

# Redes de Computadores

## Camada de Rede

Antonio Alfredo Ferreira Loureiro  
loureiro@dcc.ufmg.br

Departamento de Ciência da Computação  
Universidade Federal de Minas Gerais



# Motivação para interconexão

- Diferentes tecnologias de rede oferecem diferentes características:
  - ◆ LANs: alta velocidade, pequena distância
  - ◆ WANs: comunicação numa grande área
  - ◆ Não existe uma única tecnologia de rede que é melhor para todas as necessidades
- É comum uma grande organização ter várias redes físicas, cada uma adequada para um determinado tipo de ambiente

# Conceito de serviço universal

- Serviço universal:
  - ◆ Permite a um usuário ou aplicação em um computador trocar informações com qualquer outro usuário ou aplicação em outro computador da rede
- Um sistema de comunicação que provê serviço universal permite que pares arbitrários de computadores se comuniquem

# Conceito de serviço universal

- Incompatibilidades entre hardware de rede e endereçamento físico fazem com que não seja possível criar uma rede com diferentes tecnologias
- Solução
  - ◆ Interconexão entre redes através de hardware e software
- Sistema resultante:
  - ◆ internet

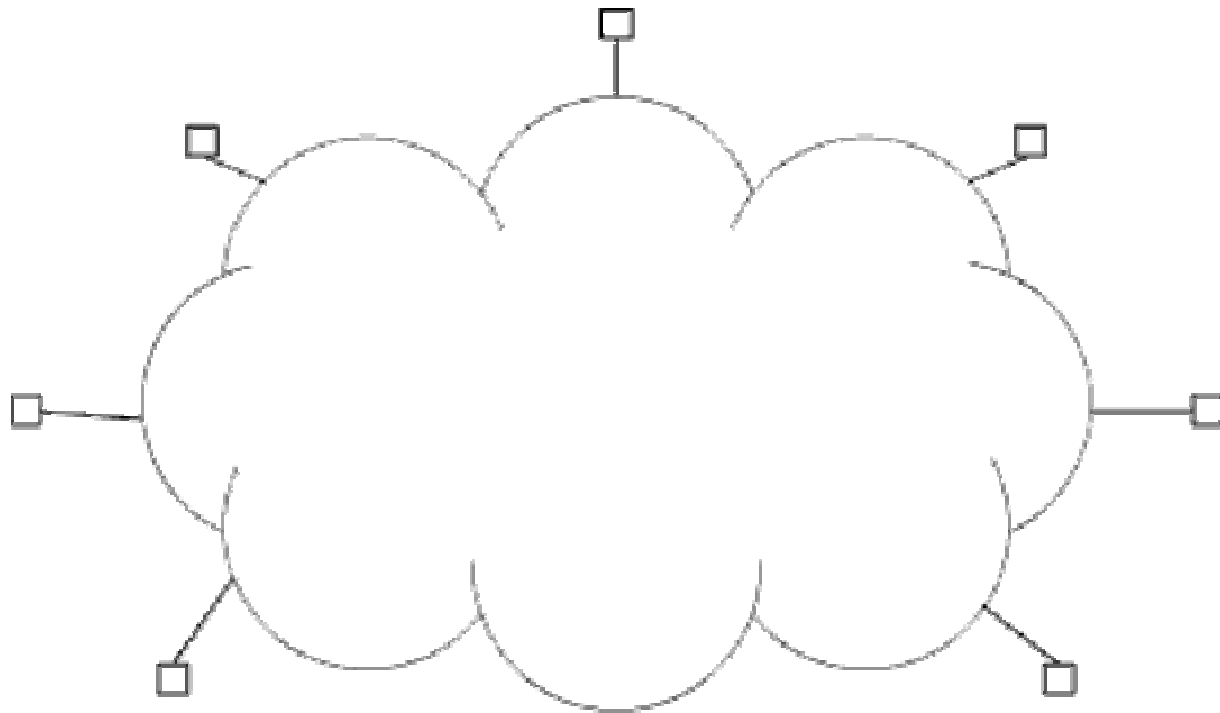
# Hardware básico de uma internet

- Roteador
  - ◆ HW especial dedicado a interconexão de redes
  - ◆ Redes podem usar diferentes tecnologias como meio físico, endereçamento, formato de pacotes
- Roteadores comerciais podem ser usados para conectar mais de duas redes
- Um único roteador não é usado para conectar todas as redes de uma organização:
  - ◆ Provável incapacidade do roteador de processar todos os pacotes entre as redes de forma apropriada
  - ◆ Falta de confiabilidade e redundância

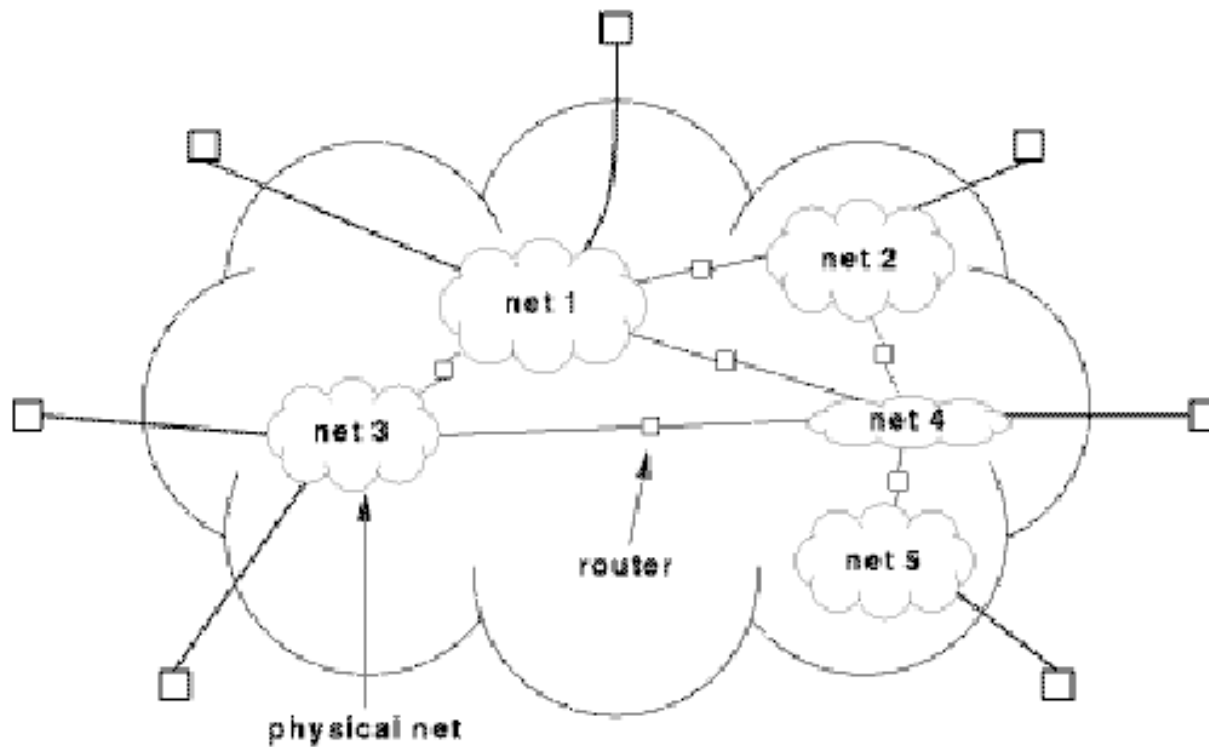
# Como obter serviço universal

- Objetivo de interconexão:
  - ◆ Ter serviço universal entre redes heterogêneas
- Como?
  - ◆ Roteadores devem negociar a sintaxe e semântica de pacotes entre origem e destino
  - ◆ Rede virtual: sistema de comunicação é uma abstração

# Rede virtual



# Rede virtual



# Protocolos para interconexão

- Mais importante de todos:
  - ➔ IP (*Internet Protocol* da arquitetura TCP/IP)
- Interconexão:
  - ◆ Conceito fundamental das redes modernas
- Interconexão entre elementos computacionais é feita pelo protocolo IP
- Roteadores também executam alguma aplicação como gerenciamento

# Endereçamento numa rede virtual

- Componente crítico da abstração fornecida por uma internet
- Independente dos endereços físicos como os usados em redes locais
- Cria a ilusão de uma rede única e integrada
- Usuários, aplicações e protocolos de alto nível usam endereços abstratos para se comunicar

# Esquema de endereçamento IP

- Na arquitetura TCP/IP, o endereçamento é especificado pelo protocolo IP
- Endereço IPv4 (ou simplesmente IP) de um computador:
  - ◆ Número binário único de 32 bits
- Dividido em duas partes:
  - ◆ Prefixo: identifica a rede física na qual o computador se encontra (número de rede)
  - ◆ Sufixo: identifica o computador na rede

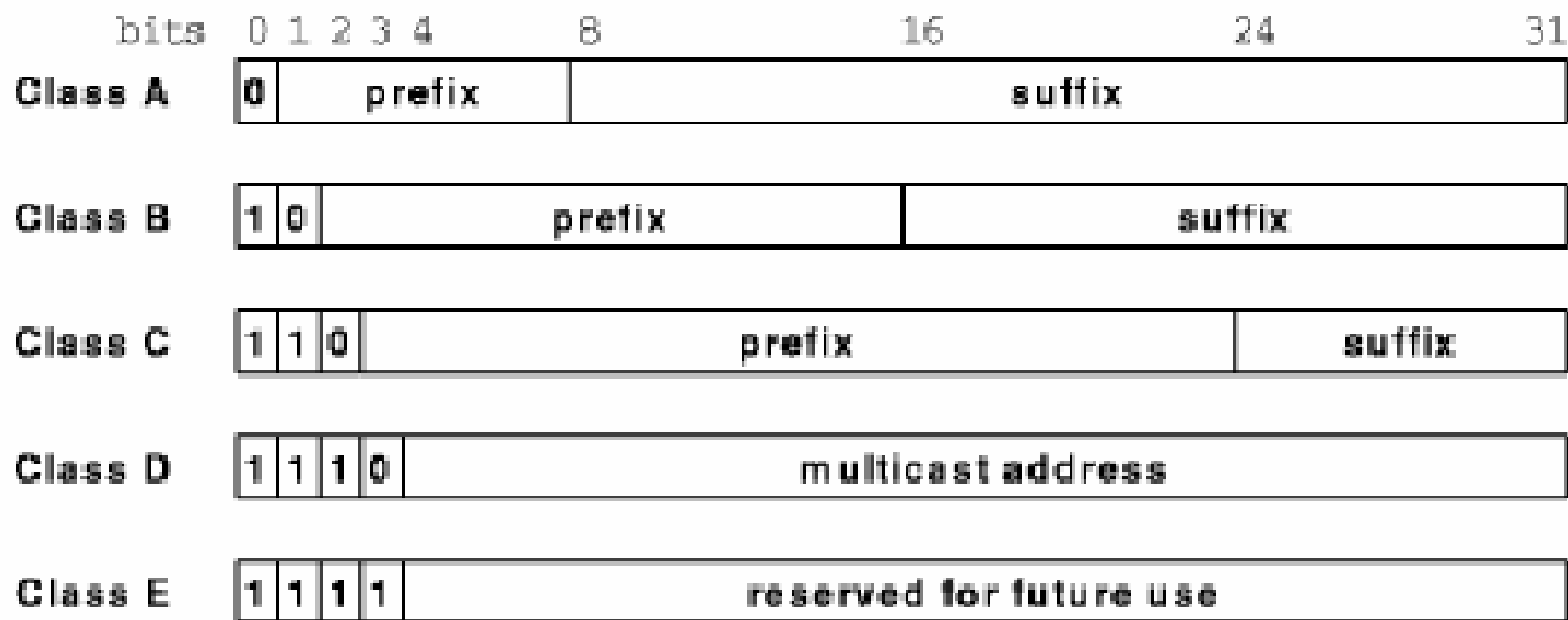
# Esquema de endereçamento IP

- Número de rede é único
- Número do prefixo deve ser controlado globalmente
- Número do sufixo pode ser controlado localmente

# Classes de endereçamento IP

- Compromisso entre tamanho de prefixo e sufixo que reflete diferentes tamanhos de rede
- Classes:
  - ◆ A, B e C: Primárias
  - ◆ D: comunicação em grupo
  - ◆ E: extensão futura (sem uso)
- É chamado de auto-identificável:
  - ◆ A classe de um endereço pode ser calculada do próprio endereço

# Classes de endereçamento IP



# Endereço IP

É auto-identificável:

- ◆ A classe de um endereço pode ser obtida a partir do próprio endereço

<b>First Four Bits Of Address</b>	<b>Table Index (in decimal)</b>	<b>Class of Address</b>
0000	0	A
0001	1	A
0010	2	A
0011	3	A
0100	4	A
0101	5	A
0110	6	A
0111	7	A
1000	8	B
1001	9	B
1010	10	B
1011	11	B
1100	12	C
1101	13	C
1110	14	D
1111	15	E

# Endereço IP

- Notação decimal com ponto:
  - ◆ 32 bits: 4 x 8 bits
- Forma usual de representar endereços
- Exemplo:
  - ◆ mica.dcc.ufmg.br: 150.164.0.134

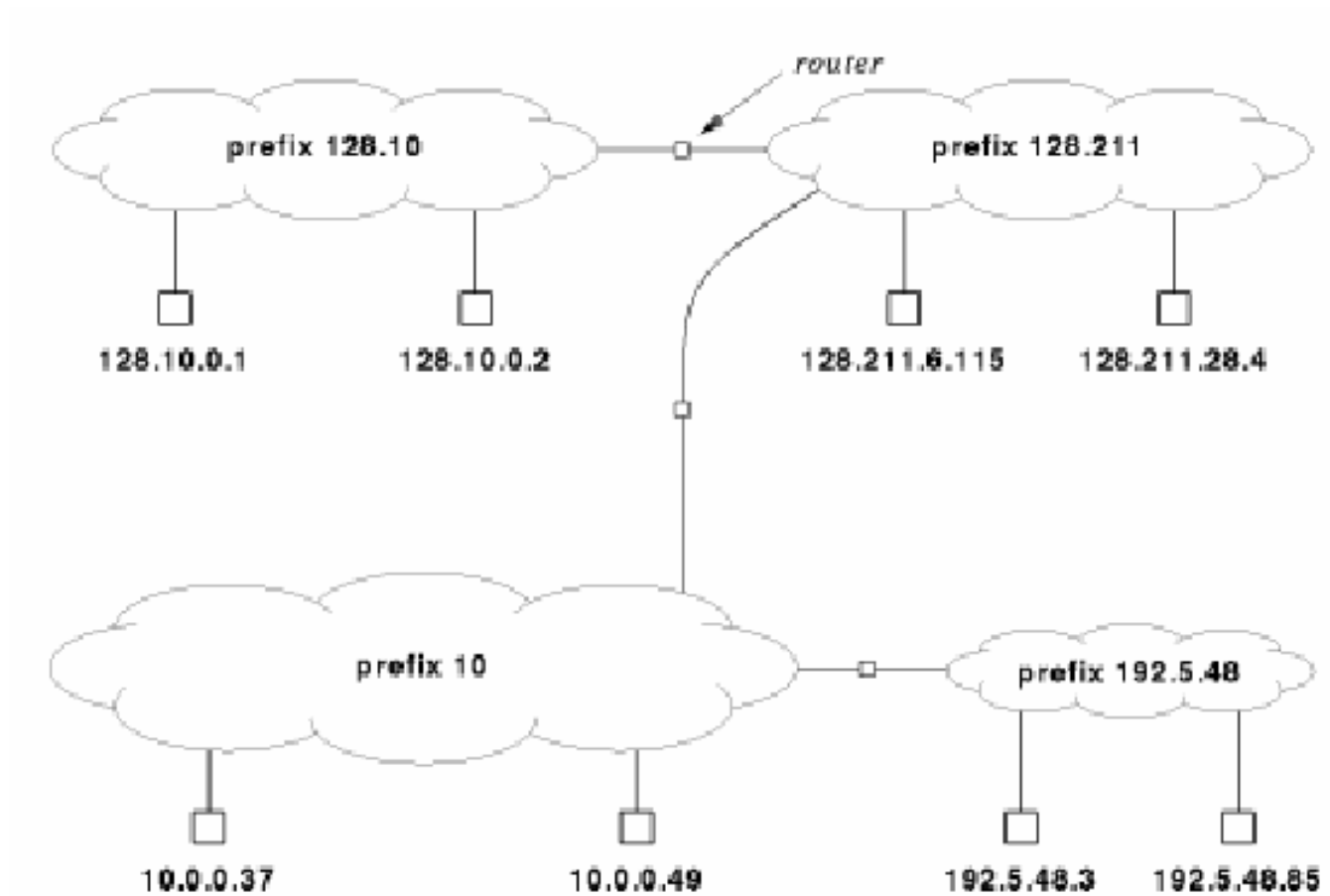
32-bit Binary Number	Equivalent Dotted Decimal
10000001 00110100 00000110 00000000	129 . 52 . 6 . 0
11000000 00000101 00110000 00000011	192 . 5 . 48 . 3
00001010 00000010 00000000 00100101	10 . 2 . 0 . 37
10000000 00001010 00000010 00000011	128 . 10 . 2 . 3
10000000 10000000 11111111 00000000	128 . 128 . 255 . 0

# Endereço IP

## Espaço de endereçamento

<b>Address Class</b>	<b>Bits In Prefix</b>	<b>Maximum Number of Networks</b>	<b>Bits In Suffix</b>	<b>Maximum Number Of Hosts Per Network</b>
<b>A</b>	<b>7</b>	<b>128</b>	<b>24</b>	<b>16777216</b>
<b>B</b>	<b>14</b>	<b>16384</b>	<b>16</b>	<b>65536</b>
<b>C</b>	<b>21</b>	<b>2097152</b>	<b>8</b>	<b>256</b>

# Exemplo de endereçamento



# Endereço IP

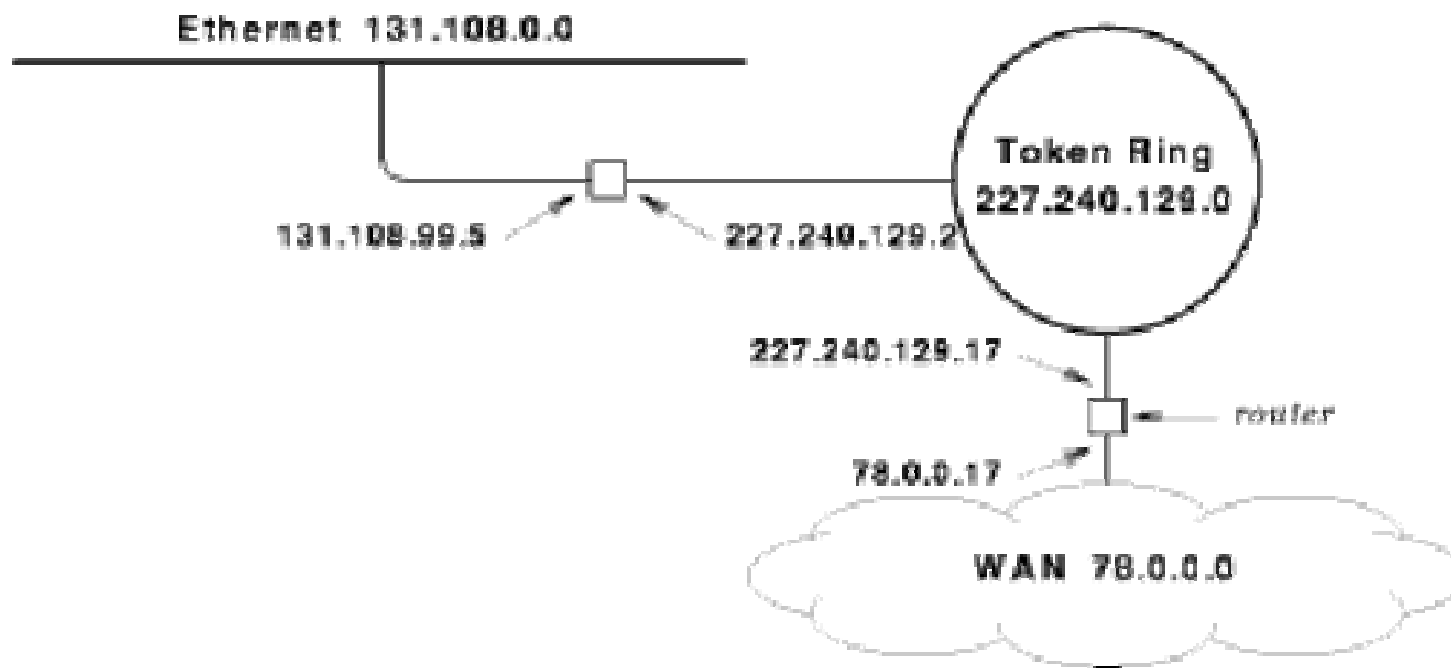
## Endereços especiais

- Existem alguns endereços que são reservados e não são atribuídos a computadores

Prefix	Suffix	Type Of Address	Purpose
all-0s	all-0s	this computer	used during bootstrap
network	all-0s	network	identifies a network
network	all-1s	directed broadcast	broadcast on specified net
all-1s	all-1s	limited broadcast	broadcast on local net
127	any	loopback	testing

# Endereços de roteadores

- Roteadores devem ter endereços IP
  - ◆ Um endereço para cada rede a qual está conectado

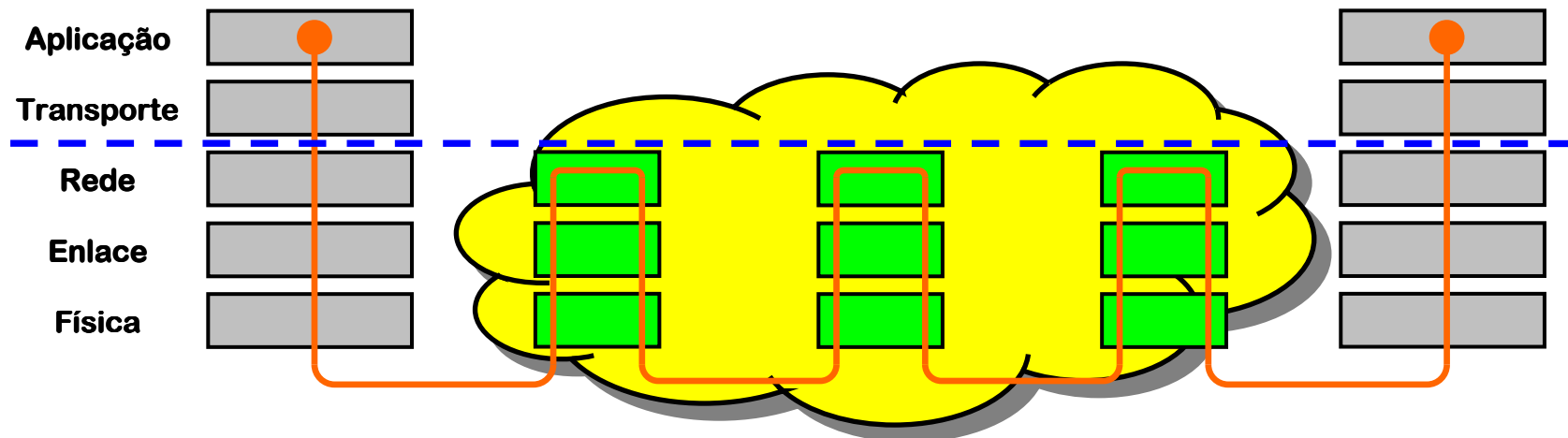


# Computadores *muti-homed*

- Computadores ligados a mais de uma rede física
- Objetivos:
  - ◆ Confiabilidade
  - ◆ Desempenho
- O computador possui um endereço IP em cada rede que se conecta

# Serviços providos pela camada de rede

- Oferecidos pela interface de rede
- Particularmente importante pois é comum ser a interface entre o cliente e uma empresa que é responsável pela sub-rede de comunicação



# Serviços providos pela camada de rede

- Princípios de projeto da camada de rede:
  - ◆ Serviços devem ser independentes da tecnologia da sub-rede de comunicação
  - ◆ Camada de transporte deve ser independente do número, tipo e topologia das sub-redes presentes
  - ◆ Endereços de rede usados pela camada de transporte devem usar um padrão uniforme para LANs e WANs

# Serviço orientado à conexão x Serviço sem conexão

- Discussão antiga
- Adeptos do serviço orientado à conexão:
  - ◆ Companhias de telecomunicações
- Adeptos do serviço sem conexão:
  - ◆ Comunidade da Internet
- Questão principal de fato:
  - ➔ Onde a complexidade deve ficar

# Organização interna da camada de rede

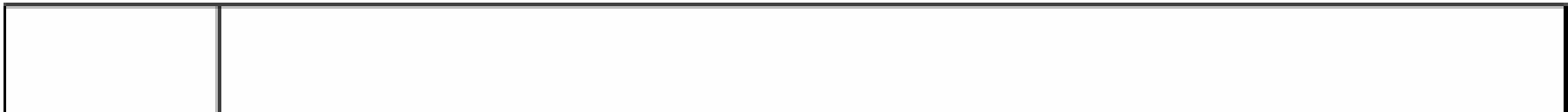
Questão	Sub-rede Datagrama	Sub-rede Circuito Virtual (CV)
Estabelecimento do circuito	Não existe	Necessário
Endereçamento	Cada pacote contém os endereços origem e destino	Cada pacote contém o número do CV
Informação de estado	Sub-rede não tem nenhuma informação	Informações sobre cada CV armazenadas numa tabela
Roteamento	Cada pacote roteado independentemente	Rota escolhida quando o CV é estabelecido; todos pacotes seguem essa rota
Efeito de falhas do roteador	Nenhum a não ser perda de pacotes	Terminam todos CVs que passam pelo roteador
Controle de congestionamento	Difícil	Fácil se houver um número suficiente de buffers

# Pacote IP

- Conhecido também como Datagrama IP
- Formato genérico:

**Header**

**Data Area**



# Pacote IP

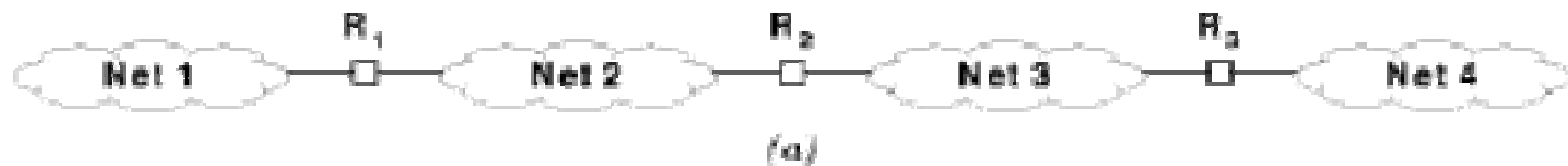
- Tamanho do pacote, respeitado o limite superior, é determinado pela aplicação
  - ◆ Torna o protocolo IP muito flexível
- IPv4 (versão corrente) permite um pacote de até 64 Kbytes

# Envio de Pacote IP

- Cada roteador tem uma tabela de roteamento
- Tabela é inicializada quando o roteador é ligado e deve ser atualizada se a topologia muda ou há uma falha de hardware
- Cada tabela contém um conjunto de entradas que especificam um destino e o próximo roteador a ser usado para alcançar esse destino

# Envio de Pacote IP

- Exemplo de uma internet com quatro roteadores, três redes e a tabela no roteador R2

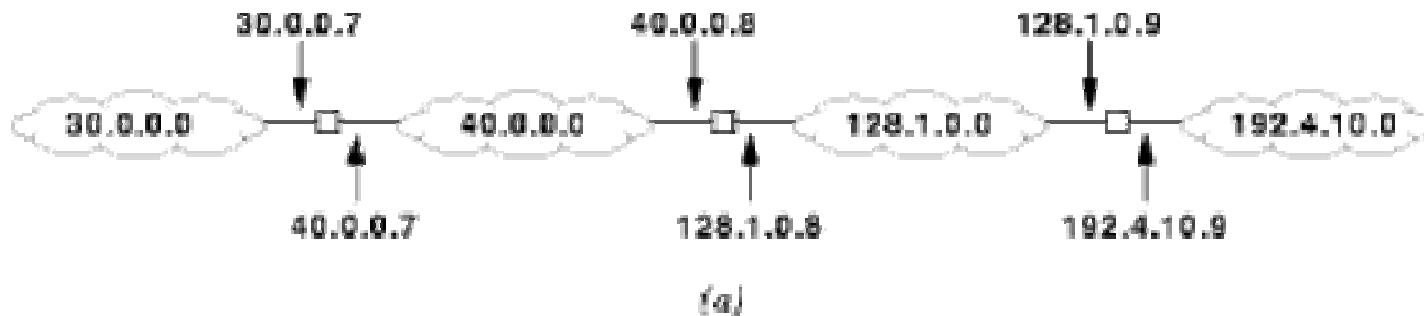


Destination	Next Hop
net 1	R <sub>1</sub>
net 2	deliver direct
net 3	deliver direct
net 4	R <sub>3</sub>

(b)

# Entradas numa tabela de roteamento

- Na prática, roteadores possuem uma rota padrão



Destination	Mask	Next Hop
30.0.0.0	255.0.0.0	40.0.0.7
40.0.0.0	255.0.0.0	deliver direct
128.1.0.0	255.255.0.0	deliver direct
192.4.10.0	255.255.255.0	128.1.0.9

Tabela de roteamento para R2

# Alguns comentários

- Dado um endereço  $D$  de destino, o roteamento é feito verificando cada entrada  $i$  da tabela da seguinte forma:

**se**  $(\text{Máscara}[ i ] \& D) = \text{Destino}[ i ]$

**então** Envie pacote para  $\text{NextHop}[ i ]$

**fimse**;

**se** Não há rota conhecida

**então** Envie pacote para rota padrão

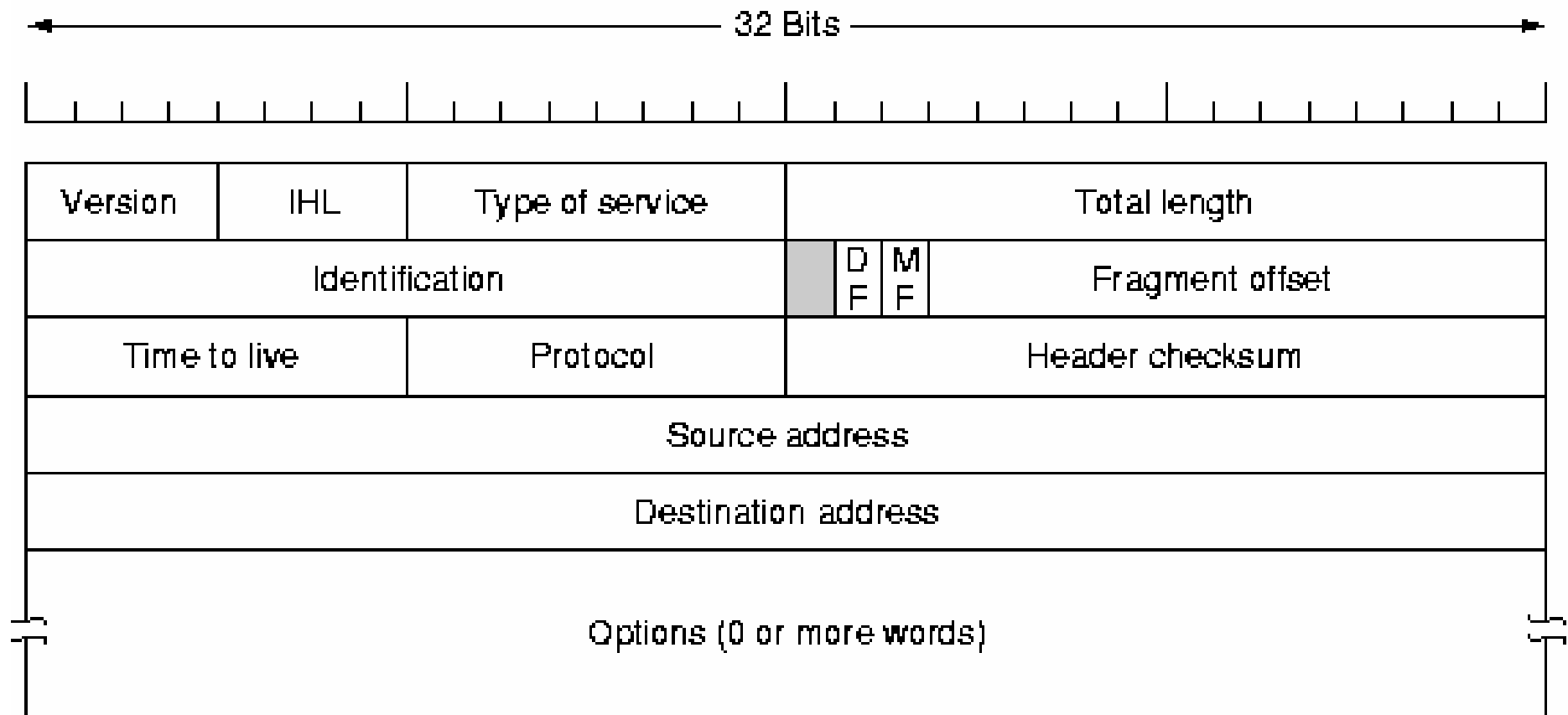
# Alguns comentários

- Endereço de destino x Next-Hop
  - ◆ Endereço de destino indica para quem deve ser entregue o pacote
  - ◆ Endereço de Next-Hop indica para que roteador o pacote deve ser enviado
  - ◆ Esse endereço não aparece no pacote

# Princípio de funcionamento do protocolo IP

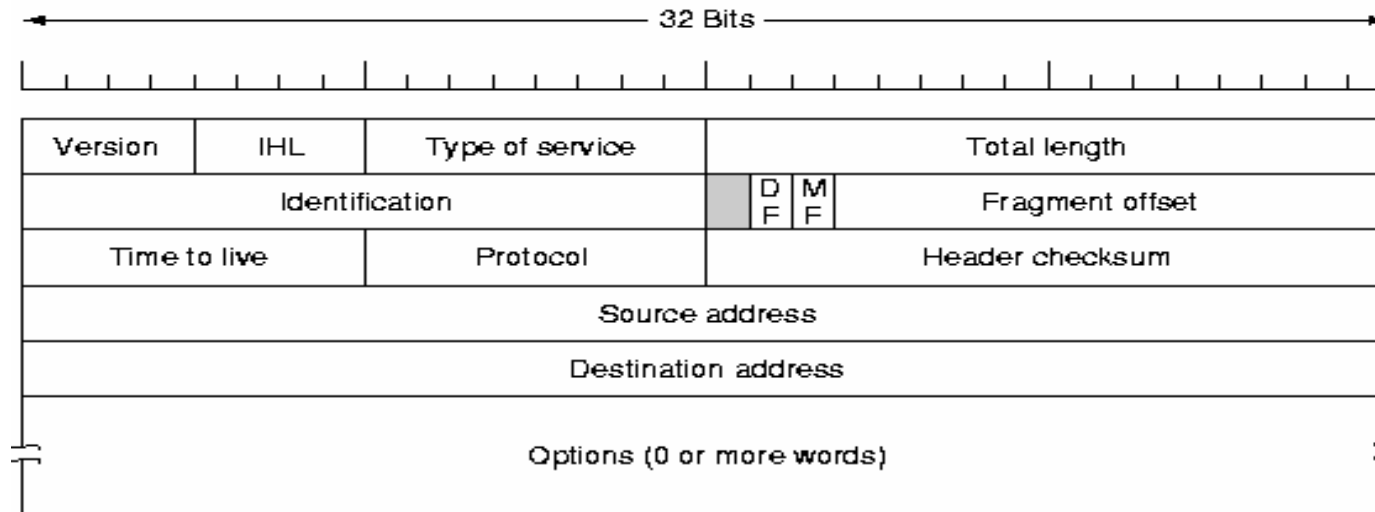
- Entrega será feita com o “maior esforço” (*best-effort delivery*)
- No entanto, IP não garante que não haja:
  - ◆ Duplicação de pacotes
  - ◆ Entrega atrasada ou fora de ordem
  - ◆ Alteração de dados
  - ◆ Perda de pacotes
  - ◆ Protocolos de outros níveis devem tratar desses problemas

# Formato do pacote IP



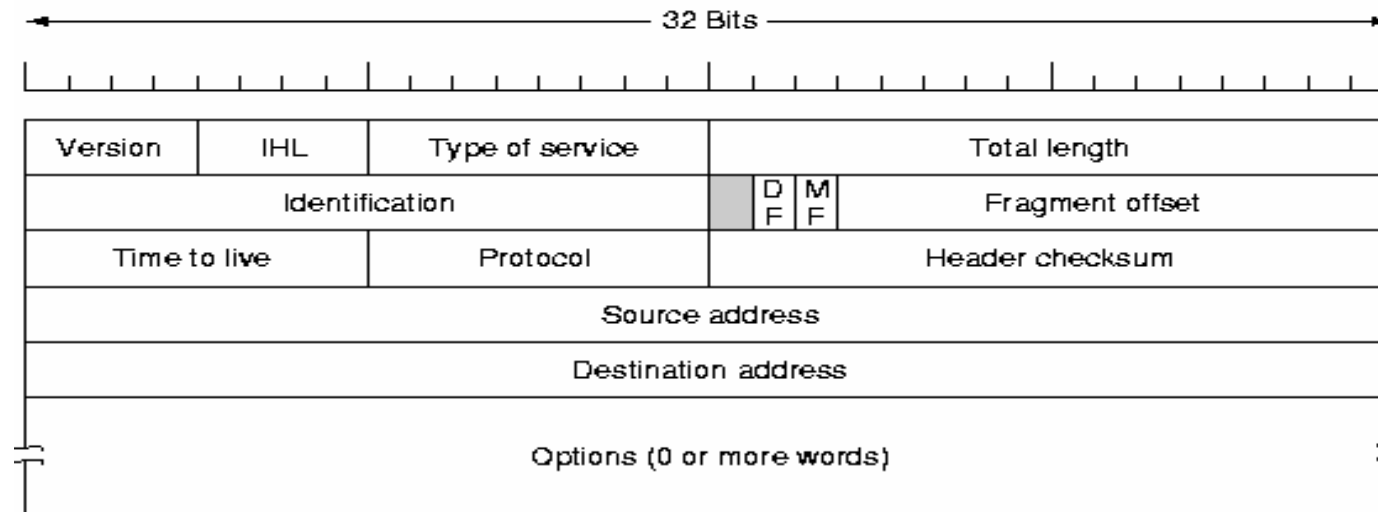
# Formato do pacote IP

- Version:
  - ◆ Indica o número da versão corrente
  - ◆ Permite uma transição “suave” entre versões



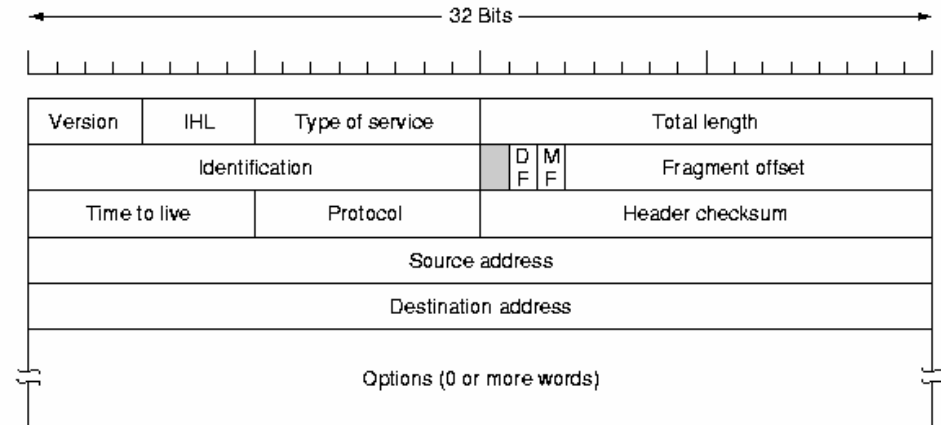
# Formato do pacote IP

- IHL (Tamanho do cabeçalho):
  - ◆ Quantidade de 32 bits presente no cabeçalho
  - ◆ Mínimo: 5 (sem nenhuma opção)
  - ◆ Máximo: 15 a 60 bytes (40 opções)



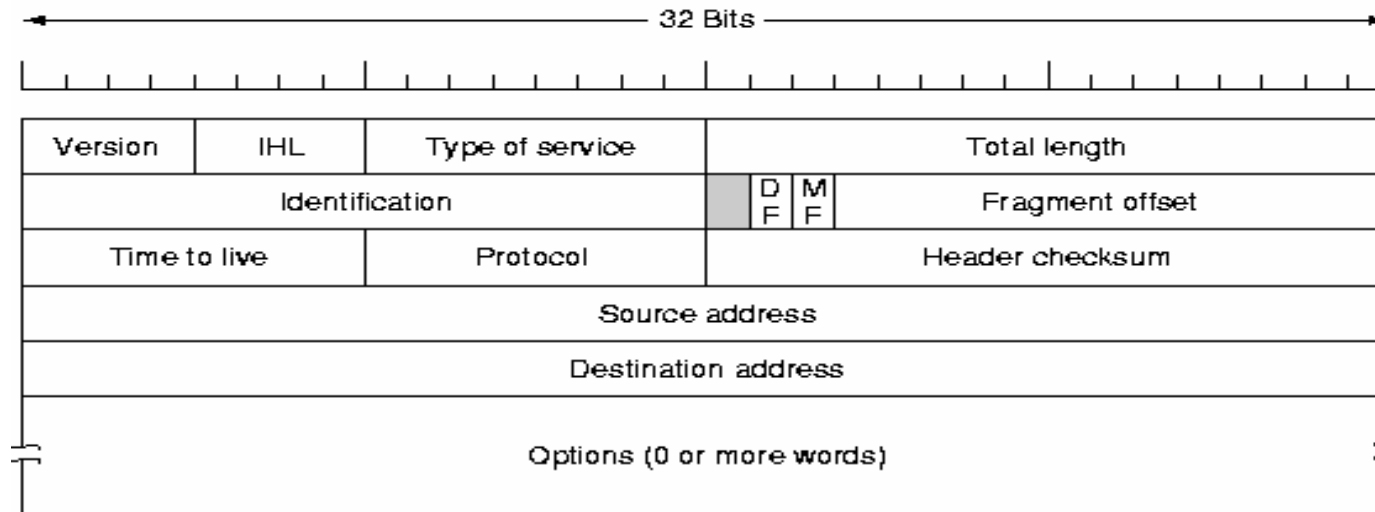
# Formato do pacote IP

- Tipo de serviço:
  - ◆ Prioridade (3 bits)
  - ◆ *Flags* indicam o que é mais importante para a aplicação: menor atraso, maior vazão, maior confiabilidade (3 bits)
  - ◆ Dois bits não usados
  - ◆ Na prática, os roteadores tendem a ignorar este campo



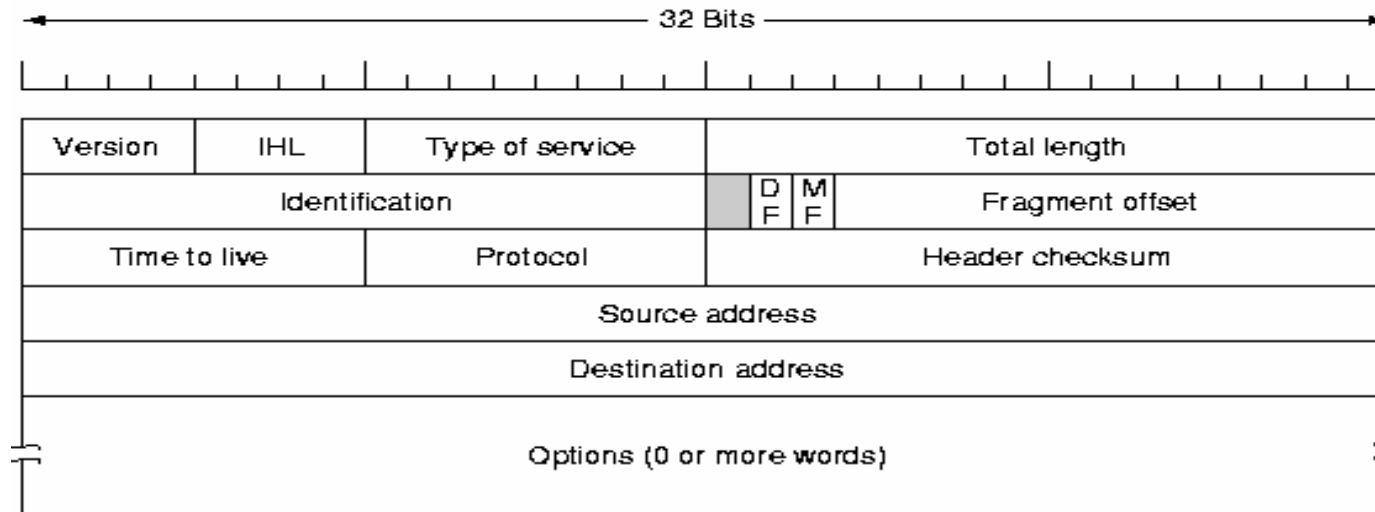
# Formato do pacote IP

- Comprimento total do pacote:
  - ◆ Pode ser até 65535 bytes (64 Kbytes)



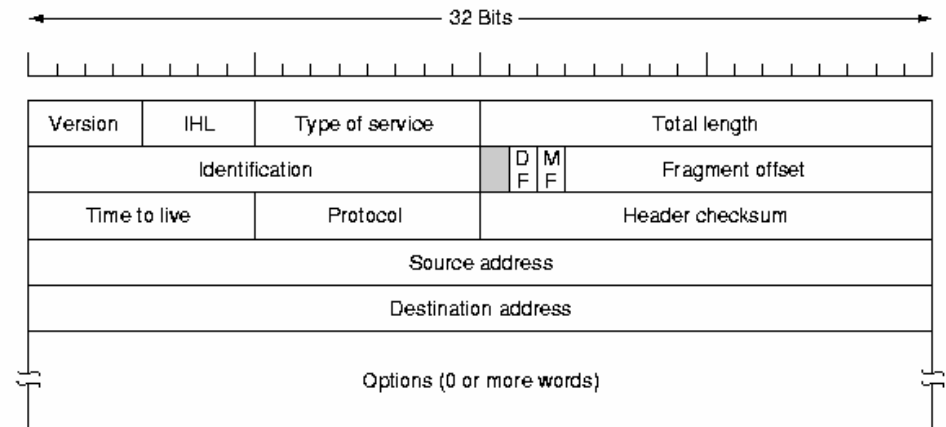
# Formato do pacote IP

- Identificação:
  - ◆ Identifica o fragmento de um datagrama e é usado pelo destinatário para remontagem



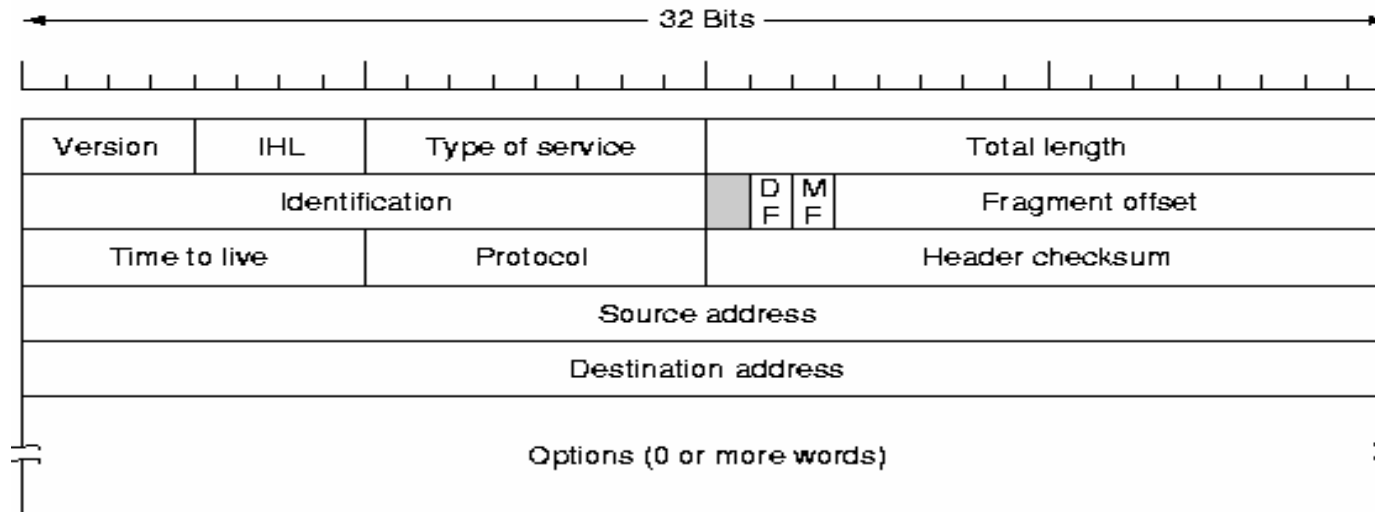
# Formato do pacote IP

- Bit DF (*don't fragment*):
  - ◆ Indica que o pacote não deve ser fragmentado
- Bit MF (*more fragments*):
  - ◆ Todos os fragmentos de um pacote, exceto o último, setam este bit



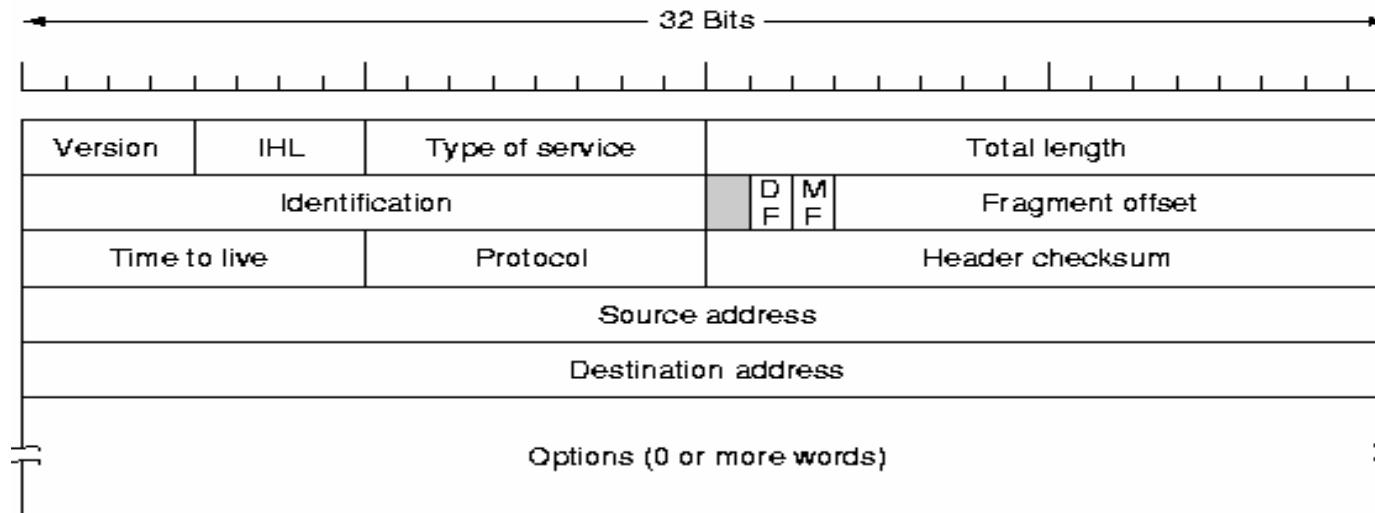
# Formato do pacote IP

- Fragment Offset:
  - ◆ Indica onde o fragmento se encaixa dentro do pacote
  - ◆ Cada fragmento, exceto o último, deve ser múltiplo de 8



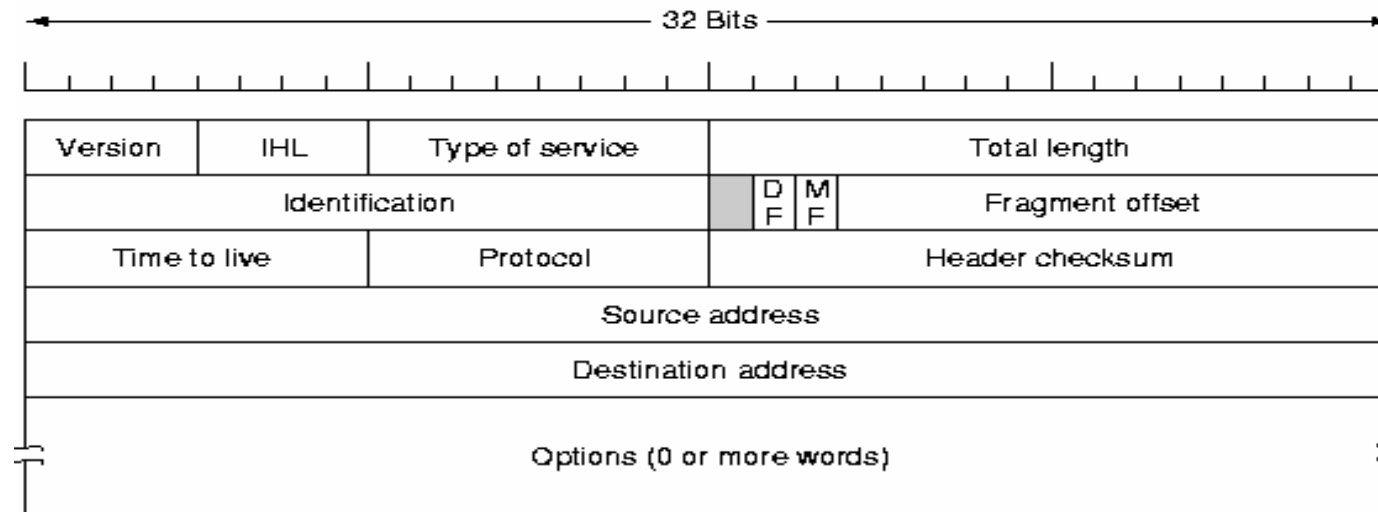
# Formato do pacote IP

- Time To Live:
  - ◆ Teoricamente, indica o tempo máximo que um pacote pode existir, i.e., 255 s
  - ◆ Na prática, indica o número máximo de roteadores que pode passar



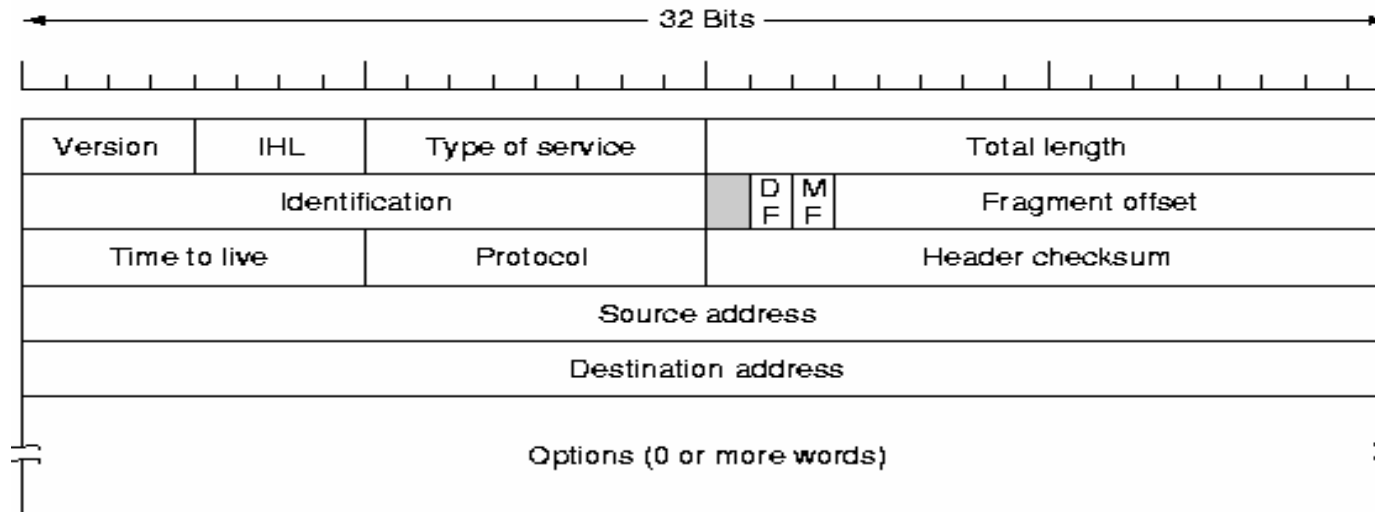
# Formato do pacote IP

- Protocolo:
  - ◆ Indica o protocolo para o qual deve-se passar o pacote
  - ◆ A identificação dos protocolos é dada pela RFC 1700



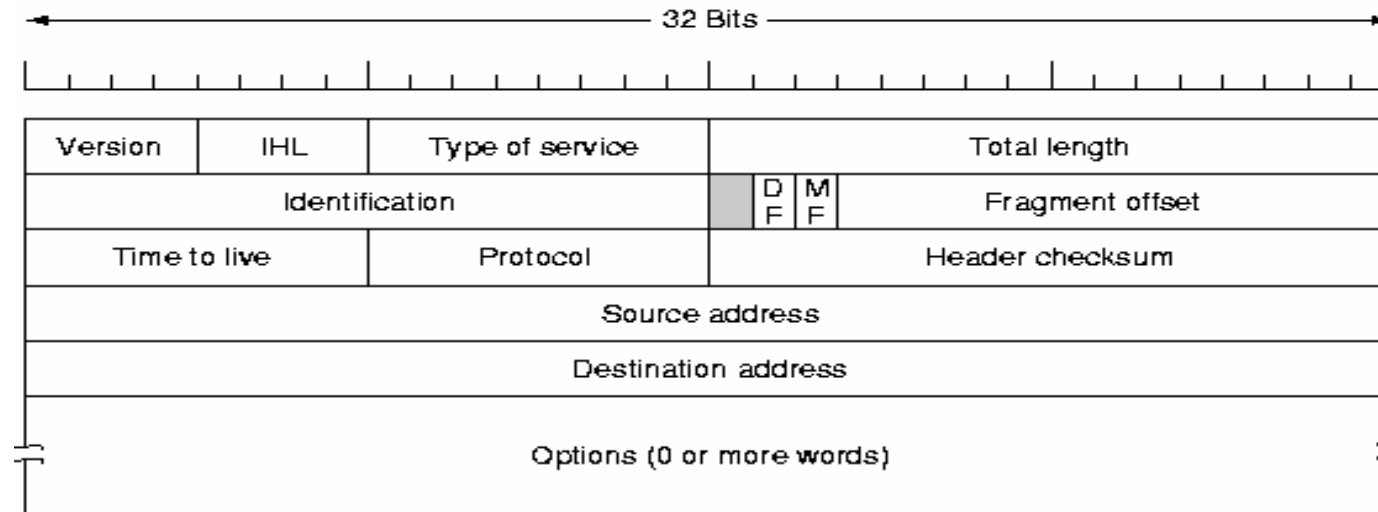
# Formato do pacote IP

- *Checksum* do cabeçalho:
  - ◆ Tem como objetivo aumentar a confiabilidade do pacote entregue às camadas superiores



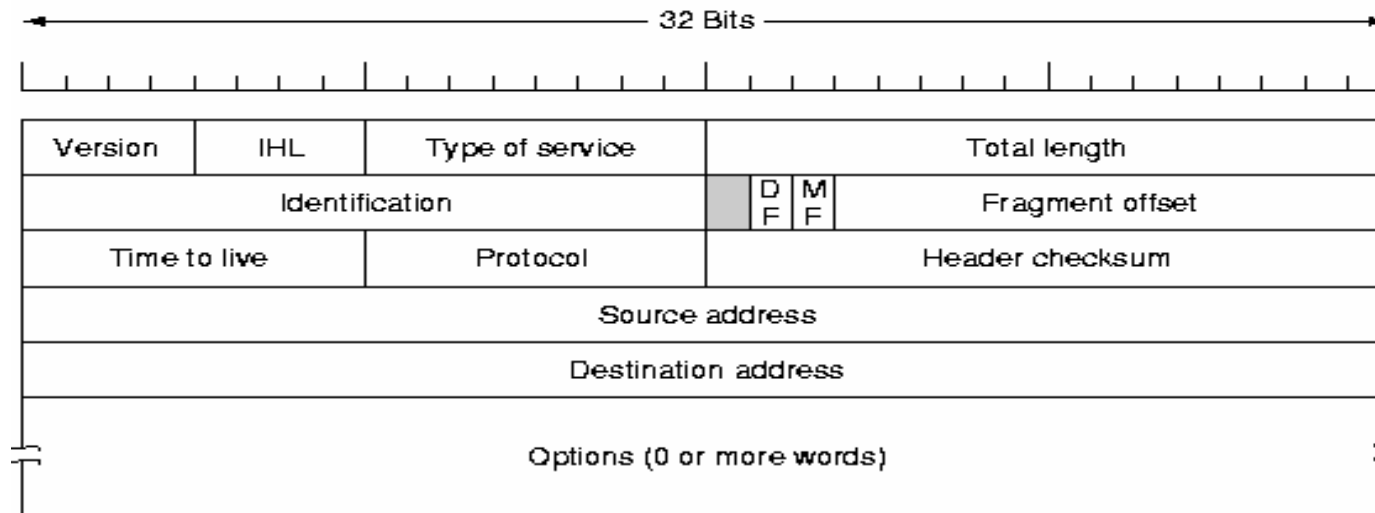
# Formato do pacote IP

- Endereços dos computadores origem e destino



# Formato do pacote IP

- Opções:
  - ◆ Forma de incluir informações não presentes na versão

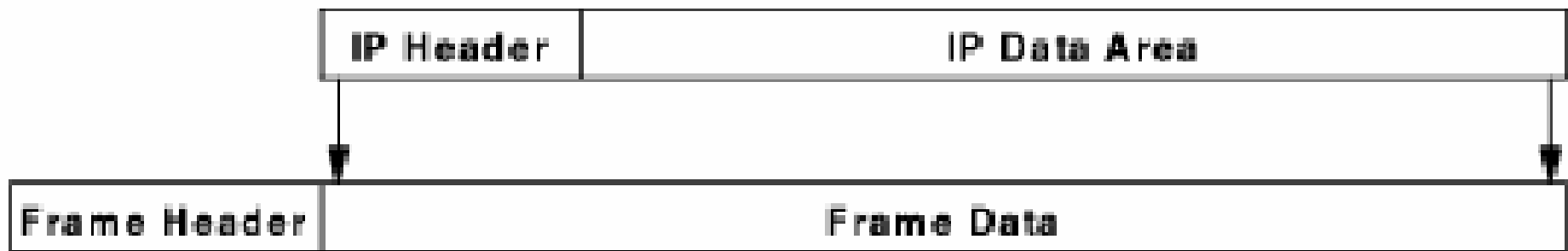


# Transmissão de pacotes e quadros

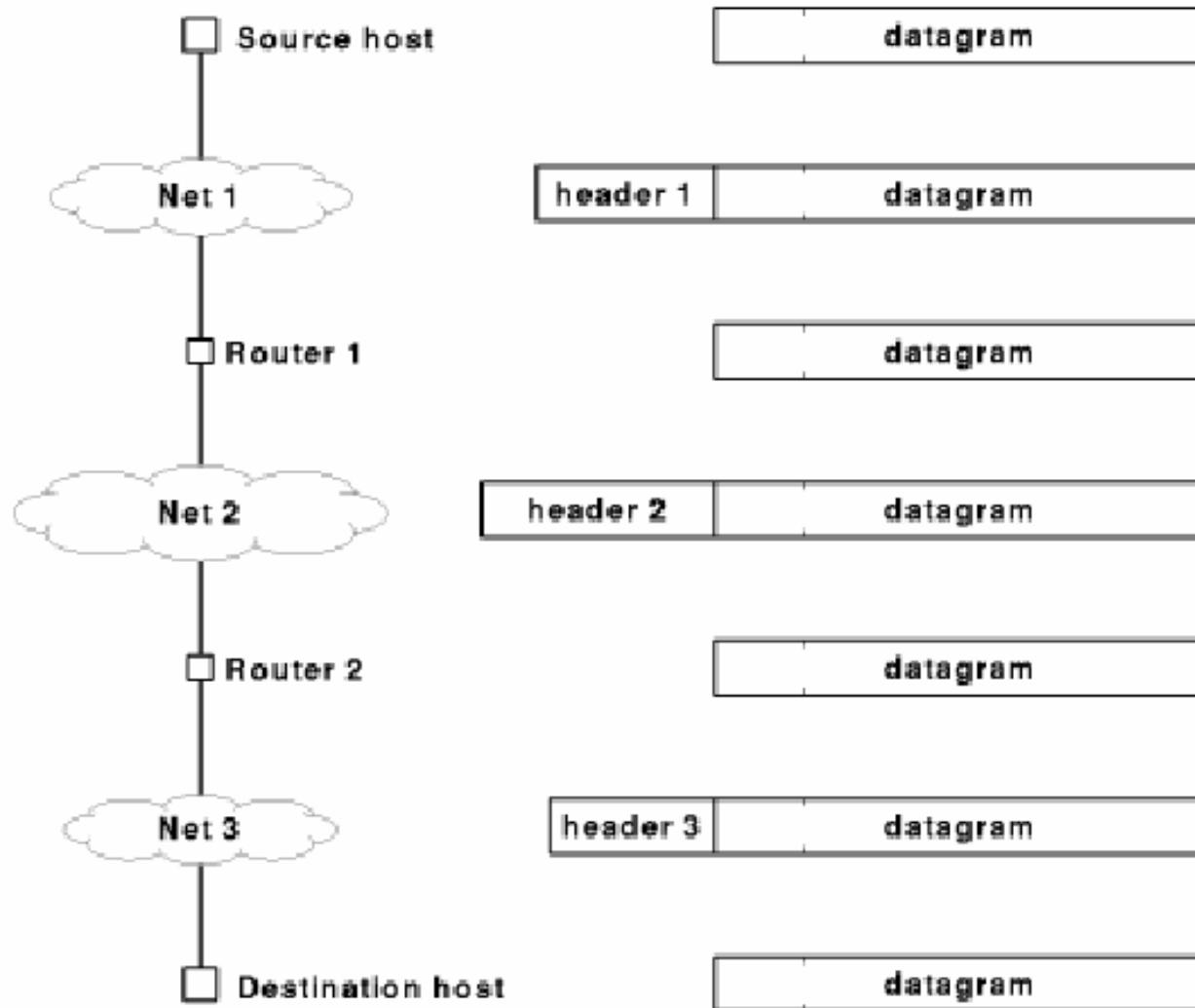
- Como ocorre:
  - ◆ Pacotes devem ser transmitidos por um meio físico
  - ◆ Cada tecnologia de comunicação define um formato de quadro e um esquema de endereçamento físico
- O que deve ser feito:
  - ◆ Pacotes podem ser entregues se seguirem o formato de quadro e o esquema de endereçamento de cada rede por onde passam

# Encapsulamento de pacotes

- Pacotes são encapsulados
- Endereços físicos de rede são obtidos através do protocolo ARP (*Address Resolution Protocol*)

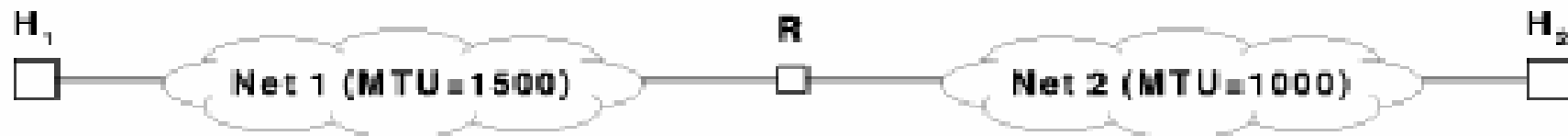


# Encapsulamento de pacotes



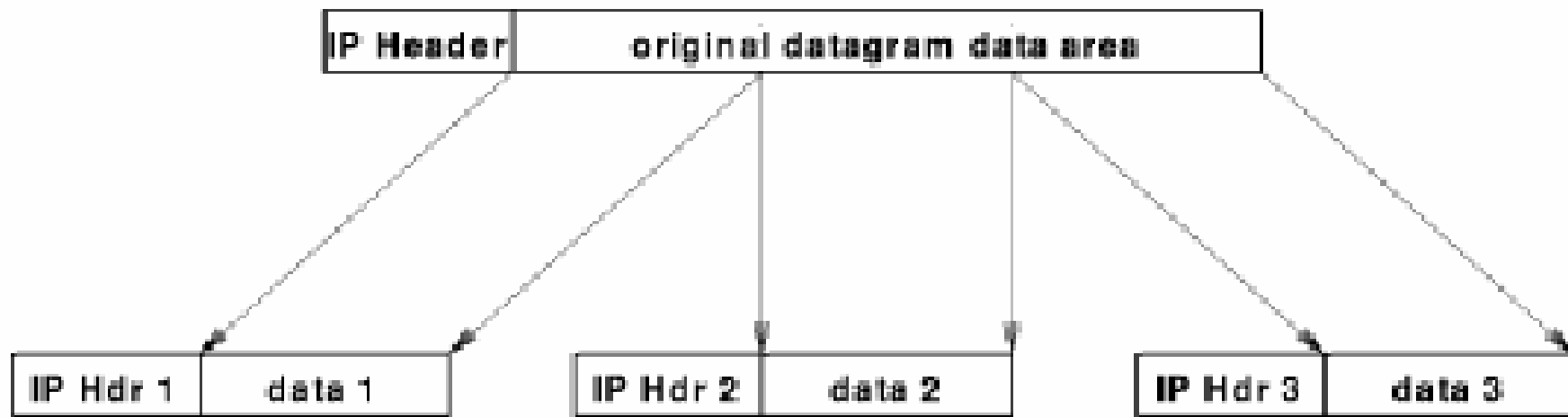
# Fragmentação de pacotes

- A camada de rede de cada protocolo especifica uma quantidade máxima de dados que pode enviar de cada vez
- Este limite é conhecido como MTU (*Maximum Transmission Unit*)
- Pacotes devem ter no máximo esse tamanho



# Fragmentação de pacotes

- Cabeçalho original deve ser preservado na fragmentação
- Bit MF e campos de Identificação e *Fragment Offset* devem ser usados



# Remontagem de pacotes

- Processo inverso ao da fragmentação
- Quem é responsável por essa tarefa?
  - ◆ Computador de destino responsável por essa tarefa
- O que ocorre se fragmentos são perdidos, chegam fora de ordem ou atrasados?
- RX não tem como informar TX para enviar um fragmento já que TX não conhece nada sobre fragmentação

# Remontagem de pacotes

- Solução:
  - ◆ RX ao receber o primeiro fragmento de um pacote inicializa um temporizador
  - ◆ Se todos os fragmentos não chegam antes do temporizador se esgotar então todos os fragmentos são ignorados

# Fragmentação de fragmentos

- É possível fragmentar fragmentos?
  - ◆ Sim. O protocolo IP não faz distinção de níveis de fragmentação
- Na prática, a remontagem pode ser feita mais rápida se todos os fragmentos forem do mesmo tamanho

# IPv6

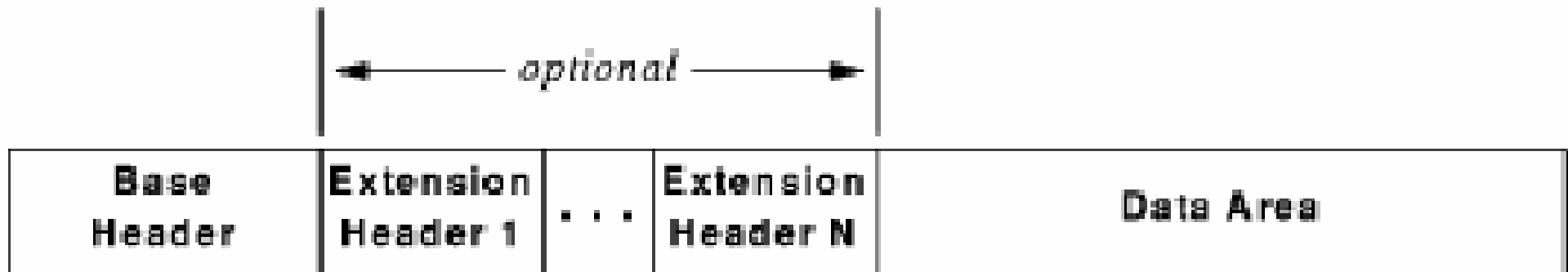
- Motivação básica para criar nova versão:
  - ◆ Falta de números IPs
- Motivação secundária:
  - ◆ Suportar novas aplicações

# Características principais

- Tamanho do campo de endereço:
  - ◆ de 32 para 128 bits
- Formato do cabeçalho:
  - ◆ Quase todos os campos mudaram
  - ◆ Alguns foram trocados

# Características principais

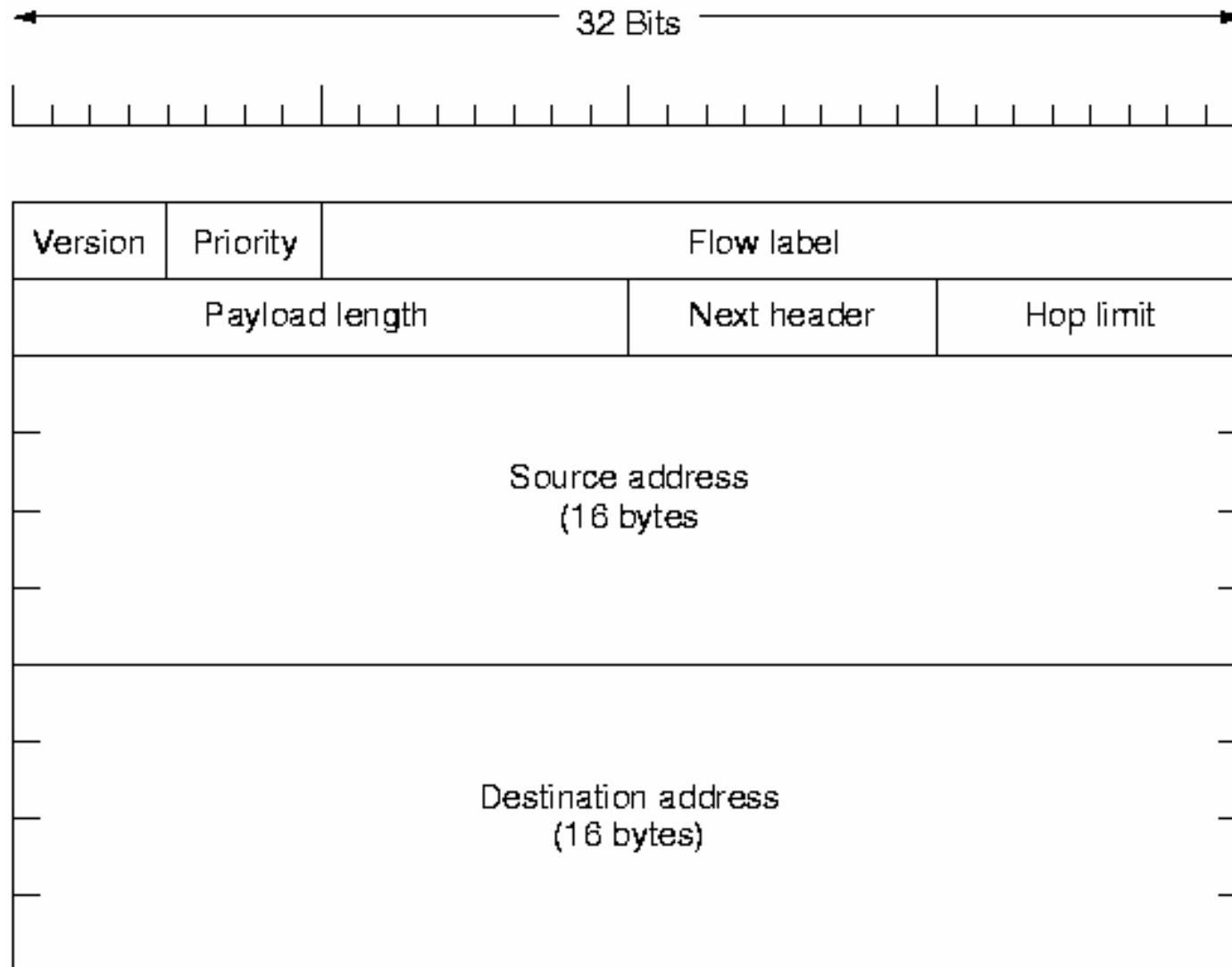
- Cabeçalhos de extensão:
  - ◆ Existe um cabeçalho básico, seguido de zero ou mais cabeçalhos de extensão



# Características principais

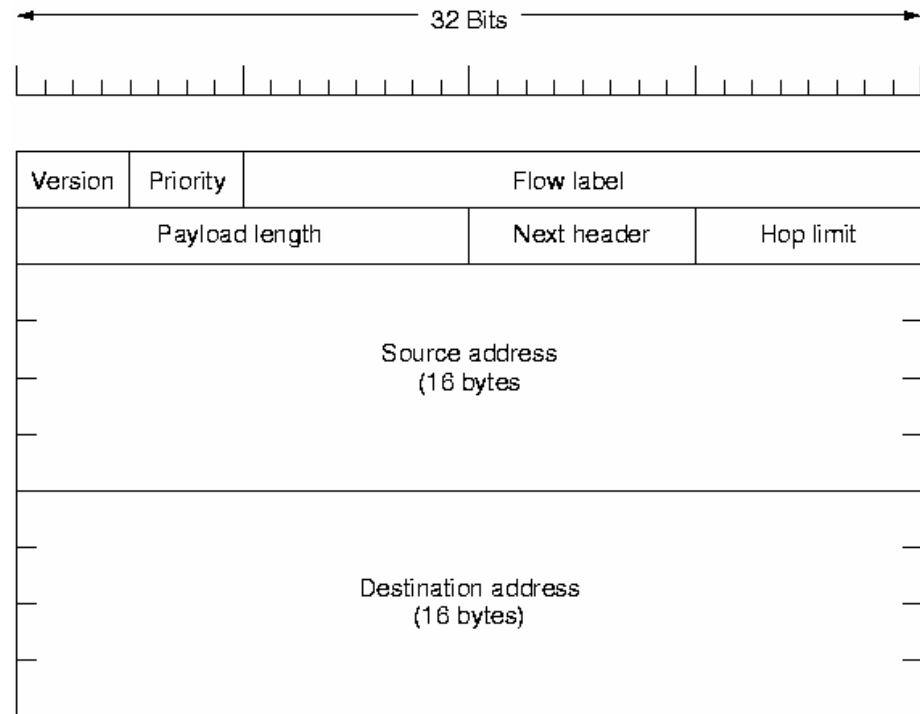
- Suporte para áudio e vídeo
  - ◆ É possível associar pacotes a um caminho específico de alta qualidade na rede
- Mecanismo que permite o protocolo evoluir :
  - ◆ É possível estender o protocolo para incluir novas funcionalidades de maneira simples

# Cabeçalho do IPv6



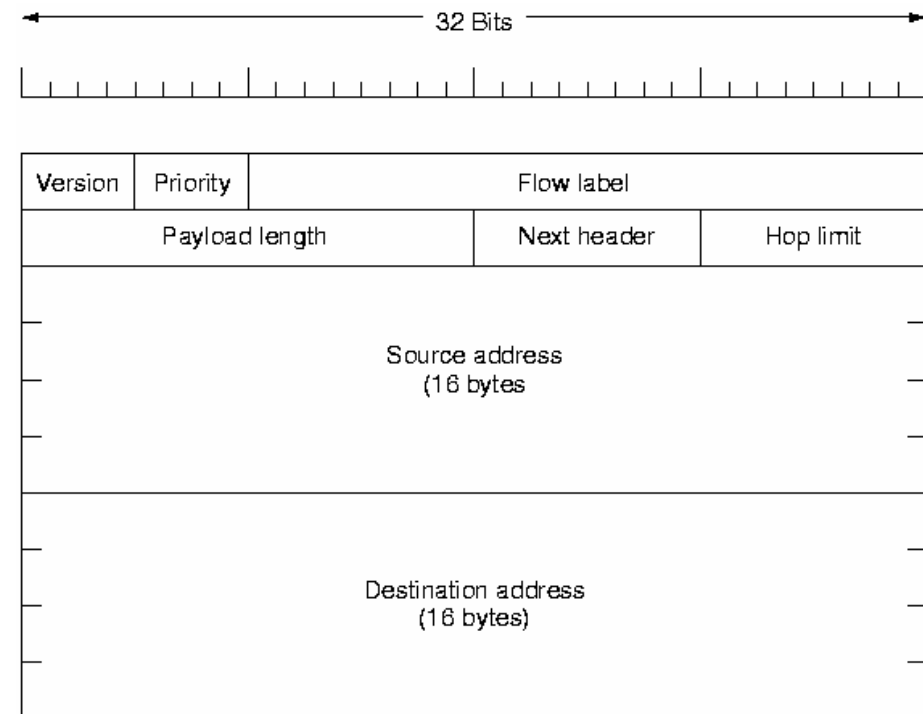
# Cabeçalho do IPv6

- Prioridade:
  - ◆ 0-7: tráfego que pode sofrer atraso
    - ▶ 1: news
    - ▶ 4: ftp
    - ▶ 6: telnet
  - ◆ 8-15: tráfego de tempo real



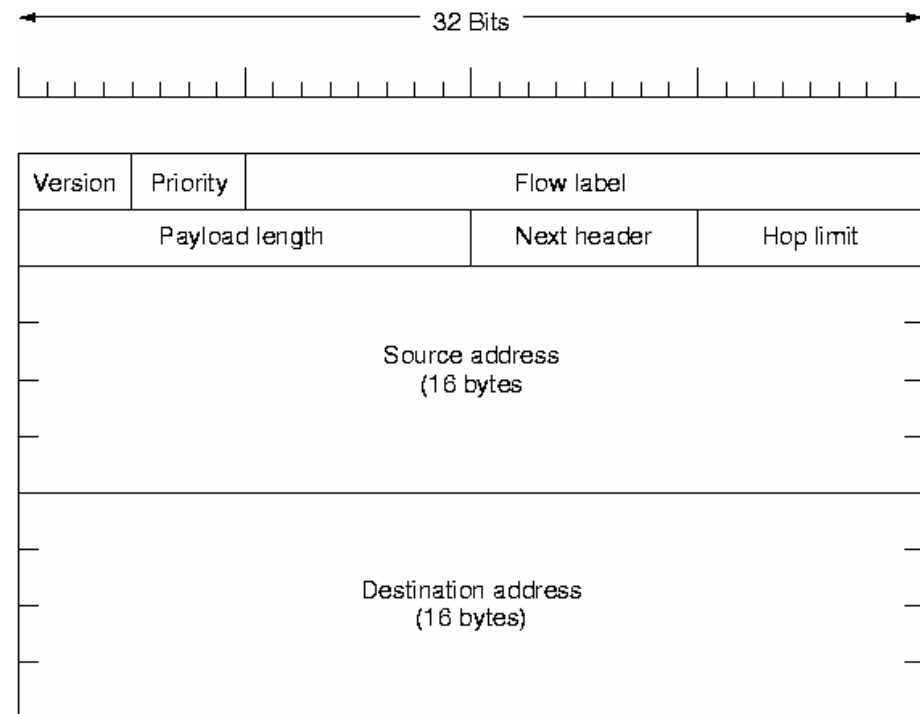
# Cabeçalho do IPv6

- Flow Label:
  - ◆ Tem como objetivo permitir que a origem e o destino estabeleçam uma pseudo-conexão com certos requisitos



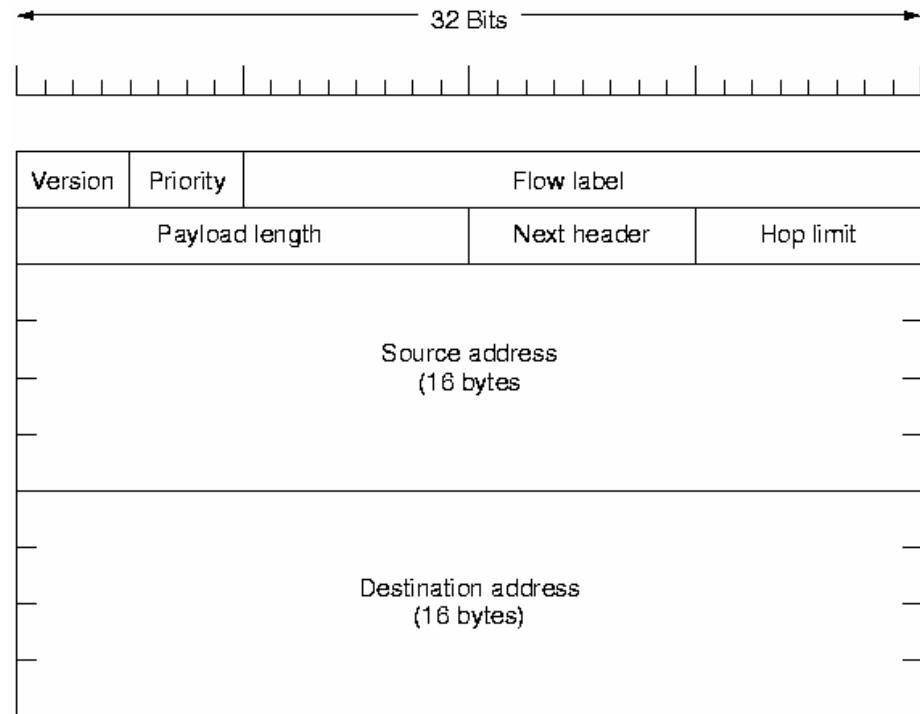
# Cabeçalho do IPv6

- Tamanho de *payload*:
  - ◆ Quantidade de bytes após o cabeçalho



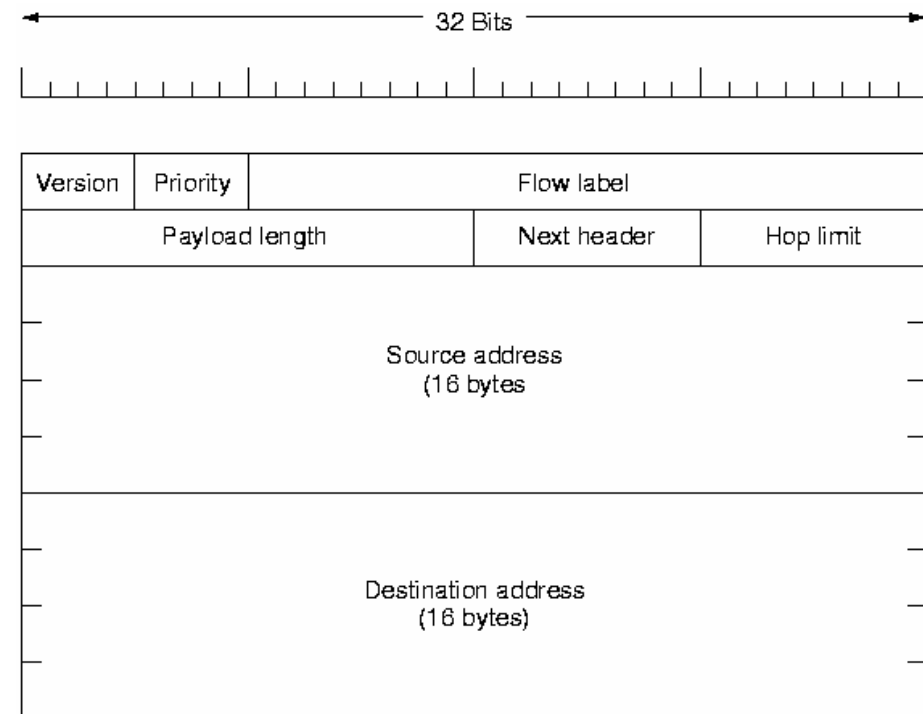
# Cabeçalho do IPv6

- Próximo cabeçalho:
  - ◆ Indica, se houver, qual cabeçalho de extensão segue o corrente



# Cabeçalho do IPv6

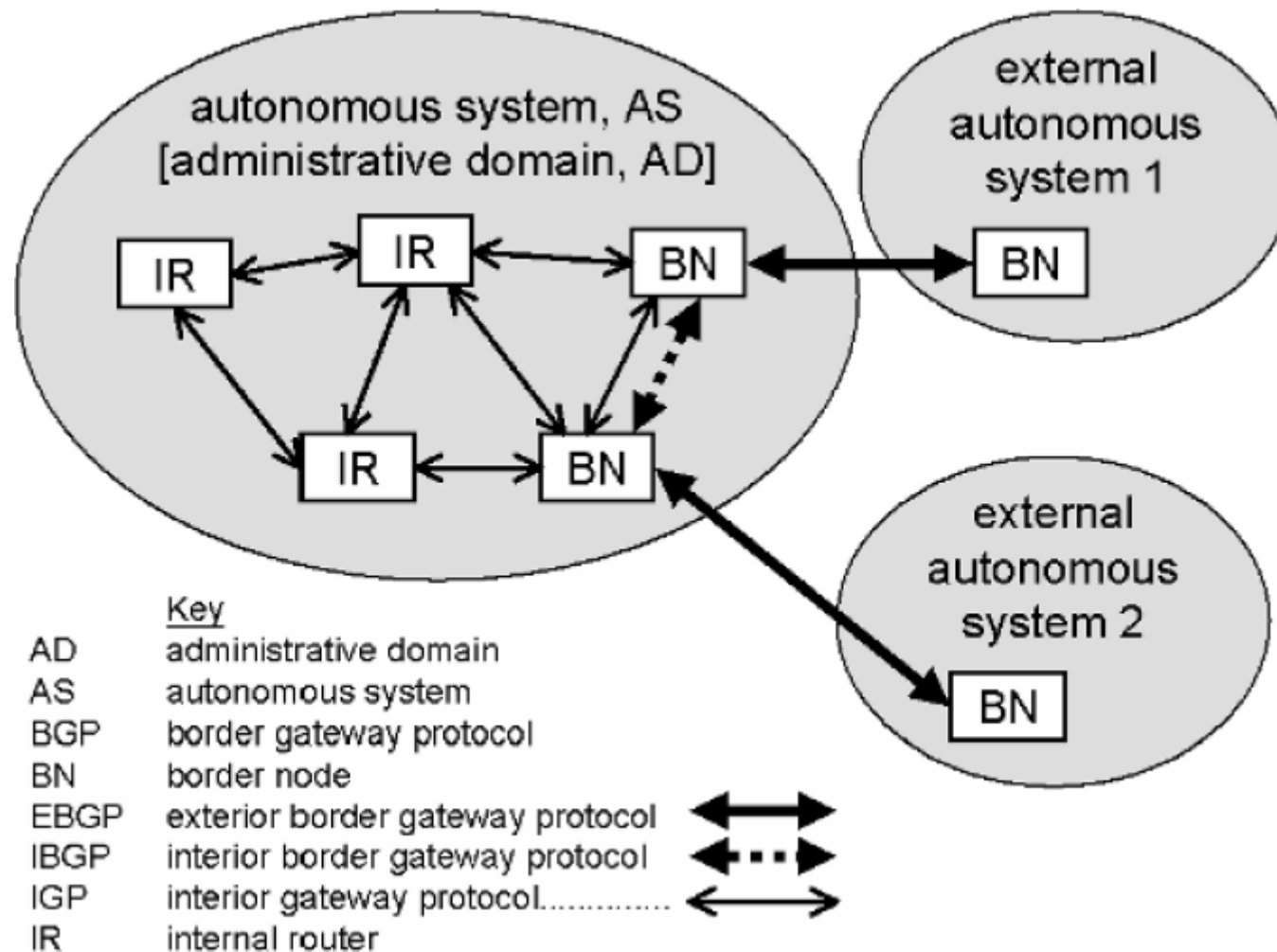
- Hop Limit:
  - ◆ Limite máximo de roteadores que um pacote pode passar



# Protocolos de roteamento da Internet

- Cada AS (*Autonomous System*) pode usar seu próprio algoritmo de roteamento
- Algoritmos de roteamento:
  - ◆ Dentro de um AS: interior gateway protocol (IGP)
  - ◆ Entre AS's: exterior gateway protocol (EGP)

# Protocolos de roteamento da Internet



# Protocolos de roteamento da Internet

## Evolução dos protocolos IGP

- RIP (protocolo de vetor de distância – *distance vector protocol*) baseado no algoritmo Bellman-Ford
- Em 1979, é substituído por um protocolo de estado de enlace (*link state protocol*)
- A partir de 1988, começa a ser substituído pelo OSPF (Open Shortest Path First) definido na RFC 2328

# Protocolos de roteamento da Internet

## Vetor de distância

- Calcula a distância e direção (i.e., *vetor* — o destino ou *next hop*) a partir de cada roteador origem para cada destino possível
- Informações são passadas para nós vizinhos, que calculam suas tabelas de roteamento
- Exemplos:
  - ◆ RIP (*Routing Information Protocol*)
  - ◆ IGRP (*Interior Gateway Routing Protocol*) da Cisco
  - ◆ BGP (*Border Gateway Protocol*)



# Protocolos de roteamento da Internet

## Estado de enlace

- Roteador envia informação sobre o estado de cada enlace para todos os roteadores (broadcast)
- Roteador constrói um “mapa” completo da topologia da rede
- Usando seu mapa, roteador pode determinar o caminho mais curto para cada destino
  - ◆ Tenta construir uma tabela de roteamento ótima
- Exemplos:
  - ◆ IS-IS (*Intermediate System-Intermediate System*)
  - ◆ OSPF

# Protocolos de roteamento da Internet

## Requisitos até chegar ao OSPF

1. Algoritmo público e não proprietário
2. Deveria tratar diferentes métricas de distância (e.g., distância física e atraso)
3. Deveria ser adaptativo, i.e., adaptar automática e rapidamente a mudanças topológicas da rede
4. Deveria suportar o roteamento baseado no tipo de serviço (campo *Type of Service* do pacote IP)
  - ◆ Campo incluído no OSPF mas não utilizado, o que levou a sua remoção

# Protocolos de roteamento da Internet

## OSPF

- Baseado no SPF (*Shortest Path First* ou algoritmo de *Dijkstra*)
- Características:
  - ◆ Usa um parâmetro de custo como base do caminho mais curto, que pode ser a largura de banda, atraso, carga, confiabilidade, etc
  - ◆ Permite o uso de múltiplos caminhos de mesmo custo por questão de compartilhamento de carga
  - ◆ Atualizações ocorrem quando há uma mudança real
  - ◆ Converge rápido com um custo mínimo em termos de mensagens
  - ◆ Demanda mais processamento e capacidade de memória
  - ◆ Tráfego OSPF é sempre autenticado — somente roteadores “confiáveis” fazem parte do processo de roteamento

# Protocolos de roteamento da Internet

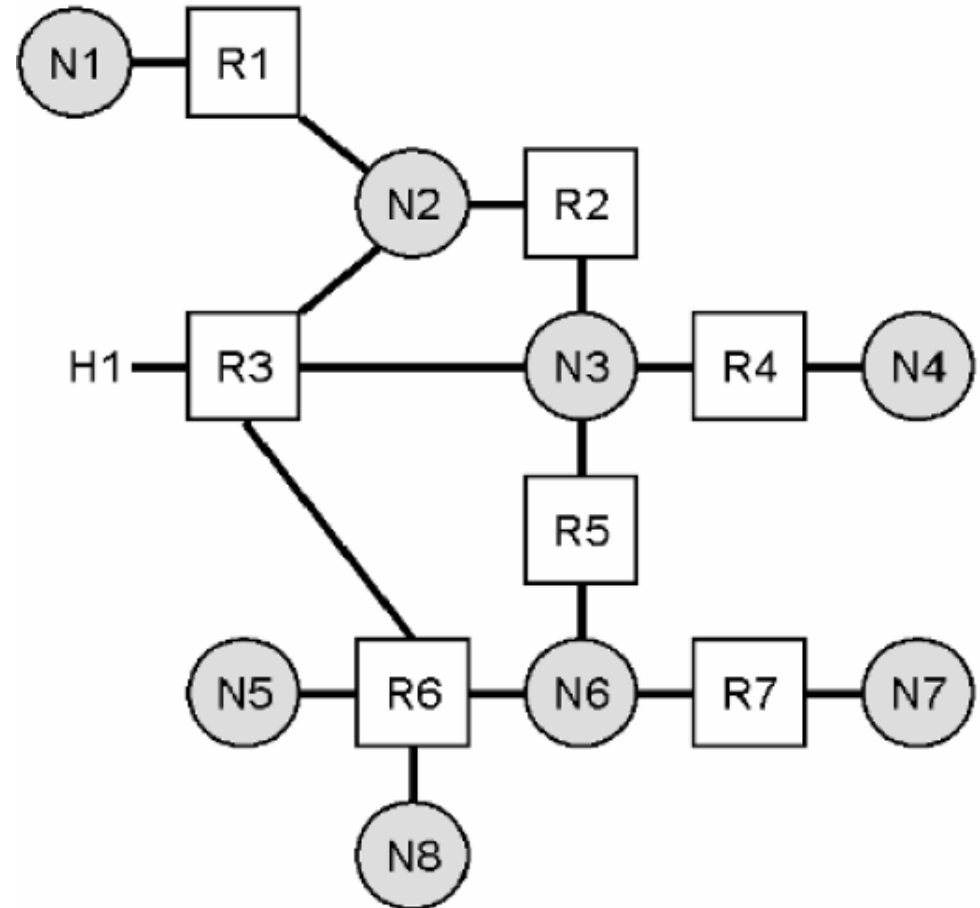
## OSPF

- Roteadores mantêm um banco de dados de estado de enlace
  - ◆ Contém informação sobre a rede como um todo e o estado (i.e., custo do enlace) de cada de link
- Usando o banco de dados, cada roteador calcula separadamente uma árvore de caminho mais curto tendo o roteador como raiz
  - ◆ Isto permite ao roteador determinar sua tabela de roteamento calculando a rota de caminho mais curto para cada nó folha da árvore

# Protocolos de roteamento da Internet

## OSPF: Exemplo

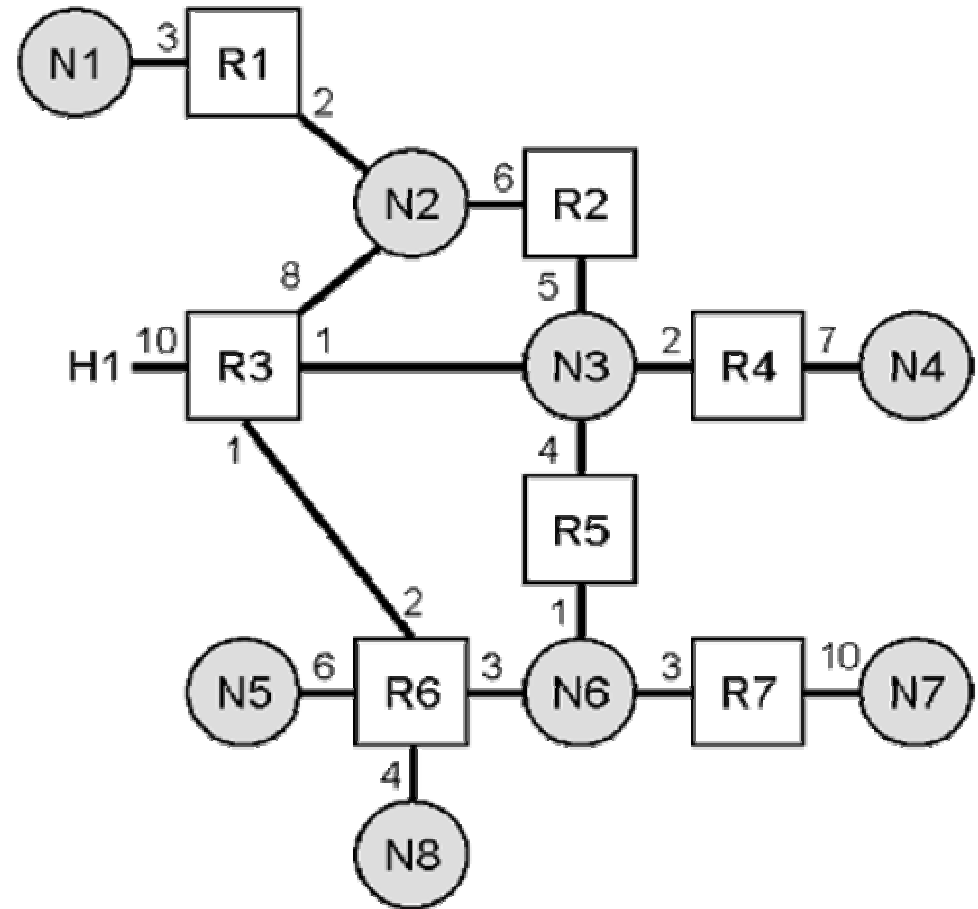
- Roteadores R1 a R7
- Redes N1 a N8
- Enlace PP entre R3 e R6
- Computador H1, dentre outros



# Protocolos de roteamento da Internet

## OSPF: Exemplo

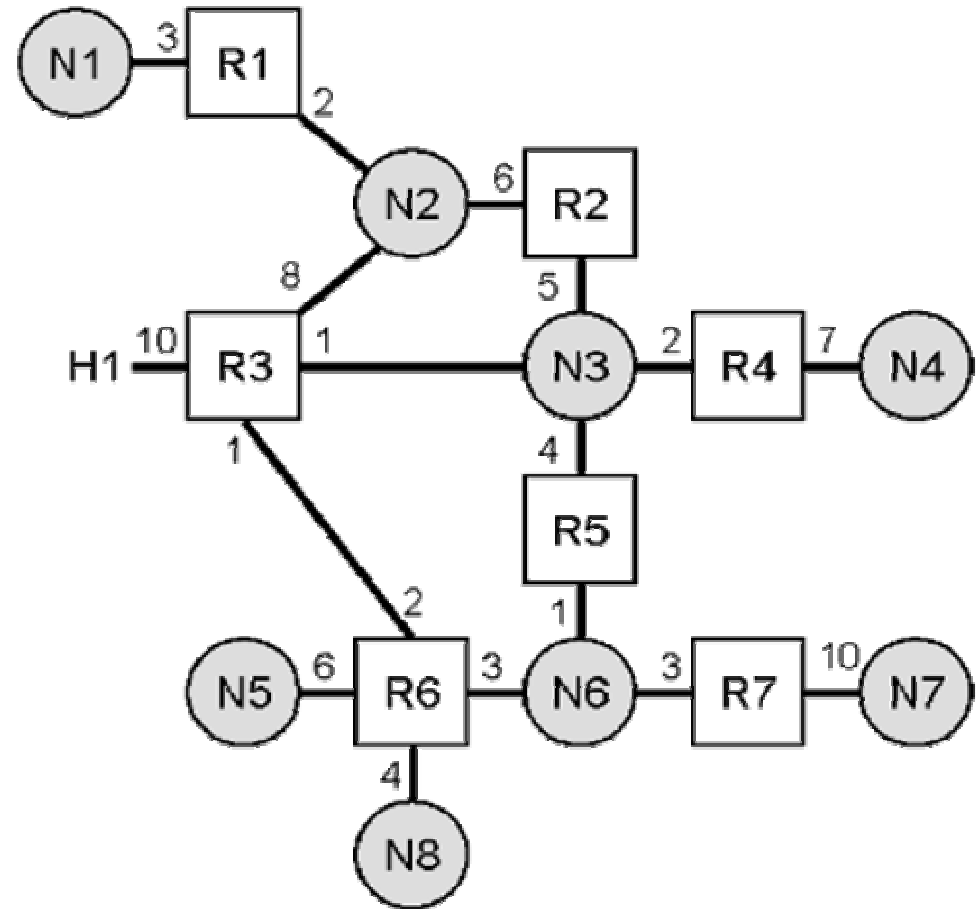
- Cada roteador atribui um custo (estado) para cada enlace de saída
- Pode ser feito de formas diferentes
- O custo *default* é  $10^9$  dividido pela taxa do canal
  - ◆ Gigabit Ethernet: 1
  - ◆ Canal 64 kbps: 15 625



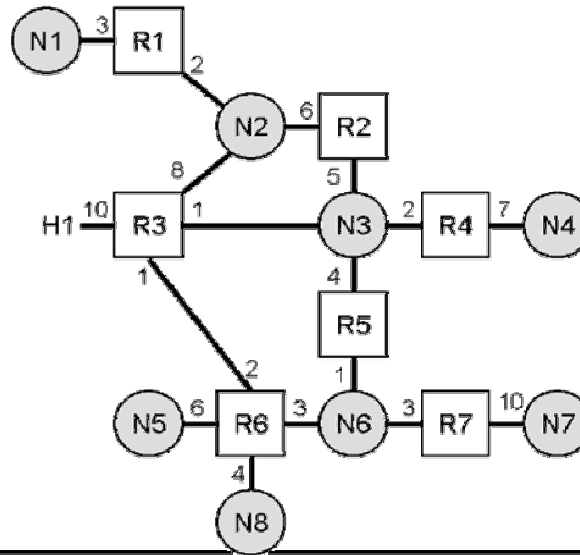
# Protocolos de roteamento da Internet

## OSPF: Exemplo

- Os custos dos roteadores R1, R2 e R3 para a rede N2 são diferentes
  - ◆ Custos representam uma grandeza adimensional
  - ◆ Valor menor representa uma rota “mais curta”
- Não há custo para sair de uma rede e entrar num roteador
  - ◆ Rota de R1 para R2 tem custo 2

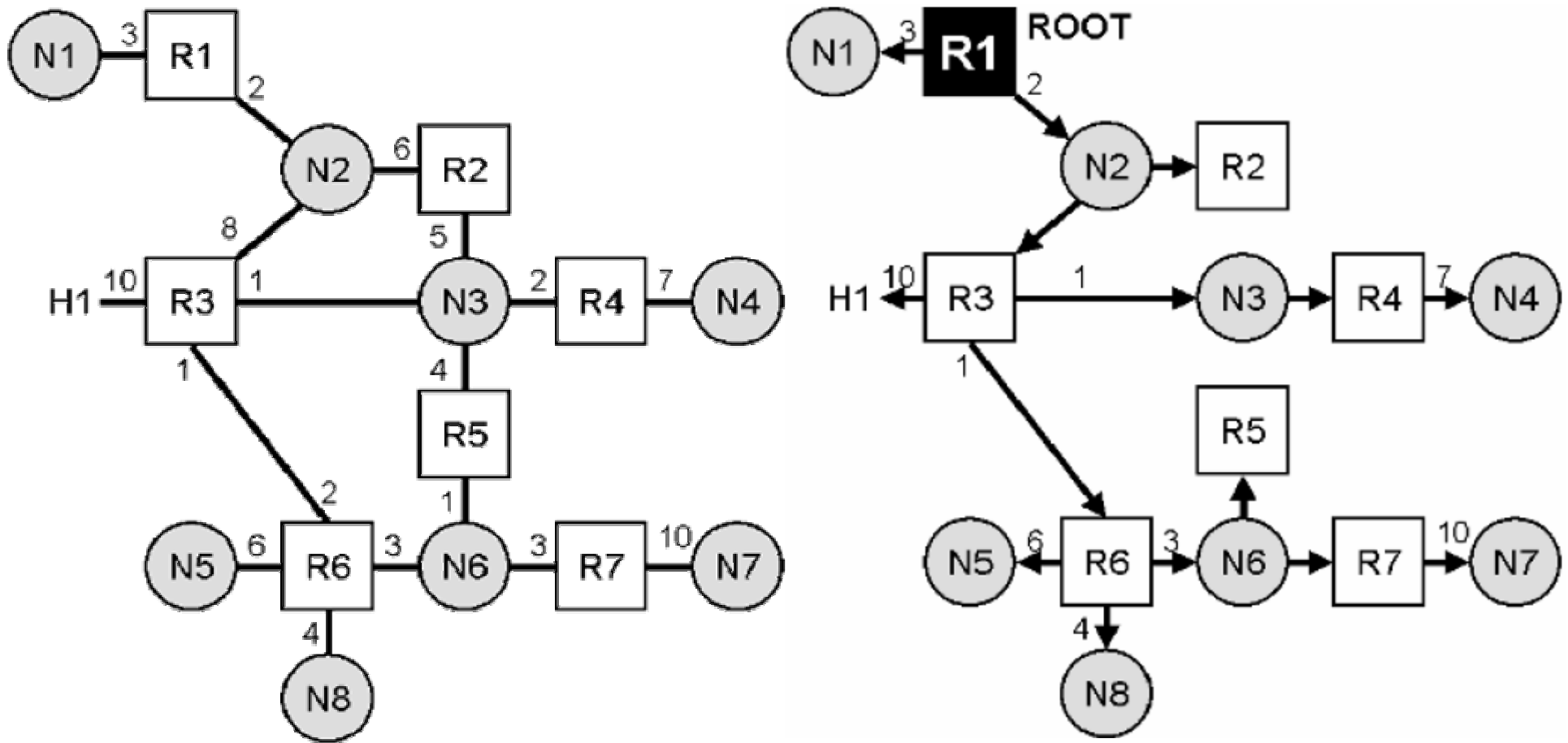


# OSPF: Exemplo

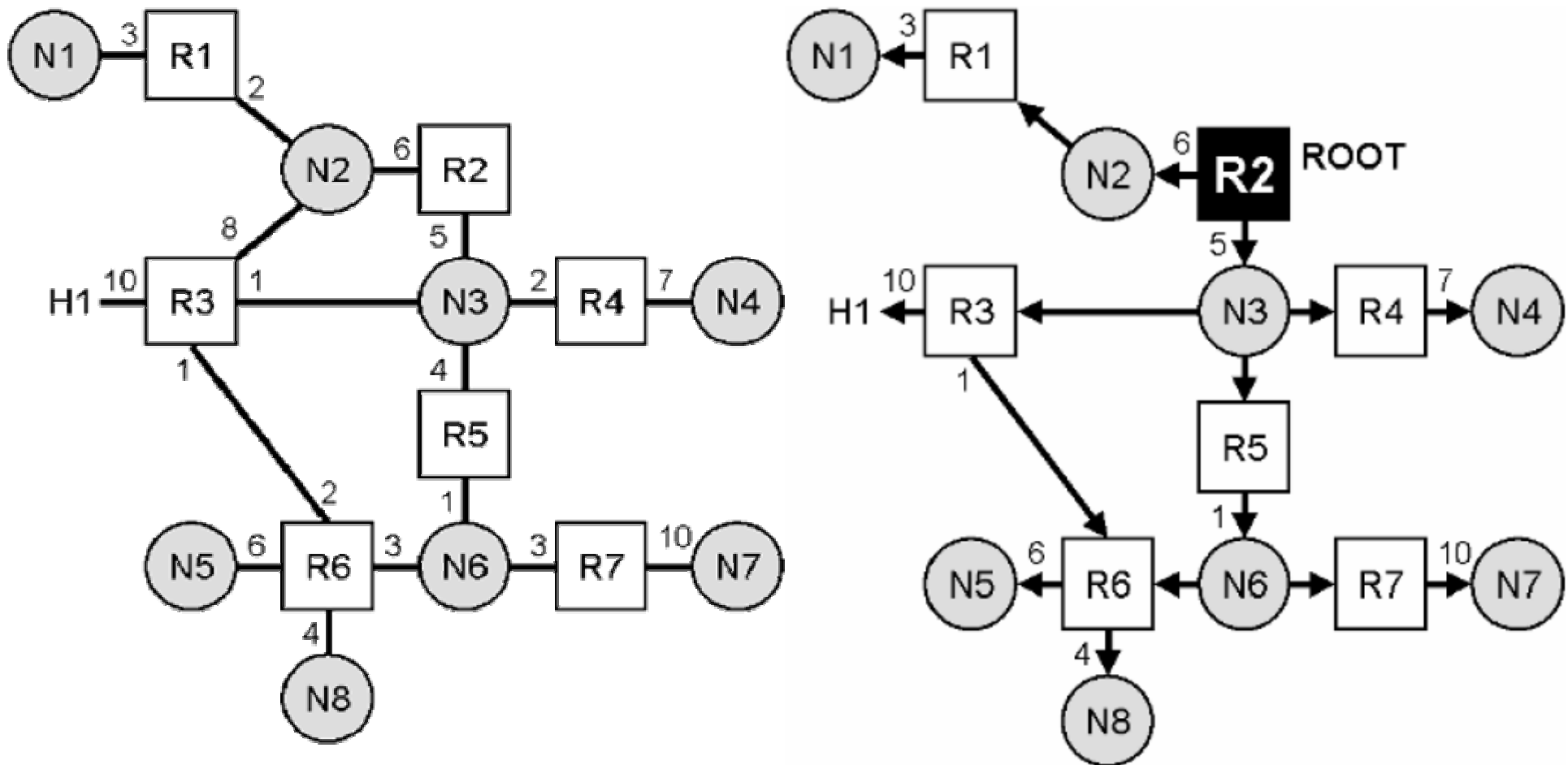


	R1	R2	R3	R4	R5	R6	R7	N1	N2	N3	N4	N5	N6	N7	N8	H1
<b>T</b>																
<b>O</b>								0	0							
R1																
R2									0	0						
R3						2			0	0						
R4										0	0					
R5										0	0		0			
R6			1									0	0		0	
R7													0	0		
N1	3															
N2	2	6	8													
N3		5	1	2	4											
N4				7												
N5						6										
N6					1	3	3									
N7							10									
N8						4										
H1			10													

# Árvore de caminho mais curto tendo R1 como raiz



# Árvore de caminho mais curto tendo R2 como raiz



# Border Gateway Protocol (BGP)

- BGP introduz dois conceitos:
  - ◆ Alcançabilidade (*reachability*): indica se uma rota para uma determinada faixa de endereços IP externos ao AS é conhecida ou não; se IP não é conhecido então esse IP não é alcançável
  - ◆ *Política* de roteamento (*routing policy*)
- Roteador de borda tenta descobrir o melhor caminho para alcançar o destino, independentemente do número de AS's que tenha que passar

# BGP

- Problema decorrente:
  - ◆ Todos AS's vão querer deixar passar tráfego de terceiros?
  - ◆ Resolvido com o routing *policy*
- Uma *import policy* indica que *routing information updates* serão consideradas e ignoradas ao calcular rotas
  - ◆ Rotas sugeridas por um “parceiro (AS) não confiável” podem ser filtradas
- De forma análoga, uma *export* ou *advertising policy* irá indicar quais destinos alcançáveis pelo AS serão divulgados para outros parceiros
  - ◆ Escondendo a alcançabilidade de alguns destinos de outros AS's, pode-se evitar tráfego não desejável

# Uma visão global

- Há necessidade de compartilhamento de informações de roteamento entre o BGP e o protocolo interno de um AS:
  - ◆ Não é um processo simples já que podem haver diferentes IGPs
  - ◆ Informação trocada também entre BNs
- ➔ *Route redistribution.*

