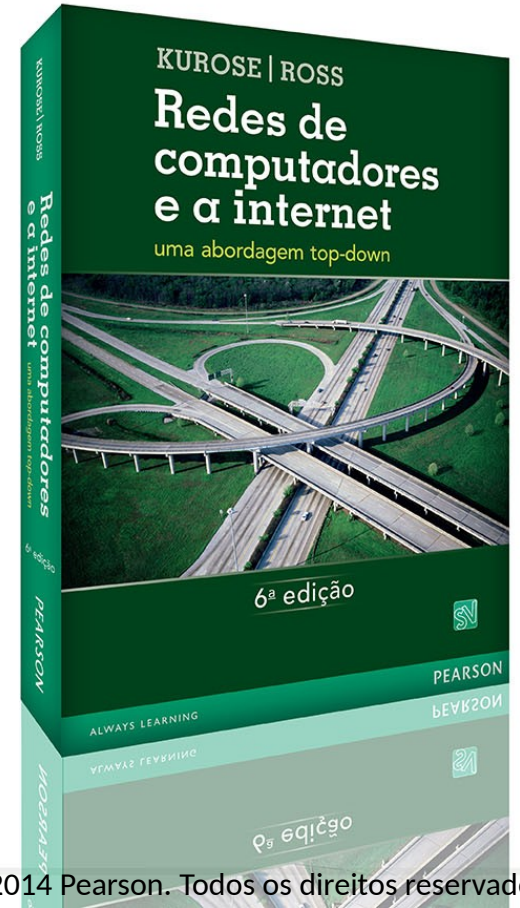
Capítulo 4

A camada de REDE

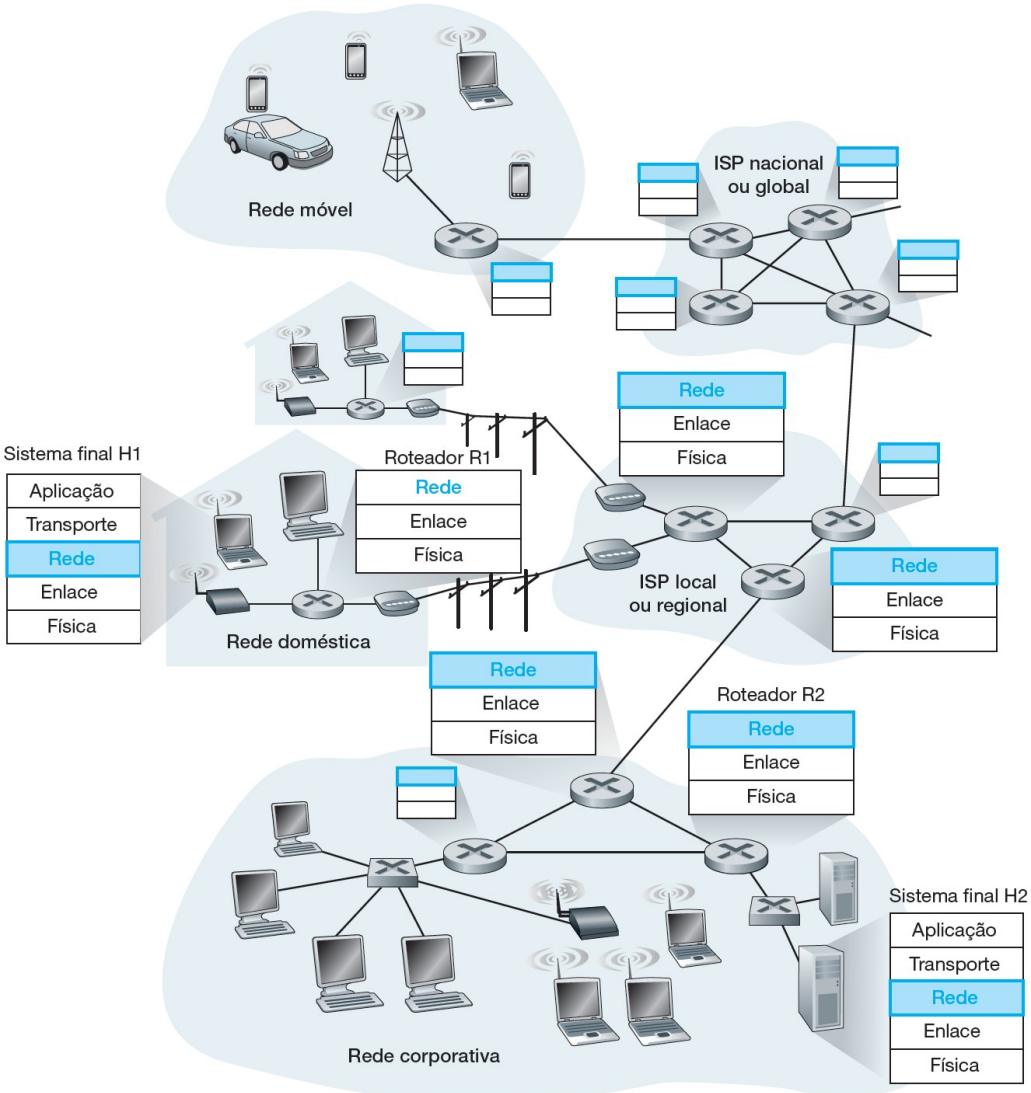


Camada de rede

Objetivos do capítulo:

- entender os princípios por trás dos serviços da camada de rede:
  - modelos de serviço da camada de rede
  - repasse *versus* roteamento
  - como funciona um roteador
  - protocolo IP
  - roteamento (seleção de caminho)
  - lidando com escala
  - tópicos avançados: IPv6
- instanciação, implementação na Internet

Introdução

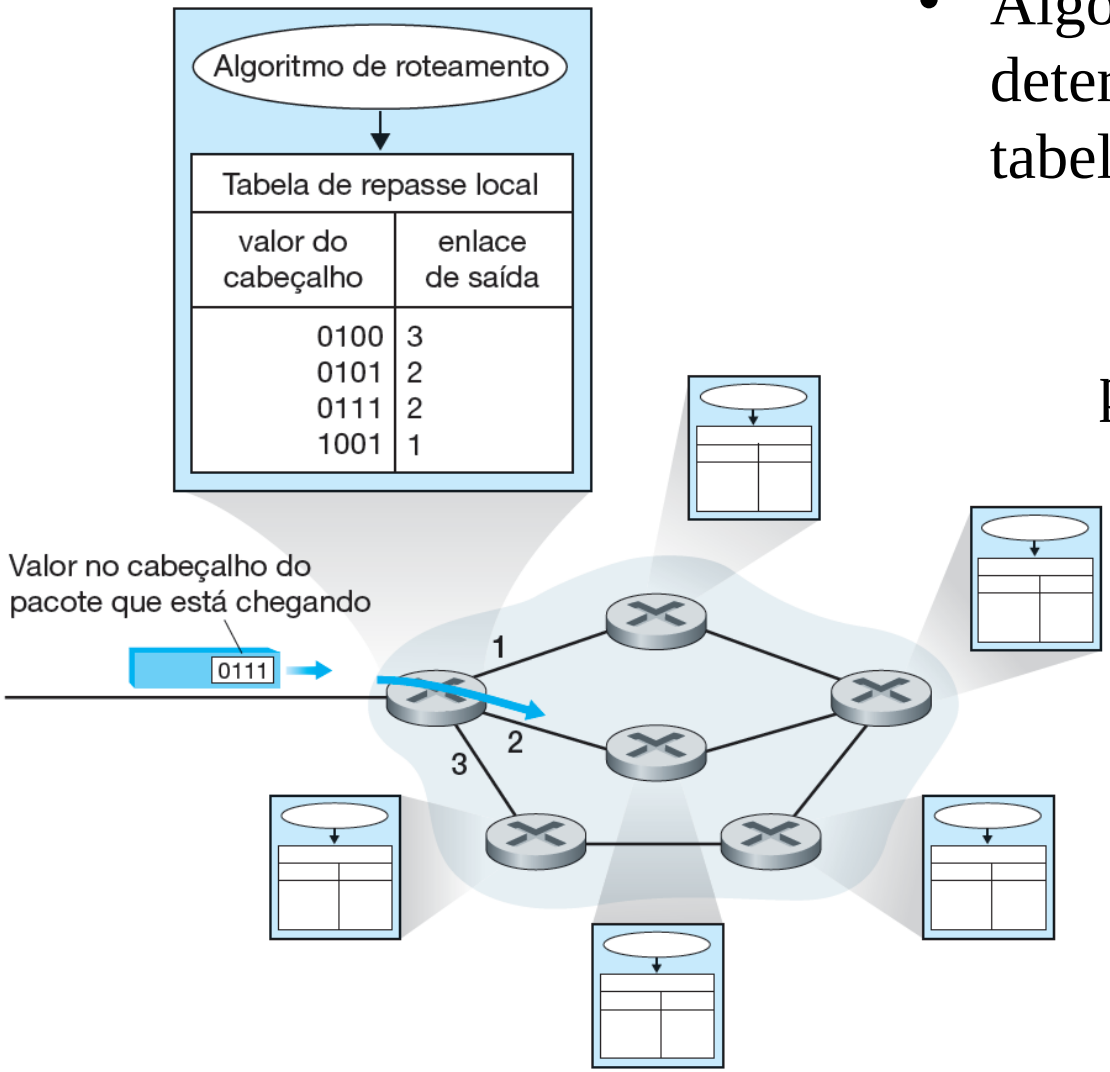


- segmento de transporte do hosp. emissor ao receptor
- o lado emissor encapsula segmentos em datagramas
- o lado receptor entrega segmentos à camada de transporte
- protocolos da camada de rede em *cada* hosp., roteador
- roteador examina campos de cabeçalho em todos os datagramas IP que passam por ele

Repasse e roteamento

- O papel da camada de rede é transportar pacotes de um hospedeiro remetente a um hospedeiro destinatário.
- **Repasse**. Quando um pacote chega ao enlace de entrada de um roteador, este deve conduzi-lo até o enlace de saída apropriado.
- **Roteamento**. A camada de rede deve determinar a rota ou o caminho tomado pelos pacotes ao fluírem de um remetente a um destinatário.
- **Estabelecimento de conexão**: ATM, *frame-relay* e MPLS, estabelecem estado antes do início da conexão.

Repasse e roteamento



- Algoritmos de roteamento determinam valores em tabelas de repasse:
  - O valor pode ser o endereço de destino do pacote ou uma indicação à qual pertence.

Modelos de serviço de rede

- O **modelo de serviço de rede** define as características do transporte de dados fim a fim entre uma borda da rede e a outra.

Alguns serviços específicos que **poderiam** ser oferecidos são:

- Entrega garantida.
- Entrega garantida com atraso limitado.
- Entrega de pacotes na ordem.
- Largura de banda mínima garantida.
- *Jitter* máximo garantido.
- Serviços de segurança.
- Alguma informação sobre congestionamento.

Modelos de serviço de rede

- Modelos de serviço das redes Internet, ATM CBR e ATM ABR

Arquitetura da rede	Modelo de serviço	Garantia de largura de banda	Garantia contra perda	Ordenação	Temporização	Indicação de congestionamento
Internet	Melhor esforço	Nenhuma	Nenhuma	Qualquer ordem possível	Não mantida	Nenhuma
ATM	CBR	Taxa constante garantida	Sim	Na ordem	Mantida	Não haverá congestionamento
ATM	ABR	Mínima garantida	Nenhuma	Na ordem	Não mantida	Indicação de congestionamento

Redes de circuitos virtuais e de datagramas

Os serviços de rede orientado e não orientado a conexão apresentam diferenças em relação aos mesmos conceitos da camada de transporte:

- Na camada de rede, são serviços de hospedeiro a hospedeiro providos pela camada de rede à de transporte. Na camada de transporte, são serviços de processo a processo oferecidos pela camada de transporte à aplicação.
- Em todas as arquiteturas importantes de redes de computadores a camada de rede oferece um dos tipos de serviços mas não ambos.
- A execução de serviço orientado para conexão na camada de rede é realizado no núcleo da rede em conjunto com os sistemas finais, diferentemente da camada de transporte.

Circuitos virtuais

"Caminho da origem ao destino comporta-se como um circuito telefônico"

- com respeito ao desempenho
- ações da rede ao longo do caminho da origem ao destino
- dados chegam em ordem
- todos os dados seguem o mesmo trajeto

- estabelecimento e término para cada chamada *antes* que os dados possam fluir
- cada pacote carrega identificador VC (não endereço do hospedeiro de destino)
- *cada* roteador no caminho origem-destino mantém "estado" para cada conexão que estiver passando
- recursos do enlace e roteador (largura de banda, *buffers*) podem ser *alocados* ao VC (recursos dedicados = serviço previsível)

Redes de circuitos virtuais

- Um circuito virtual (CV) consiste em:

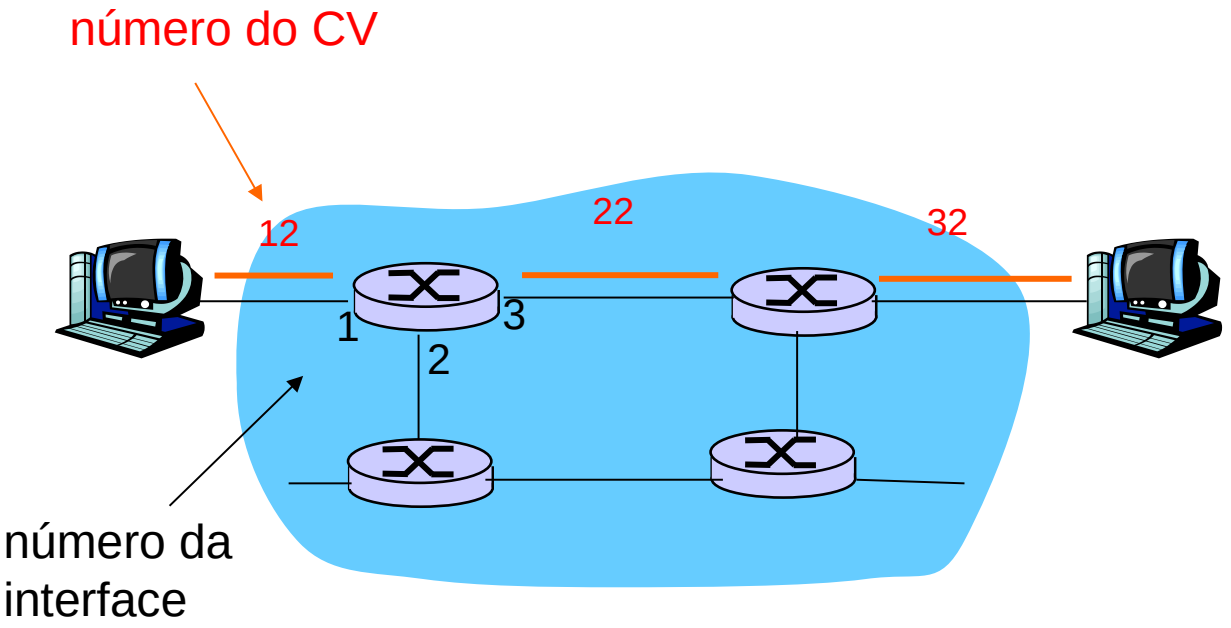
1. um caminho (isto é, uma série de enlaces e roteadores) entre hospedeiros de origem e de destino,

2. números de CVs, um número para cada enlace ao longo do caminho e

3. registros na tabela de repasse em cada roteador ao longo do caminho.

Redes de circuitos virtuais

- Uma rede de circuitos virtuais simples:

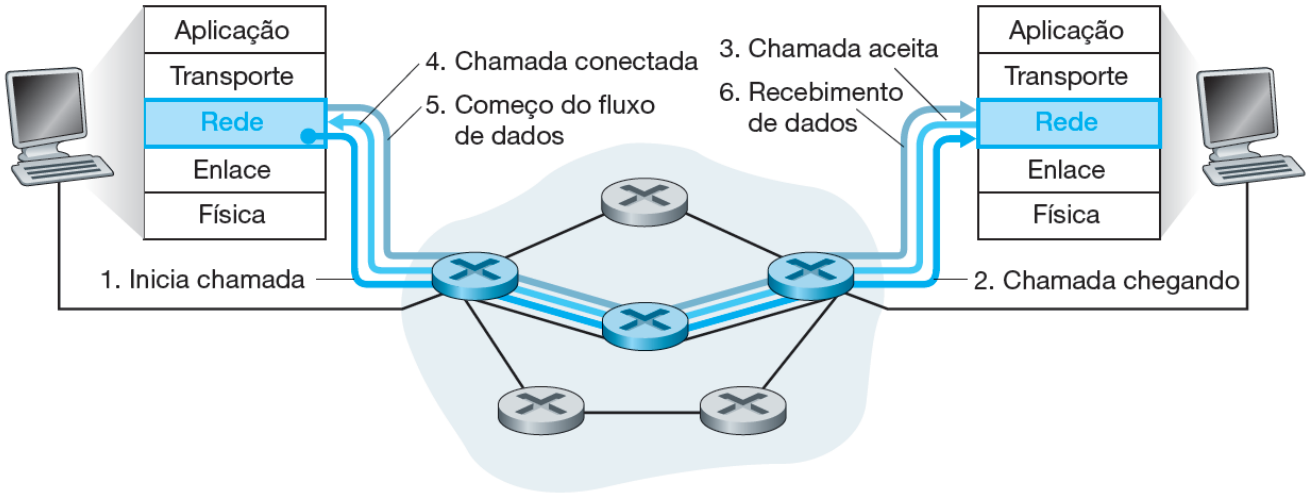


Interface de entrada	Nº do CV de entrada	Interface de saída	Nº do CV de saída
1	12	3	22
2	63	1	18
3	7	2	17
1	97	3	87
...	...	...	...

Redes de circuitos virtuais

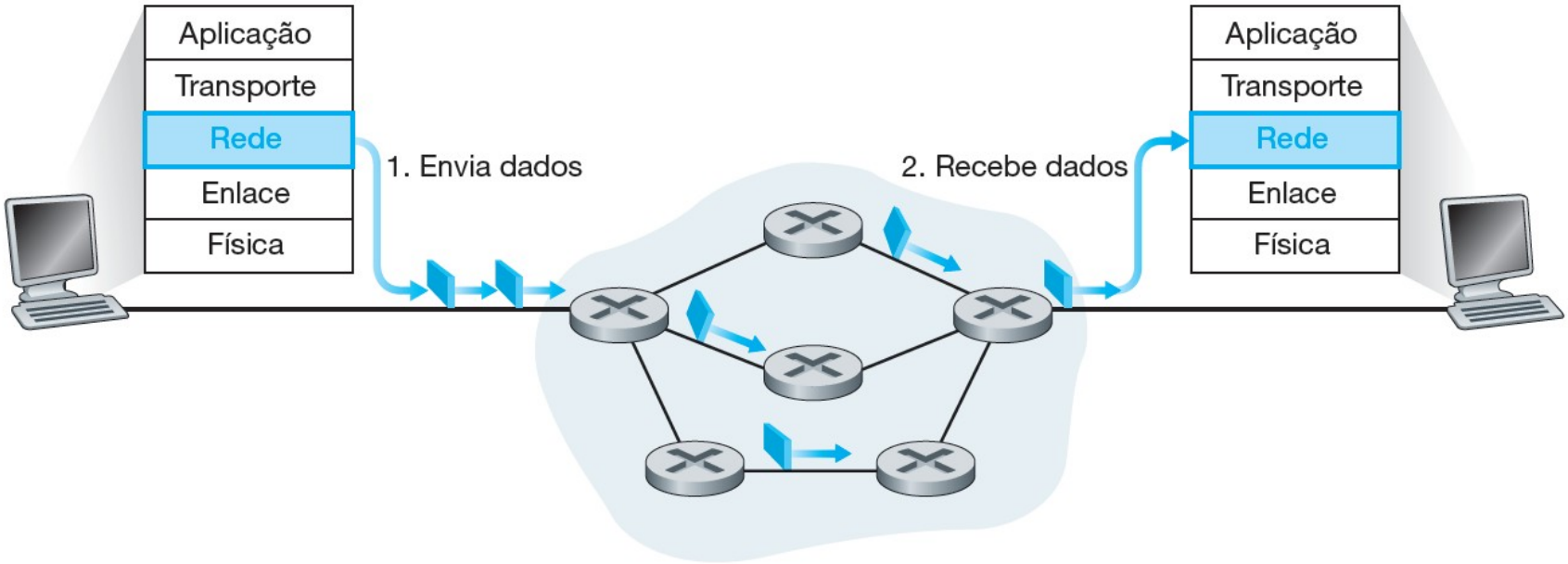
- Há três fases que podem ser identificadas em um circuito virtual:

1. Estabelecimento de CV.
2. Transferência de dados.
3. Encerramento do CV.



Redes de datagramas

- Em uma rede de datagramas, toda vez que um sistema final quer enviar um pacote, ele marca o pacote com o endereço do sistema final de destino e então o envia para dentro da rede.



Redes de datagramas

- Ao ser transmitido da origem ao destino, um pacote passa por uma série de roteadores.
- Cada um desses roteadores usa o endereço de destino do pacote para repassá-lo.
- Então, o roteador transmite o pacote para aquela interface de enlace de saída.
- A tabela de repasse de um roteador é modificada pelos algoritmos de roteamento que, em geral, atualizam as tabelas entre 1 e 5 min.

Tabela de repasse

4 bilhões de entradas possíveis

Faixa de endereços de destino	Interface de enlace
11001000 00010111 00010000 00000000 (200.23.16.0) até 11001000 00010111 00010111 11111111 (200.23.23.255)	0
11001000 00010111 00011000 00000000 (200.23.24.0) até 11001000 00010111 00011000 11111111 (200.23.24.255)	1
11001000 00010111 00011001 00000000 (200.23.25.0) até 11001000 00010111 00011111 11111111 (200.23.31.255)	2
senão	3

Concordância do prefixo mais longo

Concordância do prefixo	Interface do enlace
11001000 00010111 00010	0
11001000 00010111 00011000	1
11001000 00010111 00011	2
senão	3

Exemplos

DA: 11001000 00010111 00011010 10101010 Qual interface?

DA: 11001000 00010111 00011000 10101010 Qual interface?

DA: 11001000 00010111 00010110 10100001 Qual interface?

Rede de datagramas ou VC: por quê?

Internet (datagrama)

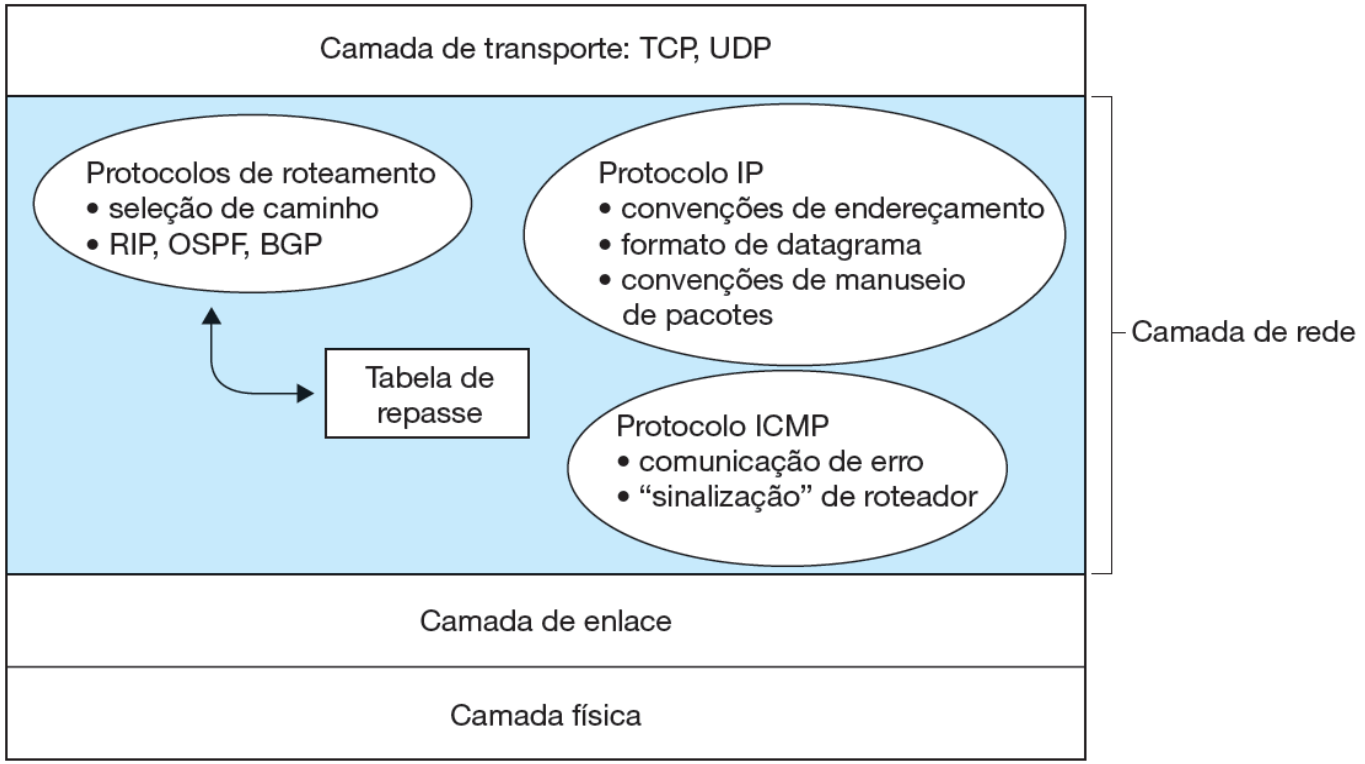
- troca de dados entre computadores
  - serviço “elástico”, sem requisitos de temporização estritos
- sistemas finais “inteligentes” (computadores)
  - pode adaptar, realizar controle, recup. de erros
  - simples dentro da rede, complexidade na “borda”
- muitos tipos de enlace
  - diferentes características
  - serviço uniforme difícil

ATM (VC)

- evoluída da telefonia
- conversação humana:
  - requisitos de temporização estritos, confiabilidade
  - necessário para serviço garantido
- sistemas finais “burros”
  - telefones
  - complexidade dentro da rede
- Homogeneidade de enlaces.

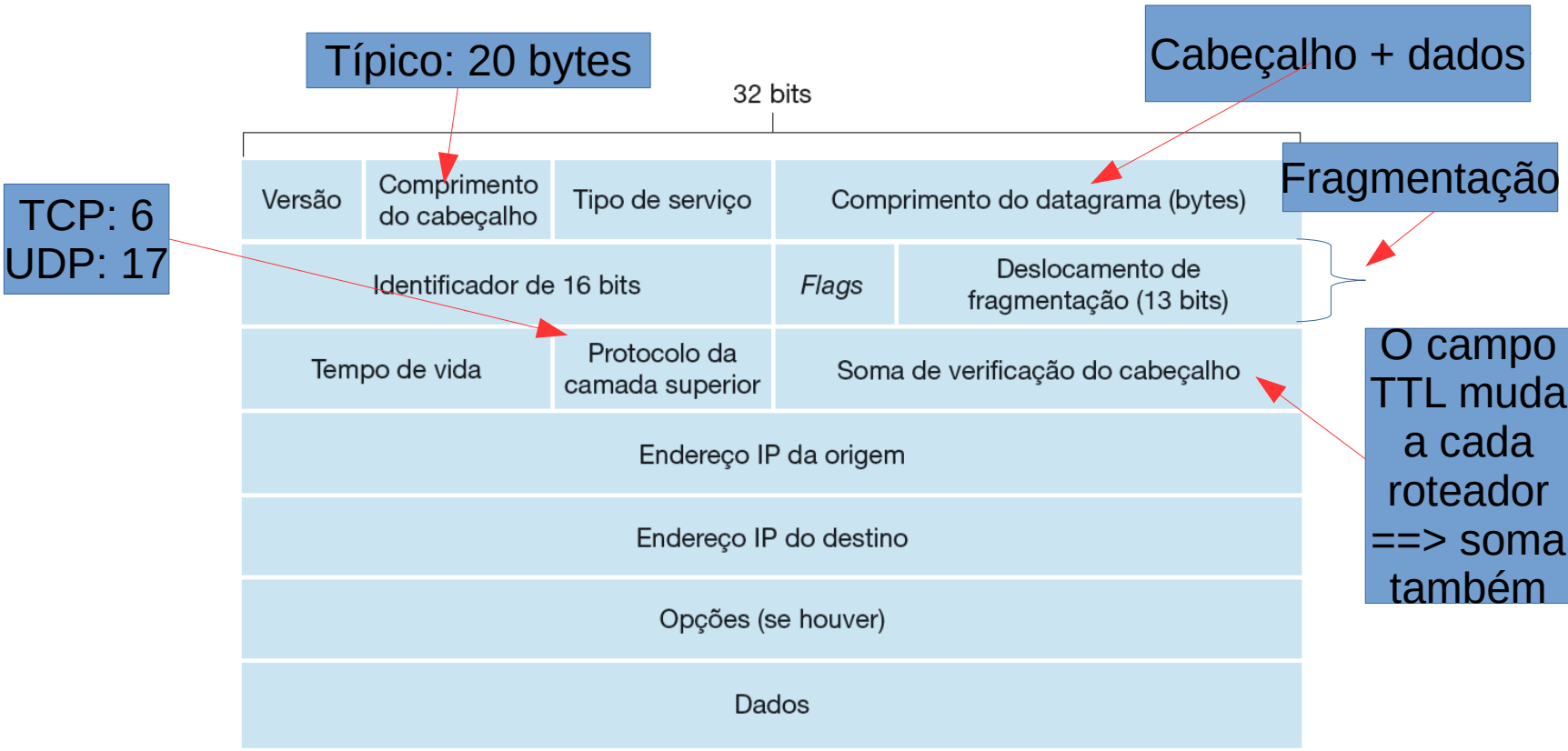
O Protocolo da Internet (IP): repasse e endereçamento na Internet

- Contemplando o interior da camada de rede da Internet



Formato de datagrama

- Formato do datagrama IPv4
  20 (TCP) + 20 (IP) = 40 bytes de cabeçalho

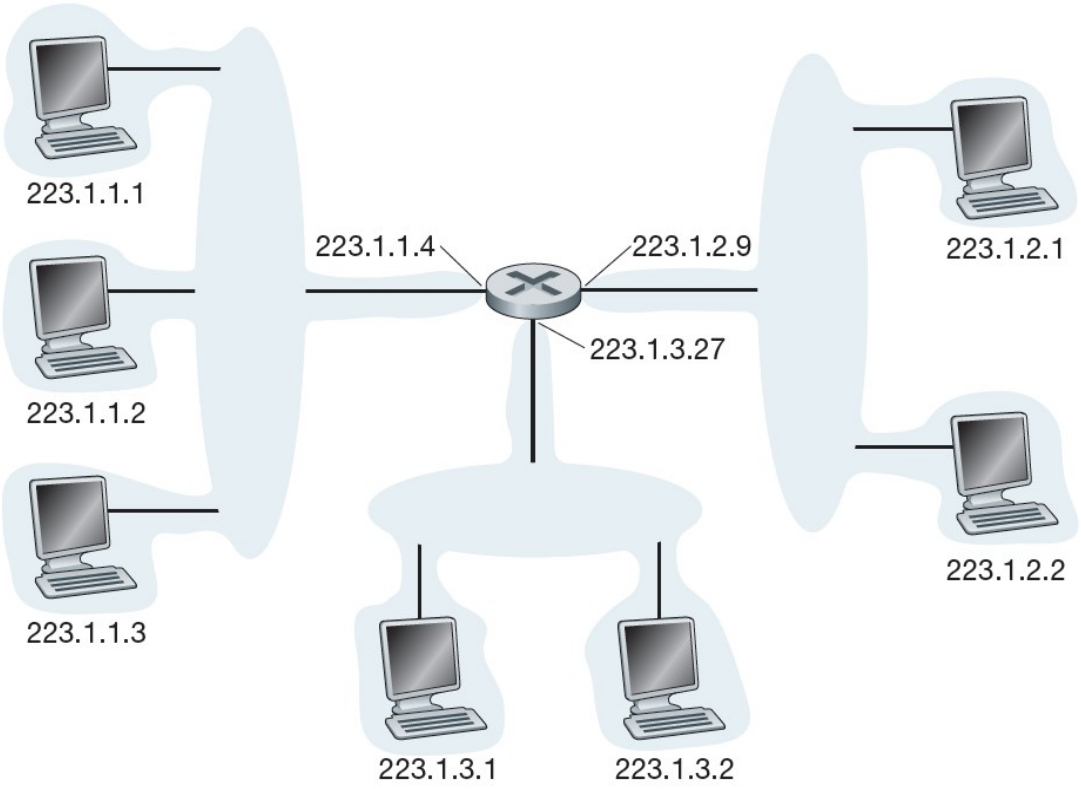


Endereçamento IPv4

- Um endereço IP está tecnicamente associado com uma interface.
- Cada endereço IP tem comprimento de 32 bits (equivalente a 4 bytes).
- Portanto, há um total de 2^{32} endereços IP possíveis.
- É fácil ver que há cerca de 4 bilhões de endereços IP possíveis.
- Esses endereços são escritos em **notação decimal separada por pontos**.

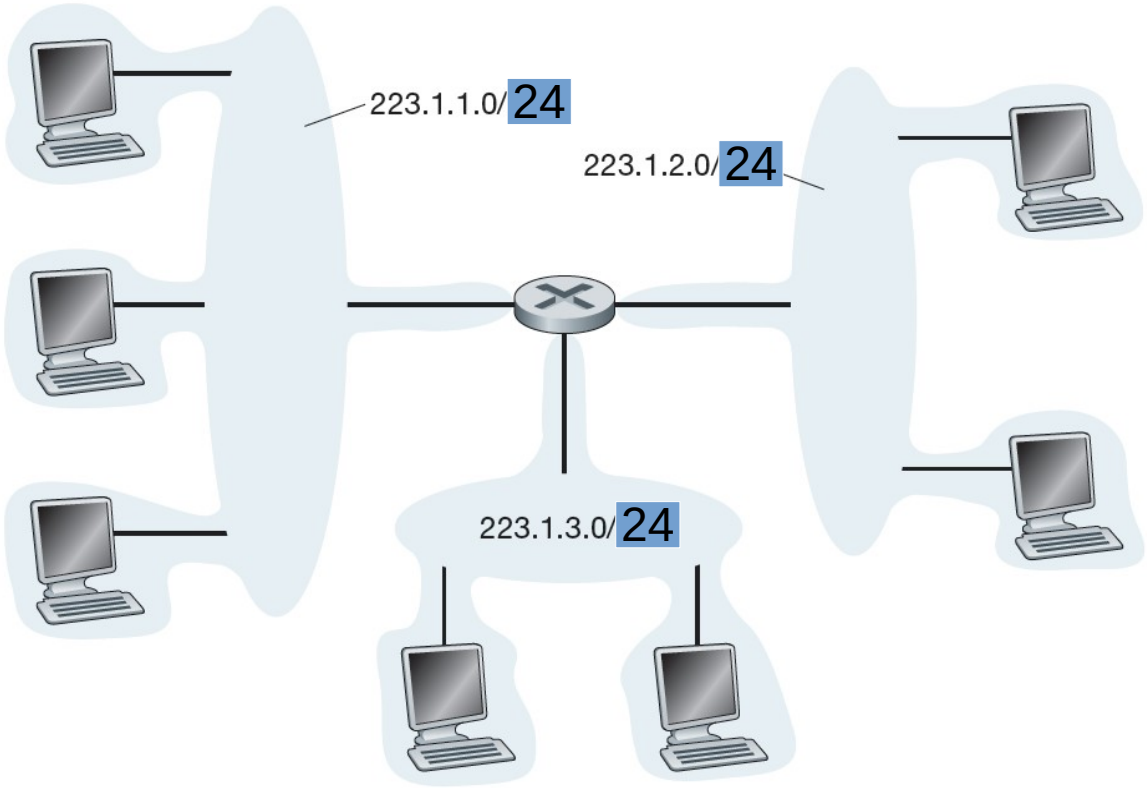
Endereçamento IPv4

- Endereços de interfaces



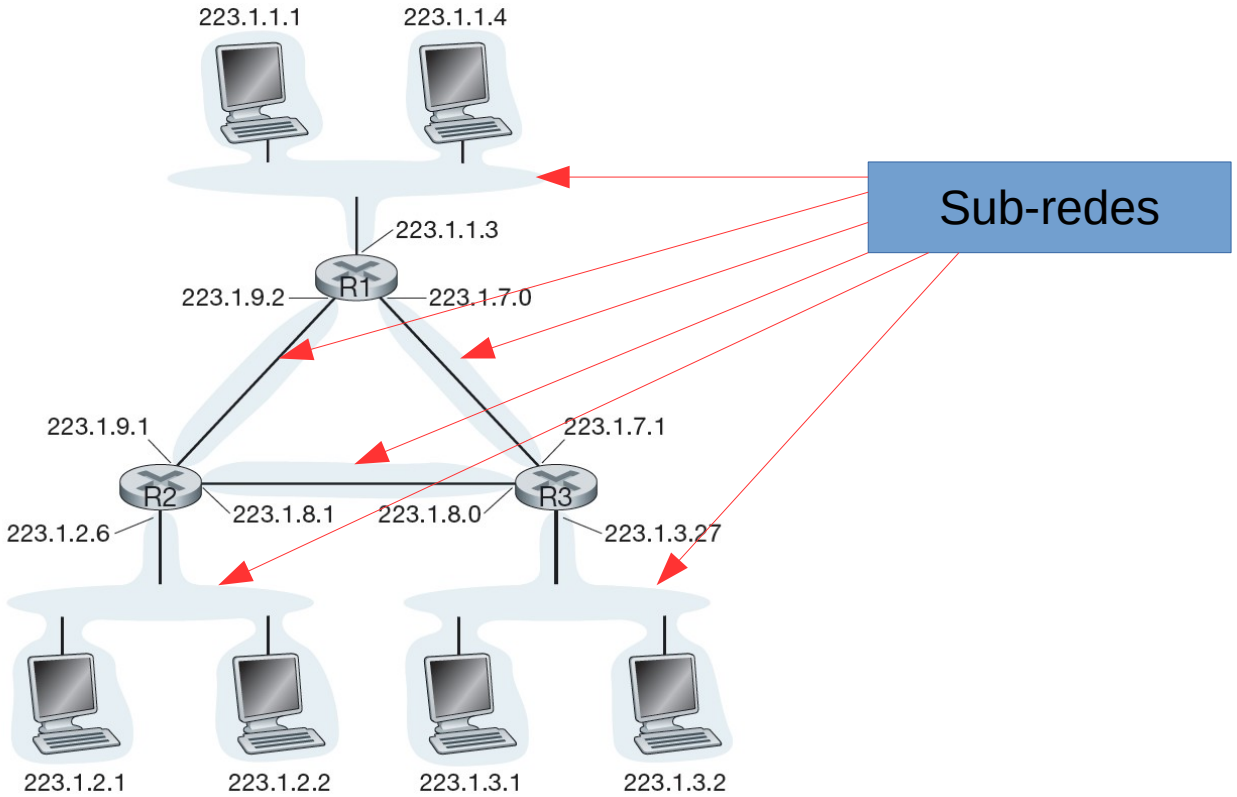
Endereçamento IPv4

- Endereços de sub-redes



Endereçamento IPv4

- Três roteadores interconectando. Quantas sub-redes?

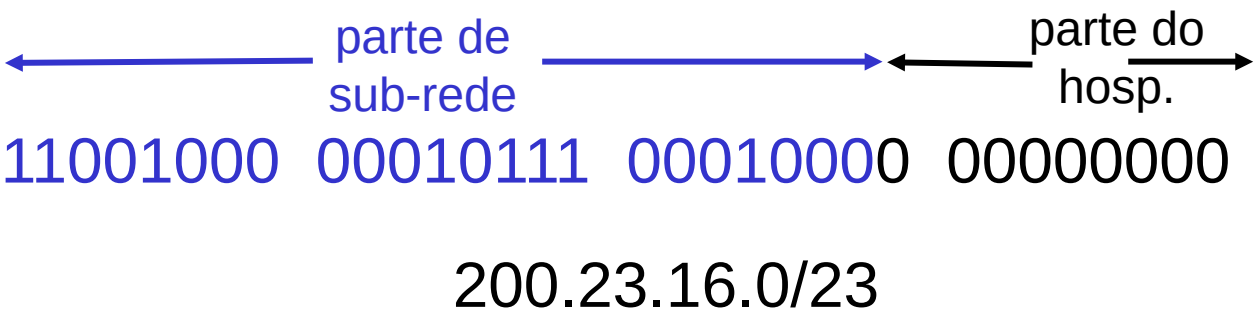


- Imunes….

Endereçamento IP: CIDR

CIDR: Classless InterDomain Routing (roteamento interdomínio sem classes)

- parte de sub-rede do endereço de tamanho arbitrário
- formato do endereço: a.b.c.d/x, onde x é # bits na parte de sub-rede do endereço



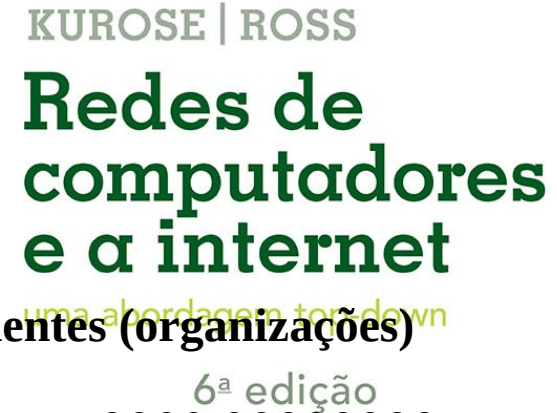
Obtenção de um bloco de endereços

- Para obter um bloco de endereços IP para utilizar dentro da sub-rede de uma organização, um administrador de rede poderia:

1. contatar seu ISP, que forneceria endereços a partir de um bloco maior de endereços que já estão alocados ao ISP.

2. O ISP, por sua vez, dividiria seu bloco de endereços em oito blocos de endereços contíguos, do mesmo tamanho, e daria um deles a cada uma de um conjunto de oito organizações suportadas por ele (veja figura a seguir):

Obtenção de um bloco de endereços

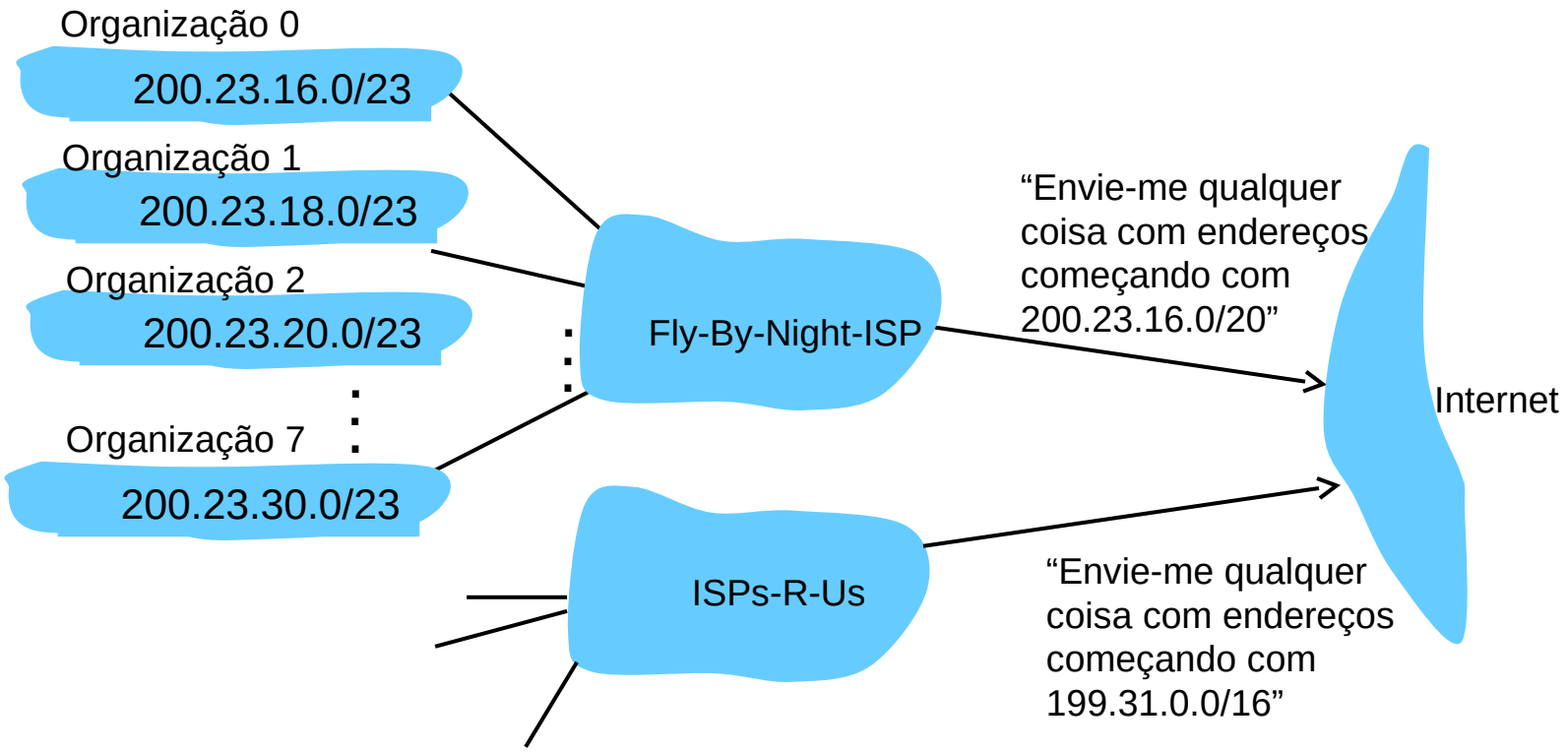


Um ISP com um bloco de endereços grande divide com 8 clientes (organizações)

• Bloco ISP:	200.23.16.0	11001000.00010111.0001		0000.00000000
• /20		11111111.11111111.1111		0000.00000000
• Cliente 0:	200.23.16.0	11001000.00010111.0001	000	0.00000000
• /23		11111111.11111111.1111	111	0.00000000
• Cliente 1:	200.23.18.0	11001000.00010111.0001	001	0.00000000
• /23		11111111.11111111.1111	111	0.00000000
• Cliente 2:	200.23.20.0	11001000.00010111.0001	010	0.00000000
• /23		11111111.11111111.1111	111	0.00000000
• Cliente 3:	200.23.22.0	11001000.00010111.0001	011	0.00000000
• /23		11111111.11111111.1111	111	0.00000000
• Cliente 4:	200.23.24.0	11001000.00010111.0001	100	0.00000000
• /23		11111111.11111111.1111	111	0.00000000
• Cliente 5:	200.23.26.0	11001000.00010111.0001	101	0.00000000
• /23		11111111.11111111.1111	111	0.00000000
• Cliente 6:	200.23.28.0	11001000.00010111.0001	110	0.00000000
• /23		11111111.11111111.1111	111	0.00000000
• Cliente 7:	200.23.30.0	11001000.00010111.0001	111	0.00000000
• /23		11111111.11111111.1111	111	0.00000000

Endereçamento hierárquico: agregação de rota

Endereçamento hierárquico permite anúncio eficiente da informação de roteamento:



Endereçamento IP: a última palavra...

P: Como um ISP recebe bloco de endereços?

R: ICANN: Internet Corporation for Assigned Names and Numbers

- aloca endereços
- administra o DNS
- atribui nomes de domínio e resolve disputas
- https://en.wikipedia.org/wiki/IPv4_address_exhaustion
- http://ipv6.br/
- https://www.nic.br/noticia/na-midia/como-funciona-a-internet/
- Buscar em Google por: faixa de IP Brasil... privado etc

Endereçamento IP: Endereços privados

Bloco CIDR	Número de IPs	Faixa de endereços IP
10.0.0.0/8	16777216	10.0.0.0 – 10.255.255.255
172.16.0.0/12	1048576	172.16.0.0 – 172.31.255.255
192.168.0.0/16	65536	192.168.0.0 – 192.168.255.255
169.254.0.0/16	65536	169.254.0.0 – 169.254.255.255

Os demais endereços são públicos, ou seja, roteáveis.

Cálculo de sub-redes

- O objetivo é determinar quais endereços IPs estarão disponíveis para alocar nos hosts (interfaces de rede)
- Cada host deve ter um endereço IP único e exclusivo
- A máscara de rede define o tamanho do bloco, sempre em potenciação de 2: 2^1, 2^2 (4), 2^3 (8)….
- O número da máscara é obtido pela subtração entre o total de bits do endereço (32) e o a potenciação para o tamanho de bloco desejado, por exemplo:
  1) /30 é a máscara de rede de um bloco de tamanho 4 (32 – 30 = 2, 2^2 = 4)
  2) /24 é a máscara de rede de um bloco de tamanho 256 (32 – 24 = 8, 2^8 = 256)
- Por regra em cada bloco de endereços **sempre**:
  - O primeiro endereço é o endereço de rede (da organização)
  - O último é o endereço de *broadcast*. Endereço normatizado que serve para encaminhar mensagens para todos os hosts do bloco.
  - Os demais endereços do bloco ficarão disponíveis para os hosts.
- Portanto, a menor sub-rede útil (2 endereços para hosts) é a que possui um bloco de 4 endereços:
  1) Endereço de rede
  2) Endereço do primeiro host
  3) Endereço do último host e
  4) Endereço de *broadcast*

Exercício

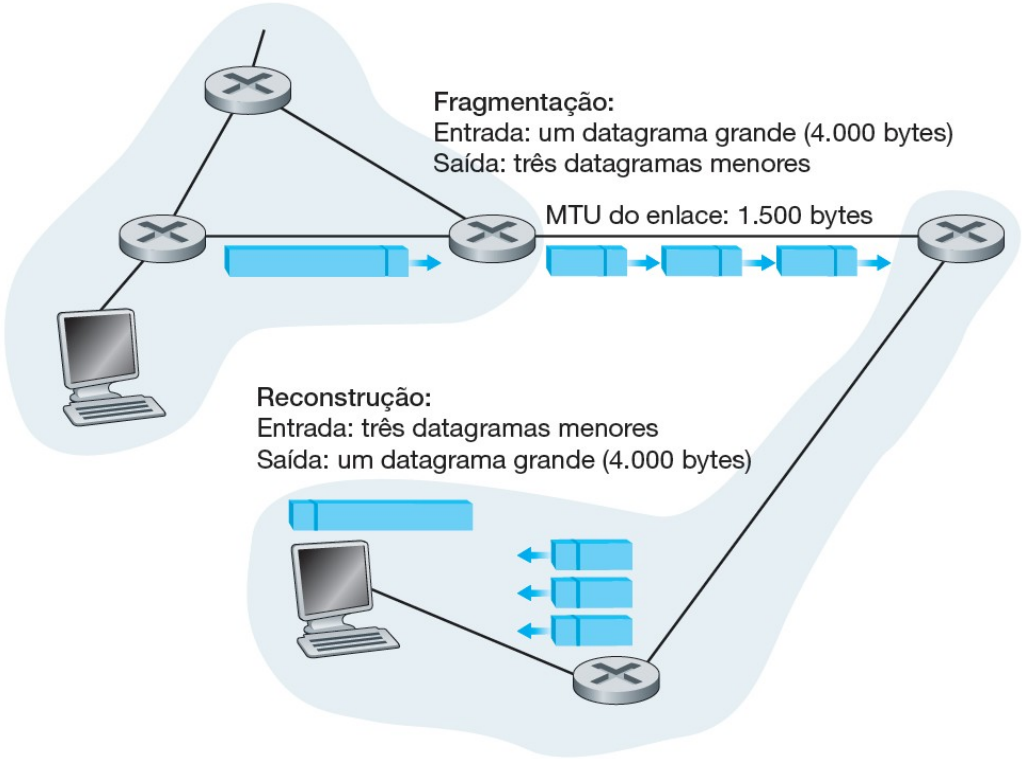
- Um provedor de acesso a Internet possui o seguinte bloco de endereços IP: 12.17.192.0/18. Os endereços IP de 12.17.192.0 até 12.17.207.255 já estão alocados para clientes deste provedor. Este provedor precisa atender a quatro novos clientes, um deles necessita de rede com pelo menos 1000 endereços de IP válidos e os outros três necessitam de rede com pelo menos 30 IP válidos. Faça uma proposta para alocação de endereços IP para estes clientes. Os endereços reservados para estes novos clientes devem ser alocados a partir da numeração mais baixa disponível. Procure também responder as questões abaixo:

  1) Qual o total de endereços que dispõe o provedor em questão?
  2) Qual o endereço identificador de cada sub-rede dos novos clientes?
  3) Qual o a máscara de rede de cada sub-rede dos novos clientes?
  4) Qual o endereço do primeiro e último *host* da sub-rede?
  5) Qual o endereço de *broadcast* de cada sub-rede dos novos clientes?
  6) Quantos endereços IP já estão utilizados?
  7) Quantos endereços IP restarão livres para alocar a novos clientes?

Fragmentação do datagrama IP

Nem todos protocolos da camada de enlace suportam pacotes de mesmo tamanho (MTU – *Maximum Transmission Unit*)

- Fragmentação e reconstrução IP



Fragmentação do datagrama IP

- Fragmentos IP

Fragmento	Bytes	ID	Deslocamento	*Flag*
1º fragmento	1.480 bytes no campo de dados do datagrama IP	identificação = 777	0 (o que significa que os dados devem ser inseridos a partir do byte 0)	1 (o que significa que há mais)
2º fragmento	1.480 bytes de dados	identificação = 777	185 (o que significa que os dados devem ser inseridos a partir do byte 1.480. Note que 185 x 8 = 1.480)	1 (o que significa que há mais)
3º fragmento	1.020 bytes de dados (= 3.980 –1.480 –1.480)	identificação = 777	370 (o que significa que os dados devem ser inseridos a partir do byte 2.960. Note que 370 x 8 = 2.960)	0 (o que significa que esse é o último fragmento)

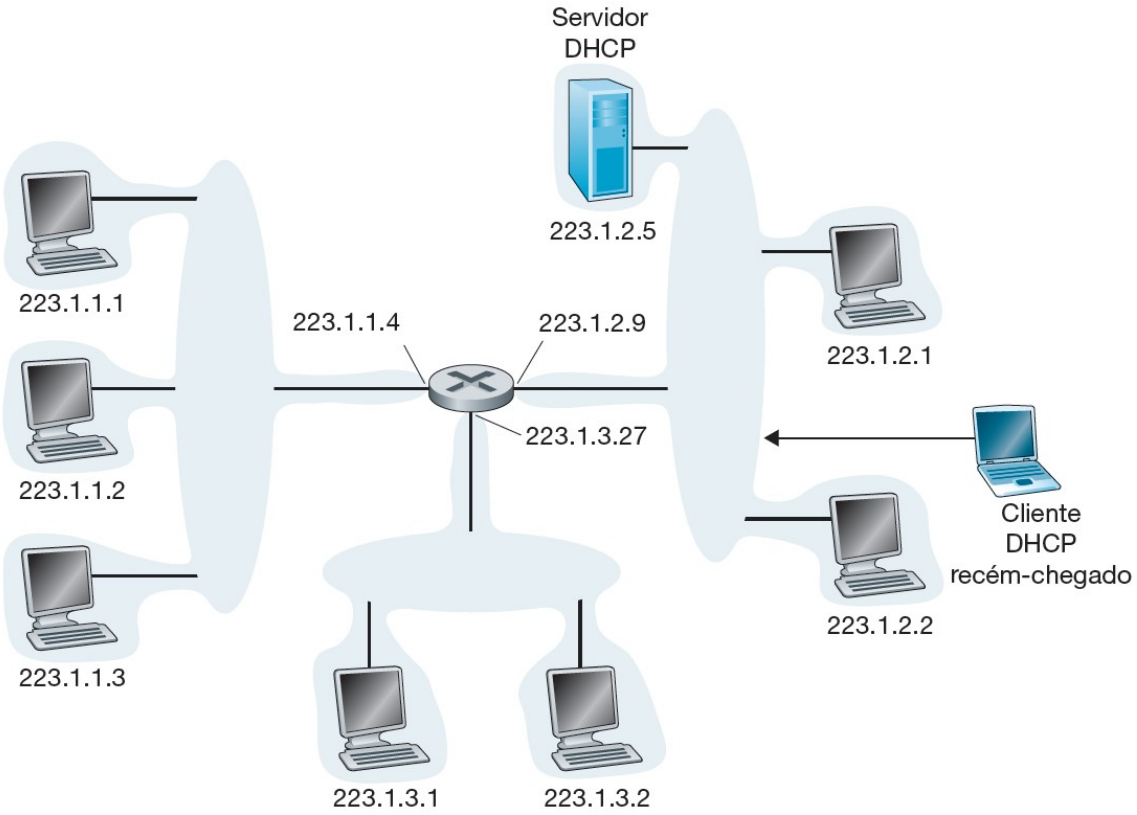
Obtenção de um endereço de hospedeiro: o Protocolo de Configuração Dinâmica de Hospedeiros (DHCP)

- O DHCP permite que um hospedeiro obtenha (seja alocado a) um endereço IP de maneira automática.
- O DHCP é em geral denominado um **protocolo *plug and play***.
- O protocolo DHCP é um processo de quatro etapas:

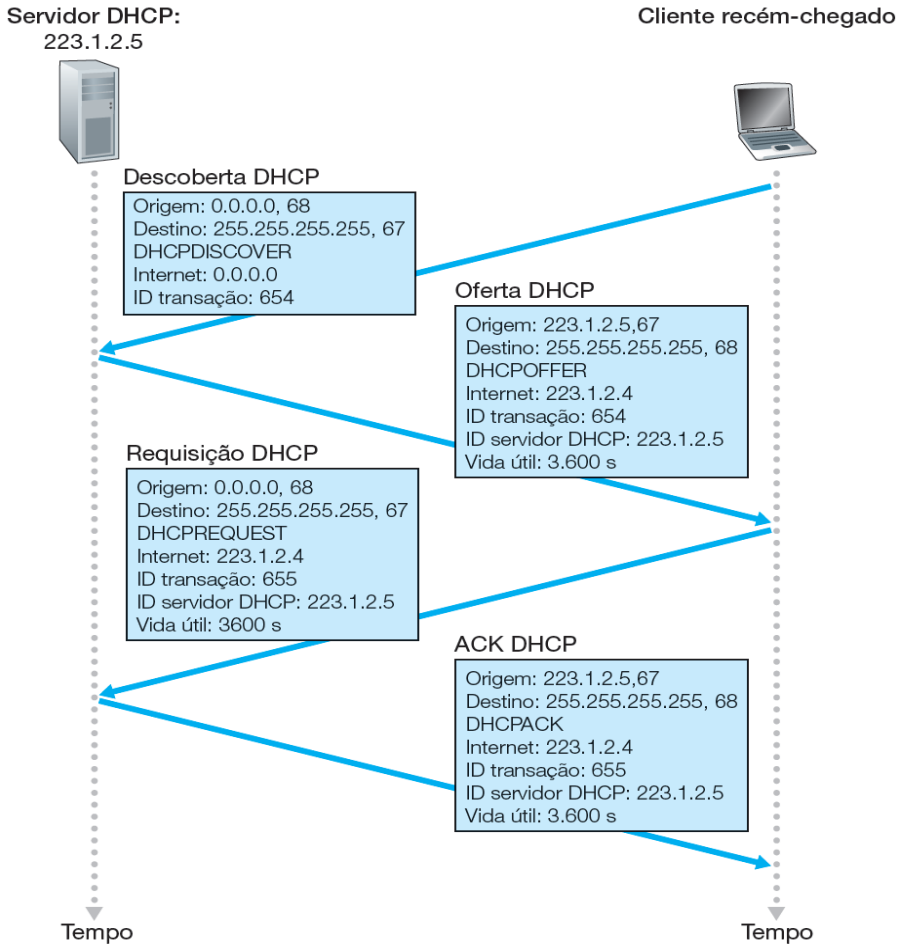
1. Descoberta do servidor DHCP.
2. Oferta(s) do(s) servidore(s) DHCP.
3. Solicitação DHCP.
4. DHCP ACK.

Obtenção de um endereço de hospedeiro: o Protocolo de Configuração Dinâmica de Hospedeiros (DHCP)

- Cenário cliente-servidor DHCP



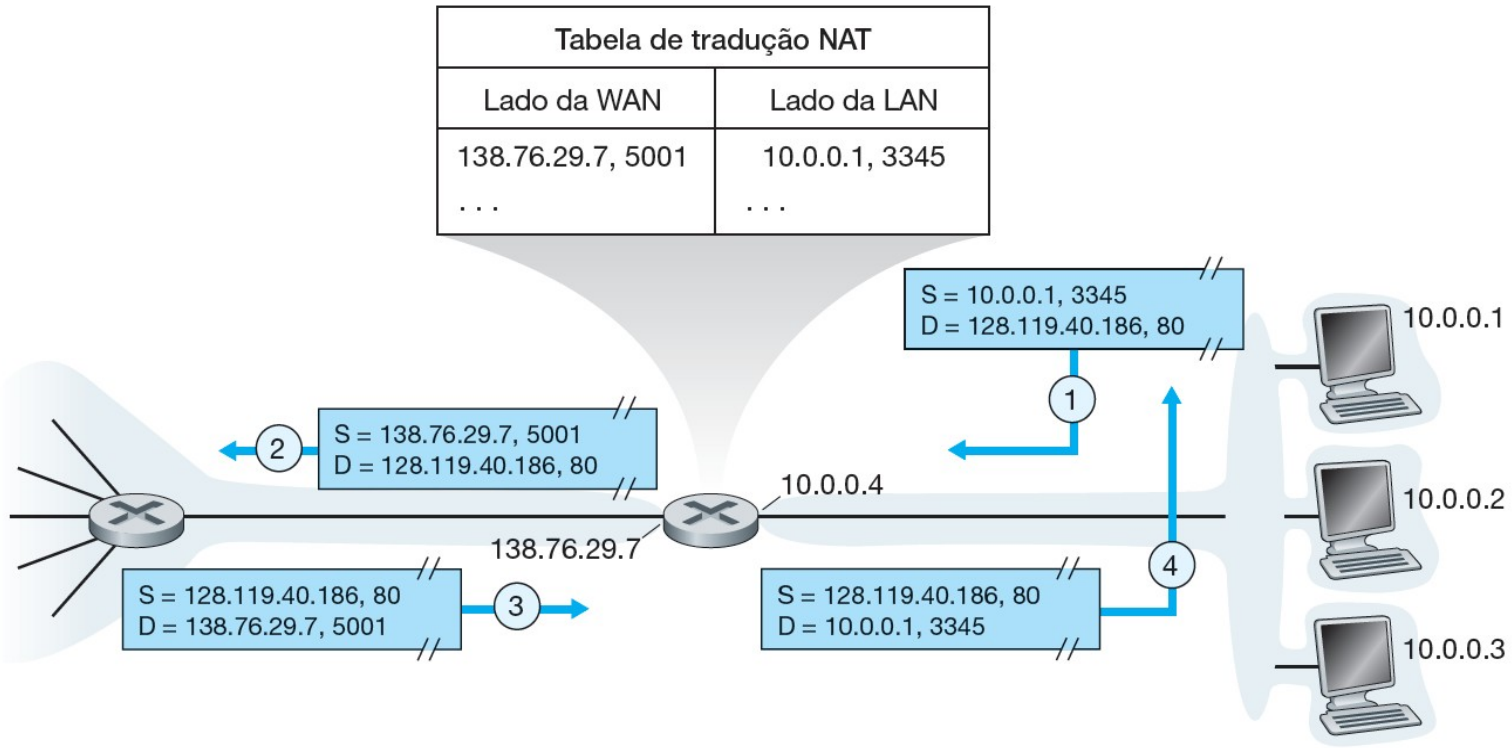
Obtenção de um endereço de hospedeiro: o Protocolo de Configuração Dinâmica de Hospedeiros (DHCP)



- Interação cliente-servidor DHCP (dhclient -v eth0:0)
- Wireshark Filter: udp.port==68 or udp.port==67

Tradução de endereços na rede (NAT)

- Tradução de endereços de rede (S = Origem, D = Destino)



Tradução de endereços na rede (NAT)

Campo de número de porta de 16 bits:

- 60.000 conexões simultâneas com um único endereço no lado da LAN!

NAT é controvertido:

- roteadores só devem processar até a camada 3
- viola argumento de fim a fim
  - a possibilidade de NAT deve ser levada em conta pelos projetistas da aplicação, p. e., aplicações P2P
- a falta de endereços deverá ser resolvida pelo IPv6

Protocolo de Mensagens de Controle da Internet (ICMP)

- O ICMP é usado por hospedeiros e roteadores para comunicar informações de camada de rede entre si.
- A utilização mais comum do ICMP é para comunicação de erros.
- Mensagens ICMP têm um campo de tipo e um campo de código.
- O conhecido programa ping envia uma mensagem ICMP do tipo 8 código 0 para o hospedeiro especificado.
- Alguns tipos de mensagens ICMP selecionadas são mostrados a seguir.

Protocolo de Mensagens de Controle da Internet (ICMP)

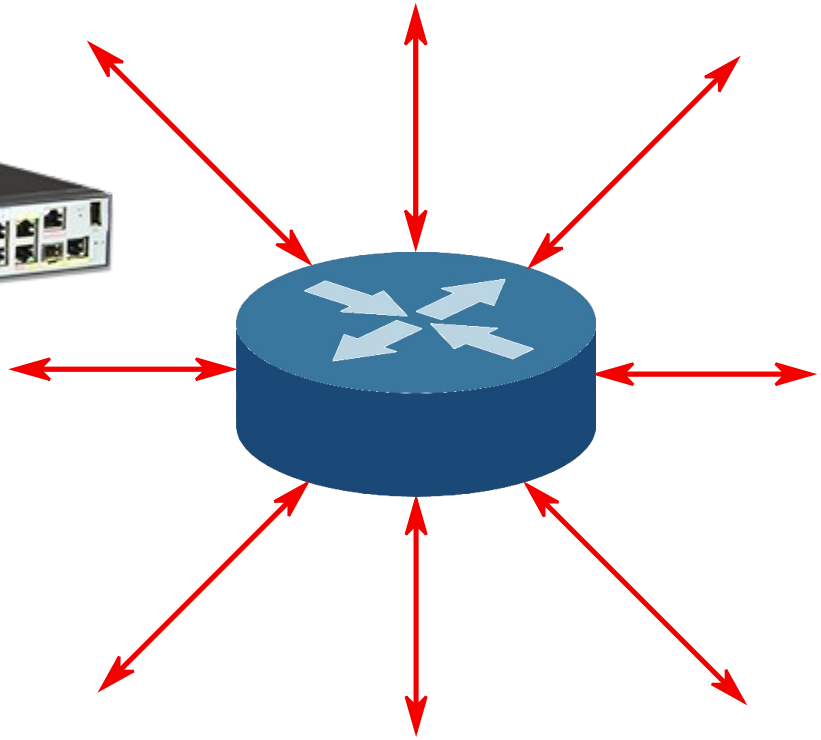
- Tipos de mensagens ICMP

Tipo ICMP	Código	Descrição
0	0	resposta de eco (para *ping*)
3	0	rede de destino inalcançável
3	1	hospedeiro de destino inalcançável
3	2	protocolo de destino inalcançável
3	3	porta de destino inalcançável
3	6	rede de destino desconhecida
3	7	hospedeiro de destino desconhecido
4	0	repressão da origem (controle de congestionamento)
8	0	solicitação de eco
9	0	anúncio do roteador
10	0	descoberta do roteador
11	0	TTL expirado
12	0	cabeçalho IP inválido

- Ping e wireshark com filtro icmp. Ping -t 1....

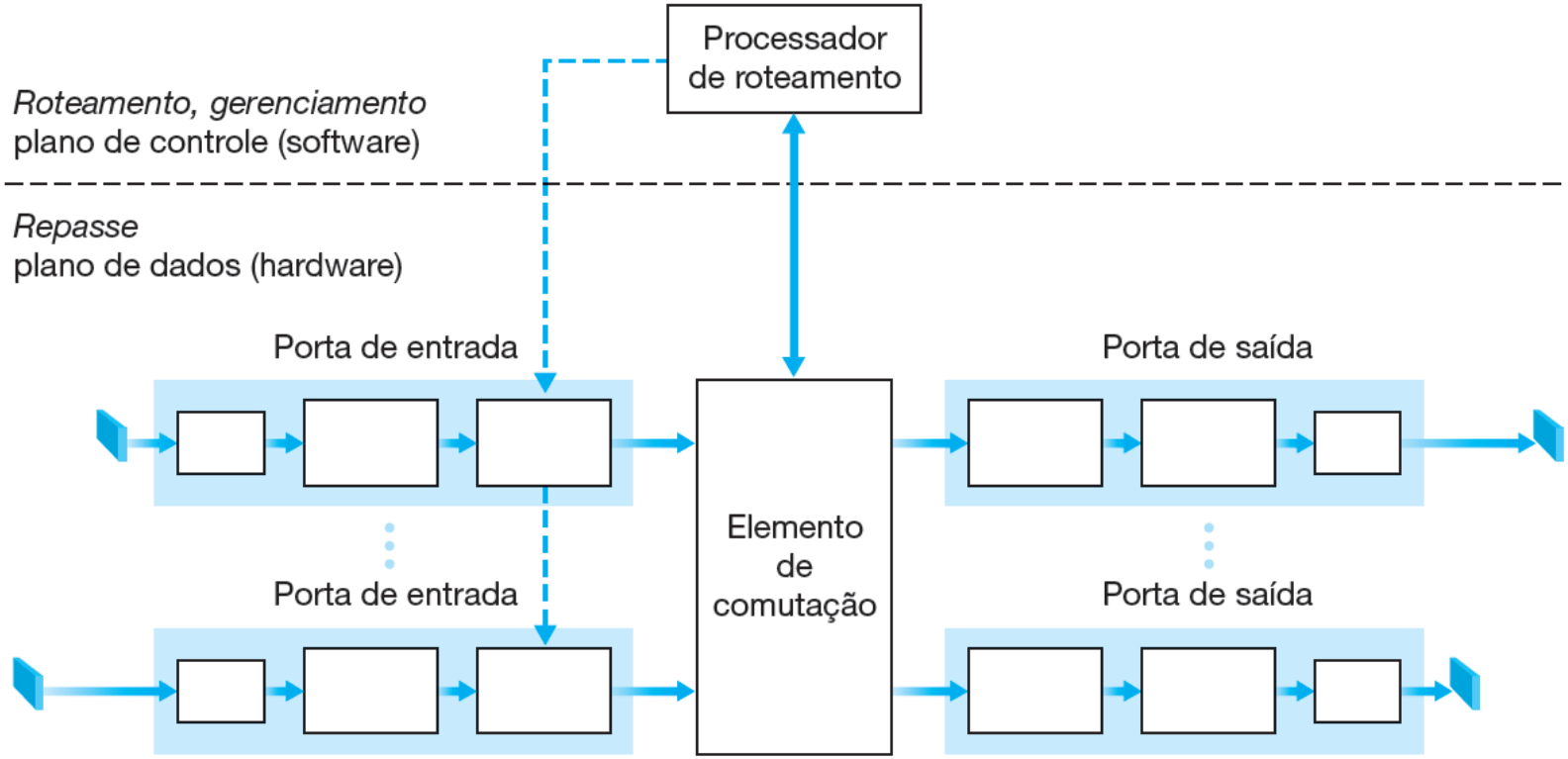
O que é um roteador?

- Equipamento que possui N portas de entrada e saída
- **Roteamento:** determina a rota ou o caminho tomado pelos pacotes ao fluírem de um remetente a um destinatário.
- **Repasse:** determina a interface de saída de determinado pacote.



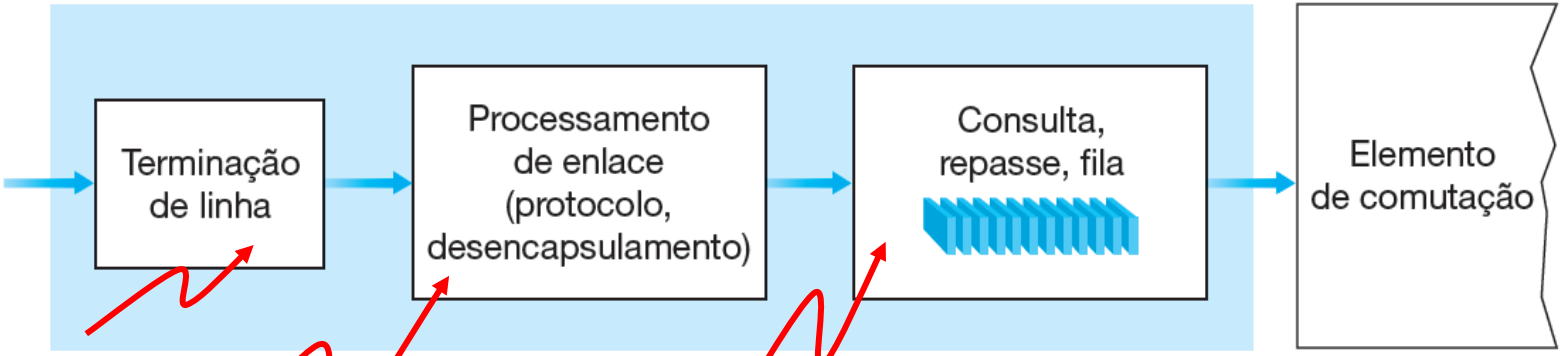
O que há dentro de um roteador?

- Arquitetura de roteador
- Processador de roteamento em *software*: ~milissegundos
- Repasse em *hardware*: ~nanosegundos.



Processamento de entrada

- Processamento na porta de entrada



Camada física:
recepção por bit

Camada de enlace de dados:
p. e., Ethernet
ver Capítulo 5

Comutação descentralizada:

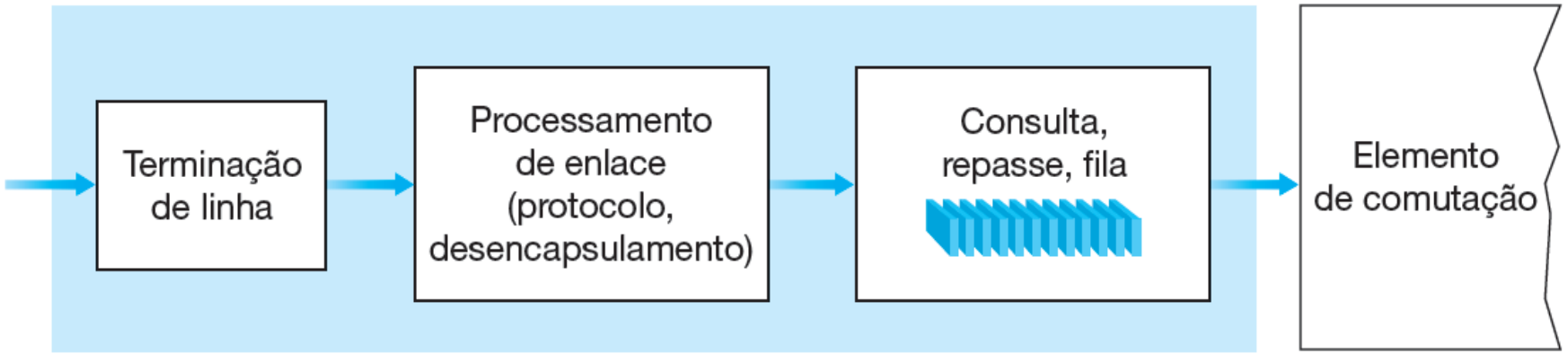
dado destino do datagrama, determina a porta de saída pesquisando a tabela de repasse na memória da porta de entrada

objetivo: processamento completo da porta de entrada na 'velocidade de linha'

fila: se datagramas chegarem mais rápido que taxa de repasse no elemento de comutação

Processamento de entrada

- Processamento na porta de entrada



- Ex: Enlace de 10 Gbits/s, datagrama de 64 bytes ==> 51,2 ns para processar o datagrama antes da chegada do seguinte.
- Se N portas forem combinadas em uma placa de linha (prática) ==> deve ser N vezes mais rápido, o que é muito para software, portanto é feito em hardware.

Elemento de comutação

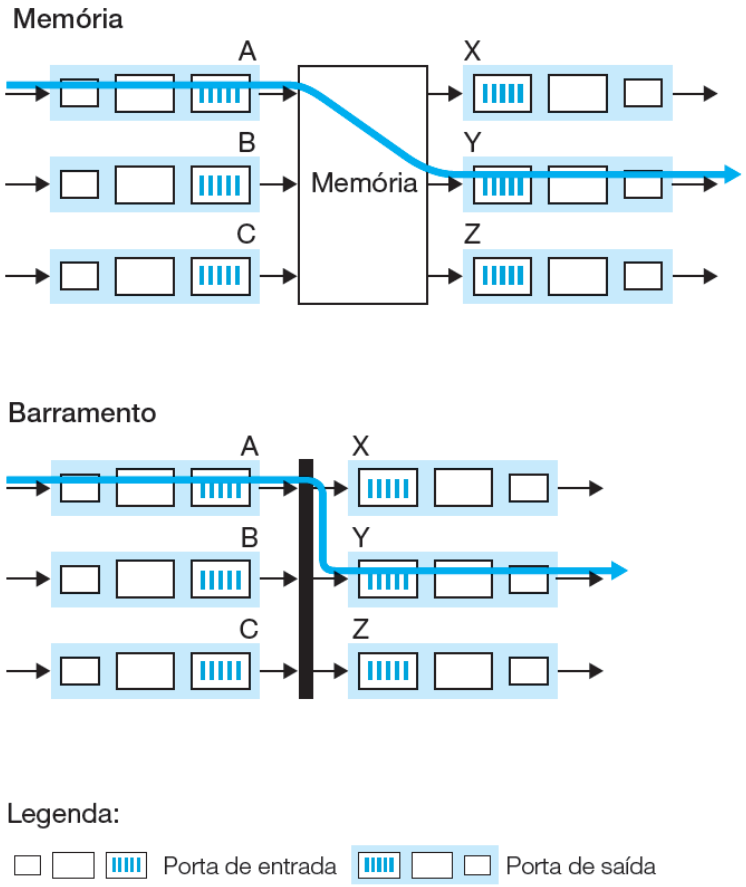
É por meio do elemento de comutação que os pacotes são comutados de uma porta de entrada para uma porta de saída.

A comutação pode ser realizada de inúmeras maneiras:

- Comutação por memória. Única operação por vez, seja entrada ou saída.
- Comutação por um barramento. Um único pacote atravessando o barramento por vez.
- Comutação por uma rede de interconexão. Vários pacotes em paralelo desde que o destino não seja a mesma porta de saída.

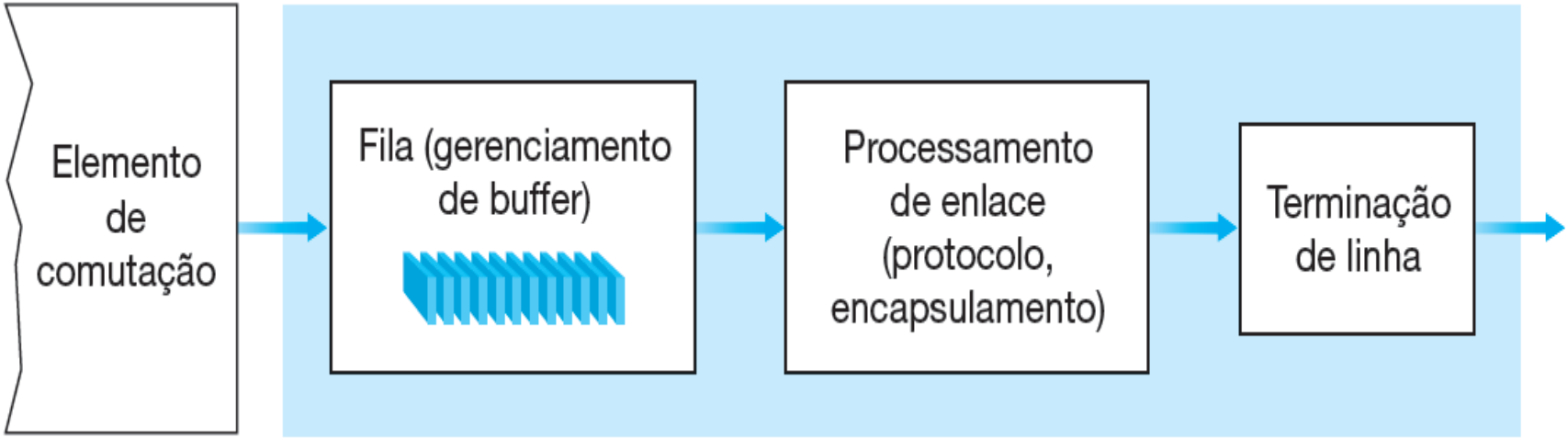
Aumento da velocidade, complexidade e custo.

Elemento de comutação



Processamento de saída

- Processamento de porta de saída



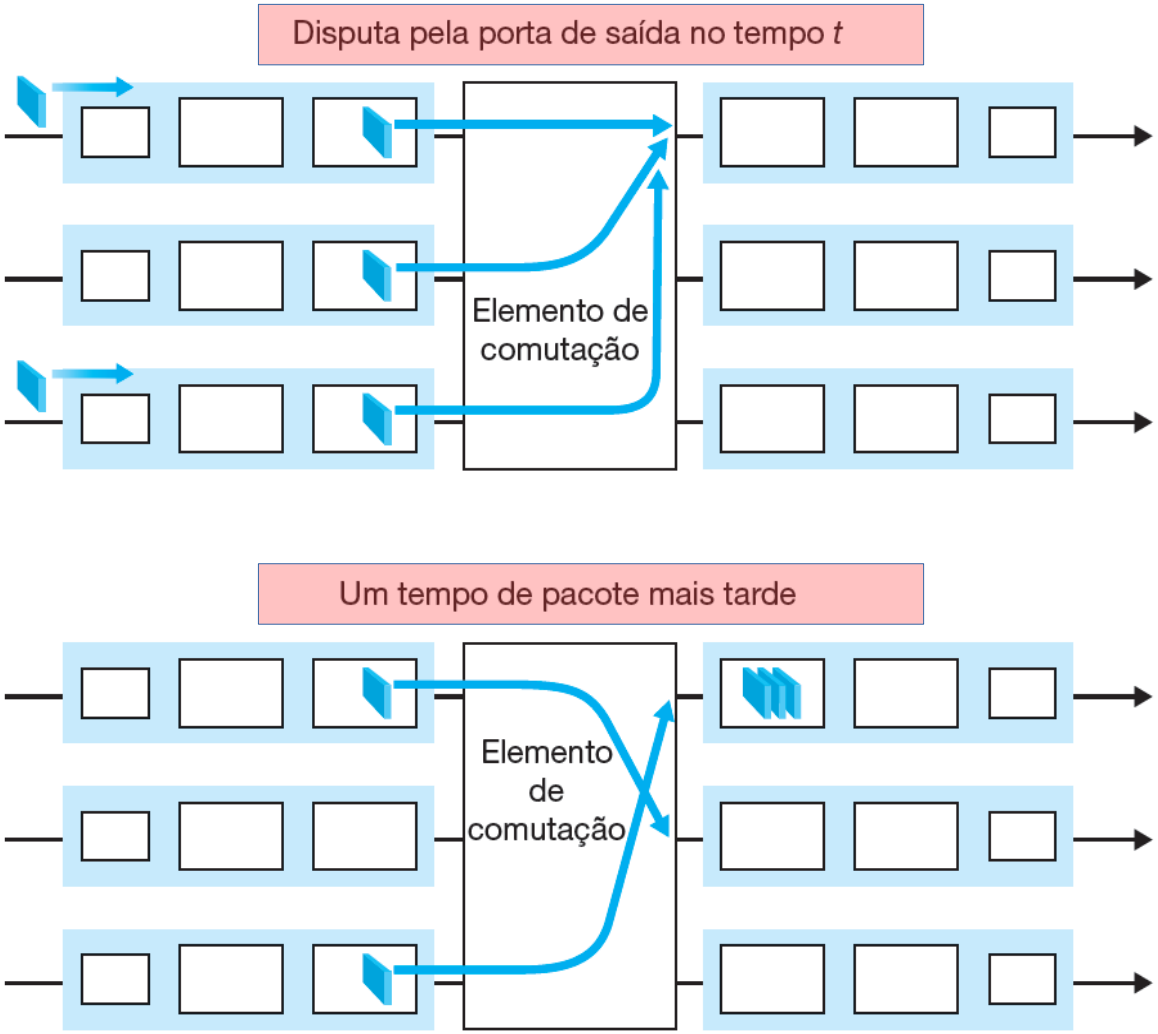
Onde ocorre formação de fila?

Filas de pacotes podem se formar tanto nas portas de entrada como nas de saída.

O local e a extensão da formação de fila dependerão:

- da carga de tráfego,
- da velocidade relativa do elemento de comutação e
- da taxa da linha.

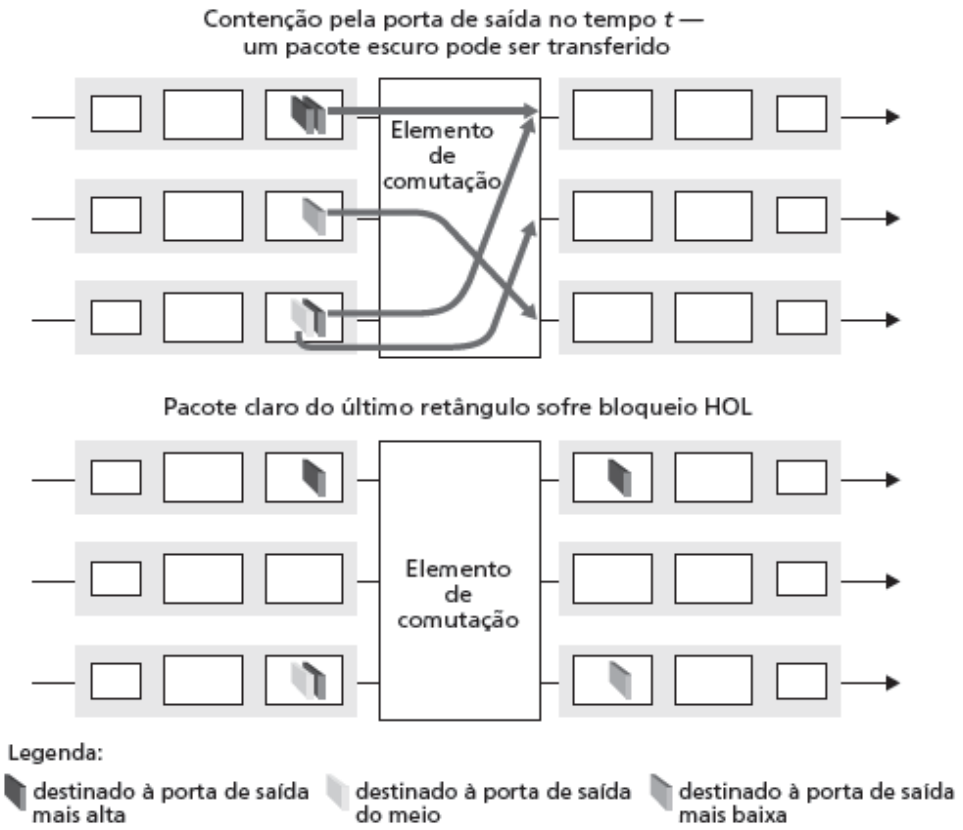
Onde ocorre formação de fila?



- Formação de fila na porta de saída
- Ao formar fila -- **escalonador de pacotes: FCFS** (*First-come, first-served*) **ou WFQ** (*Weighted fair queueing*) ==> QoS

Enfileiramento da porta de entrada

- elemento de comutação mais lento que portas de entrada combinadas -> enfileiramento possível nas filas de entrada
- bloqueio de cabeça de fila (HOL - *Head Of Line blocking*) : datagrama enfileirado na frente da fila impede que outros na fila sigam adiante
- *atraso de enfileiramento e perda devidos a estouro no buffer de entrada*



Quanto armazenamento em *buffer*?

- regra prática da RFC 3439: armazenamento médio em buffer igual à RTT “típica” (digamos, 250 ms) vezes capacidade do enlace C
  - p. e., C = enlace de 10 Gbps: *buffer* de **2,5 Gbit**
- recomendação recente: com *N* fluxos, armazenamento deve ser igual a

$$\frac{N \cdot RTT \cdot C}{\sqrt{N}}$$

  - p. e., C = 10 enlaces de 10 Gbps: *buffer* de **7,9 Gbit**

Algoritmos de roteamento

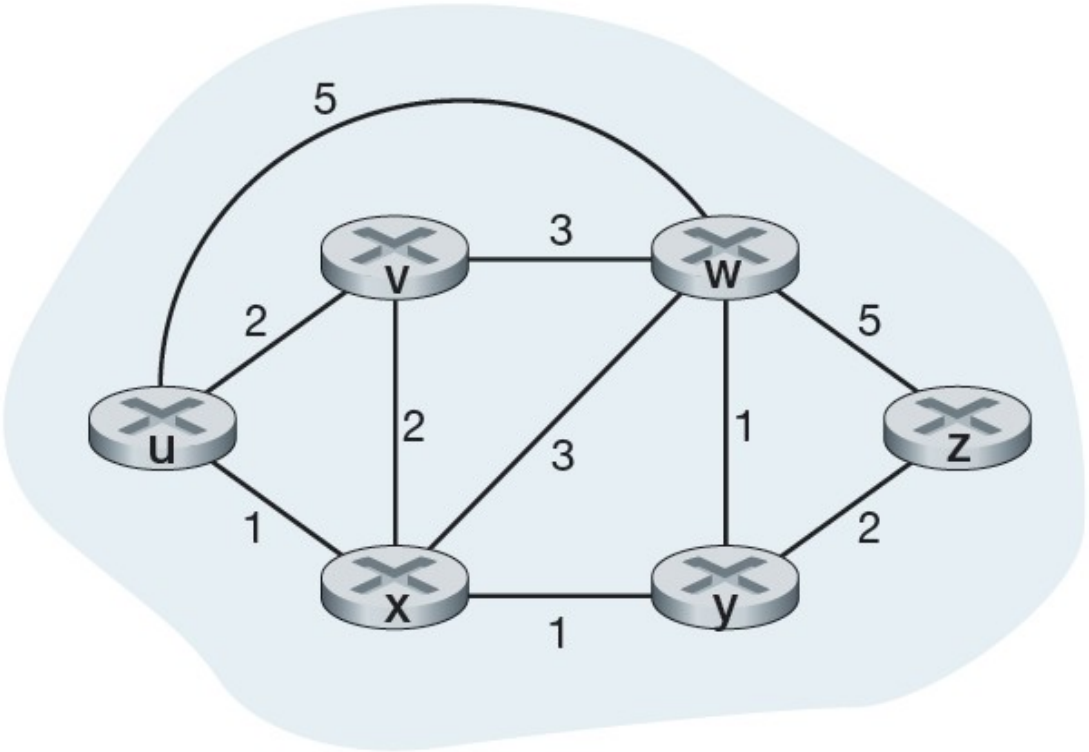
- Em geral um hospedeiro está ligado diretamente a um roteador, o **roteador *default*** para esse hospedeiro.

- Denominamos **roteador de origem** o roteador *default* do hospedeiro de origem e **roteador de destino** o roteador *default* do hospedeiro de destino.

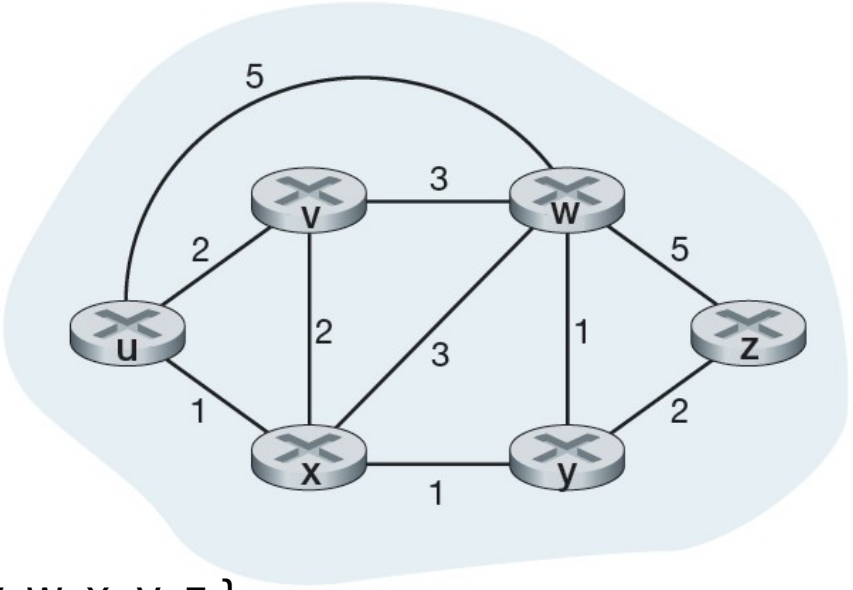
- O problema de rotear um pacote do hospedeiro de origem até o hospedeiro de destino se reduz, claramente, ao problema de direcionar o pacote do roteador de origem ao roteador de destino.

Algoritmos de roteamento

- Um grafo é usado para formular problemas de roteamento.
- Qual o melhor caminho entre u e z?
- Quantos possíveis caminhos entre u e $z_{(17)}$?



Abstração de grafo



Grafo: G = (N,E)

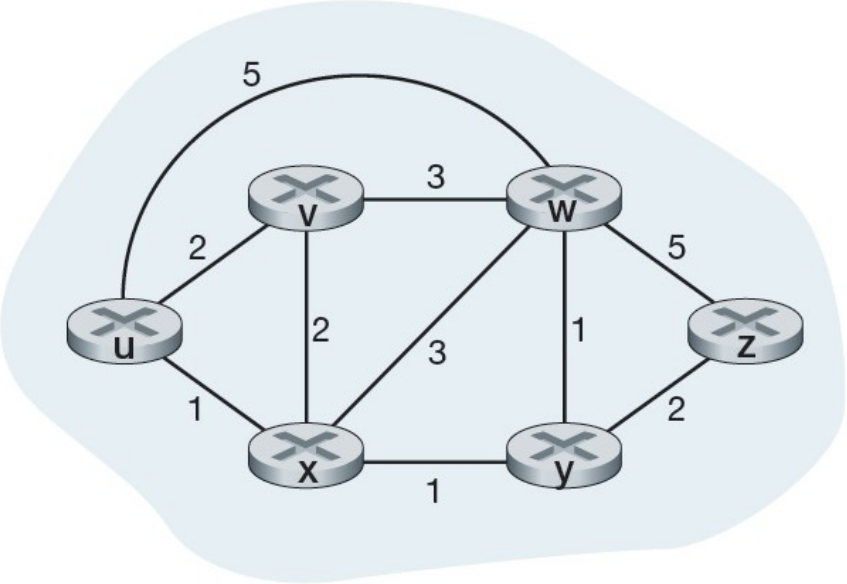
N = conjunto de roteadores = { u, v, w, x, y, z }

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



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

  - p. e., c(w,z) = 5

- custo poderia ser sempre 1, ou inversamente relacionado à largura de banda ou inversamente relacionado ao congestionamento

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

Pergunta: Qual é o caminho de menor custo entre u e z?

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

Algoritmos de roteamento

- Um **algoritmo de roteamento global** calcula o caminho de menor custo entre uma origem e um destino usando conhecimento completo e global sobre a rede. **Algoritmos de estado de enlace** (*link state* – **LS, OSPF e IS IS**). (ser humano olhando a figura como um todo)

- Em um **algoritmo de roteamento descentralizado**, o cálculo do caminho de menor custo é realizado de modo iterativo e distribuído. **Algoritmo de vetor de distâncias** (*distance-vector* – **DV, RIP**).

Algoritmos de roteamento - Taxionomia

- Em **algoritmos de roteamento estáticos**, as rotas mudam muito devagar ao longo do tempo, muitas vezes como resultado de intervenção humana. Fazer exemplo de rotas estáticas.

Rede	Roteador
200.10.20.1/30	eth0

- **Algoritmos de roteamento dinâmicos** mudam os caminhos de roteamento à medida que mudam as cargas de tráfego ou a topologia da rede.

- **Algoritmos in/sensível à carga. Sensível**: custos de enlace variam dinamicamente para refletir o nível corrente de congestionamento no enlace subjacente.

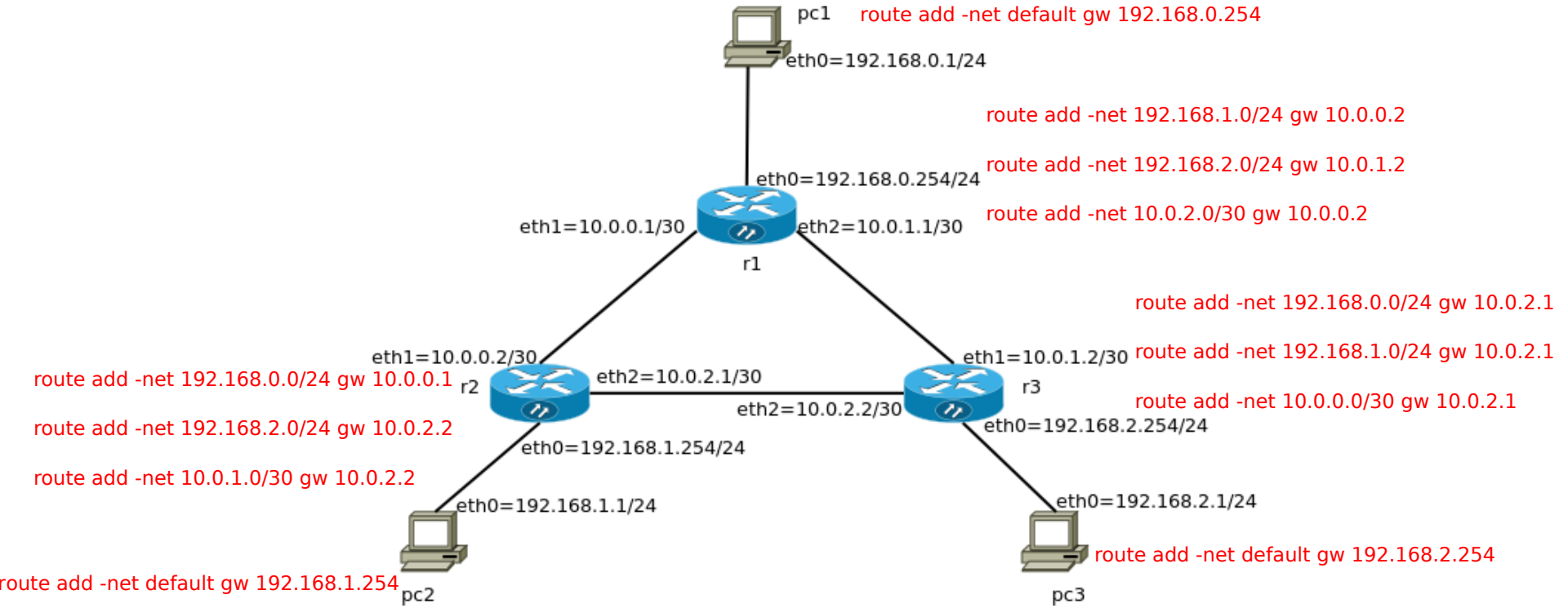
Analisando tabela de roteamento real

- root@r1# route -n

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
10.0.0.0	0.0.0.0	255.255.255.252	U	0	0	0	eth1
10.0.1.0	0.0.0.0	255.255.255.252	U	0	0	0	eth2
10.0.2.0	10.0.0.2	255.255.255.252	UG	0	0	0	eth1
192.168.0.0	0.0.0.0	255.255.255.0	U	0	0	0	eth0
192.168.1.0	10.0.0.2	255.255.255.0	UG	0	0	0	eth1
192.168.2.0	10.0.1.2	255.255.255.0	UG	0	0	0	eth2

Interpretação

- Gateway = 0.0.0.0 e sem *flag* G ==> entrega direta
- Gateway explícito e *flag* G ==> entrega direta para a interface explicíta de um roteador vizinho
- Iface = interface de saída



Algoritmo de roteamento de estado do enlace (LS ==> OSPF, IS IS)

algoritmo de Dijkstra - LS

- nova topologia, custos de enlace conhecidos de todos os nós (conhecimento global)
  - 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
- iterativo: após k iterações, sabe caminho de menor custo para k destinos
- *Open Shortest Path First* (**OSPF**) e *Intermediate System to Intermediate System* (**IS IS**)

notação:

c(x,y): custo do enlace do nó x até y; Inicialmente = ∞ se não forem vizinhos diretos

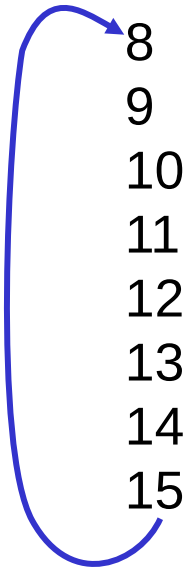
D(v): valor atual do custo do caminho da origem ao destino v

p(v): nó predecessor ao longo do caminho da origem até v

N': conjunto de nós cujo caminho de menor custo é definitivamente conhecido

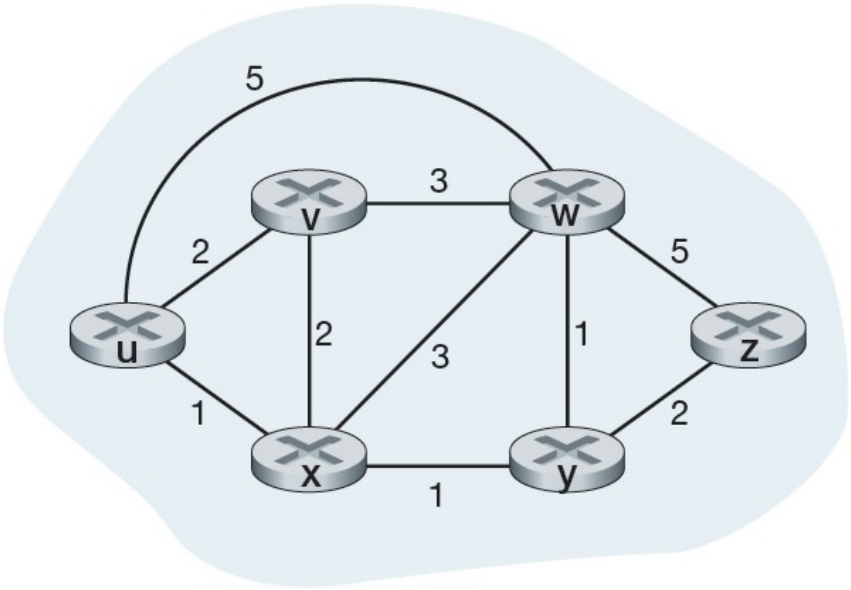
Algoritmo de Dijkstra

```
1  Inicialização:
2    N' = {u}   #origem
3    para todos os nós v
4      se v adjacente a u
5          então D(v) = c(u,v)
6      senão D(v) = ∞
7
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' :
12       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'
```



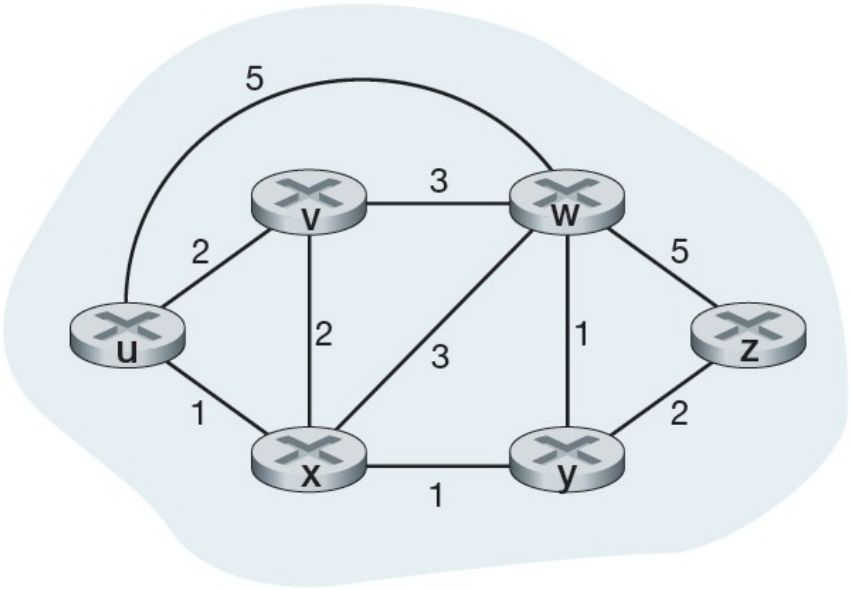
Algoritmo de Dijkstra: exemplo a partir de **u**

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



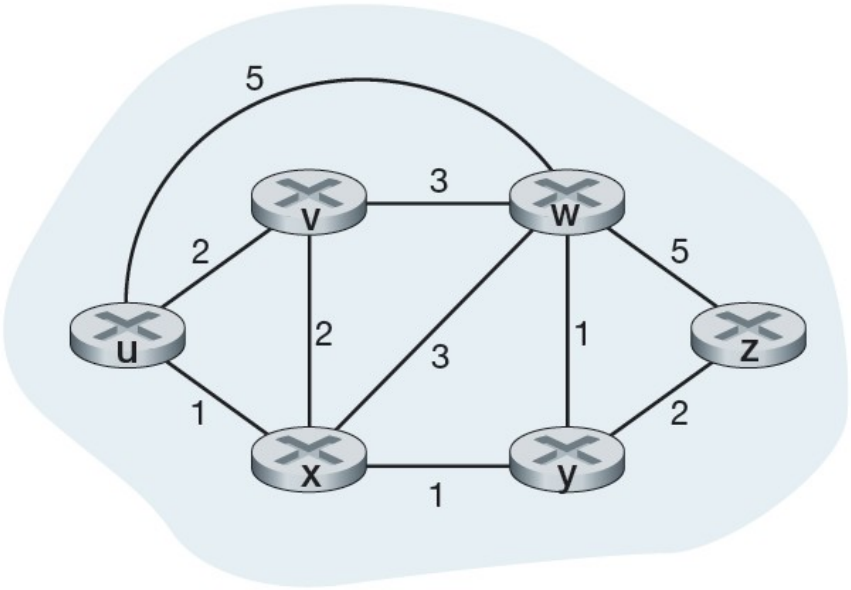
Algoritmo de Dijkstra: exemplo a partir de **u**

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



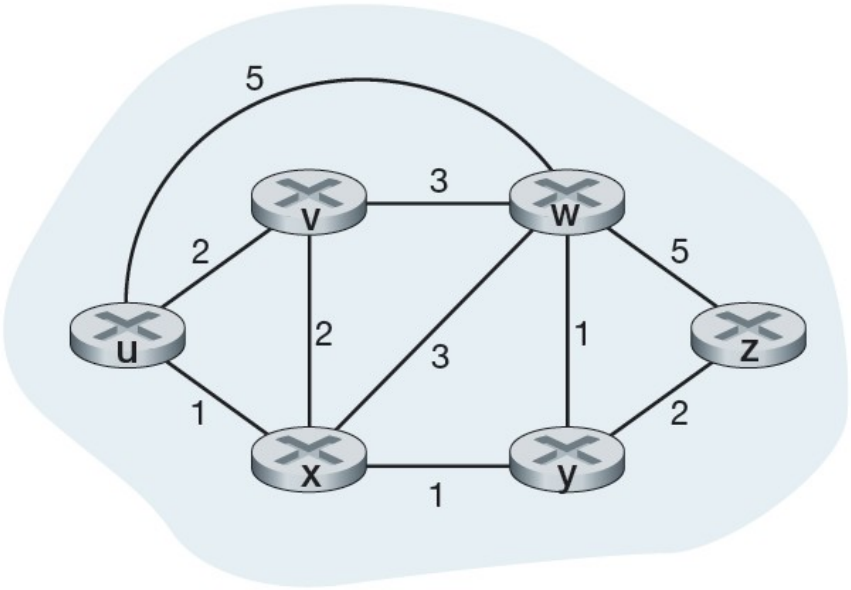
Algoritmo de Dijkstra: exemplo a partir de **u**

Etapa	N'	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(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



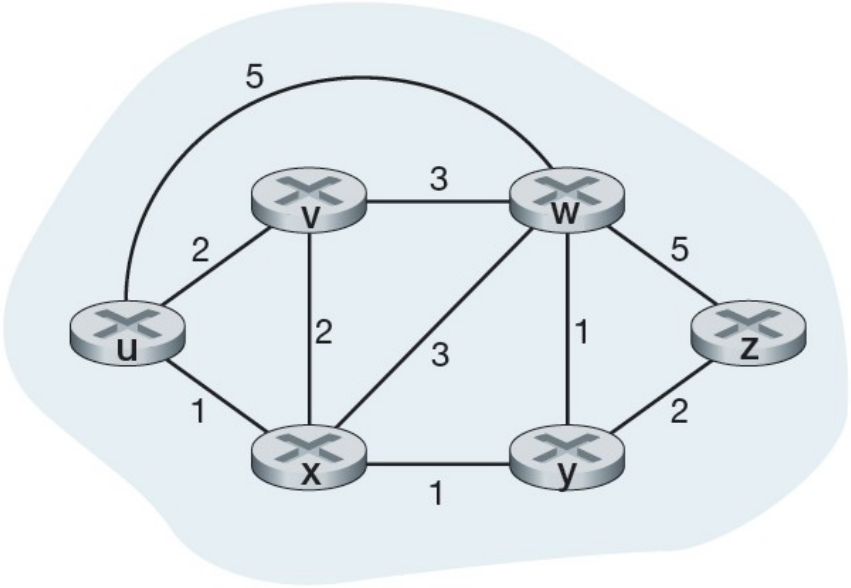
Algoritmo de Dijkstra: exemplo a partir de **u**

Etapa	N'	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(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



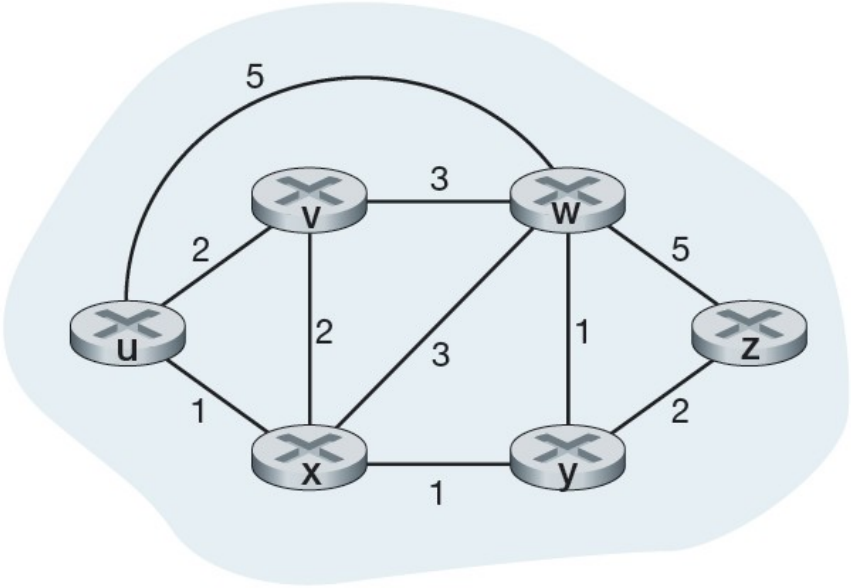
Algoritmo de Dijkstra: exemplo a partir de **u**

Etapa	N'	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(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



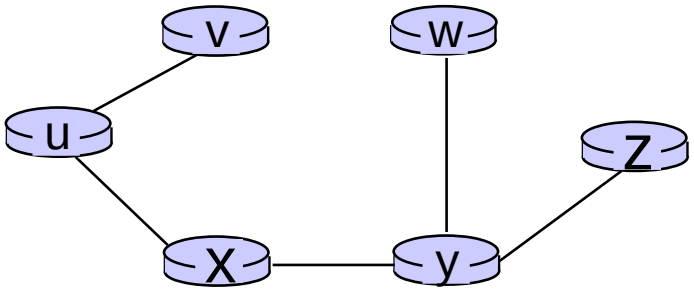
Algoritmo de Dijkstra: exemplo a partir de **u**

Etapa	N'	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(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:

- árvore resultante do caminho mais curto a partir de u:

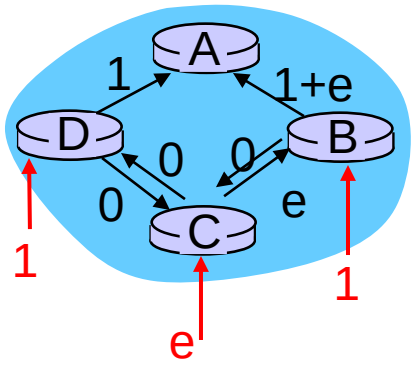


- tabela de repasse resultante em u:

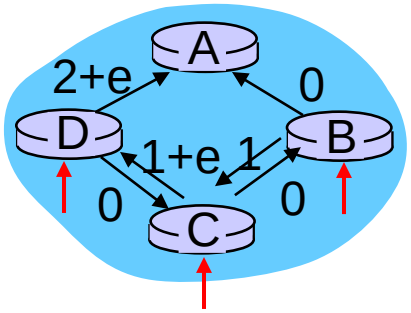
destino	enlace
v	(u,v)
x	(u,x)
y	(u,x)
w	(u,x)
z	(u,x)

Algoritmo de Dijkstra,discussão

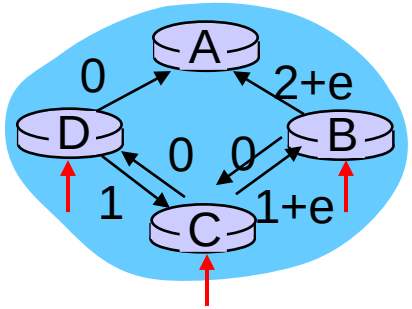
- complexidade do algoritmo: n nós
- cada iteração: precisa verificar todos os nós, w, não em N
- n(n+1)/2 comparações: $O(n^2)$
- implementações mais eficientes possíveis: O(nlogn)
- oscilações possíveis:
  - custo do enlace = quantidade de tráfego transportado
  - A é o destino de todos os tráfegos



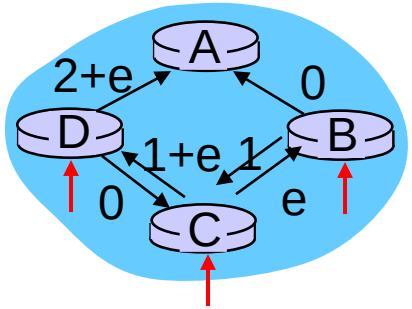
inicialmente



B e C detectam que o menor custo é no sentido horário



B, C e D detectam que o menor custo é no sentido anti-horário



… recalcula

Algoritmo de vetor de distância

Iterativo, assíncrono e distribuído

RIP: *Routing Information Protocol*

Equação de Bellman-Ford (programação dinâmica)

- defina

$d_x(y)$ = custo do caminho de menor custo de x para y

$$d_x(y) = \min_v \{c(x,v) + d_v(y)\}$$

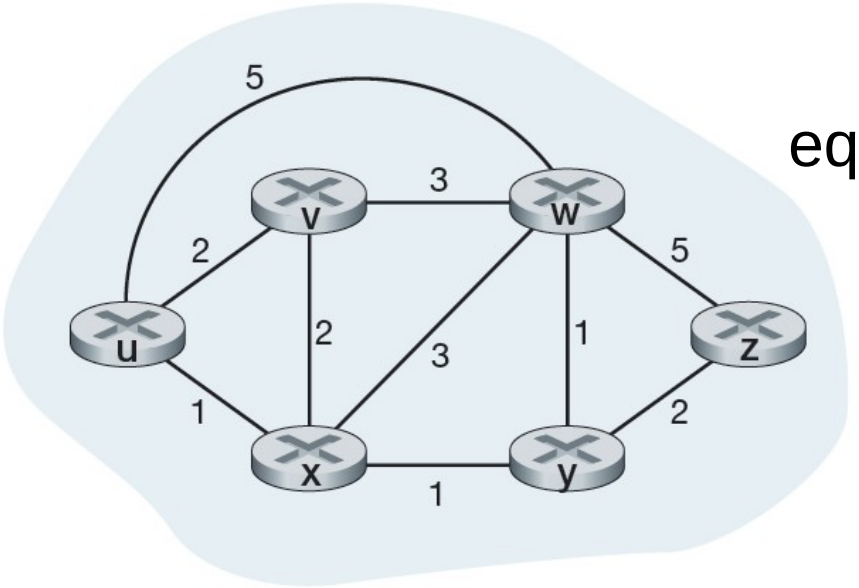
onde $\min_v$ assume todos os vizinhos v de x

Algoritmo de vetor de distância

- $D_x(y)$ = estimativa do menor custo de x para y
- nó x sabe custo de cada vizinho v: $c(x,v)$
- nó x mantém vetor de distância $\mathbf{D}_x = [D_x(y): y \in N]$
- nó x também mantém vetor de distância de seus vizinhos
  - para cada vizinho v, x mantém $\mathbf{D}_v = [D_v(y): y \in N]$

Exemplo de Bellman-Ford

Tendo u como origem e z como destino:
claramente, $d_v(z) = 5$, $d_x(z) = 3$, $d_w(z) = 3$



equação B-F diz:

$$d_u(z) = \min \{ c(u,v) + d_v(z),$$
$$c(u,x) + d_x(z),$$
$$c(u,w) + d_w(z) \}$$
$$= \min \{2 + 5,$$
$$1 + 3,$$
$$5 + 3\} = 4$$

- nó que alcança mínimo é o próximo salto no caminho mais curto ➜ tabela de repasse

Algoritmo de vetor de distância

ideia básica:

- de tempos em tempos, cada nó envia sua própria estimativa de vetor de distância aos vizinhos
- assíncrono
- quando um nó x recebe nova estimativa DV do vizinho, ele atualiza seu próprio DV usando a equação de B-F:

  $D_x(y) \leftarrow min_v\{c(x,v) + D_v(y)\}$ *para cada nó* $y \in N$

- sob condições modestas, naturais, a estimativa $D_x(y)$ *converge para o menor custo real* $d_x(y)$

Algoritmo de vetor de distância

iterativo, assíncrono: cada iteração local causada por:

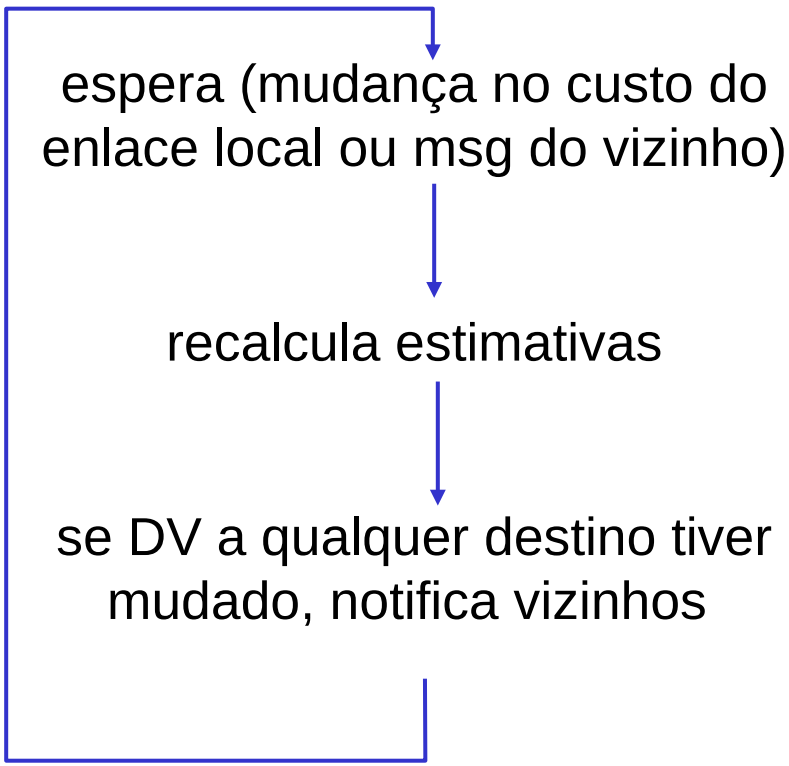
- mudança de custo do enlace local
- mensagem de atualização do DV do vizinho

distribuído:

cada nó notifica os vizinhos *apenas* quando seu DV muda

- vizinhos, então, notificam seus vizinhos, se necessário

Cada nó:



$D_x(y) = \min\{c(x,y) + D_y(y), c(x,z) + D_z(y)\}$
$= \min\{2+0 , 7+1\} = 2$

tabela nó x

de	custo para x	custo para y	custo para z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

de	custo para x	custo para y	custo para z
x	0	2	3
y	2	0	1
z	7	1	0

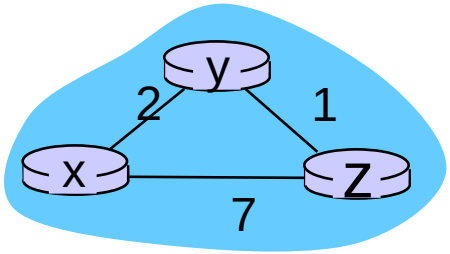
$D_x(z) = \min\{c(x,y) + D_y(z), c(x,z) + D_z(z)\}$
$= \min\{2+1 , 7+0\} = 3$

tabela nó y

de	custo para x	custo para y	custo para z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

tabela nó z

de	custo para x	custo para y	custo para z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0



tempo

tabela nó x

custo para

de	x	y	z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

custo para

de	x	y	z
x	0	2	3
y	2	0	1
z	7	1	0

custo para

de	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

tabela nó y

custo para

de	x	y	z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

custo para

de	x	y	z
x	0	2	7
y	2	0	1
z	7	1	0

custo para

de	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

tabela nó z

custo para

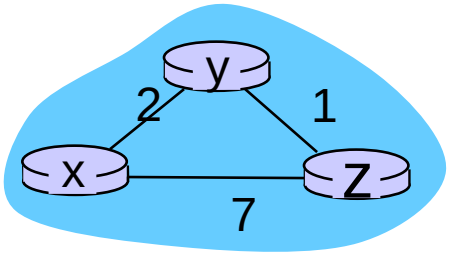
de	x	y	z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0

custo para

de	x	y	z
x	0	2	7
y	2	0	1
z	3	1	0

custo para

de	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

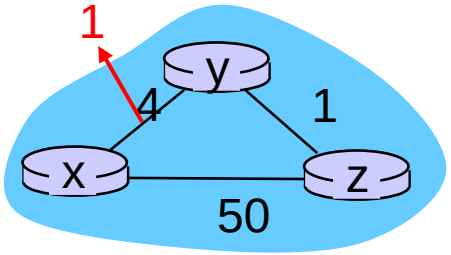


tempo

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"

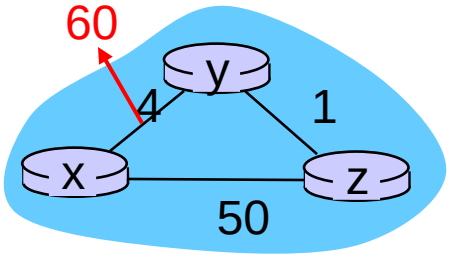
- no tempo t_0, *y* detecta a mudança do custo do enlace, atualiza seu DV e informa aos seus vizinhos.
- no tempo t_1, *z* recebe a atualização de *y* e atualiza sua tabela. Calcula um novo custo mínimo para *x* e envia seu DV aos vizinhos.
- no tempo t_2, *y* recebe a atualização de *z* e atualiza sua tabela de distância. Menores custos de *y* não mudam, e daí *y* *não* envia qualquer mensagem a *z*.

mudanças de custo do enlace:

- boas notícias correm rápido
- más notícias correm lento – problema da “contagem até o infinito”!
- 44 iterações antes que o algoritmo estabilize: ver texto ao lado

reverso envenenado:

- se Z passa por Y para chegar a X:
  - *Z diz a Y que sua distância (de Z) até X é infinita (de modo que Y não roteará para X passando por Z)*
- isso solucionará completamente o problema da contagem até o infinito? NÃO: loops com 3 ou + nós



1) $D_y(x)=4$, $D_y(z)=1$, $D_z(y)=1$ e $D_z(x)=5$
2) Y detecta 4 → 60
3) $D_y(x)=min\{60+0,1+5\}=6$
4) Y anuncia seu novo custo
5) Z recebe a nova informação
6) $D_z(x)=min\{50+0,1+6\}=7$
7) Z anuncia seu novo custo
8) Y recebe a nova informação
9) $D_y(x)=min\{60+0,1+7\}=8$
10)…. 44 iterações
11)Z percebe: custo por Y é maior que 50 e roteará direto para X.
Se c(y,x) 4 → 10000 e c(z,x) fosse 9999? 10 mil iterações (∞)

OSPF versus RIP

- O rápido crescimento e expansão das redes atuais levou o protocolo RIP aos seus limites. O RIP possui certas limitações que podem causar problemas em grandes redes, como:

1) RIP possui o limite de 15 hops (saltos). Uma rede RIP com dimensão maior do que 15 hops (15 roteadores) é considerada inalcançável.
2) RIP não pode trabalhar com máscaras de subrede de tamanho variável (CIDR). Dado a falta de endereços IP, e a grande flexibilidade proporcionada por CIDRs na atribuição de endereços IP, esta é considerada uma das maiores falhas.
3) O *broadcast* periódico de toda a tabela de roteamento consome uma grande quantidade de banda. Este é um grande problema com redes grandes, especialmente em links lentos.

OSPF versus RIP

4)O RIP converge mais lentamente que o OSPF. Em grandes redes, a convergência deve se dar na ordem de minutos.

5)No RIP não há o conceito de atraso da rede e custo do enlace. As decisões de roteamento são baseadas na contagem de hops. A rota com menor número de hops até o destino é sempre a escolhida, mesmo que a rota mais longa possua melhor relação entre largura de banca do enlace e atraso.

6)Redes RIP são planas, não há o conceito de áreas ou fronteiras.

7)Alguns avanços foram introduzidos na nova versão do RIP, chamada de RIP2. Esta nova versão trata da questão de CIDRs, autenticação, e atualizações de roteamento simultâneas (multicast). RIP2 não apresenta avanços expressivos em relação ao RIP porque ainda apresenta limitações na contagem de hops e lenta convergência, o que é essencial nas grandes redes atuais.

OSPF é melhor que RIP (!?)

- OSPF apresenta algumas características atrativas, como:

1)Não há limite na contagem de hops.

2)O uso eficiente de CIDR é muito útil na alocação de endereços IP.

3)OSPF usa multicast de IP para enviar atualizações de link-state. Isto garante menor processamento nos roteadores que não estão escutando os pacotes OSPF. Além disto, as atualizações só são enviadas nos casos em que mudanças ocorrem, ao invés de periodicamente; isso garante uma menor utilização da banda

4)Apresenta melhor convergência que o RIP. Isto porque as mudanças na rota são enviadas instantaneamente, e não periodicamente.

5)Permite um melhor balanceamento de carga.

OSPF versus RIP

6)Permite uma divisão lógica da rede, onde roteadores podem ser divididos em áreas. Isto limita a explosão de atualizações de *link state* por toda a rede. Também fornece um mecanismo para agregar roteadores e limita a propagação desnecessária de informações de sub-rede.

7)Permite a autenticação de rota utilizando diferentes métodos para a autenticação de senha.

8)Permite a transferência e marcação de rotas externas inseridas em um AS. Isto rastreia rotas externas inseridas por protocolos externos como o BGP.

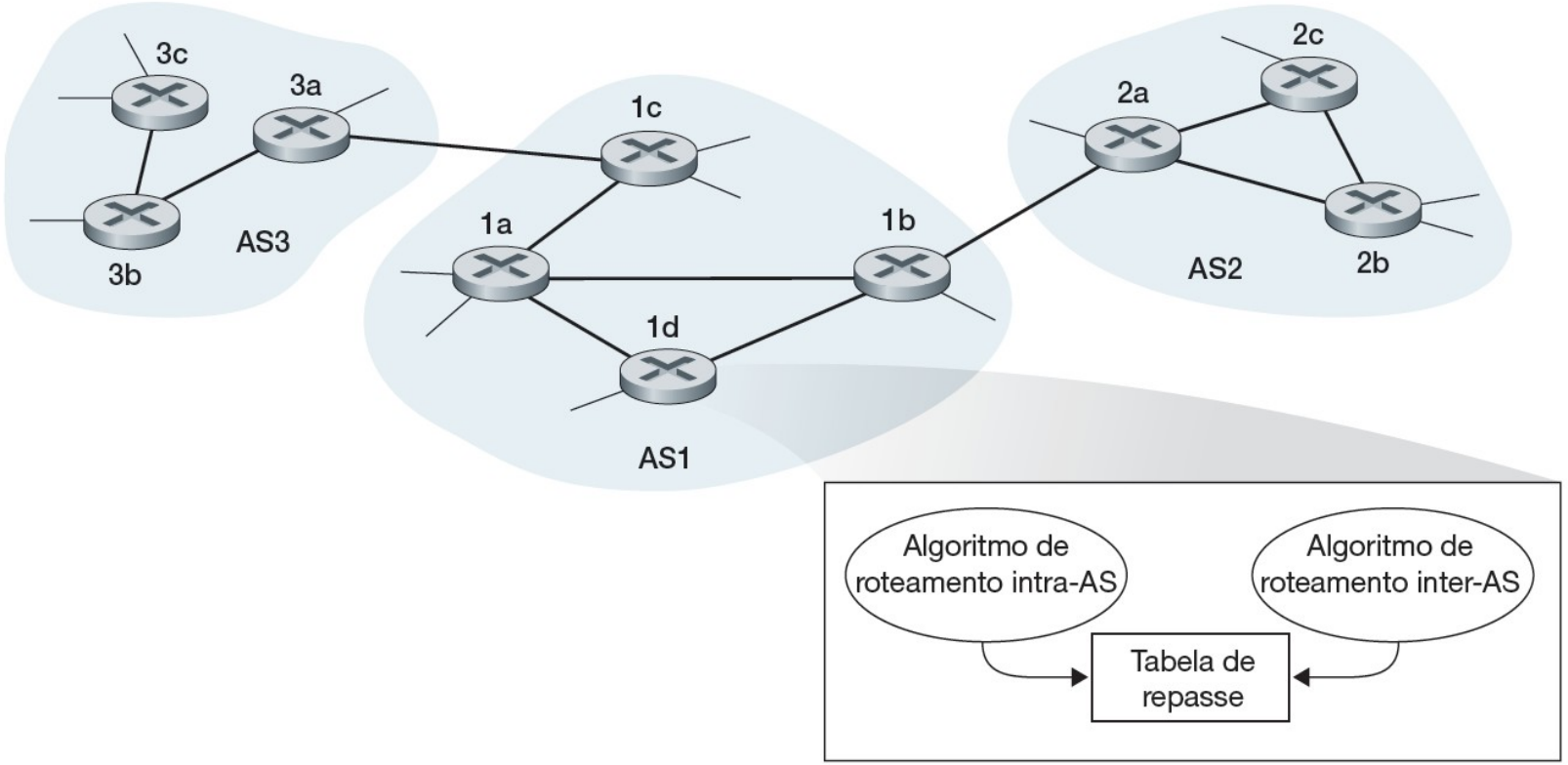
9)Estes pontos obviamente levam a maior complexidade na configuração e resolução de problemas. Administradores acostumados com a simplicidade do RIP ficam assustados com a quantidade nova de informação requerida para ficar a par de redes OSPF. Também adiciona mais *overhead* na alocação de memória e utilização da CPU.

Roteamento hierárquico

- Todos os roteadores dentro do mesmo AS rodam o mesmo algoritmo de roteamento e dispõem das informações sobre cada um dos outros.
- O algoritmo de roteamento que roda dentro de um AS é denominado um **protocolo de roteamento intrassistema autônomo**.
- A figura a seguir ilustra um exemplo simples com três ASs: AS1, AS2 E AS3.

Roteamento hierárquico

- Um exemplo de sistemas autônomos interconectados:



Roteamento intra-AS na Internet

- Um protocolo de roteamento intra-AS é usado para determinar como é rodado o roteamento dentro de um sistema autônomo (AS).

- Historicamente, dois protocolos de roteamento têm sido usados para roteamento dentro de um sistema autônomo na Internet:

1. o protocolo de informações de roteamento, **RIP** (Routing Information Protocol) (vetor de distância - DV) http://www.gta.ufrj.br/grad/02_2/ospf/rip.html e
2. o **OSPF** (Open Shortest Path First) (~estado de enlace – LS). http://www.gta.ufrj.br/grad/02_2/ospf/ospf.html
3. Comparativo: http://www.gta.ufrj.br/grad/02_2/ospf/comparacao.html

Roteamento inter-AS: BGP

- O BGP (*Border Gateway Protocol*) oferece a cada AS meios de:

1. Obter de ASs vizinhos informações de alcançabilidade de sub-redes.

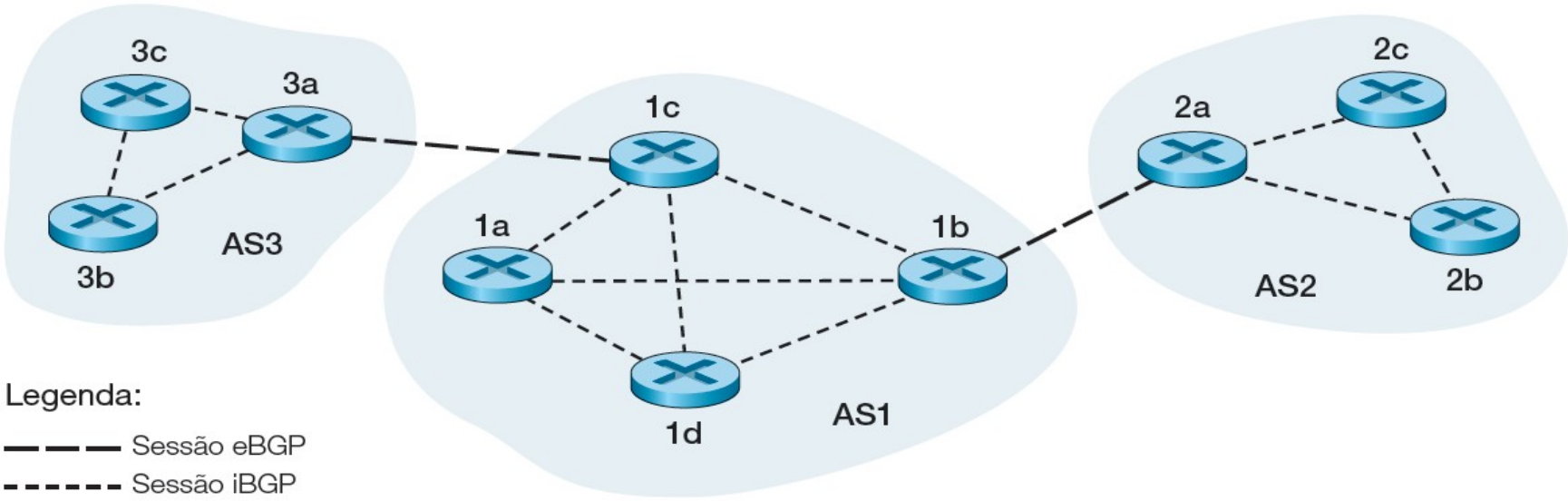
2. Propagar a informação de alcançabilidade a todos os roteadores internos ao AS.

3. Determinar rotas “boas” para sub-redes com base na informação de alcançabilidade e na política do AS.

Roteamento inter-AS: BGP

- No BGP, pares de roteadores trocam informações de roteamento por conexões TCP semipermanentes usando a porta 179.

- Sessões eBGP e iBGP (**e**xternal, **i**nternal)



Roteamento inter-AS: BGP

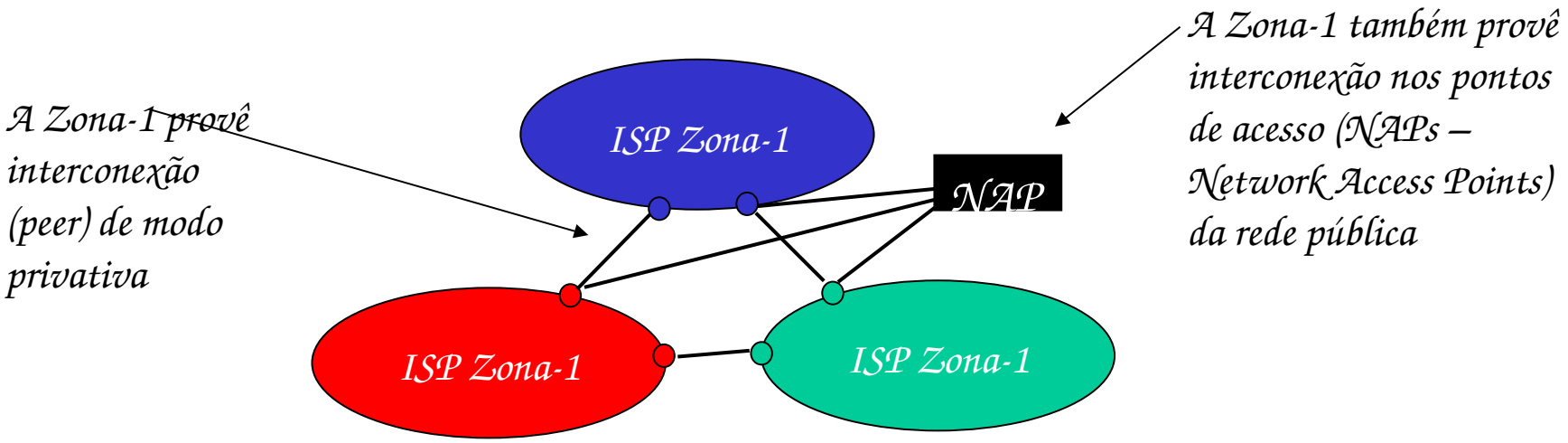
- O BGP permite que cada AS conheça quais destinos podem ser alcançados por meio de seus ASs vizinhos.
- No BGP, um sistema autônomo é identificado por seu **número de sistema autônomo** (ASN) globalmente exclusivo [RFC 1930].
- Quando um roteador anuncia um prefixo para uma sessão BGP, inclui vários atributos BGP juntamente com o prefixo.
- O BGP usa eBGP e iBGP para distribuir rotas a todos os roteadores dentro de ASs.

Internet no Brasil em números (CGI.br)

- 8923 ASNs
  - 2º pais no mundo (1º EUA)
  - 69% dos ASNs da América Latina e Caribe
- Provedores de Acesso (ISPs)
  - 12.826 (estimado)

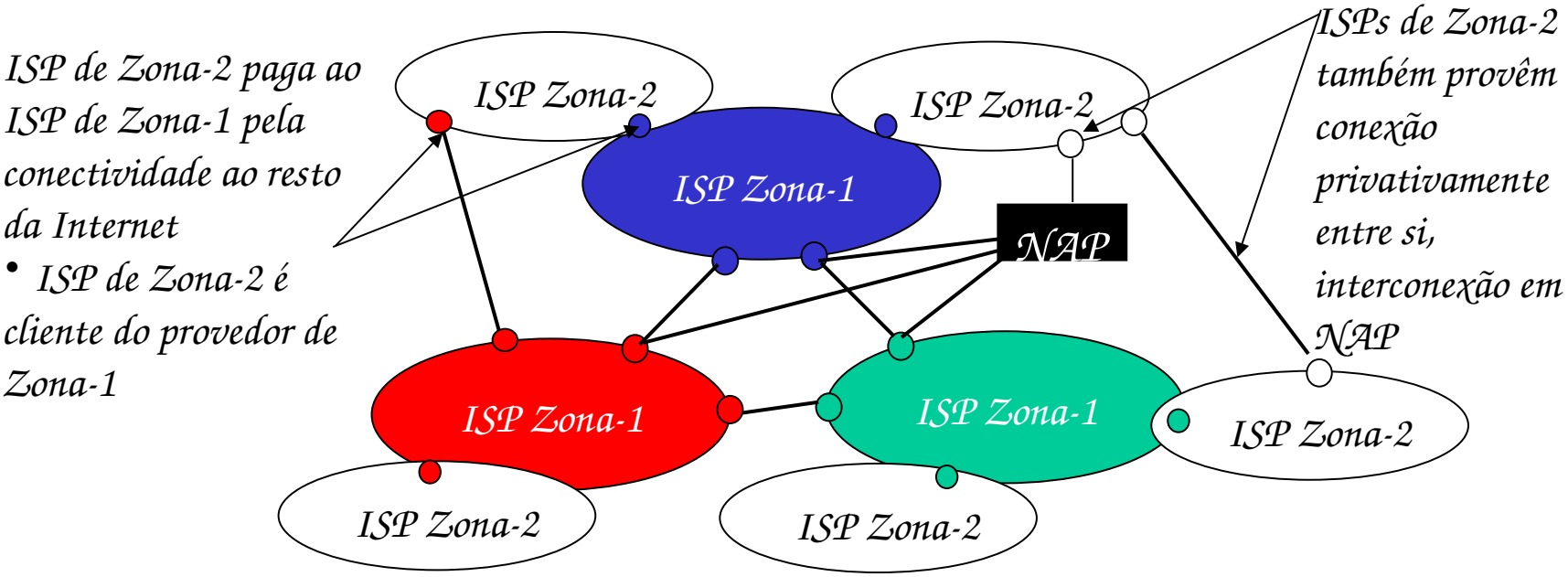
Estrutura da Internet: rede de redes

- Grosseiramente hierárquica
- No centro: ISPs de “zona-1” (ex.: UUNet, BBN/Genuity, Sprint, AT&T), cobertura national/international
  - Os outros são igualmente tratados



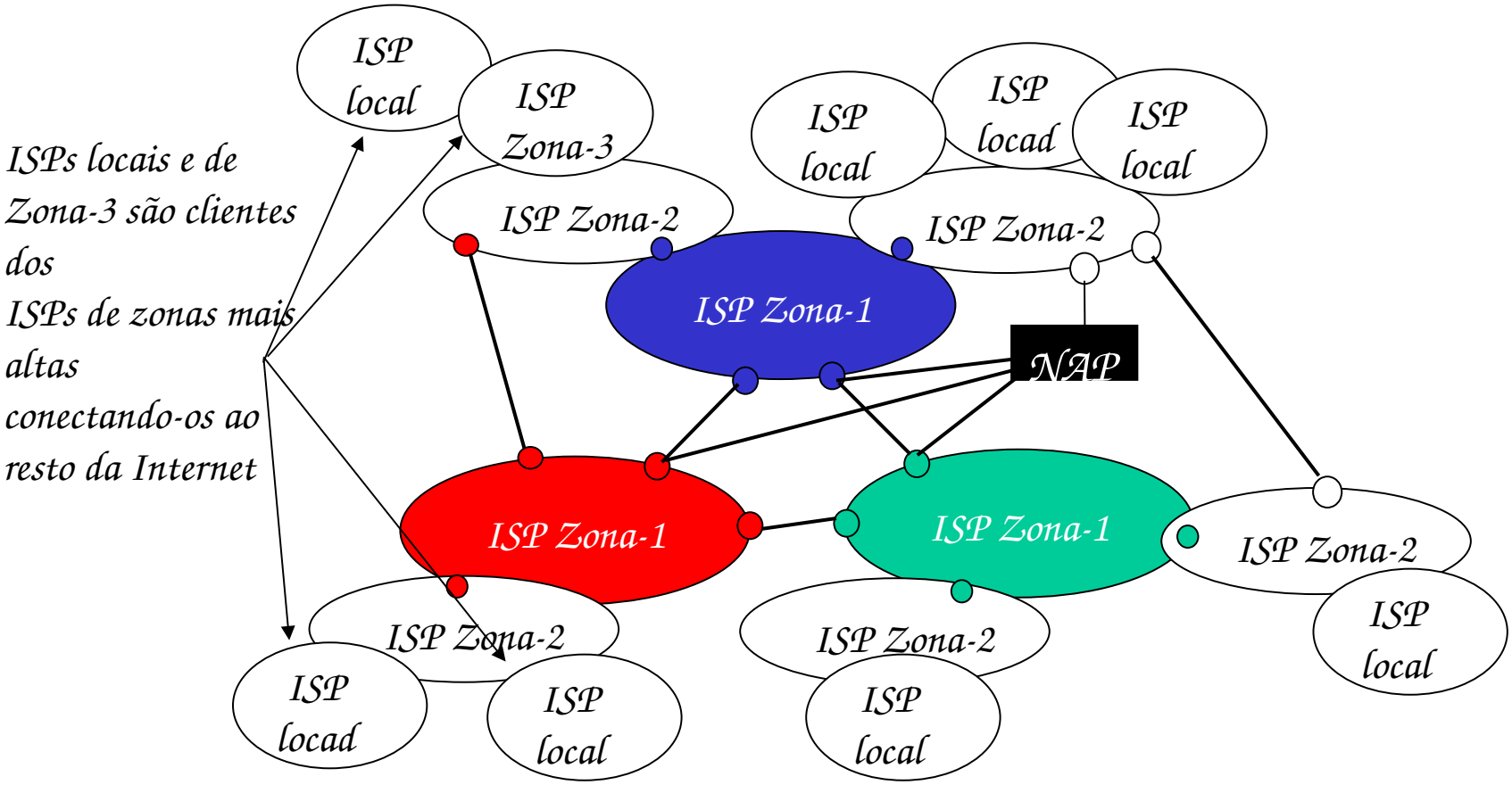
Estrutura da Internet: rede de redes

- ISPs de ”Zona-2”: ISPs menores (freqüentemente regionais)
  - Conectam-se a um ou mais ISPs de Zona-1, possivelmente a outros ISPs de Zona-2



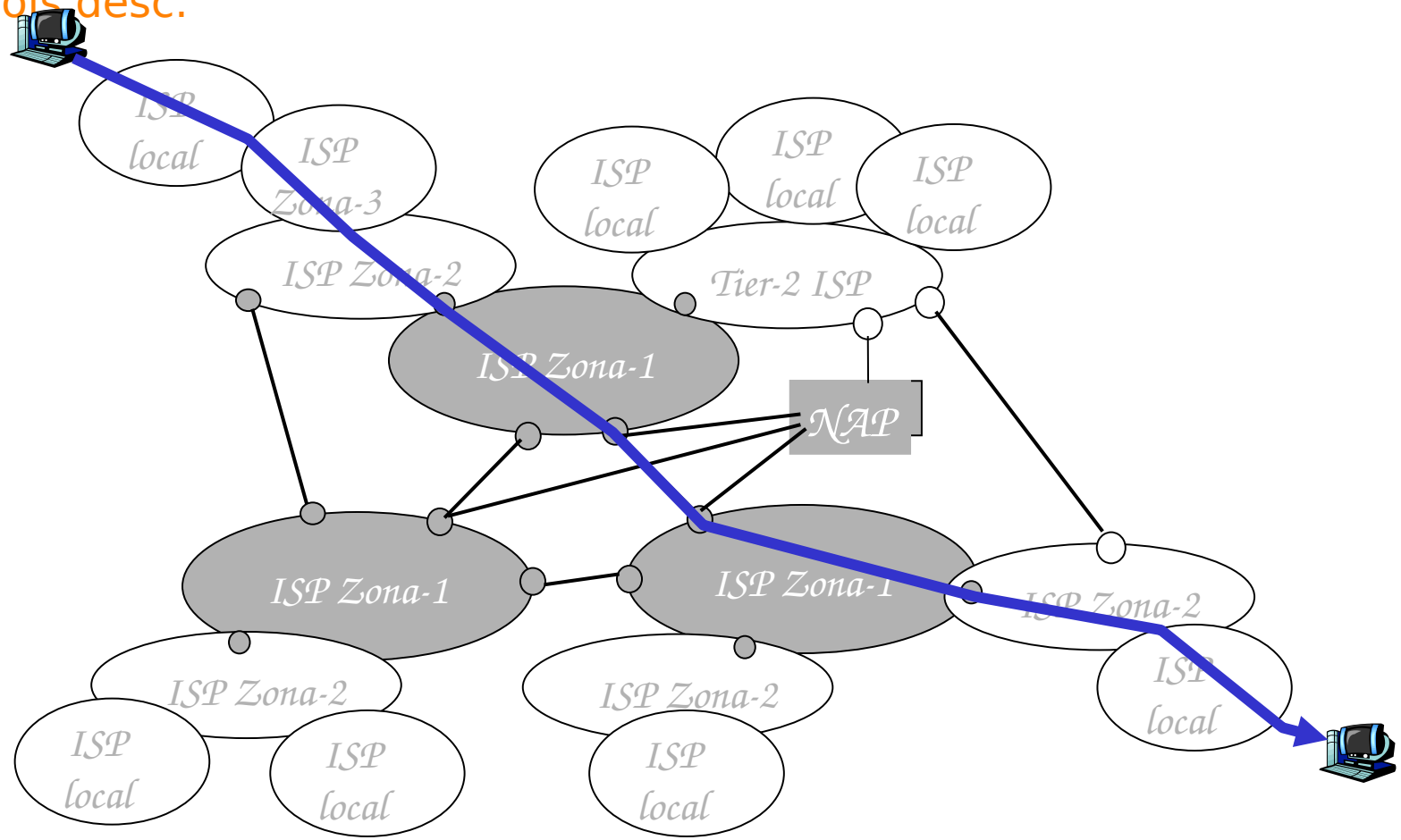
Estrutura da Internet: rede de redes

- ISPs de “Zona-3” e ISPs locais
  - Última rede de acesso (“hop”) (mais próxima dos sistemas finais)



Estrutura da Internet: rede de redes

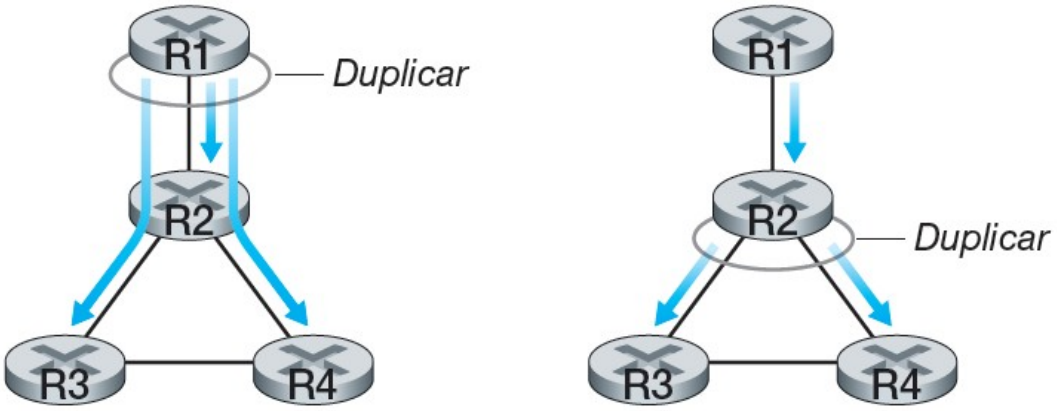
- Um pacote passa através de muitas redes
- Roteamento Hirérquico: Vai subindo até chegar a uma zona comum e depois desc.



Algoritmos de roteamento por difusão (*broadcast*)

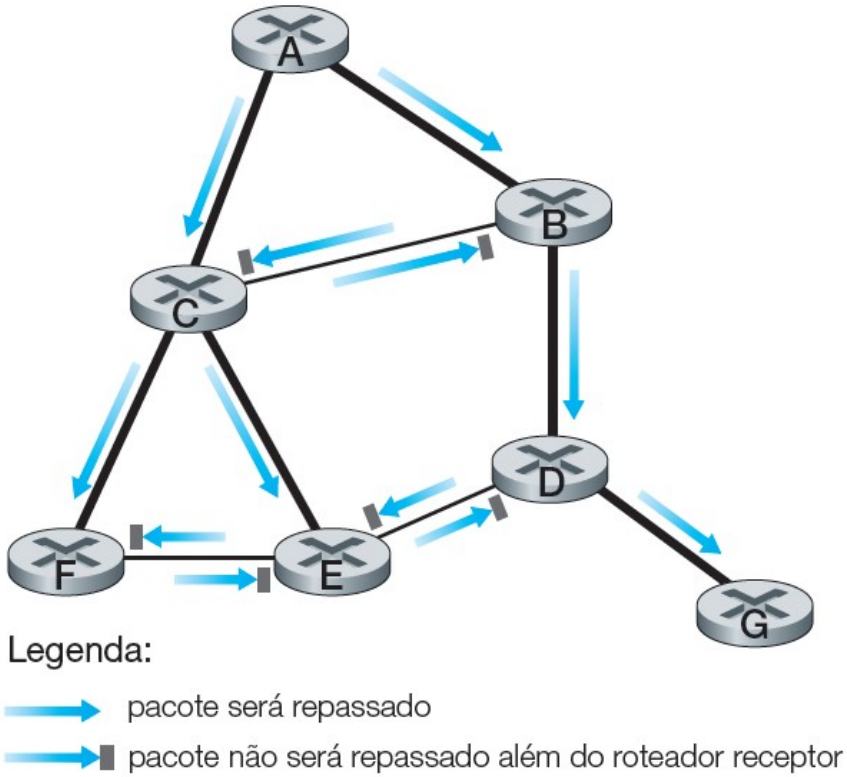
- Talvez o modo mais direto de conseguir comunicação por difusão seja o nó remetente enviar uma cópia separada do pacote para cada destino, Figura (a) – **duplicar na origem**.
- A técnica mais óbvia, na inundação controlada por número de sequência, um nó de origem coloca seu endereço, bem como um número de sequência de difusão em um pacote de difusão e então envia o pacote a todos os seus vizinhos, Figura (b) – **duplicar dentro da rede**. Problemas: ciclos: R2-R3-R4-R2... e tempestade de difusão.

Criação/transmissão de duplicatas



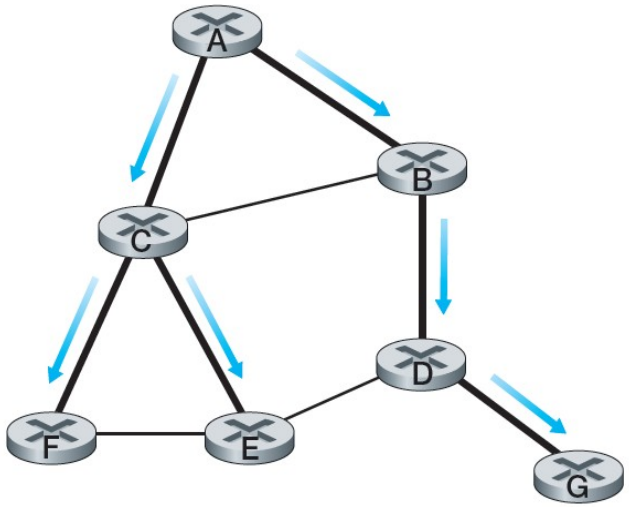
Algoritmos de roteamento por difusão (*broadcast*)

- Para evitar tempestade de difusão usa-se o repasse pelo caminho inverso: quando um roteador recebe um pacote de difusão com determinado endereço de origem, ele transmite o pacote para todos os seus enlaces de saída somente se chegou pelo enlace mais "próximo" ao endereço de origem.
- Perceba que ainda há problemas com transmissões desnecessárias, B-C, F-E....

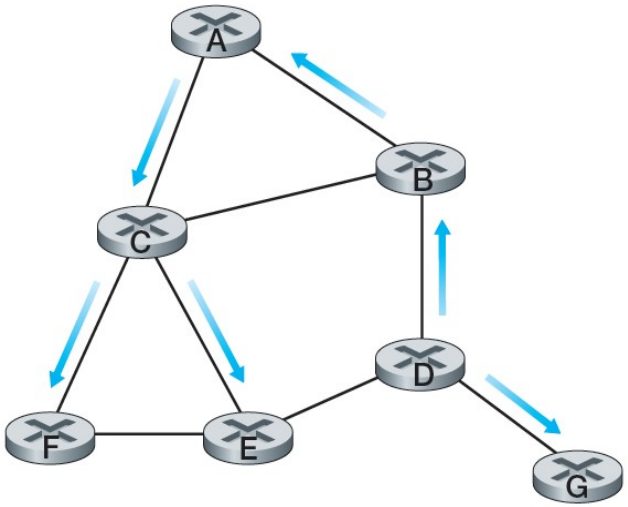


Algoritmos de roteamento por difusão (*broadcast*)

- Assim, outra abordagem para o fornecimento de difusão é os nós da rede construírem uma ***spanning tree***, em primeiro lugar.
- Na **abordagem de nó central** da construção de uma *spanning tree*, é definido um nó central.
  a) formação da árvore (ramos em negrito), partindo de A
  b) uso da mesma, partindo de D.



a. Difusão iniciada em *A*



b. Difusão iniciada em *D*

Serviço para um grupo (*multicast*)

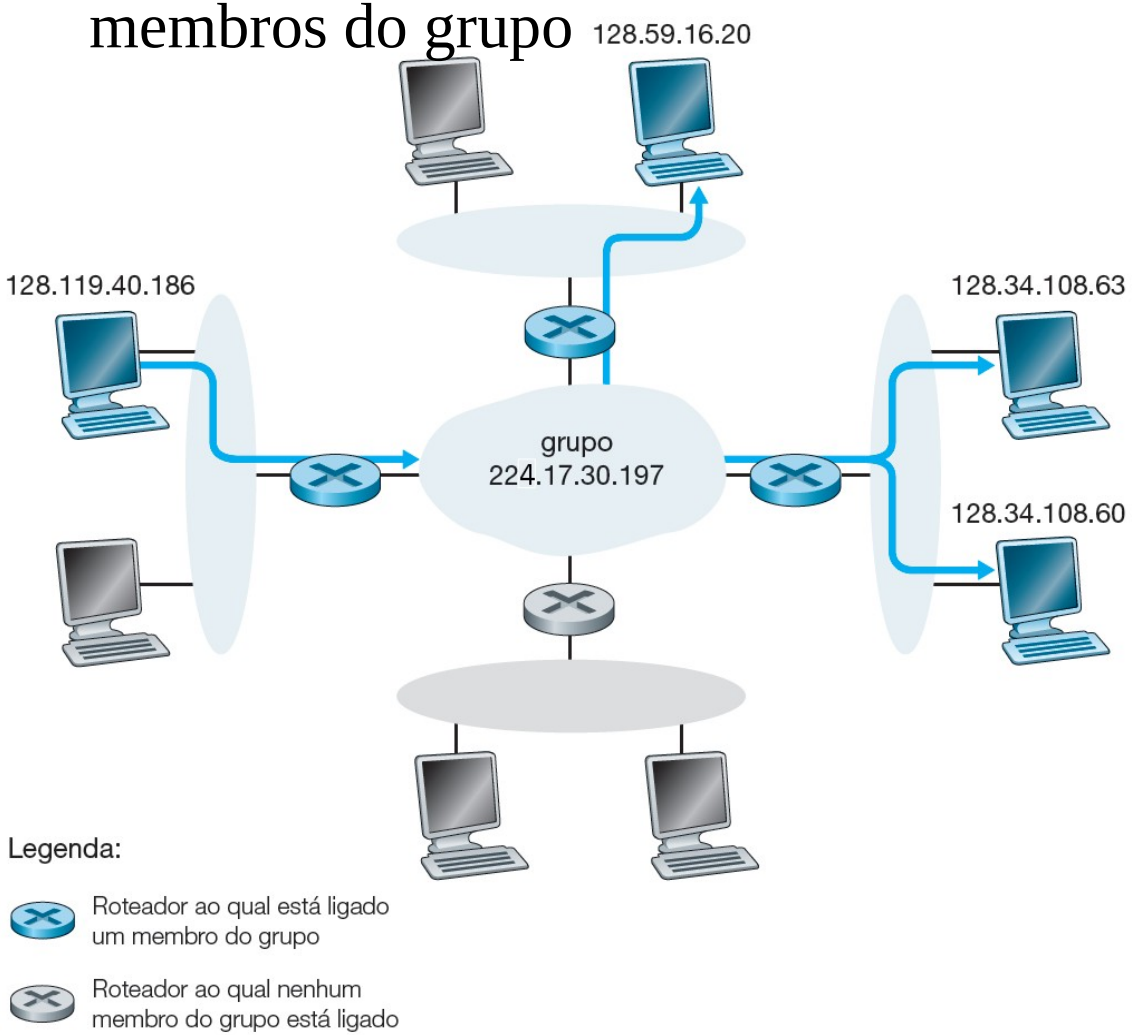
- Na comunicação para um grupo, enfrentamos imediatamente dois problemas:
  1. como identificar os destinatários de um pacote desse tipo e
  2. como endereçar um pacote enviado a um desses destinatários.

- Um pacote para um grupo é endereçado usando **endereço indireto**.

- O grupo de destinatários associados a um endereço classe D é denominado **grupo *multicast***.
  https://en.wikipedia.org/wiki/Reserved_IP_addresses
  https://en.wikipedia.org/wiki/Multicast_address

Serviço para um grupo (*multicast*)

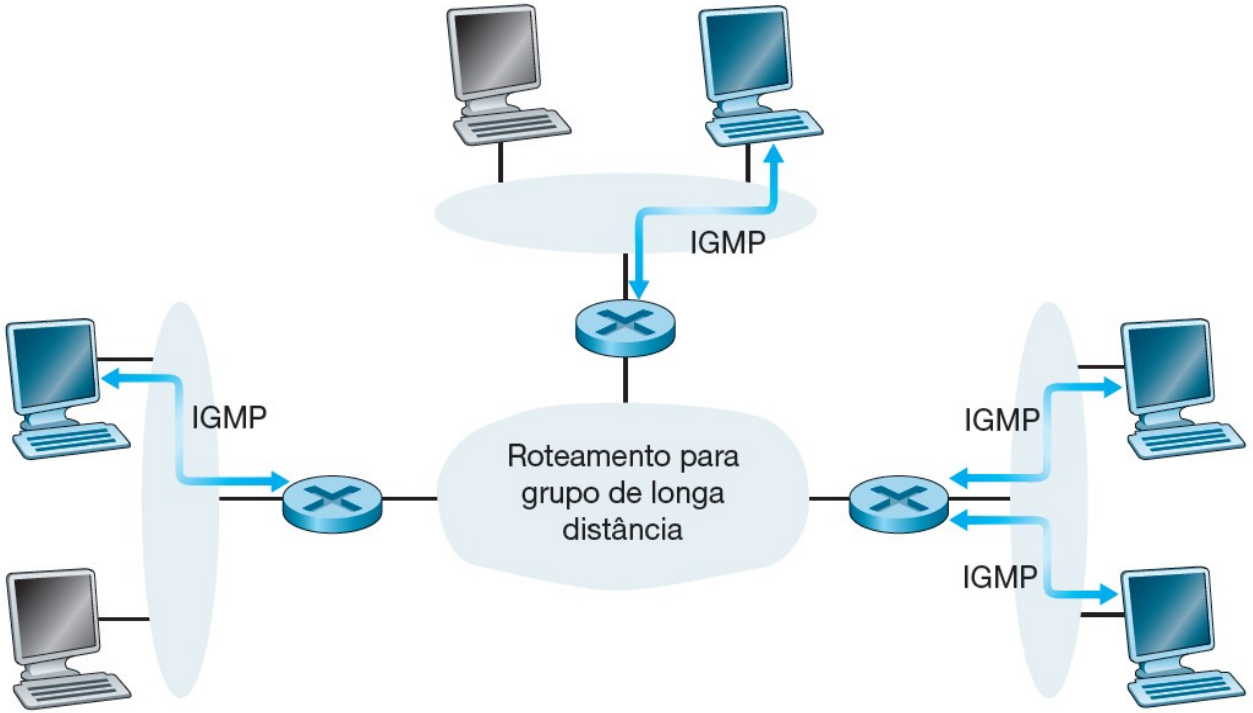
- O serviço para um grupo: um datagrama endereçado ao grupo (endereço classe D = /24) é entregue a todos os membros do grupo



1) Como o grupo começa e termina?
2) Como é escolhido o endereço?
3) Como um novo hospedeiro é incorporado?
4) Qualquer um pode se juntar ao grupo?
5) Os membros do grupo podem se conhecer mutuamente?
6) Como os nós da rede interagem para entregar um datagrama a todos os membros do grupo?

Serviço para um grupo (*multicast*)

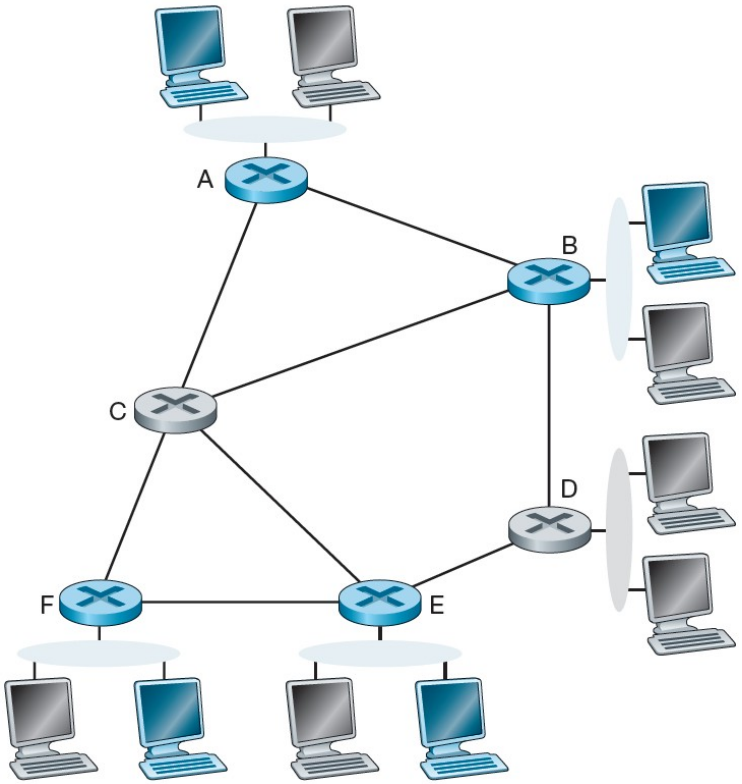
- Os dois componentes de grupo da camada de rede: IGMP (*Internet Group Management Protocol*) e protocolos de roteamento para um grupo



- Router ==> membership_query
- Host ==> membership_report
- Host ==> leave_group

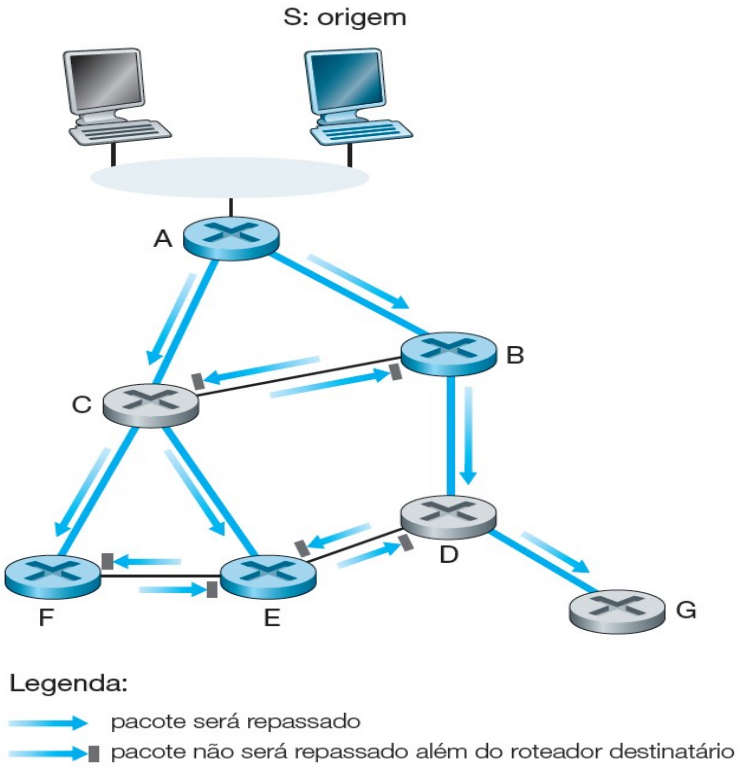
Algoritmos de roteamento para grupo

- Hospedeiros do grupo, seus roteadores conectados e outros roteadores



Serviço para um grupo (*multicast*)

- Roteamento para um grupo usando uma árvore baseada na origem.
- Repasse pelo caminho inverso, no caso do serviço para um grupo



- **Poda**. Roteadores sem interesse devolvem MSGs ….