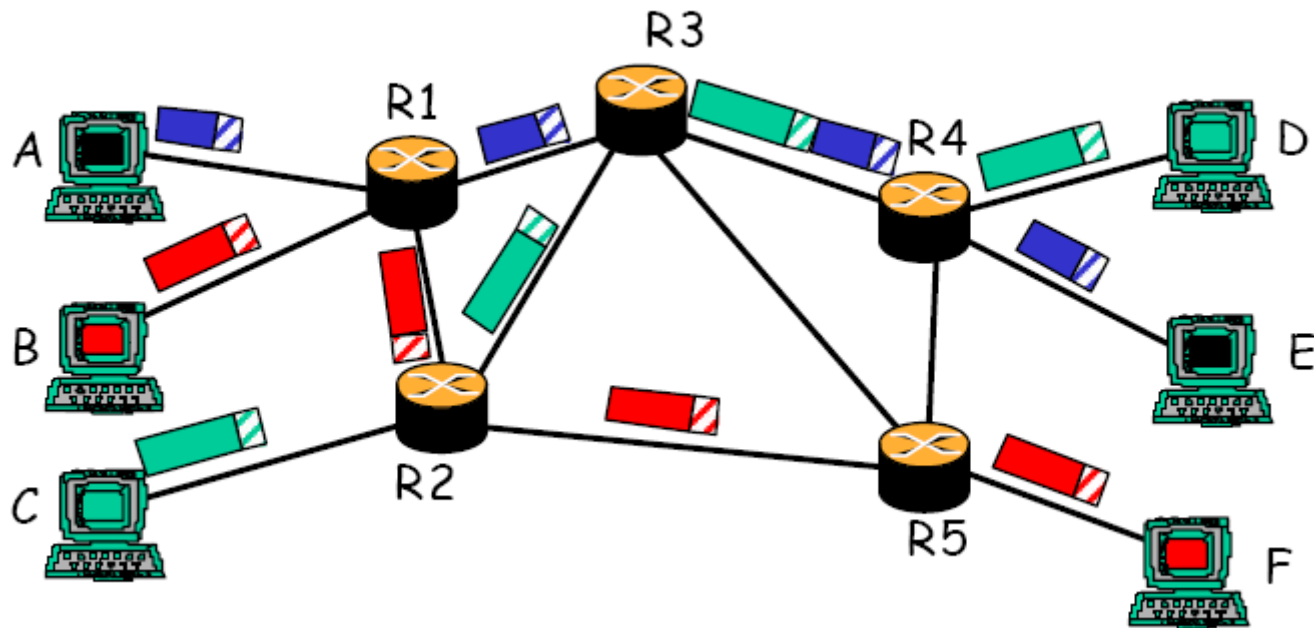
A Camada de Rede (cont.)

Links desta videoaula - P1: <https://youtu.be/r8cGJwBK4so>
P2: <https://youtu.be/SrgM2udDX3k>

Referências:

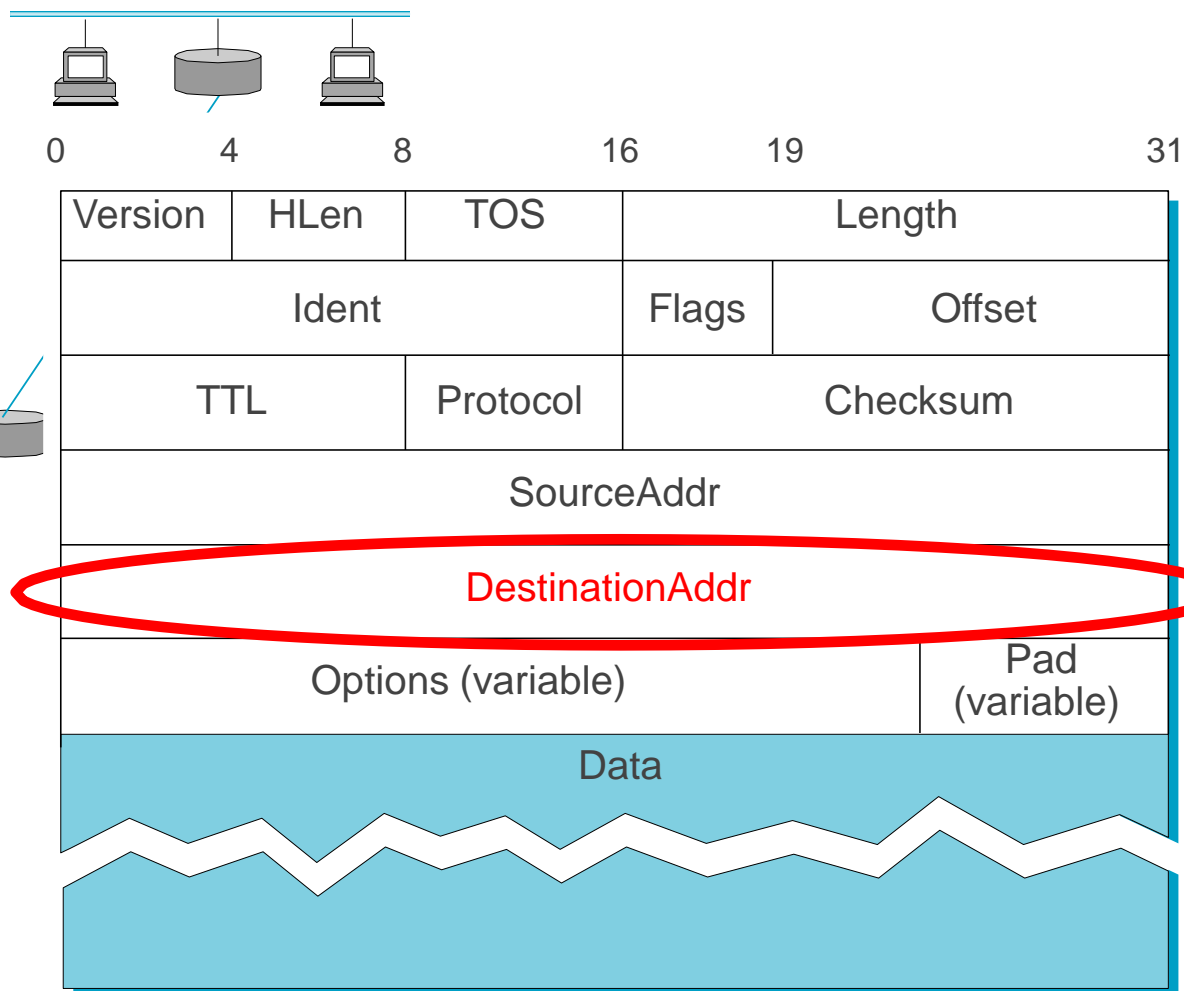
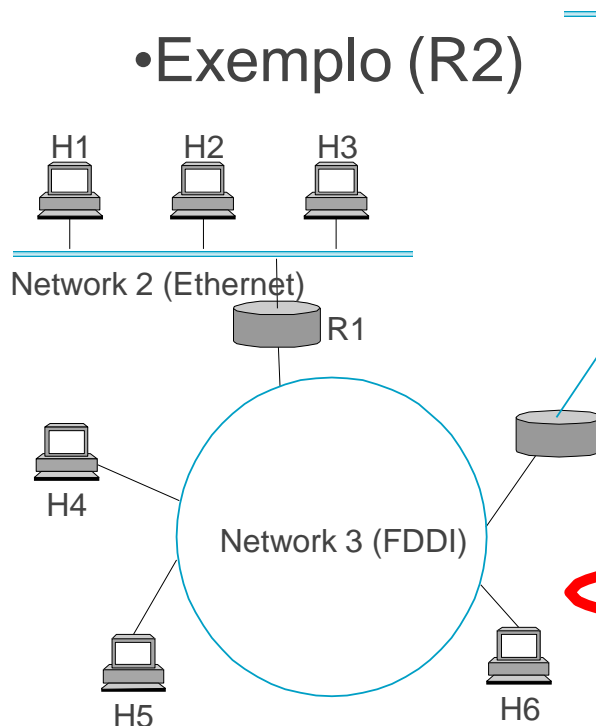
- Tanenbaum, A.; Wetherall, D., Redes de Computadores, 5ª edição, Pearson, 2011 - Capítulo 5
- Kurose, J.; Ross, K., Redes de Computadores e a Internet. Pearson, 2010 - Capítulo 4

O que é roteamento?

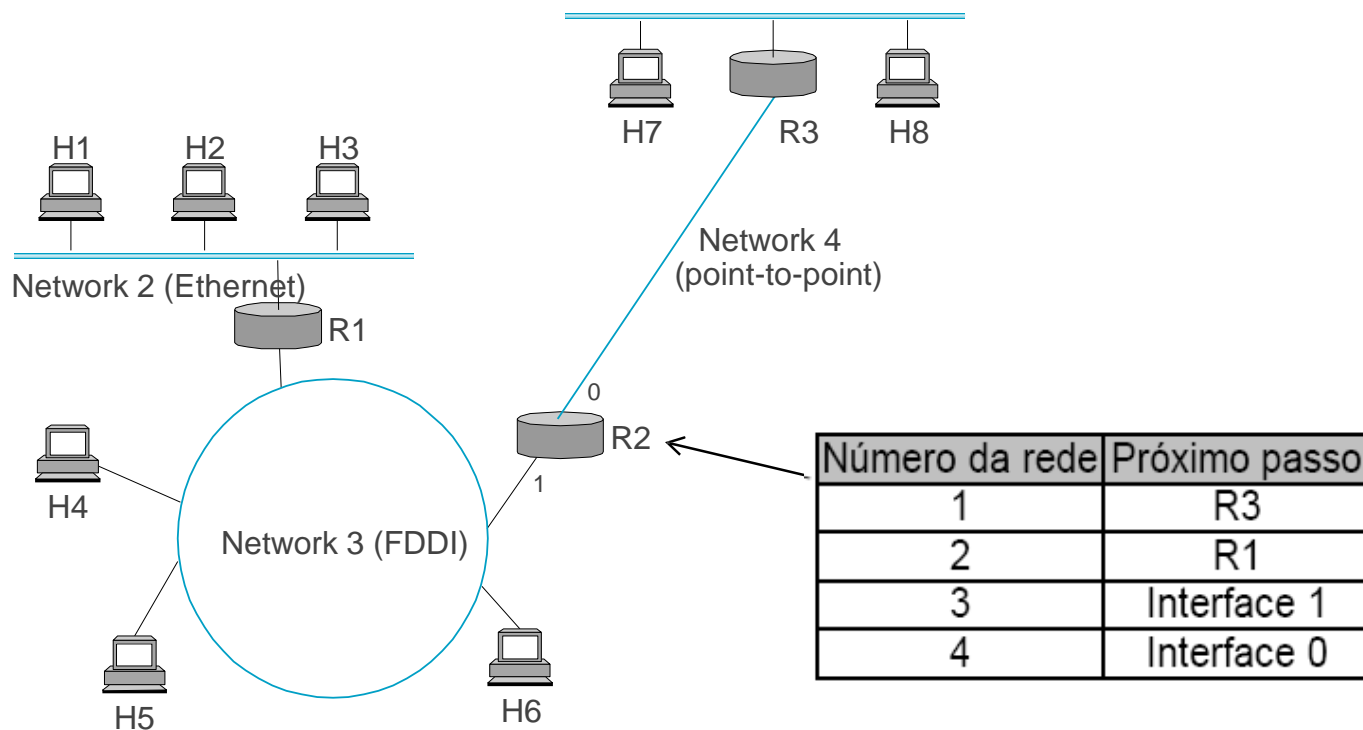


O que é roteamento?

•Exemplo (R2)



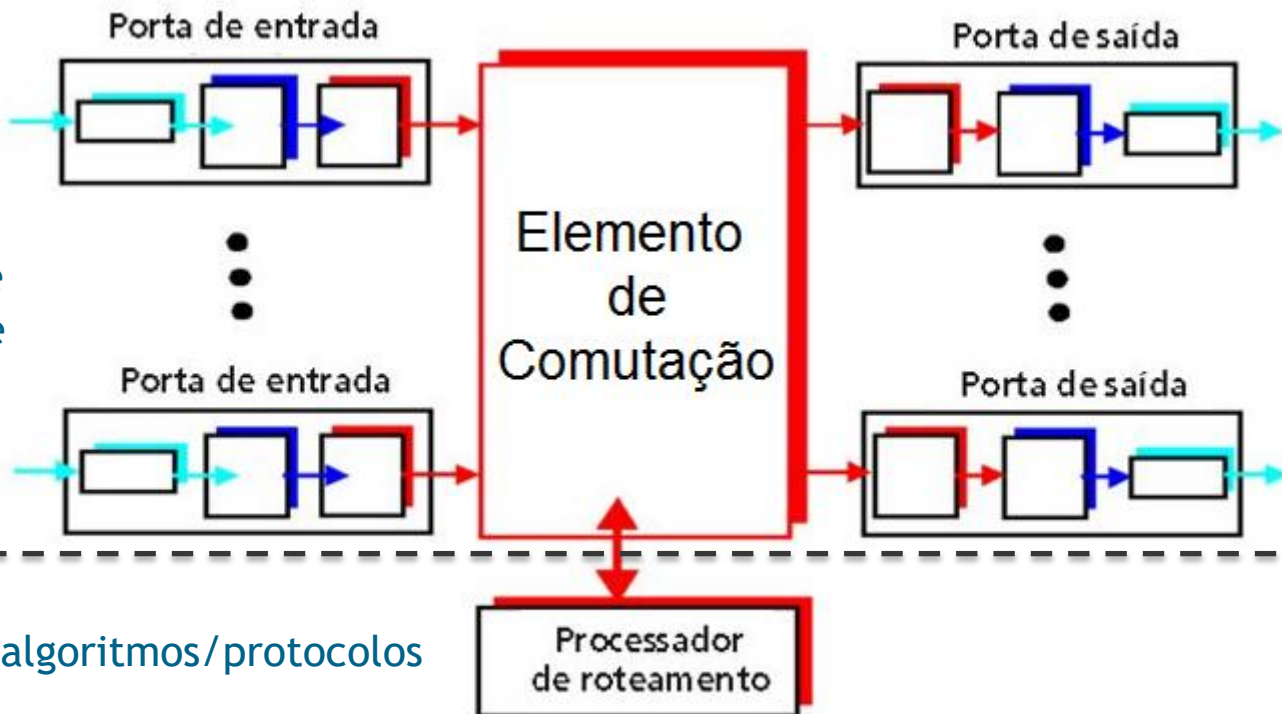
O que é roteamento?



Arquitetura Básica do Roteador IP

Duas funções-chave do roteador:

- **Dados:** Comutar os datagramas do link de entrada para o link de saída



- **Controle:** Executar algoritmos/protocolos (RIP, OSPF, BGP)

- **Portas de entrada:** diversas funções
 - Funções da camada física, enlace e rede (repasse e roteamento).
 - Ocorre a escolha da porta de saída, baseada em uma tabela de repasse.
- **Elemento de comutação:**
 - Conecta as portas de entrada às portas de saída;
 - Crossbar (interno no roteador)
- **Portas de saída:**
 - Armazena os pacotes que foram repassados e transmite-os ao enlace de saída.
- **Processador de roteamento:**
 - Executa os protocolos de roteamento;
 - Mantém as tabelas de roteamento e informações de estado do enlace;
 - Funções de gerenciamento de rede.

- Operação de repasse executada em uma escala de nanossegundos!
- **Analogia:** carros entrando e saindo de um pedágio:
 - Suposição: antes do carro entrar no pedágio, o veículo para em uma estação de entrada e indica seu destino final.
 - Atendente examina o destino final e determina a saída do pedágio que leva a esse destino.
 - Carro entra no pedágio (pode estar cheio) e segue pela saída indicada.
- **Pista de entrada e cabine de entrada** → porta de entrada
- **Pedágio** → elemento de comutação
- **Pista de saída** → porta de saída
- **Engarrafamentos poderão ocorrer** → E se a maioria dos carros que entram quiserem sair do pedágio na mesma pista de saída?

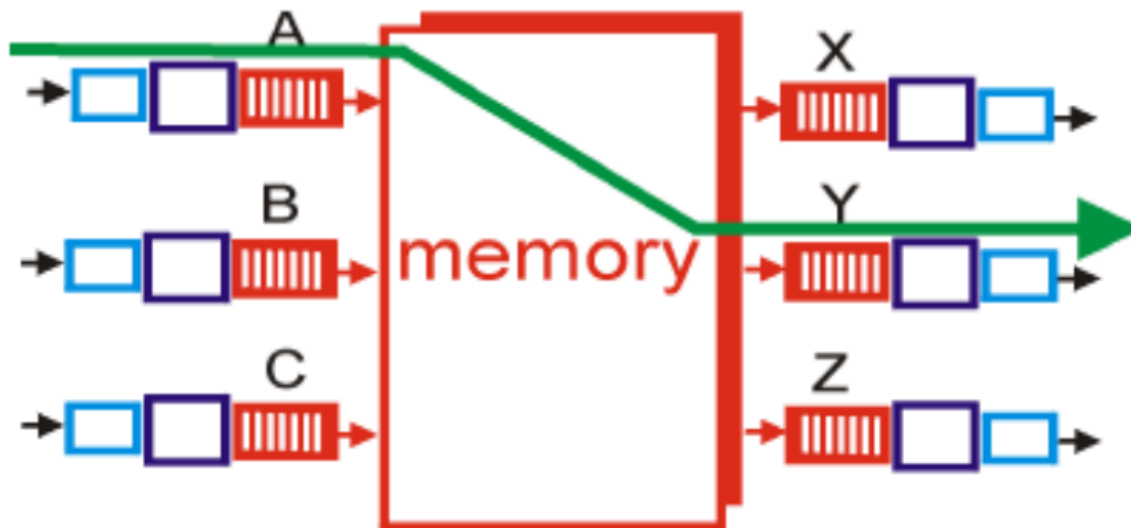
- **É o coração do roteador;**
 - É assim que os pacotes são comutados de uma porta de entrada para uma porta de saída.
- **Pode ser feito de várias formas:**
 - Comutação por memória
 - Comutação por Barramento
 - Comutação por uma rede de interconexão

Elemento de comutação



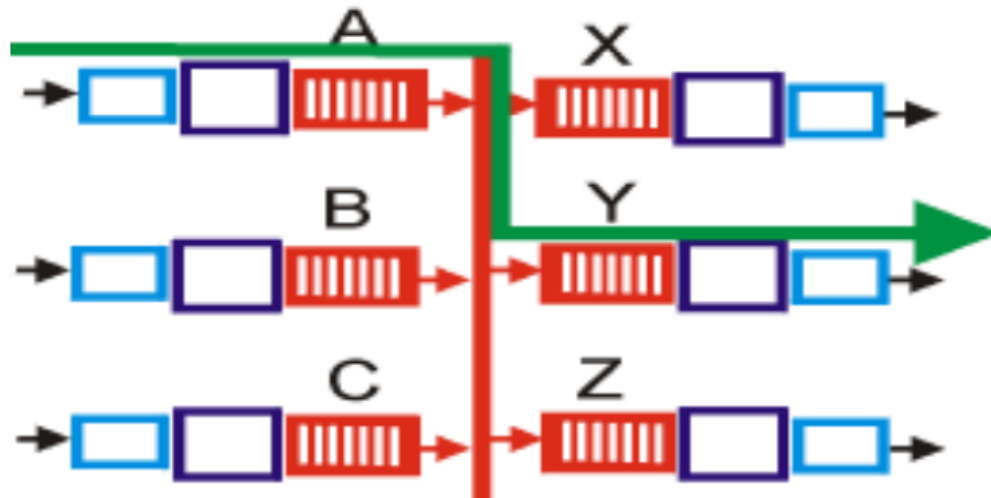
Comutação por memória

- Modelo mais simples de comutação.
- Primeiros roteadores eram computadores tradicionais e a comutação entre as portas E/S era realizada sob o controle direto da CPU.
- As portas de entrada e saída funcionam como dispositivos tradicionais de E/S de um sistema operacional tradicional.



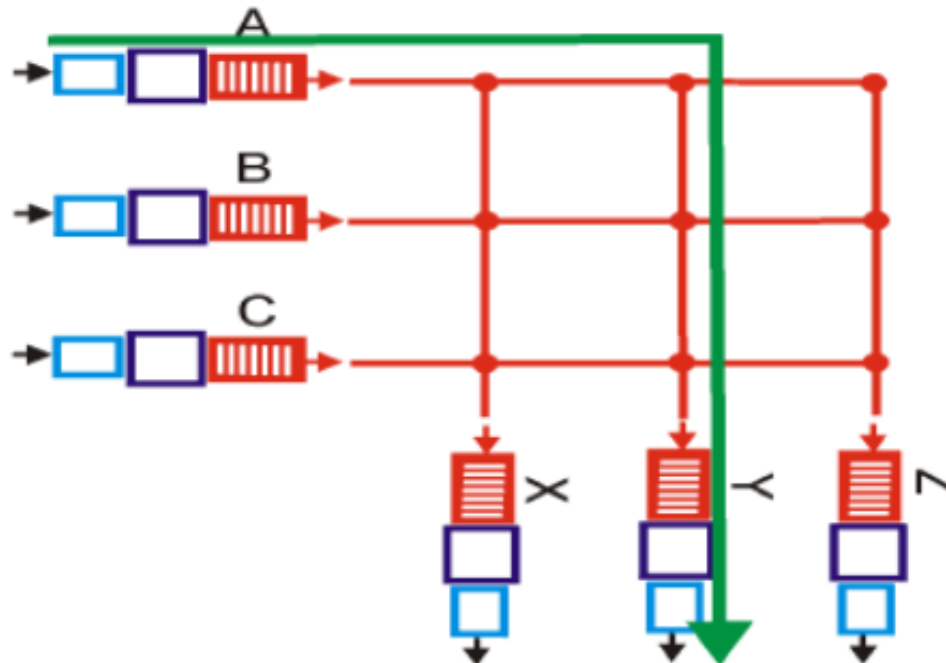
Comutação por um Barramento

- Portas de entrada transferem um pacote direto para a porta de saída por um barramento compartilhado sem a intervenção do processador.
- Barramento é compartilhado → somente um pacote por vez pode ser transferido por meio do barramento.
- BW de comutação (roteador) fica limitada à velocidade do barramento.
- A comutação por barramento muitas vezes é suficiente para roteadores que operam em redes pequenas.



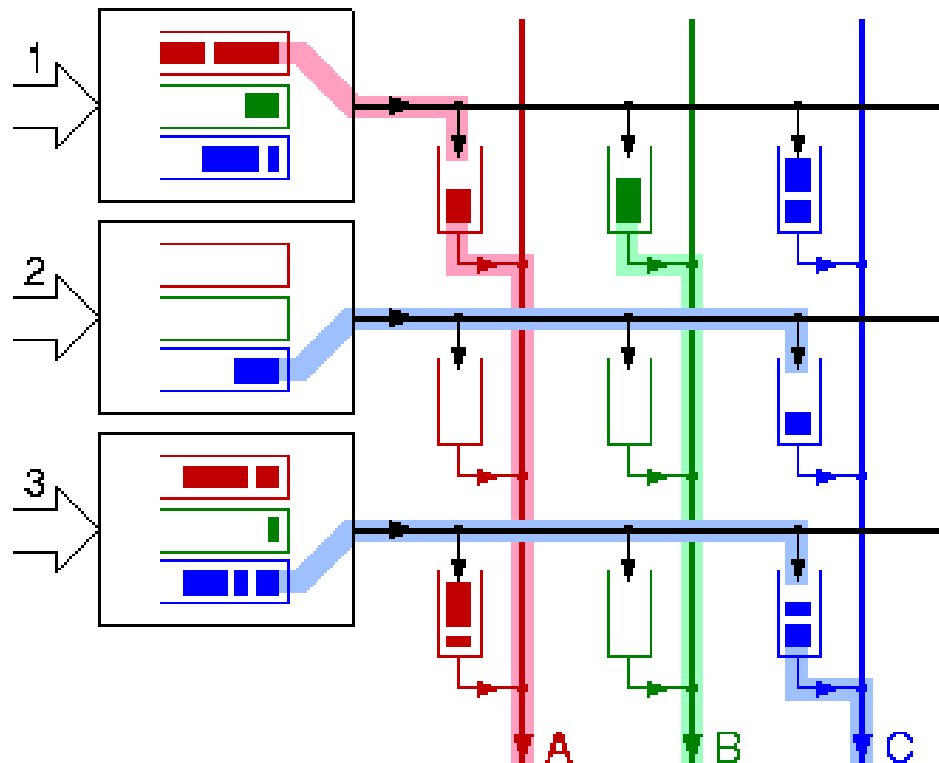
Comutação por uma rede de Interconexão (Crossbar)

- Basicamente uma rede dentro do roteador.
- Desenvolvida para vencer a limitação da largura de banda da comutação por barramento.
- É uma rede de interconexão que consistem em 2^n barramentos, conectando n portas de entrada a n portas de saída.



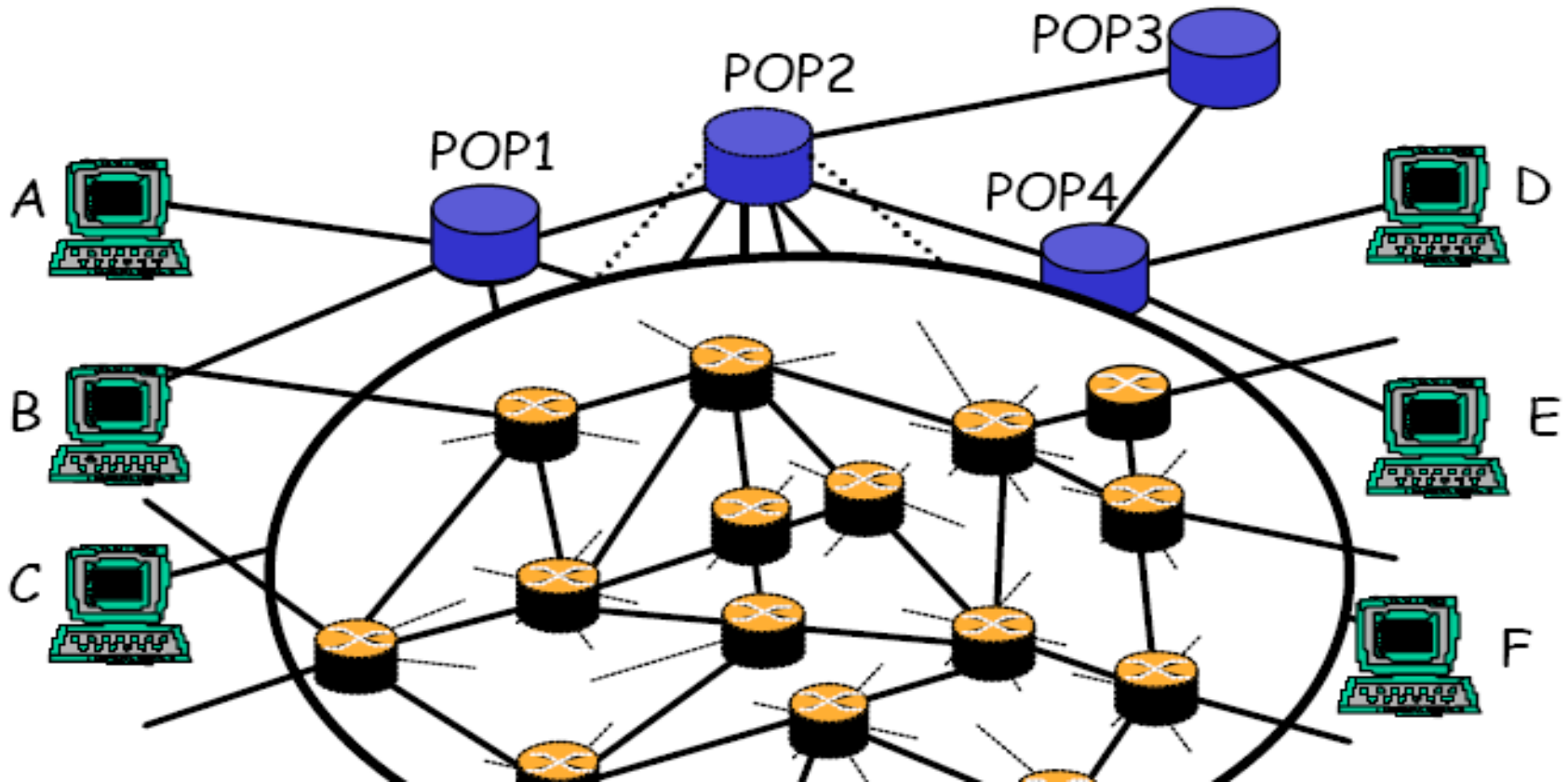
Formação de filas

- Quando o tamanho da fila supera a capacidade de armazenamento do buffer ocorre a perda de pacotes.
- O local real da perda do pacote dependerá da carga do tráfego, da velocidade relativa do elemento de comutação e da taxa da linha.



- **Formas de evitar estouro do buffer:**
 - Gerenciamento ativo de fila (Active Queue Management – AQM).
 - Detecção antecipada aleatória (Random Early Detection - RED).

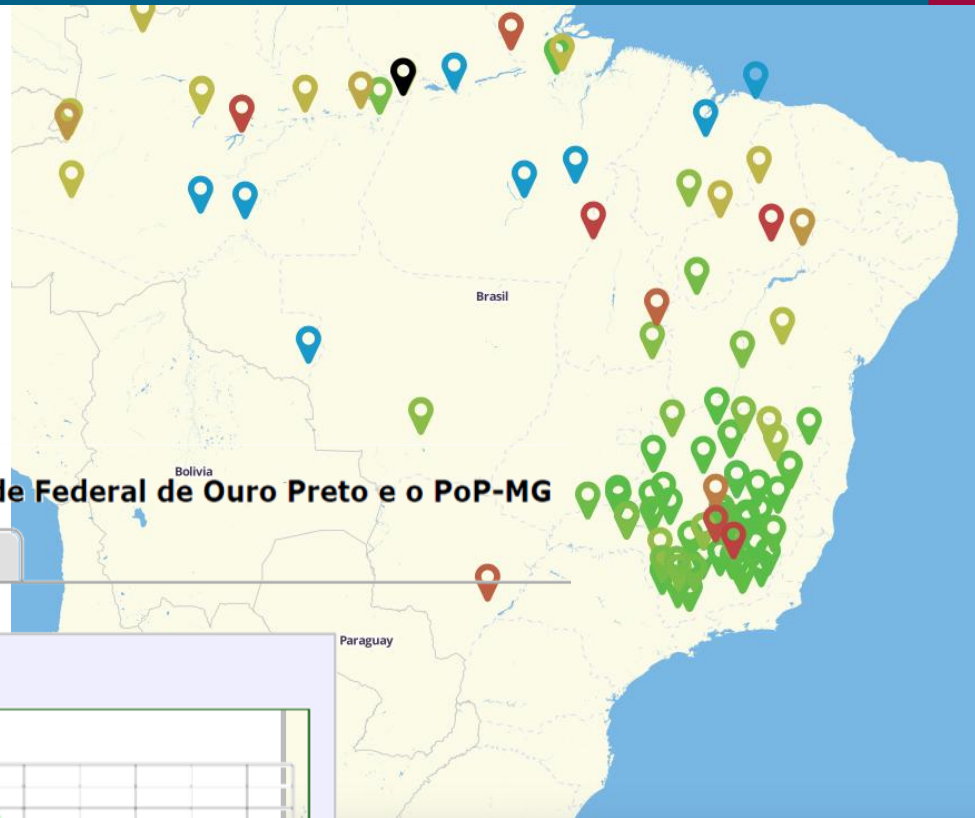
Point of Presents (POPs)



- POP - ponto físico de uma determinada rede de comunicação em um determinado local ou cidade.
- Pontos em que um [ISP](#) se conecta a outros ISPs → [roteadores](#) na rede do ISP com os quais roteadores em outros ISPs podem se conectar.

POP-MG -- RNP

Região ou Estado	Instituições	Campi
Minas Gerais	23	108
Região Norte	12	27
Região Nordeste	5	9
Região Centro-Oeste	5	5
Totais	45	149



Estatísticas da conexão entre UFOP - Universidade Federal de Ouro Preto e o PoP-MG

Tráfego

Pacotes

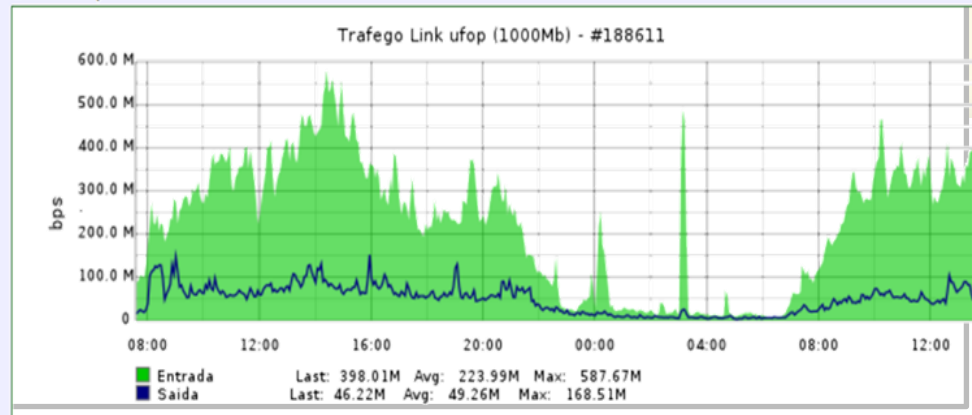
Latência

Erros

Descartes

Tráfego

Link ufop



Uso da rede do ICEA em um dia típico...



Estatísticas da conexão entre o PoP-MG e UFOP campus João Monlevade

De	16/10/2023, 06:14:56
Até	17/10/2023, 06:14:56

Selecione uma região dos gráficos para detalhar ou use duplo clique para visualizar um período maior de

Taxa Download p/ hora

41.2 Mb/s

Taxa Upload p/ hora

791.3 kb/s

Taxa Upload p/ ...

0.2%

Taxa Upload p/ ...

0.0%

i Qualidade da conexão

Boa

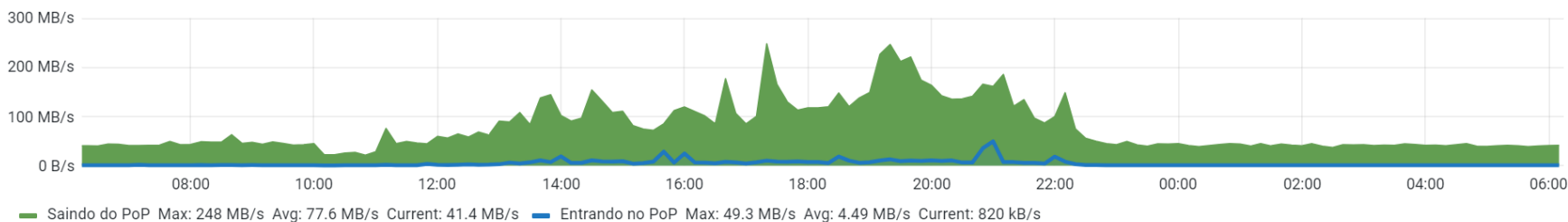
Perda de pacotes

0.6%

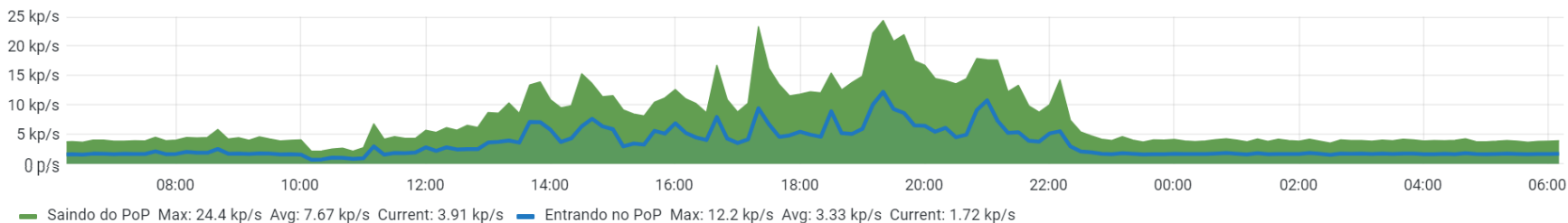
Latência ▾

2.0 ms

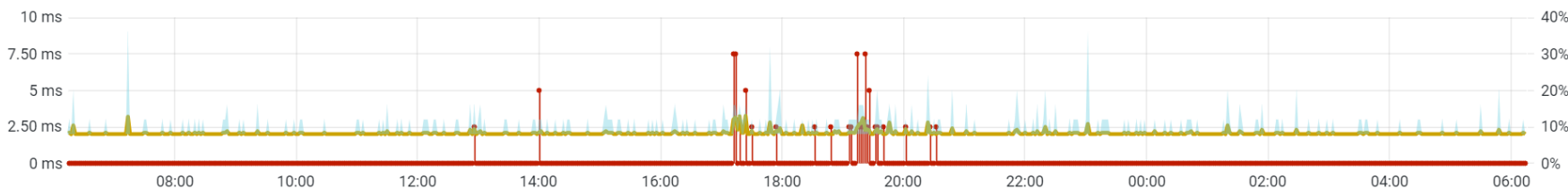
Tráfego total



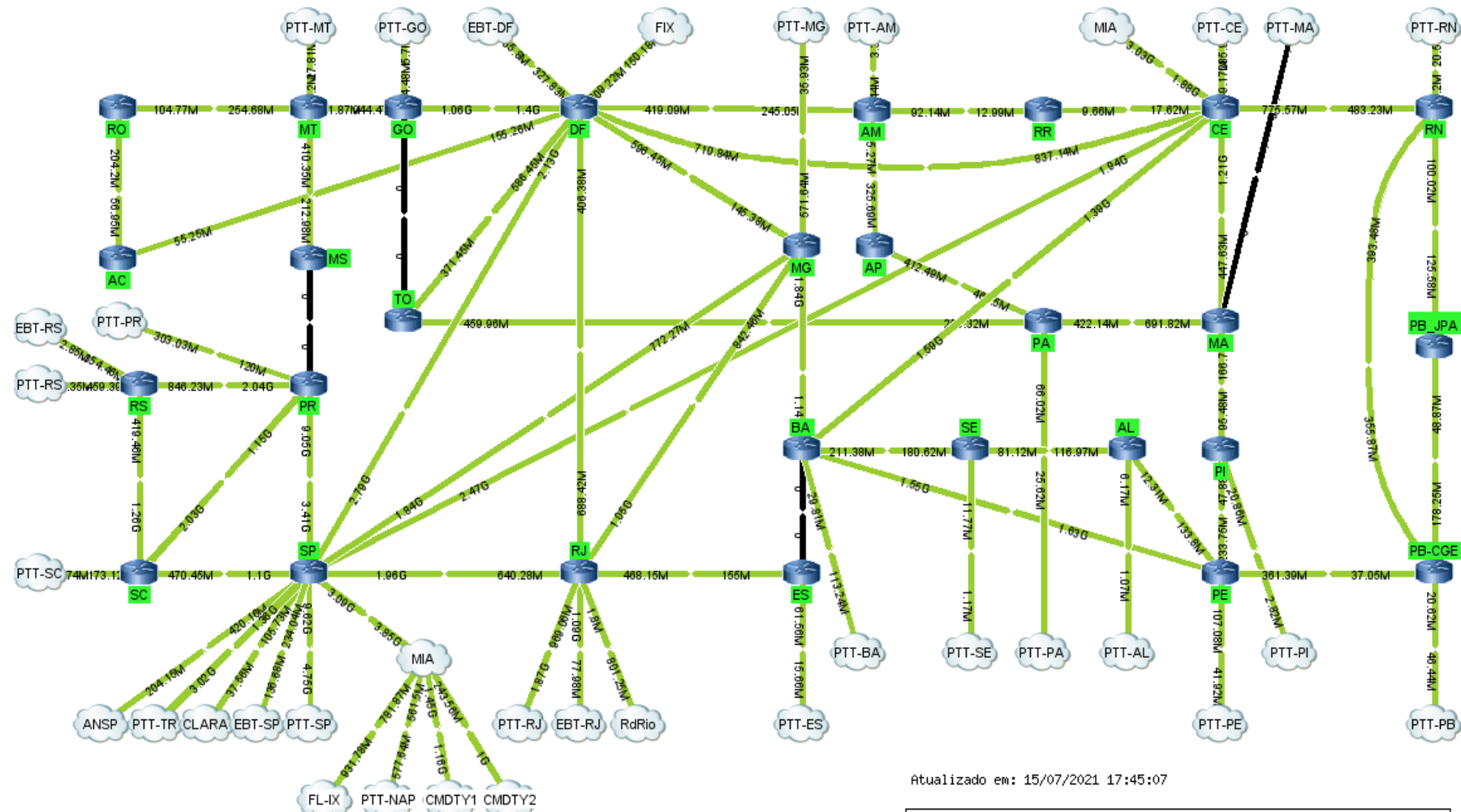
Pacotes total



Latência 395697

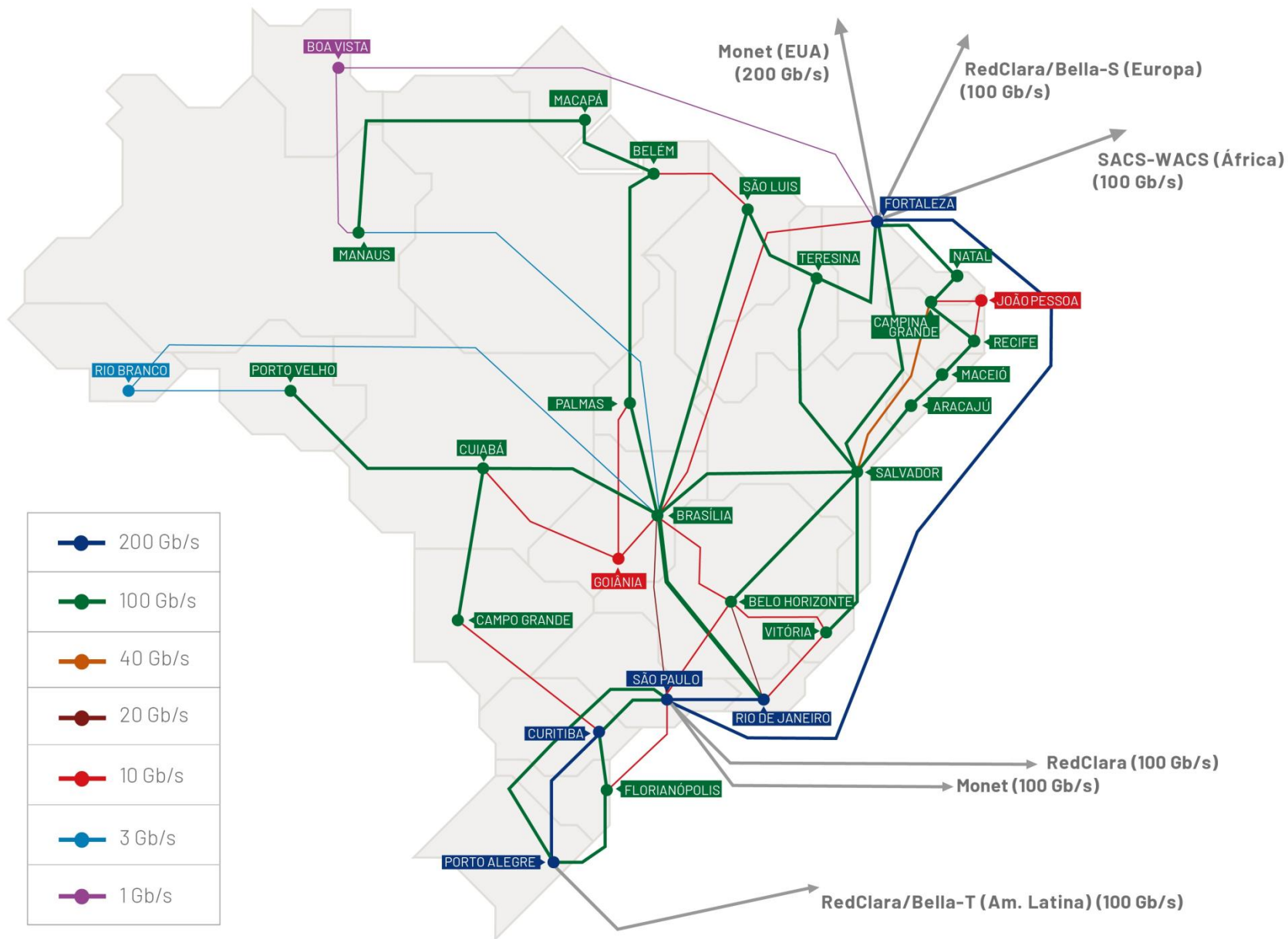


Tráfego Backbone - RNP



Atualizado em: 15/07/2021 17:45:07

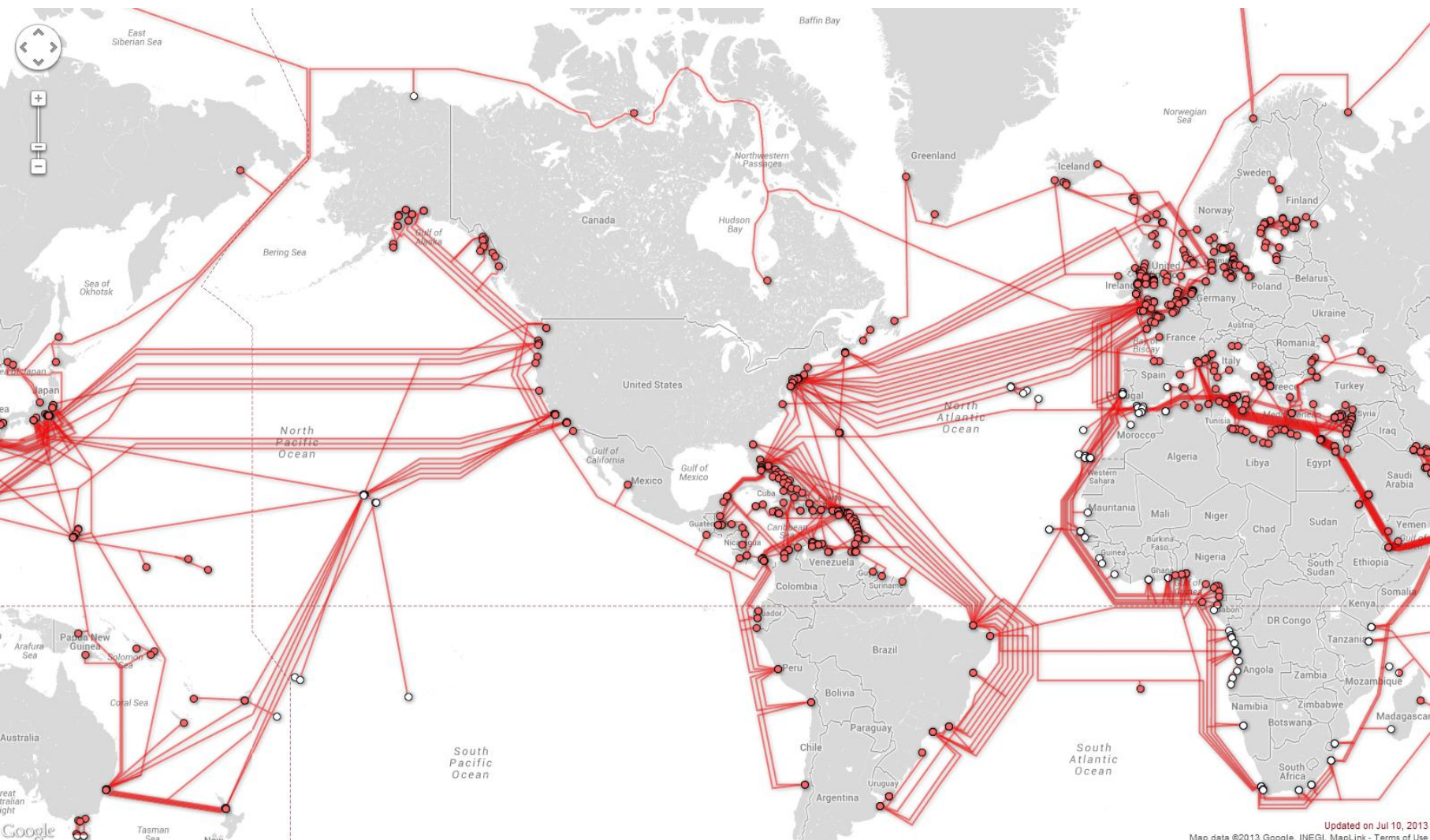




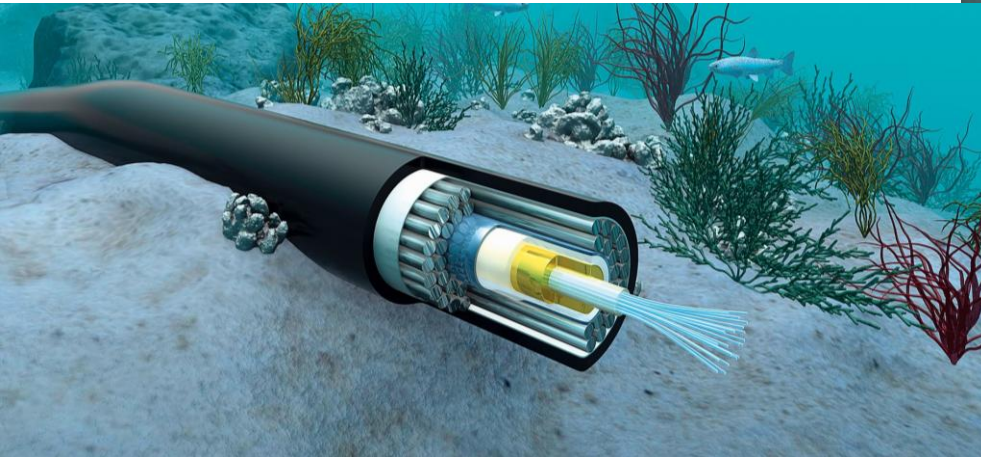
Conectividade Brasil - Mundo



Conectividade internacional



Cabos submarinos



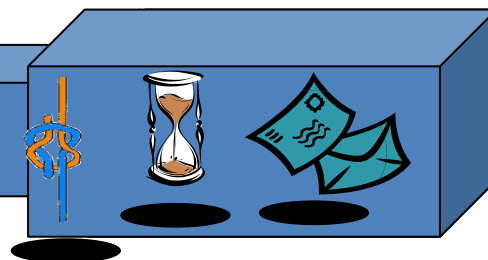
Processamento por pacote no roteador IP



Data

01000111100010101001110100011001

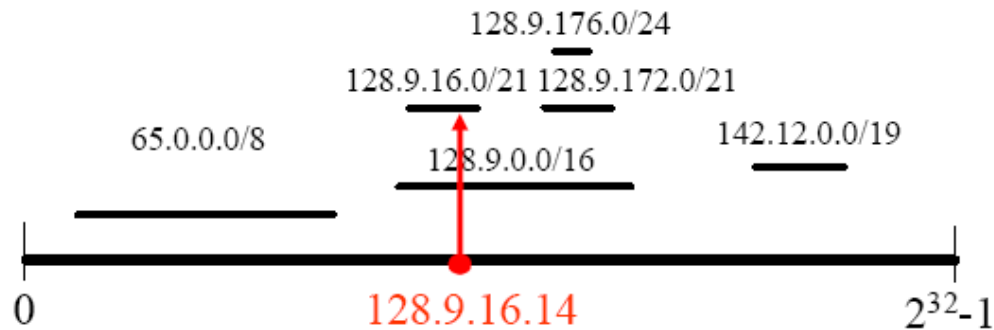
Header



1. Endereço Internet
2. Idade (age)
3. Checksum para proteger cabeçalho



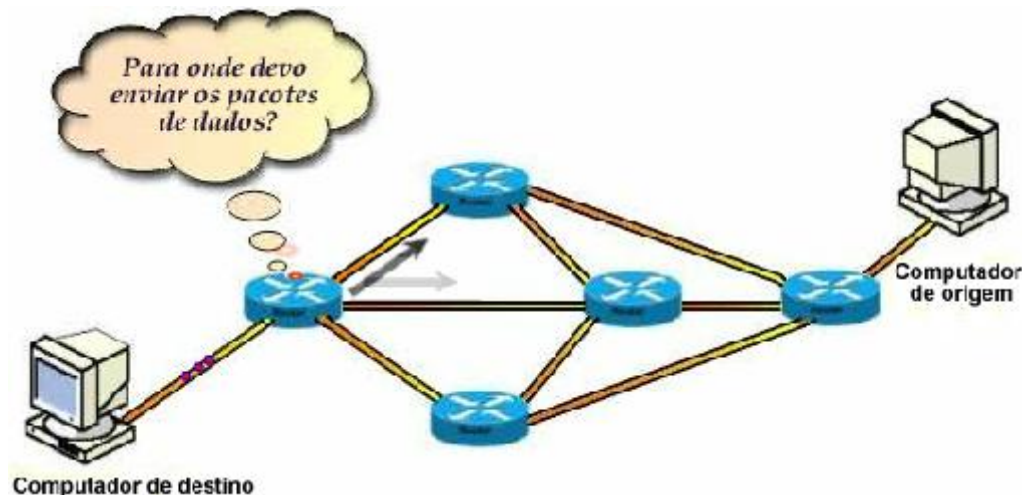
Procurar endereço



Achar o prefixo mais longo entre todos da tabela que case com o endereço de destino.

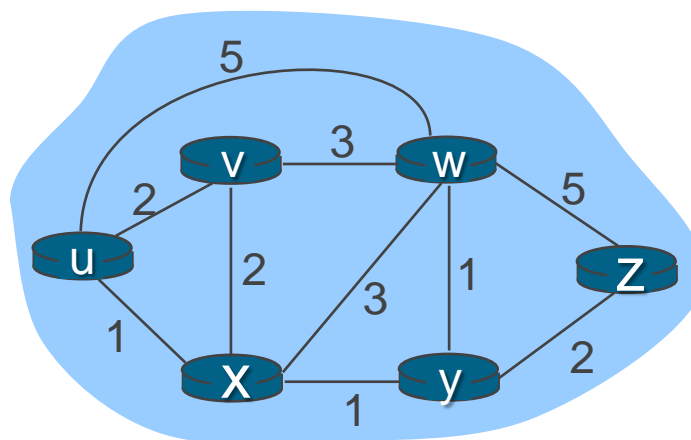
- **4. 1 Introdução**
- **4.2 Redes de circuitos virtuais e de datagramas**
- **4.3 IP: Internet Protocol**
 - formato do datagrama
 - endereçamento IPv4
 - ICMP
 - IPv6
- **4.4 Algoritmos de roteamento**
 - O que é roteamento
 - estado de enlace
 - vetor de distâncias
 - roteamento hierárquico

- **A finalidade de um algoritmo de roteamento é simples:**
 - Dado um conjunto de roteadores conectados por enlaces, um algoritmo de roteamento descobre um 'bom' caminho entre um roteador fonte e roteador destino.
- **Importante entender o conceito de grafos**
 - Grafos são usados para formular problemas de roteamento



Abstração de grafo

Um Grafo: $G = (N, E)$ é um conjunto N de nós e uma coleção E de arestas, onde cada aresta é um par de nós do conjunto N .



N = conjunto de roteadores = $\{ u, v, w, x, y, z \}$

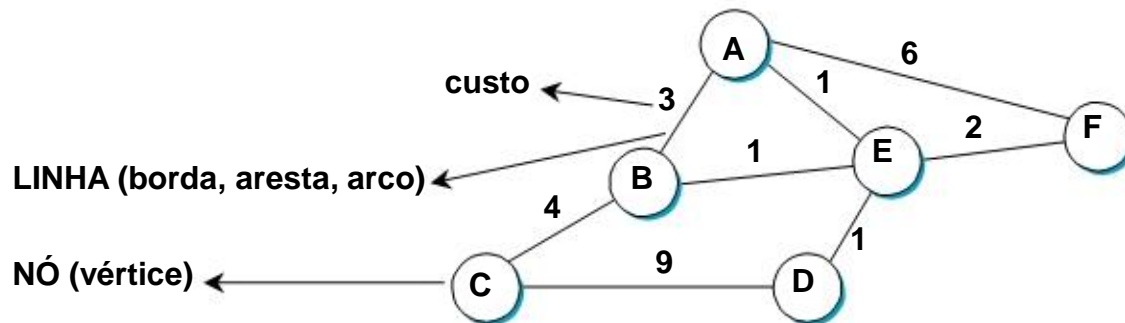
E = conjunto de enlaces = $\{ (u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z) \}$

Comentário: Abstração de grafo é útil em outros contextos de rede

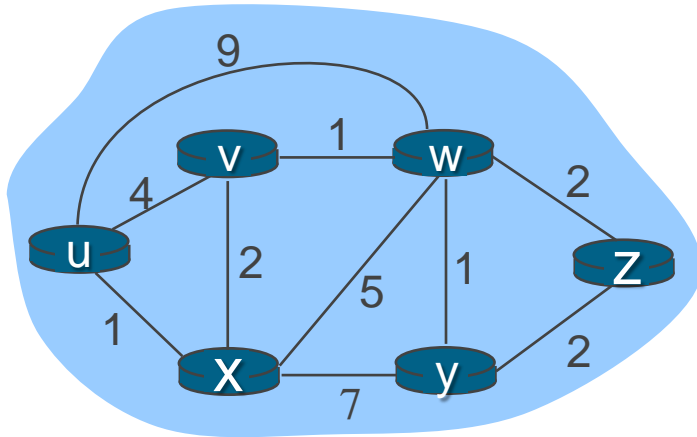
Exemplo: P2P, onde N é conj. de pares e E é conj. de conexões TCP

Abstração de grafo: custos

- As arestas de um grafo pode ser direcionada ou não.
- Arestas podem ser valoradas, onde esse valor representa seu custo, que pode refletir:
 - O tamanho físico do enlace
 - Velocidade do enlace
 - Custo monetário
 - Utilização / carga do enlace



Abstração de grafo: custos



- $c(x,x')$ = custo do enlace (x,x')

- p. e., $c(w,z) = 2$

- custo poderia ser sempre 1, ou relacionado à BW ou inversamente relacionado ao congestionamento

Custo do caminho $(x_1, x_2, x_3, \dots, x_p) = c(x_1, x_2) + c(x_2, x_3) + \dots + c(x_{p-1}, x_p)$

algoritmo de roteamento: algoritmo que encontra o caminho de menor custo

→ Informação global ou descentralizada?

global: calcula o caminho de menor custo usando conhecimento completo e global da rede.

- todos os roteadores têm topologia completa, informação de custo do enlace;
- algoritmos de “estado do enlace”

descentralizada: calculo do menor caminho é realizado de modo iterativo e distribuído.

- roteador conhece vizinhos conectados fisicamente, custos de enlace para vizinhos
- processo de computação iterativo, troca de informações com vizinhos
- algoritmos de “vetor de distância”

→ Estático ou dinâmico?

estático:

- rotas mudam lentamente com o tempo
 - Muitas vezes por intervenção humana

dinâmico:

- rotas mudam mais rapidamente
 - atualização periódica
 - em resposta a mudanças no custo do enlace

→ Algoritmo sensível à carga

- Custos de enlace variam dinamicamente para refletir o nível corrente de congestionamento no enlace subjacente.

É desejável nos algoritmos roteamento...

Simplicidade: ser eficiente sem sobrecarregar o roteador.

Robustez: Quando a rede entra em operação, ela deve permanecer assim durante muito tempo, sem que ocorram falhas no sistema.

Precisão: algoritmo de roteamento tem de calcular rotas corretas para todos os destinos.

- Não basta que o algoritmo descubra uma rota para um destino, é um dos objetivos que ele descubra a melhor rota possível.

Velocidade: O algoritmo de roteamento deve convergir rapidamente.

Estabilidade: Um algoritmo de roteamento deve ser capaz de se recuperar (criar novas rotas) em caso de falhas na rede.

Ex.: Quando ocorrem modificações na topologia da rede, as tabelas de repasse de alguns roteadores apresentarão informações erradas.

- **4. 1 Introdução**
- **4.2 Redes de circuitos virtuais e de datagramas**
- **4.3 IP: Internet Protocol**
 - formato do datagrama
 - endereçamento IPv4
 - ICMP
 - IPv6
- **4.4 Algoritmos de roteamento**
 - vetor de distâncias
 - estado de enlace
 - roteamento hierárquico
- **4.5 Roteamento na Internet**
 - RIP
 - OSPF
 - BGP

Os algoritmos de roteamento pode ser classificados como:

- **Algoritmos não-adaptativos (roteamento estático):**
 - Flooding
- **Algoritmo adaptativos (roteamento dinâmico):**
 - Distance Vector
 - Link State

Inundação (flooding)

- Algoritmo estático, no qual cada pacote de entrada é enviado para todas as linhas de saída, exceto aquela em que chegou.
- Problema: gera grande quantidade de pacotes duplicados, na verdade um número infinito.
- Solução 1: Ter um contador de hops no cabeçalho de cada pacote
 - contador decrementado em cada hop
 - pacote descartado quando hop igual a 0 (zero).
- O ideal é iniciar o hop com o comprimento desde a origem até o destino.
 - Se não souber tamanho do caminho, emissor inicializa contador com o valor referente ao pior caso (diâmetro total da sub-rede).

Inundação (flooding)

- Não é muito prático nas aplicações, mas utilizado em:
 - **aplicações militares:**
 - Ex: onde muitos enlaces / roteadores destruídos
 - robustez do algoritmo de inundação é desejável
 - sempre encontra um caminho se existir para levar o pacote até o destino.
 - **banco de dados distribuídos:**
 - é necessário atualizar todos os bancos de dados ao mesmo tempo.
 - **redes sem fio:**
 - todas as mensagens transmitidas podem ser recebidas por todas outras estações dentro do alcance de rádio.

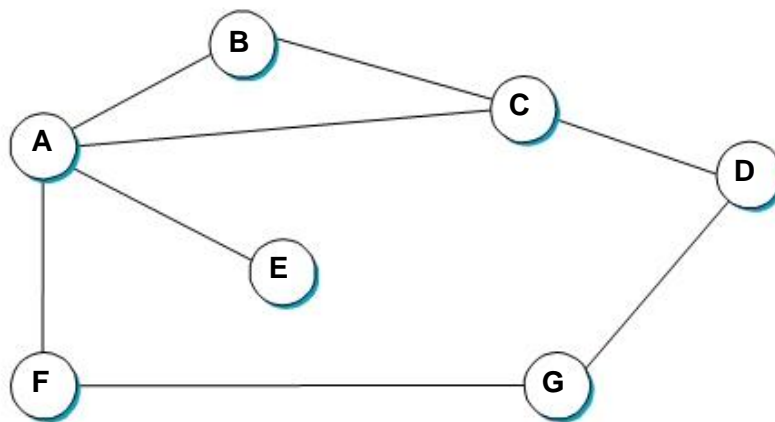
- **4. 1 Introdução**
- **4.2 Redes de circuitos virtuais e de datagramas**
- **4.3 IP: Internet Protocol**
 - formato do datagrama
 - endereçamento IPv4
 - ICMP
 - IPv6
- **4.4 Algoritmos de roteamento**
 - **vetor de distâncias**
 - estado de enlace
 - roteamento hierárquico
- **4.5 Roteamento na Internet**
 - RIP
 - OSPF
 - BGP

Vetor de Distância

- **Iterativo, assíncrono, distribuído e finito.**
- Cada nó monta um vetor com as “distâncias” (custos) até todos os outros nós e distribui esse vetor aos nós vizinhos diretos.
- Supõe que o nó conhece o custo do enlace para seus vizinhos.
 - Um enlace que não seja alcançado recebe custo infinito.
- Tabela local mantém linhas “Destino-Custo-PróximoSalto”.
- Vizinhos trocam atualizações regularmente (em períodos de vários segundos). Atualização = par “Destino-NovoCusto”.
 - Se atualização recebida implica em menor custo, atualiza linha da tabela local com “Destino-MenorCusto-NovoPróximoSalto”.
 - Se atualização traz nova entrada, acrescenta esta à tabela.

Vetor de Distância - Exemplo

Neste exemplo o custo de qualquer enlace é 1 (métrica = “saltos”).



Nó A - Tabela Rot. Inicial

Destino	Custo	PróxSalto
A	0	--
B	1	B
C	1	C
D	∞	--
E	1	E
F	1	F
G	∞	--

Vetores de Atualização vindos de:

Nó B	Nó C	Nó E	Nó F
1	1	1	1
0	1	∞	∞
1	0	∞	∞
∞	1	∞	∞
∞	∞	0	∞
∞	∞	∞	0
∞	∞	∞	1

Nó A - Tabela Rot. Final

Destino	Custo	PróxSalto
A	0	--
B	1	B
C	1	C
D	2	C
E	1	E
F	1	F
G	2	F

- Não há um nó na rede que tenha todas as informações da tabela.
 - Cada nó só conhece o conteúdo de sua própria tabela de roteamento.
 - Um algoritmo distribuído permite que todos os nós atinjam uma visão coerente da rede, sem uma autoridade centralizada.
- Duas circunstâncias diferentes que um nó envia uma atualização a seus vizinhos:
 - Atualização periódica
 - Atualização acionada

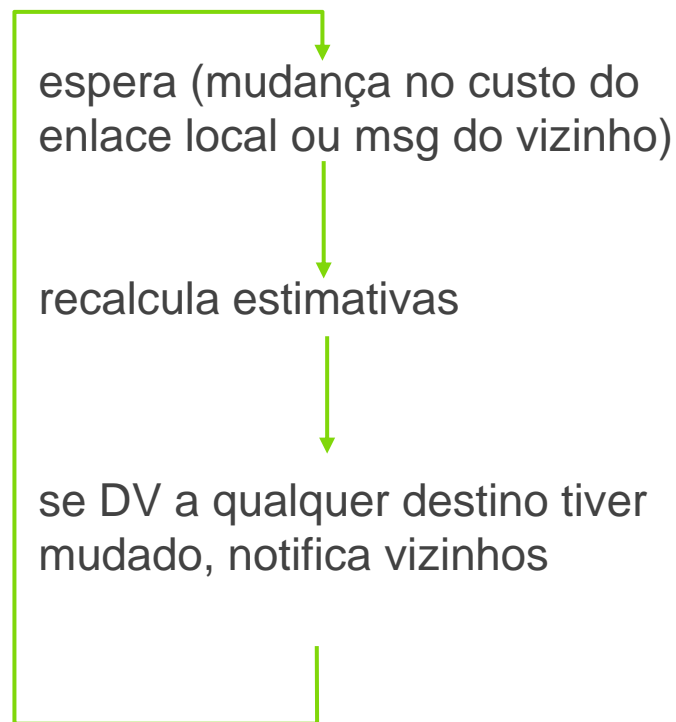
Algoritmo de vetor de distância (5)

- iterativo, assíncrono:** cada iteração local causada por:
- mudança de custo do enlace local
 - mensagem de atualização do DV do vizinho

distribuído:

- cada nó notifica os vizinhos *apenas* quando seu DV muda
 - vizinhos, então, notificam seus vizinhos, se necessário

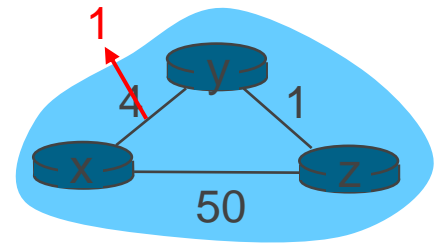
Cada nó:



Vetor de distância: mudanças de custo do enlace

mudanças de custo do enlace:

- ❑ nó detecta mudança de custo no enlace local
- ❑ atualiza informação de roteamento, recalcula vetor de distância
- ❑ se DV mudar, notifica vizinhos



“boas
notícias
correm
rápido”

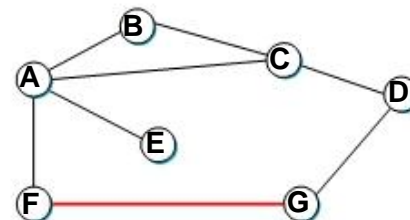
no tempo t_0 , y detecta a mudança do custo do enlace, atualiza seu DV e informa aos seus vizinhos.

no tempo t_1 , z recebe a atualização de y e atualiza sua tabela. Calcula um novo custo mínimo para x e envia seu DV aos vizinhos.

no tempo t_2 , y recebe a atualização de z e atualiza sua tabela de distância. Menores custos de y não mudam, e daí y *não* envia qualquer mensagem a z.

Vetor de Distância - Atualização Acionada

Suponha que ocorre falha no Enlace F-G.



Nó A -Tabela Roteamento			Nó A -Tabela Roteamento			Nó A -Tabela Roteamento				
Destino	Custo	PróxSalto	Nó F		Destino	Custo	PróxSalto	Destino	Custo	PróxSalto
A	0	--	1		A	0	--	A	0	--
B	1	B	2		B	1	B	B	1	B
C	1	C	2		C	1	C	C	1	C
D	2	C	2		D	2	C	D	2	C
E	1	E	2		E	1	E	E	1	E
F	1	F	0		F	1	F	F	1	F
G	2	F	∞		G	∞	--	G	3	C

+

=

+

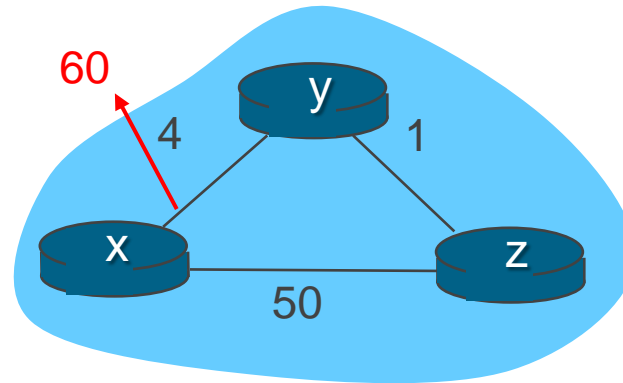
=

Nó F -Tabela Roteamento			Nó F -Tabela Roteamento				
Destino	Custo	PróxSalto	Nó A		Destino	Custo	PróxSalto
A	1	A	0		A	1	A
B	2	A	1		B	2	A
C	2	A	1		C	2	A
D	∞	--	2		D	3	A
E	2	A	1		E	2	A
F	0	--	1		F	0	--
G	∞	--	3		G	4	A

Vetor de distância: mudanças de custo do enlace

mudanças de custo do enlace:

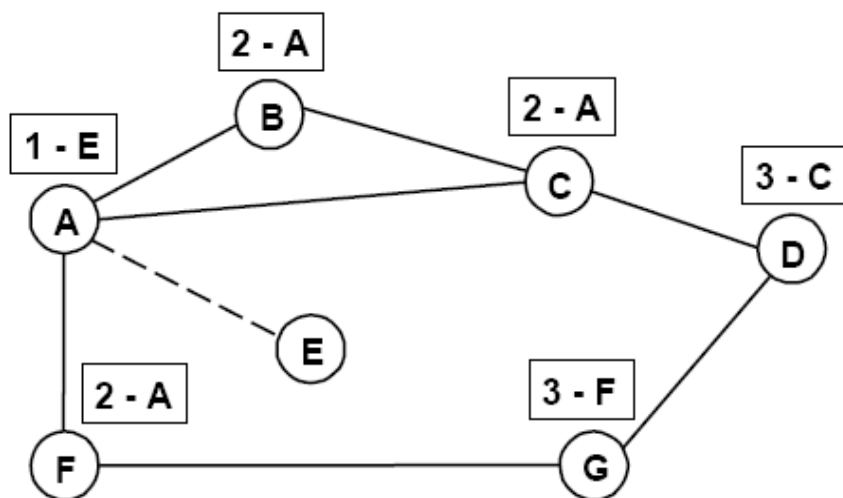
- ❑ boas notícias correm rápido
- ❑ más notícias correm lento
- ❑ 44 iterações antes que o algoritmo estabilize



- ❑ problema da “contagem até o infinito”!

Vetor de Distância – Contagem até infinito

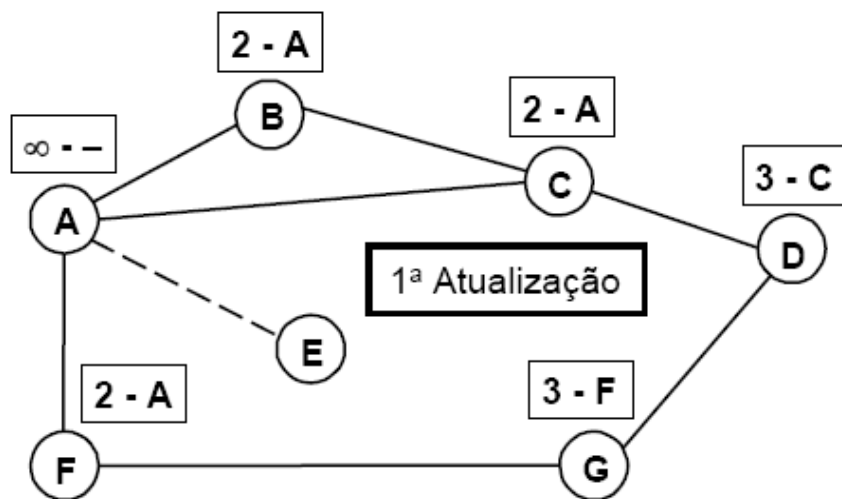
- O Distance Vector apresenta uma **convergência lenta**.



→ Estado da rede no momento de queda do enlace A-E.

OBS.1: pares “n-X” indicam a entrada da tabela local de cada nó para “Destino = nó E”, onde “n” é o “Custo” e “X” é o “PróximoSalto”.

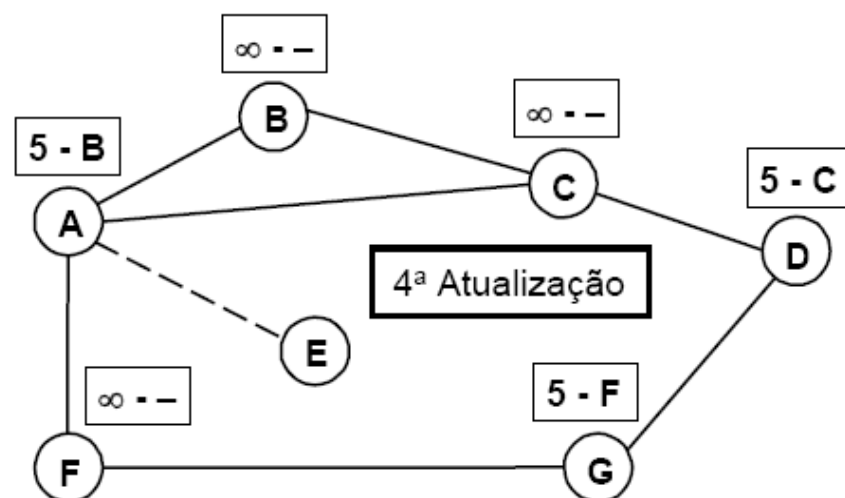
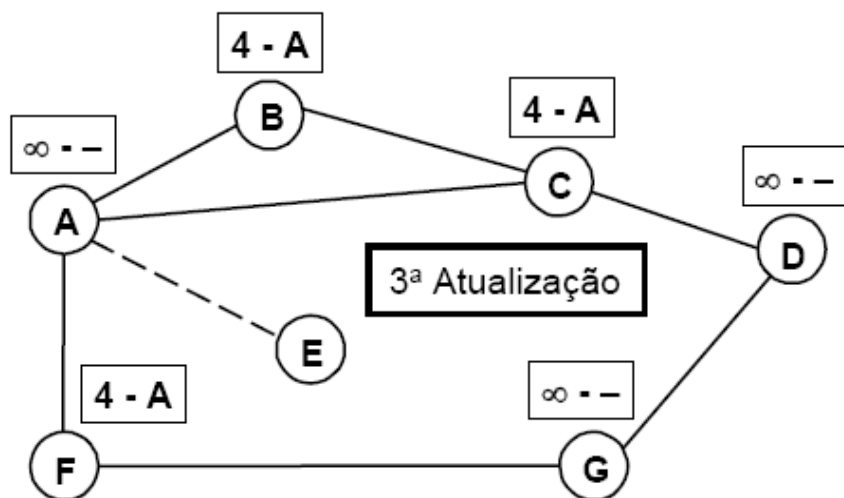
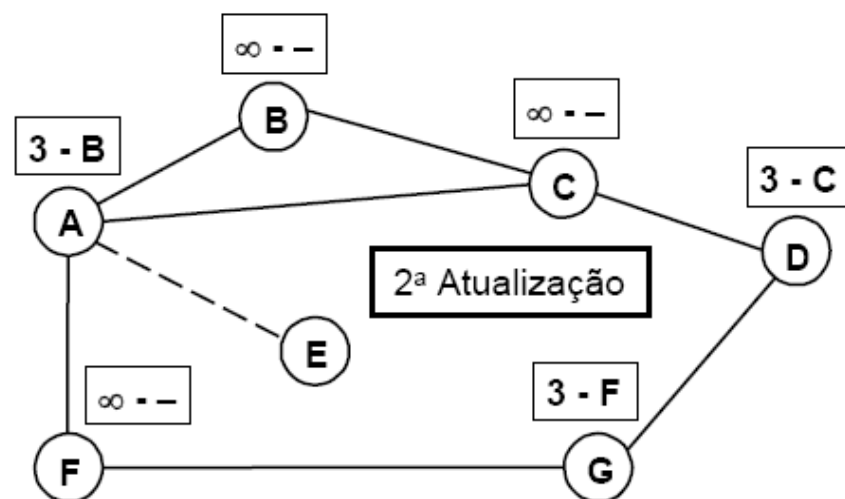
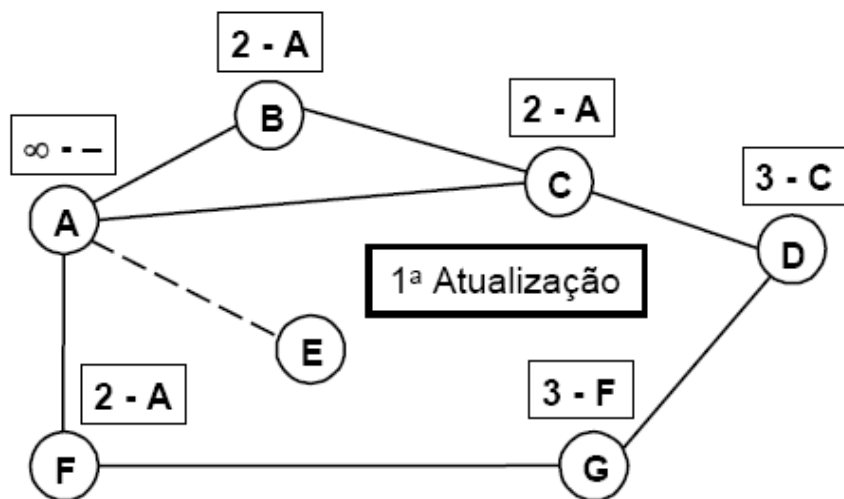
OBS.2: um nó só aceita atualização com “Custo = ∞ ” se esta vier de nó igual a “PróximoSalto”.



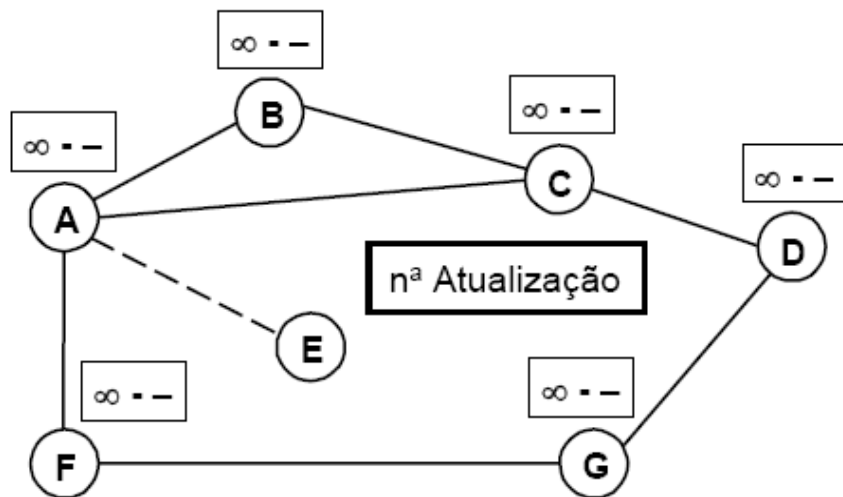
→ Estado da rede após a 1ª Atualização Periódica.

OBS.: ao não receber nenhuma atualização do nó E, o nó A conclui que o nó E está inalcançável (Custo = ∞).

Vetor de Distância – Contagem até infinito



Vetor de Distância – Contagem até infinito



→ Estado final da rede
(após queda do enlace A-E).

Problema da Convergência Lenta
(“Contagem até Infinito”)

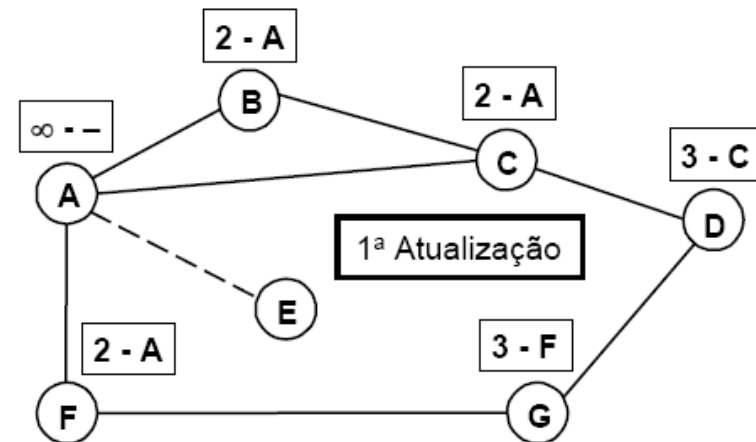
Convergência Lenta - Soluções:

- valor pequeno para “infinito”
- horizonte dividido (*split horizon*)
- envenenamento reverso (*poison reverse*)
- espera (*hold down*)
- atualizações acionadas (*triggered updates*)

Contagem até infinito - soluções

- Número pequeno como aproximação para o infinito.
 - Ex: 16 saltos
 - Limita a quantidade de tempo gasta para contagem até o infinito
 - Problema: se a rota crescesse a mais de 16 hops...
- Split Horizon:
 - Quando um nó enviar uma atualização de roteamento a seus vizinhos, não enviar de volta a esse vizinho aquelas rotas descobertas por ele.
- Envenenamento Reverso
 - B envia a rota de volta para A, mas coloca informações negativas na rota.

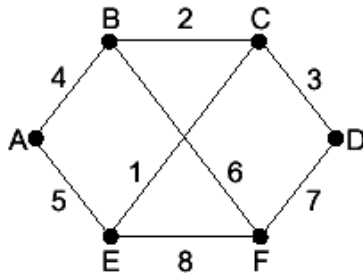
Problema: não funciona para loops de roteamento que envolve 3 ou + nós.



- **4. 1 Introdução**
- **4.2 Redes de circuitos virtuais e de datagramas**
- **4.3 IP: Internet Protocol**
 - formato do datagrama
 - endereçamento IPv4
 - ICMP
 - IPv6
- **4.4 Algoritmos de roteamento**
 - vetor de distâncias
 - **estado de enlace**
 - roteamento hierárquico
- **4.5 Roteamento na Internet**
 - RIP
 - OSPF
 - BGP

Algoritmo de Estado de Enlace

- **Funcionamento do Link State:**
 - Descobrir seus vizinhos e aprender seus endereços de rede.
 - Medir o retardo ou o custo até cada um dos seus vizinhos.
 - Criar um pacote que informe tudo o que ele acabou de aprender.



A	B	C	D	E	F
Seq.	Seq.	Seq.	Seq.	Seq.	Seq.
Age	Age	Age	Age	Age	Age
B 4	A 4	B 2	C 3	A 5	B 6
E 5	C 2	D 3	F 7	C 1	D 7
	F 6	E 1		F 8	E 8

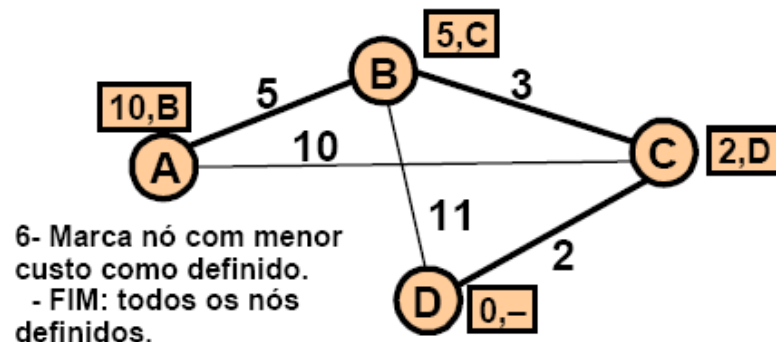
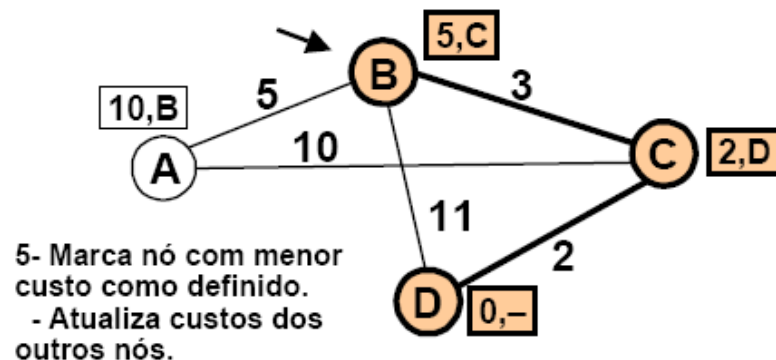
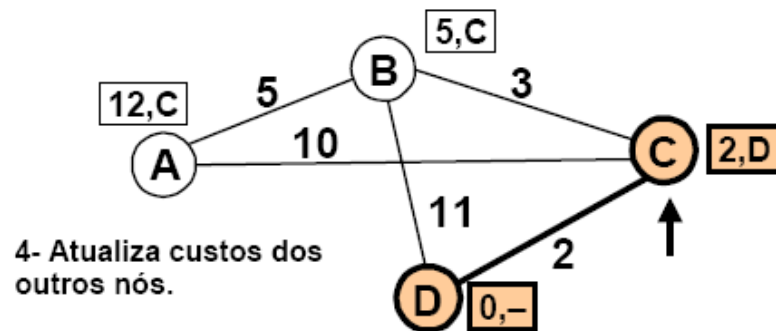
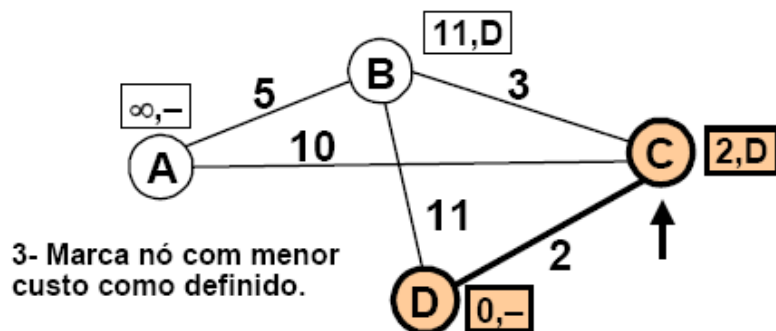
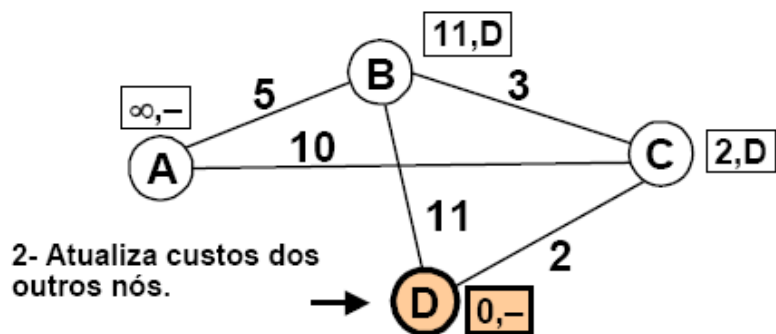
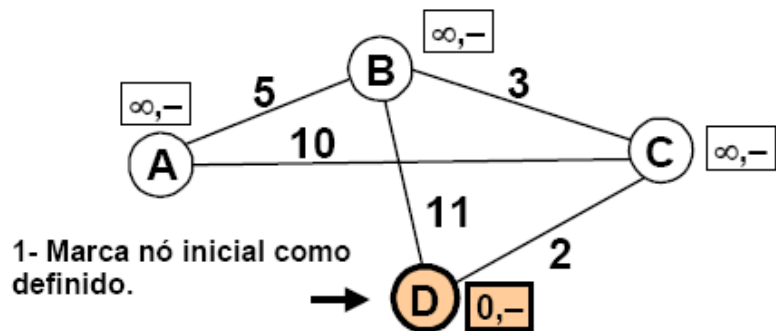
- Enviar esse pacote a todos os outros roteadores
- Calcular o caminho mais curto até cada um dos outros roteadores

O algoritmo Dijkstra



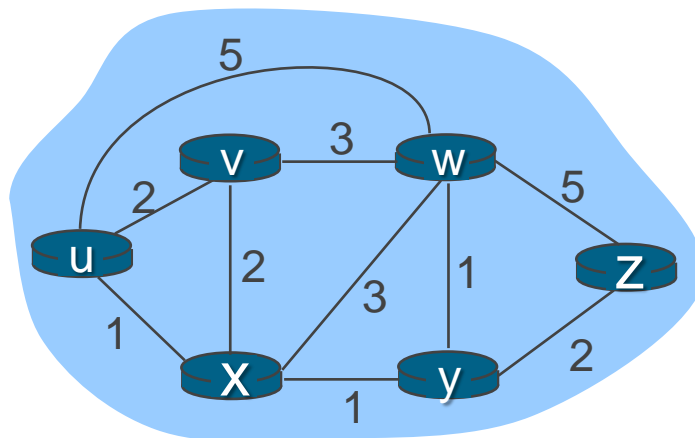
- **Todos roteadores devem ter informação completa da topologia da rede e custo dos enlaces.**
 - realizado por “broadcast de estado do enlace”
 - todos os nós têm a mesma informação
- **calcula caminhos de menor custo de um nó (“origem”) para todos os outros nós**
 - Cria a **tabela de repasse** para esse nó
- **iterativo:** após k iterações, sabe caminho de menor custo para k destinos.
- **As rotas calculadas pelo Dijkstra são sempre livres de loops.**

Algoritmo de Dijkstra



Algoritmo de Dijkstra: exemplo

Etapa	N'	D(v)	D(w)	D(x)	D(y)	D(z)
0	u	2,u	5,u	1,u	∞	∞
1	ux	2,u	4,x		2,x	∞
2	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					



Algoritmo de Dijkstra: exemplo

árvore resultante do caminho mais curto a partir de u:

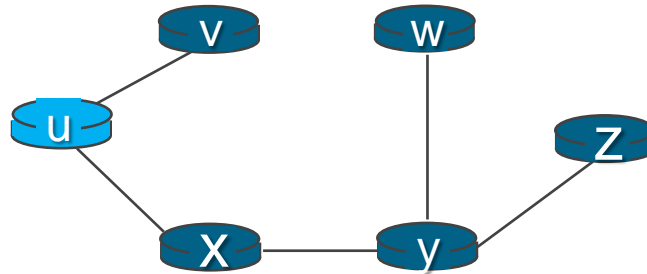
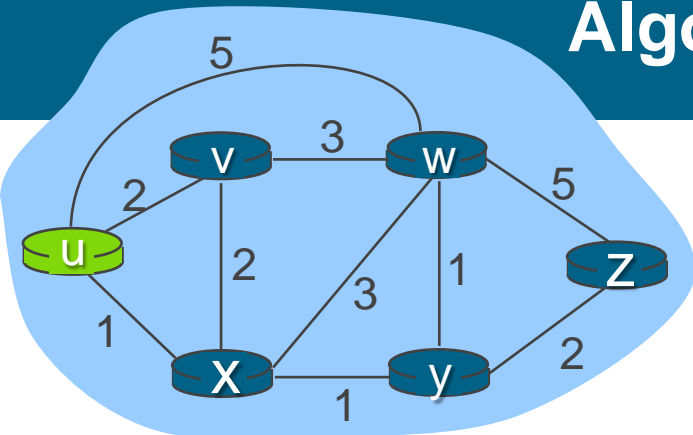


Tabela de repasse resultante em u:

destino	enlace
v	(u,v)
x	(u,x)
y	(u,x)
w	(u,x)
z	(u,x)

Algoritmo de Dijkstra



notação:

- **$c(x,y)$** : custo do enlace do nó x até y ; $= \infty$ se não forem vizinhos diretos
- **$D(v)$** : valor atual do custo do caminho da origem ao destino v
- **$p(v)$** : nó predecessor ao longo do caminho da origem até v
- **N'** : conjunto de nós cujo caminho de menor custo é definitivamente conhecido

1 Inicialização:

- 2 $N' = \{u\}$
- 3 para todos os nós v
- 4 se v adjacente a u
- 5 então $D(v) = c(u,v)$
- 6 senão $D(v) = \infty$

8 Loop

- 9 acha w não em N' tal que $D(w)$ é mínimo
- 10 acrescenta w a N'
- 11 atualiza $D(v)$ para todo v adjacente a w e não em N' :
 $D(v) = \min(D(v), D(w) + c(w,v))$
- 13 /* novo custo para v é custo antigo para v ou custo conhecido
- 14 do caminho mais curto para w + custo de w para v */
- 15 **até todos os nós em N'**

complexidade do algoritmo:

Em termos de otimização, o Dijkstra é um **algoritmo guloso** e sua complexidade é de **$O(n^2)$** , para o pior caso.

robustez: o que acontece se roteador der defeito?

LS:

- nó pode anunciar custo do *enlace* incorreto
- cada nó calcula apenas sua *própria* tabela

DV:

- nó DV pode anunciar custo do *caminho* incorreto
- tabela de cada nó usada por outros
 - erro se propaga pela rede

Vetor de Distância: cada nó conta a seus vizinhos o que sabe sobre todos nós.

Estado de Enlace: cada nó conta a todos os nós o que sabe sobre seus vizinhos.

- 4. 1 Introdução
- 4.2 Redes de circuitos virtuais e de datagramas
- 4.3 IP: Internet Protocol
 - formato do datagrama
 - endereçamento IPv4
 - ICMP
 - IPv6
- **4.4 Algoritmos de roteamento**
 - vetor de distâncias
 - estado de enlace
 - roteamento hierárquico
- 4.5 Roteamento na Internet
 - RIP
 - OSPF
 - BGP

nosso estudo de roteamento até aqui – o ideal:

- ❑ A rede como uma coleção de roteadores interconectados
- ❑ todos os roteadores idênticos

... *não acontece na prática*

escala: + de 21 bilhões de destinos!!

- não é possível armazenar todos os destinos nas tabelas de roteamento!
- troca de tabela de roteamento atolaria os enlaces!
- Um algoritmo DV interagindo com um n^o tão grande de roteadores jamais convergiria!

autonomia administrativa

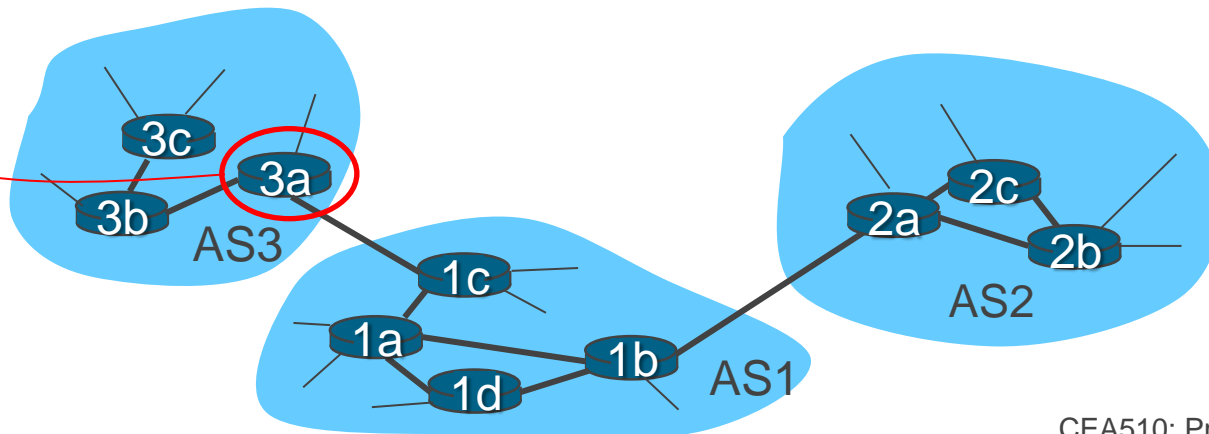
- Internet = rede de redes;
- cada administrador de rede pode querer controlar o roteamento em sua própria rede;

Roteamento hierárquico

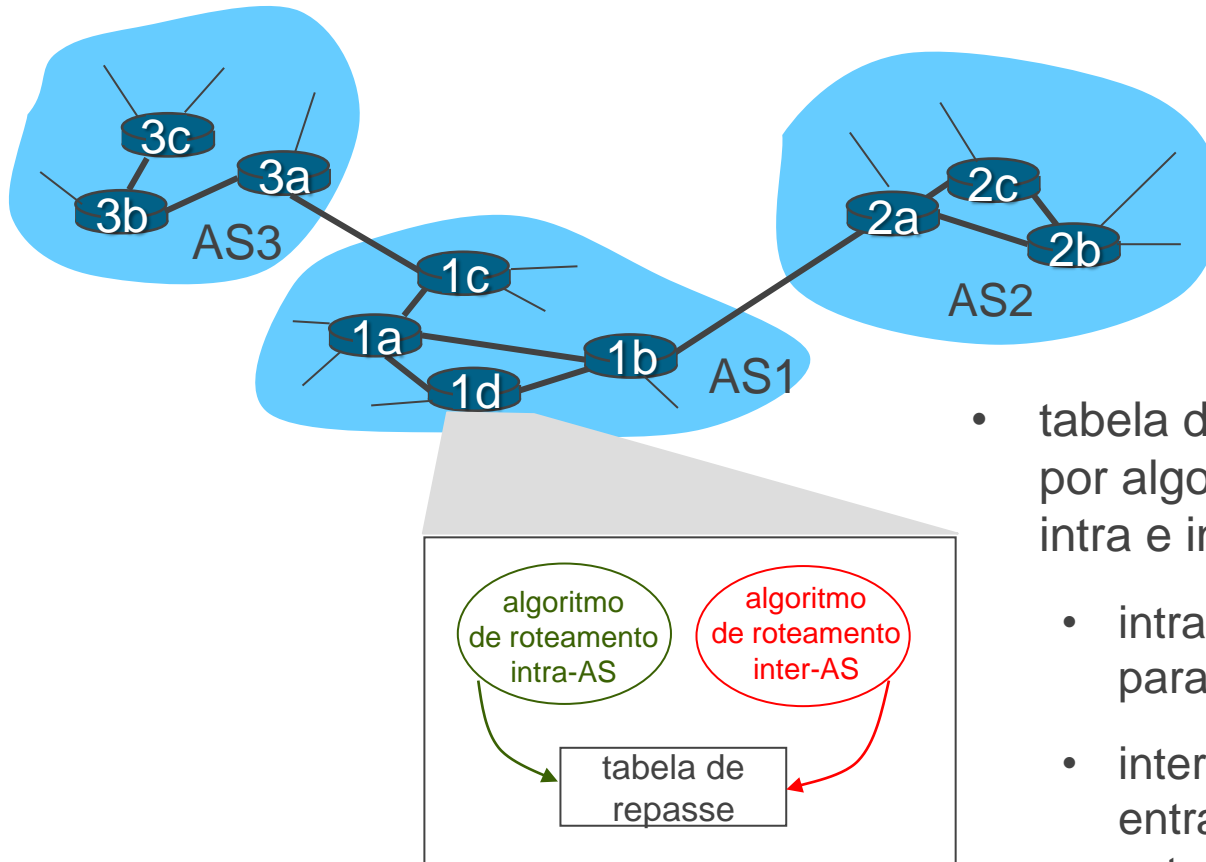
- roteadores agregados em regiões, “**sistemas autônomos**” (AS)
 - Ex. de AS: redes de universidade, empresas, ISP;
- roteadores no mesmo AS rodam o mesmo protocolo de roteamento
 - protocolo de **roteamento “intra-AS”**
 - roteadores em ASs diferentes podem executar protocolo de roteamento intra-AS diferente

roteador de borda

- Enlace direto com roteador em outro AS



ASs interconectados



- tabela de repasse configurada por algoritmo de roteamento intra e inter-AS
 - intra-AS define entradas para destinos internos
 - inter-AS & intra-AS definem entradas para destinos externos

Tarefas inter-AS

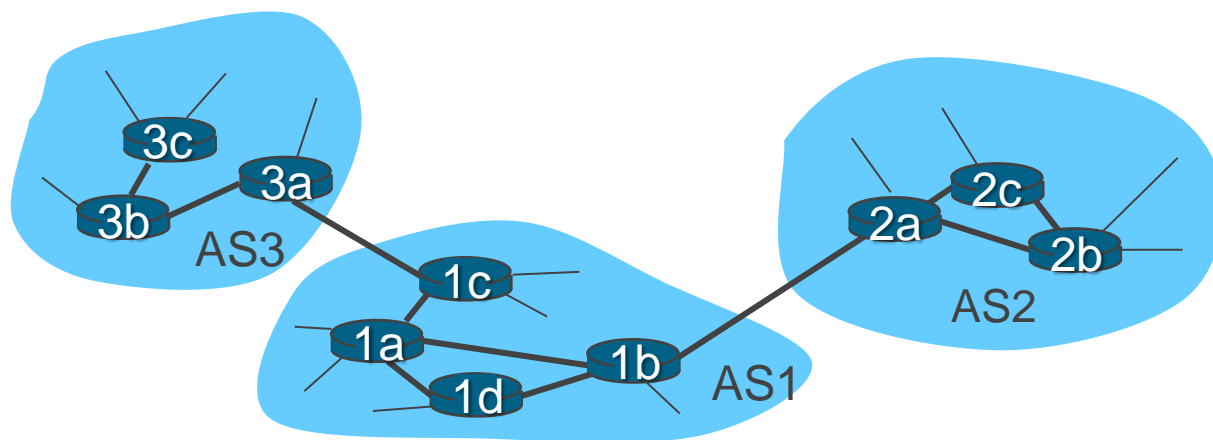
Como um roteador de um AS sabe como rotear um pacote para outro AS?

- suponha que roteador no AS1 recebe datagrama destinado para fora do AS1:
 - roteador deve encaminhar pacote ao roteador de borda, mas qual?

AS1 deve:

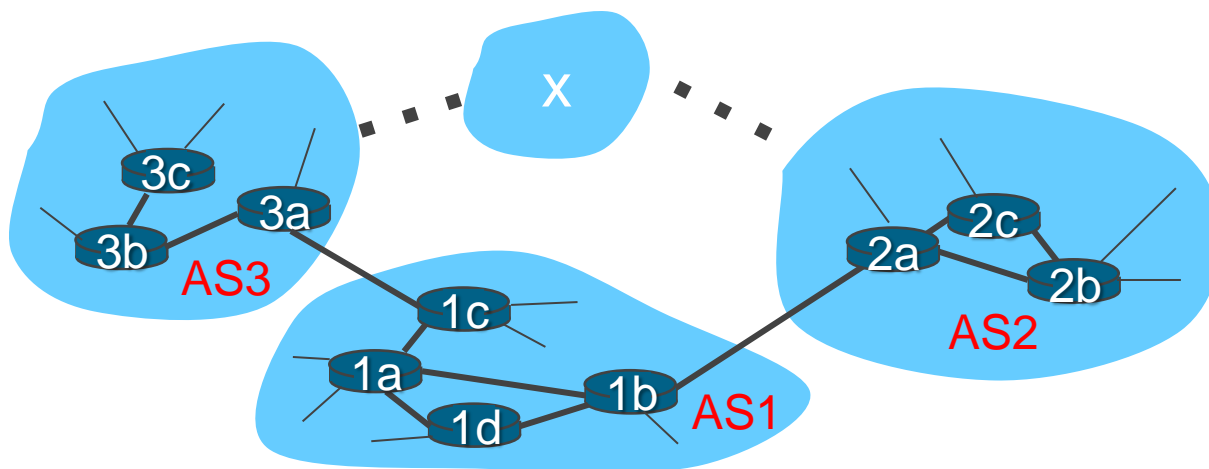
1. descobrir quais destinos são alcançáveis por AS2 e quais por AS3
2. propagar essa informação de acessibilidade a todos os roteadores no AS1

Tarefa do roteamento inter-AS!



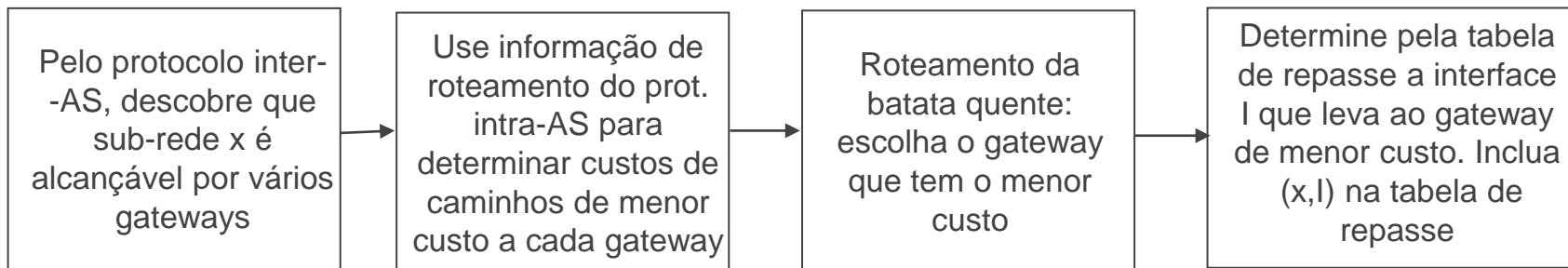
Ex: escolhendo entre múltiplos ASes

- agora suponha que o AS1 descubra pelo protocolo inter-AS que a sub-rede **x** pode ser alcançada por AS3 e por AS2.
- para configurar a tabela de repasse, roteador *1a* deve determinar para que gateway ele deve repassar os pacotes para o destino **x**.
 - isso também é tarefa do protocolo de roteamento inter-AS!



Ex: escolhendo entre múltiplos ASes

- O AS se livra do pacote o + rápido possível! (com o menor custo).
- **roteamento da batata quente:** envia pacote para o mais próximo dos dois roteadores.



- **4. 1 Introdução**
- **4.2 Redes de circuitos virtuais e de datagramas**
- **4.3 IP: Internet Protocol**
 - formato do datagrama
 - endereçamento IPv4
 - ICMP
 - IPv6
- **4.4 Algoritmos de roteamento**
 - vetor de distâncias
 - estado de enlace
 - roteamento hierárquico
- **4.5 Roteamento na Internet**
 - RIP
 - OSPF
 - BGP

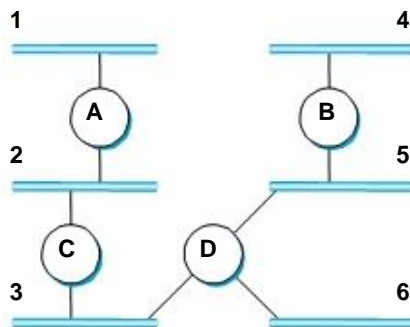
- **Conheça um roteamento mais inteligente:**
 - hosts conhecem o roteador local;
 - roteadores locais conhecem roteadores do site;
 - roteadores do site conhecem roteador de borda;
 - roteadores de borda se conhecem.
- **Hierarquia de propagação de rota em dois níveis**
 - IGP - protocolo de gateway interior (cada AS seleciona o seu):
RIP, OSPF, etc.
 - EGP - protocolo de gateway exterior (padrão de toda a Internet):
BGP.

- também conhecido como **IGP** (Interior Gateway Protocols)
- protocolos de roteamento intra-AS mais comuns:
 - RIP: Routing Information Protocol
 - OSPF: Open Shortest Path First
 - IGRP: Interior Gateway Routing Protocol (proprietário da Cisco)

- **4. 1 Introdução**
- **4.2 Redes de circuitos virtuais e de datagramas**
- **4.3 IP: Internet Protocol**
 - formato do datagrama
 - endereçamento IPv4
 - ICMP
 - IPv6
- **4.4 Algoritmos de roteamento**
 - vetor de distâncias
 - estado de enlace
 - roteamento hierárquico
- **4.5 Roteamento na Internet**
 - **RIP**
 - OSPF
 - BGP

Protocolo RIP

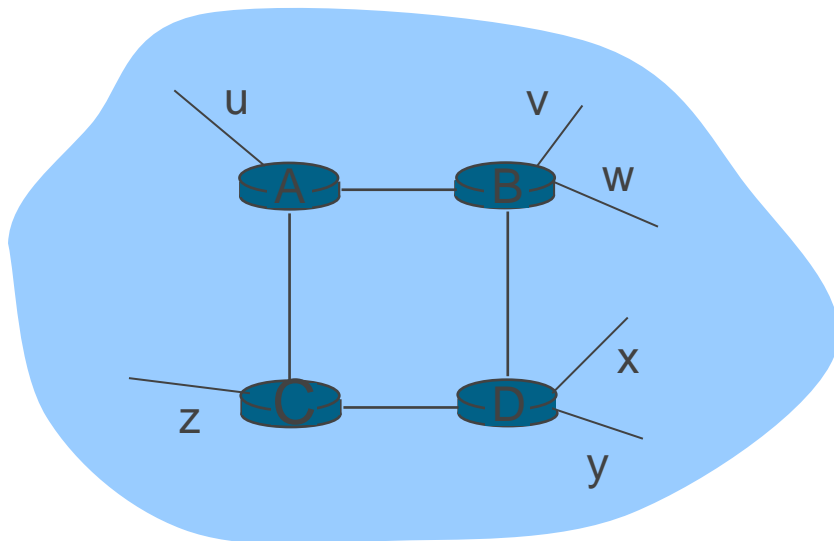
- **RIP (Routing Information Protocol):**
 - usado em inter-redes IP (mas admite outras famílias)
 - popular (distribuição grátis com Unix)
 - implementação simples do “vetor de distância”
 - roteadores anunciam quantidade de saltos para atingir as redes



- atualizações a cada 30 s ou quando tabela local muda
- métrica = quantidade de saltos (custo do enlace = 1)
- distâncias válidas = 1 a 15 (infinito = 16)
- limitado a redes com diâmetro de 15 saltos (15 enlaces)

RIP (Routing Information Protocol)

- algoritmo de vetor de distância
- métrica de distância: # de saltos (máx. = 15 saltos)



Do roteador A às sub-redes:

destino	saltos
u	1
v	2
w	2
x	3
y	3
z	2

- *vetores de distância*: trocados entre vizinhos a cada 30 s por meio de mensagem de resposta (também conhecida como **anúncio**)
- cada anúncio: lista de até 25 sub-redes de destino dentro do AS

se nenhum anúncio for ouvido após 180 s --> vizinho/enlace declarado morto

- rotas via vizinho invalidadas
- novos anúncios enviados aos vizinhos
- vizinhos por sua vez enviam novos anúncios (se não houver tabelas alteradas)
- informação de falha do enlace rapidamente (?) se propaga para rede inteira
- *reversão envenenada* usada para impedir loops de ping-pong (distância infinita = 16 saltos)

- **4. 1 Introdução**
- **4.2 Redes de circuitos virtuais e de datagramas**
- **4.3 IP: Internet Protocol**
 - formato do datagrama
 - endereçamento IPv4
 - ICMP
 - IPv6
- **4.4 Algoritmos de roteamento**
 - vetor de distâncias
 - estado de enlace
 - roteamento hierárquico
- **4.5 Roteamento na Internet**
 - RIP
 - OSPF
 - BGP

OSPF (Open Shortest Path First)



- **“open”**: publicamente disponível
 - padrão aberto do IETF (*Internet Engineering Task Force*);
- **usa algoritmo Link State**
 - disseminação de pacote LS
 - mapa de topologia em cada nó
 - cálculo de rota usando algoritmo de Dijkstra (ou Prim)
- **anúncio OSPF transporta uma entrada por roteador vizinho**
- **anúncios disseminados ao AS inteiro (com inundação)**
 - transportados nas mensagens OSPF diretamente por IP (em vez de TCP ou UDP)

Recursos “avançados” do OSPF



- Mais poderoso que o RIP;
- **segurança:** todas as mensagens OSPF autenticadas (para impedir intrusão maliciosa)
- **múltiplos caminhos** de mesmo custo permitidos (apenas um caminho no RIP)
- para cada enlace, suporta várias métricas simultaneamente (facilita implementação de QoS);
- suporta balanceamento de carga entre rotas (evita oscilações de roteamento);

- **4. 1 Introdução**
- **4.2 Redes de circuitos virtuais e de datagramas**
- **4.3 IP: Internet Protocol**
 - formato do datagrama
 - endereçamento IPv4
 - ICMP
 - IPv6
- **4.4 Algoritmos de roteamento**
 - vetor de distâncias
 - estado de enlace
 - roteamento hierárquico
- **4.5 Roteamento na Internet**
 - RIP
 - OSPF
 - BGP

Roteamento interdomínio

- O problema de roteamento entre domínios é **tão difícil**, que os objetivos são mais modestos:
- **1º objetivo:** encontrar algum caminho para o destino, **livre de loops**.
 - a maior preocupação é o alcance do que a rota ideal.
- Localizar um caminho que seja um pouco mais próximo do ideal é considerado uma grande realização!
- A principal dificuldade que torna o roteamento interdomínio complexo é a escala da Internet.
- Embora o CIDR tenha ajudado a controlar o nº de prefixos distintos que são transportados no roteamento da Internet é **enorme**.
- **Aprox. 380.000 prefixos diferentes.**



Roteamento interdomínio

- O segundo desafio no roteamento interdomínio vem da natureza autônoma dos AS.
 - Cada domínio pode executar seus próprios protocolos de roteamento.
 - Isso significa dizer que é quase impossível calcular custos de um caminho para outro que cruze vários AS.
 - Um custo de 1.000 pode ser ótimo para um provedor, mas péssimo para outro AS.
- Como resultado, o roteamento interdomínio anuncia apenas o “alcance”.
 - Você pode alcançar esta rede através desse AS.
- Assim, pode-se escolher bons caminhos, mas o escolher o **melhor caminho** torna-se uma missão quase impossível.

- Padrão: **BGP** (Border Gateway Protocol)
- BGP oferece a cada AS um meio de:
 1. obter informação de acessibilidade da sub-rede a partir de ASs vizinhos.
 2. propagar informação de acessibilidade a todos os roteadores internos ao AS.
 3. determinar rotas “boas” para sub-redes com base na informação e política de acessibilidade.
- permite que a sub-rede anuncie sua existência ao resto da Internet: “*Estou aqui*”



- O BGP foi definido para ser executado em cima do TCP.
 - *Significa que um roteador executando o BGP não precisa reenviar mensagens, já que o TCP é confiável.*
- O número de nós participantes no BGP está na ordem no n^0 de AS, que é muito menor que o número de redes.
- Encontrar uma boa rota é uma questão de encontrar um caminho correto.
- Assim, o problema de roteamento foi subdividido:
 - **Roteamento interdomínio:** na ordem no n^0 de roteadores do AS
 - **Roteamento intradomínio:** está na ordem no n^0 de redes em um AS.

SERIA ÓTIMO SE TIVESSE COMO TER
O ALGORITMO DE ROTEAMENTO
ATÉ AMANHÃ, COM TODOS ESSES
REQUISITOS DE ROTAS QUE
O CLIENTE QUER...



JÁ TEMOS TODOS OS MAPAS,
SÓ FALTA VOCÊ FAZER O
ALGORITMO...



SERIA TÃO BOM SE O PÉ DE
LARANJAS LÁ DE CASA
PRODUZISSE MAÇÃS ATÉ
AMANHÃ DE MANHÃ...
DEU VONTADE DE COMER
MAÇÃS...

