Laboratório de Pesquisa em Redes e Multimídia

O Protocolo IP

Universidade Federal do Espírito Santo
Departamento de Informática

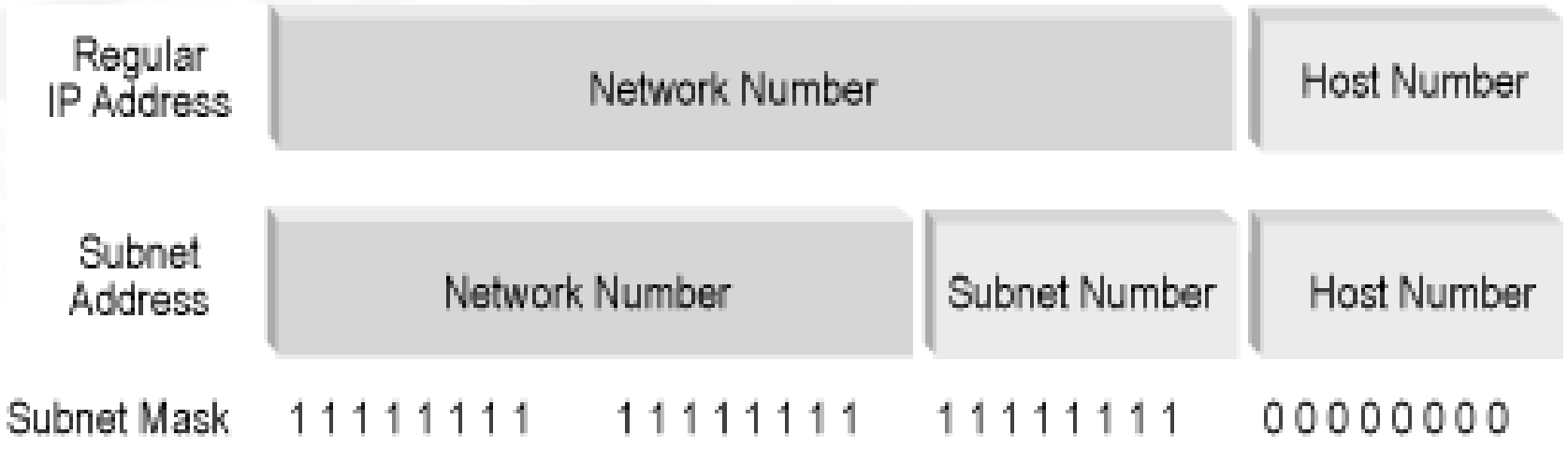
O IP e a Máscara de Sub-Rede

- O IP se baseia em duas estruturas para efetuar o roteamento de datagramas:
  - Máscara de Sub-Rede
  - Tabela de Rotas
- Como já visto, a máscara de sub-rede é um número de 32 bits usado para distinguir um prefixo de rede estendido (*Netid* + *SubnetID*) em um endereço IP.
- A máscara de sub-rede também é usada para determinar se um endereço IP está localizado na rede local ou em uma rede remota.

A Máscara de Sub-Rede (cont.)

- Seja a seguinte situação: computador *A*, com endereço IP 130.15.12.131 e máscara de sub-rede 255.255.255.0, deseja transmitir dados para o computador *B,* cujo IP é 130.15.12.22.
- Passos:
  - O computador A (o software IP) executa um AND lógico entre a máscara e cada um dos endereços IP. O efeito dessa operação é que os 0's da máscara zeram a parte de *host* dos endereços, deixando os campos de rede e sub-rede intactos.
  - Se a parte de rede e sub-rede dos dois endereços são iguais, então ambos estão na mesma sub-rede, caso contrário estão em sub-rede diferentes.

A Máscara de Sub-Rede (cont.)



A Tabela de Rotas

- Provê um meio de se dizer como encaminhar pacotes para máquinas que não estão conectadas à rede local. Cada máquina na rede local possui a sua própria tabela de rotas.
- Na sua forma mais simples, a tabela de rotas é uma estrutura que contém um conjunto de pares (*N*,*G*), onde:
  - *N* é o endereço IP da rede destino
  - *G* é o endereço IP do próximo roteador no caminho da rede N
- Seja a seguinte situação: computador *A*, com endereço IP 130.15.12.131 e máscara 255.255.255, deseja transmitir dados para o computador *C* , cujo IP é 92.45.89.5

A Tabela de Rotas (cont.)

- Um rápido exame da máscara mostra que o *host* destino não está na mesma sub-rede. Neste caso, o IP consulta a sua tabela de rotas local.
- Observe que as informações contidas na tabela de rotas incluem o endereço de rede ao invés do endereço físico de cada *host* destino.
- A quantidade de informação que um roteador necessita guardar na sua tabela de rotas é diretamente proporcional ao número de redes e não ao número de *hosts*.

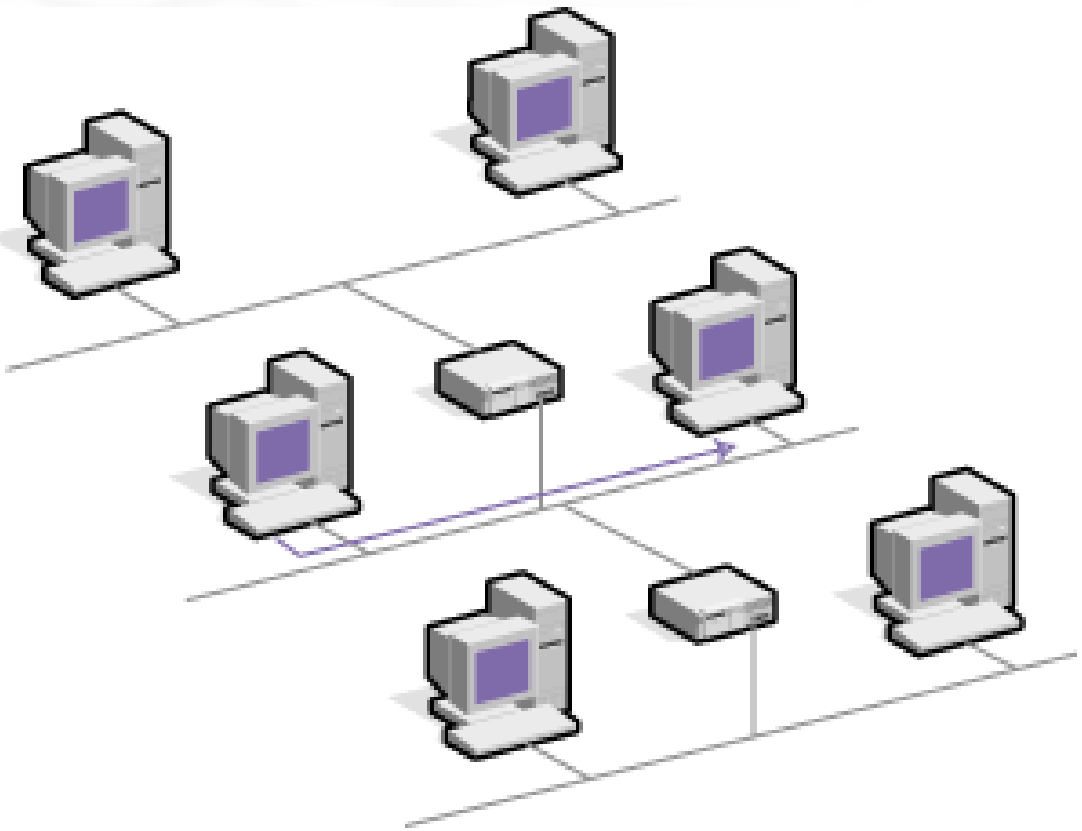
A Tabela de Rotas (cont.)

- A tabela de rotas pode ser mantida tão simples porque o IP não necessita de uma rota completa até o destino. É guardado apenas o endereço do próximo roteador.
- Os roteadores não conhecem o caminho completo até a máquina destino mas, sim, o próximo passo (*hop*) em direção àquela rede.
- As tabelas de rotas sempre apontam para roteadores que estão na mesma rede física.

Roteamento Direto

- Ocorre quando as máquinas participando de uma conversação estão na mesma rede física.
- O transmissor encapsula o pacote IP no quadro do nível de enlace, mapeia o endereço IP destino no endereço físico de destino (via ARP) e envia o quadro diretamente ao destinatário.
- O transmissor sabe que o destinatário está na mesma rede física examinando a porção *NetId* do endereço IP destino, que deve ser igual ao próprio *NetId* (usando para isso a máscara de sub-rede).

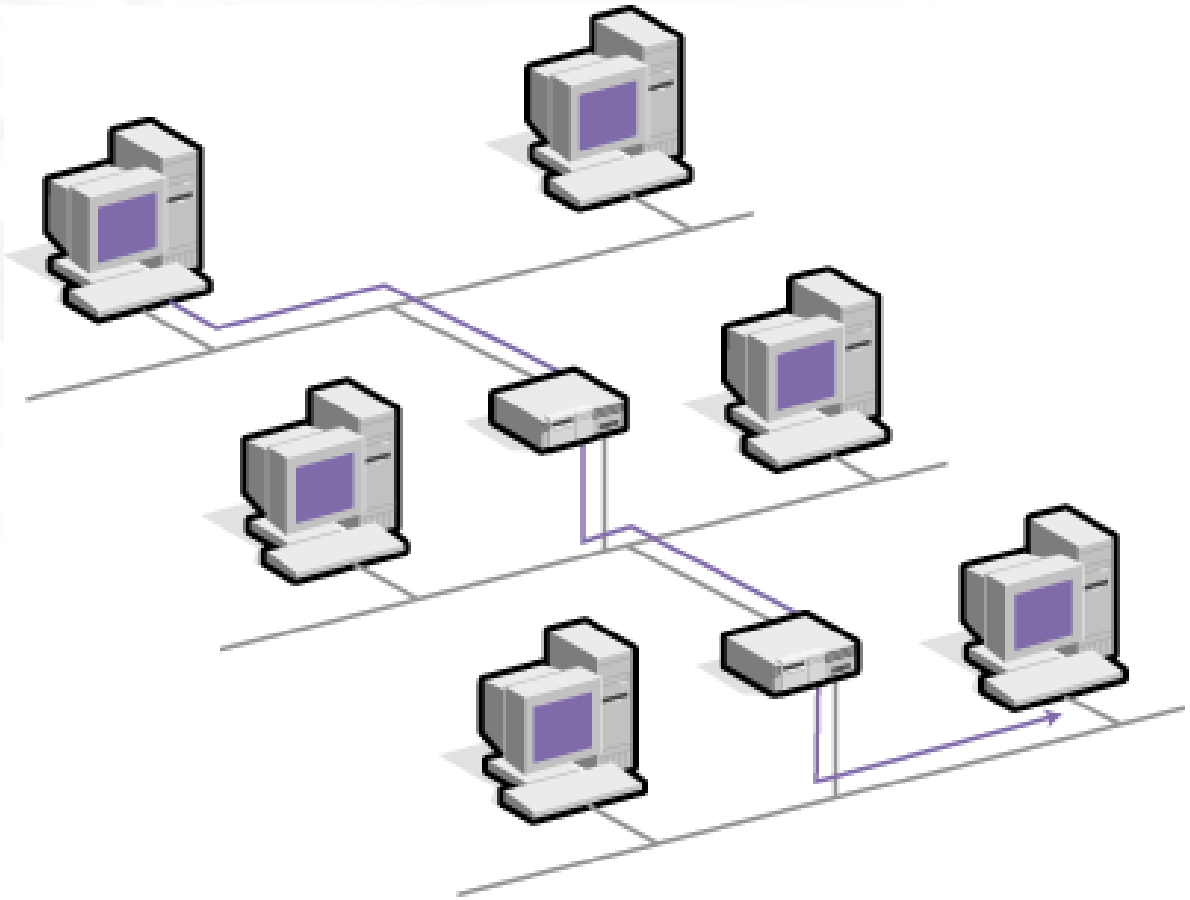
Roteamento Direto (cont.)



Roteamento Indireto

- Ocorre quando duas máquinas participando de uma conversação não estão na mesma rede física.
- Neste caso, *gateways* intermediários terão que manusear o datagrama até que ele chegue ao seu destino.
- O transmissor deve identificar um *gateway* para onde enviar o datagrama. Este *gateway* precisa estar, necessariamente, na mesma rede física do transmissor.

Roteamento Indireto (cont.)



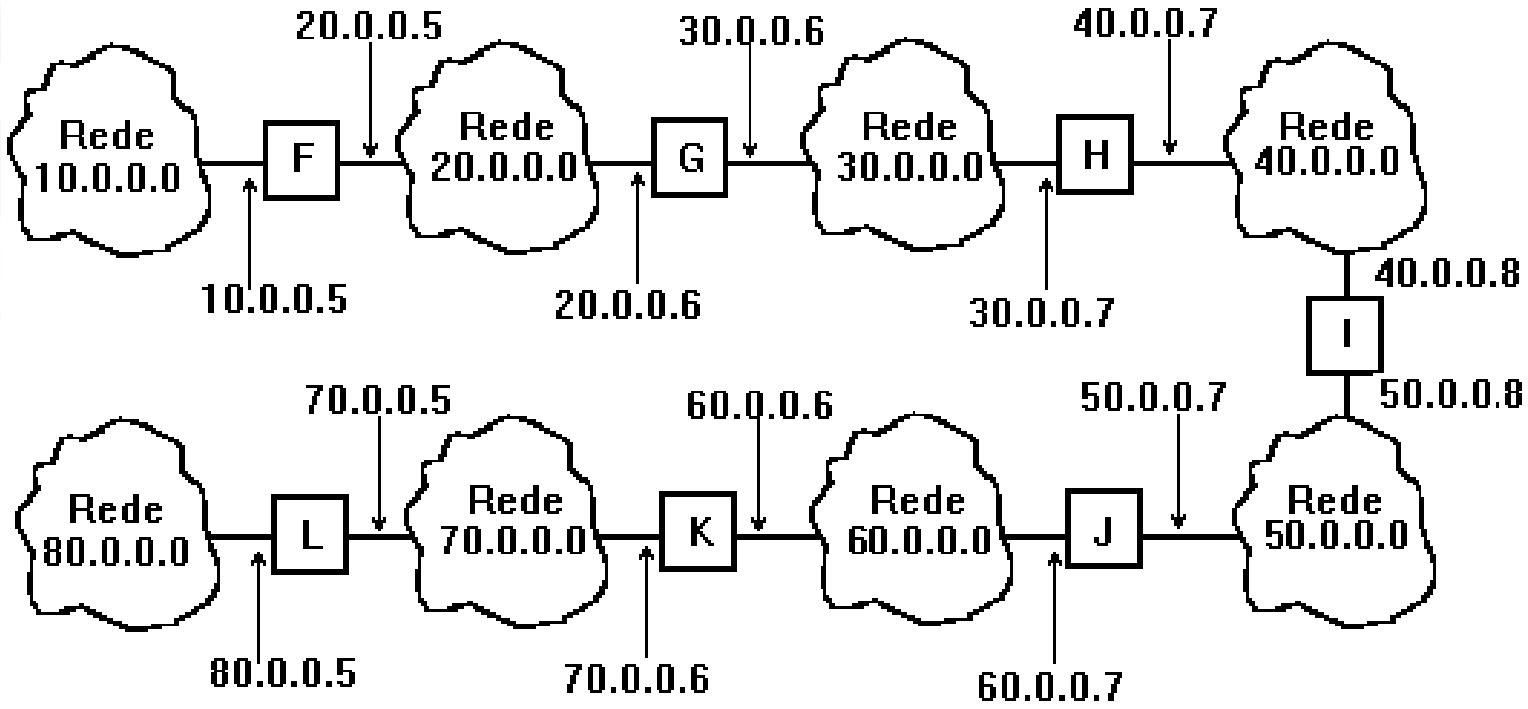
Decisão dos *Hosts* e Roteadores

- Hosts:
  - O roteamento será direto ou não?
  - Caso não seja, para qual roteador deverá ser enviado o pacote?
- Roteadores:
  - A máquina destino está na mesma rede física e pode ser alcançada diretamente? (neste caso ele é o roteador final)
  - Caso contrário, para qual outro roteador entregar o pacote?

Rota Default

- A especificação TCP/IP permite que seja definida uma rota padrão, que será usada sempre que não for encontrada nenhuma rota para a rede destino.
- Normalmente, o endereço classe A *0.0.0.0* é usado como sinônimo de *default* nas tabelas de roteamento.
- Essa entrada diz ao IP para "*redirecionar qualquer datagrama não-local para o roteador default, cujo endereço é x.x.x.x.*"
- No Unix: */etc/defaultrouter*

Rota Default (cont.)



Adicionando uma Rota

- O comando "*route*" é usado para acessar a tabela de roteamento e aceita diversas opções, tais como *add* e *delete*.

```
route [opção] [rede destino] [roteador] [métrica]
```

- O parâmetro *métrica* indica o custo para se atingir um destino através de uma rota.
  - O significado da métrica varia entre os protocolos de roteamento (número de roteadores, volume de tráfego, etc.).
- Podem ser usados nomes em vez de números para identificar redes e máquinas nas rotas.

Adicionando uma Rota (cont.)

```
# route add 164.41.0.0  200.23.98.10  1

# route add unb guarani 1

# route add default 164.41.1.11
```

- No Unix:
  - Os nomes das máquinas e os endereços IP são associados em um arquivo que normalmente se chama */etc/hosts*.
  - Os nomes das redes e os endereços IP são associados em um arquivo que normalmente se chama */etc/networks*.

Examinando a Tabela de Rotas

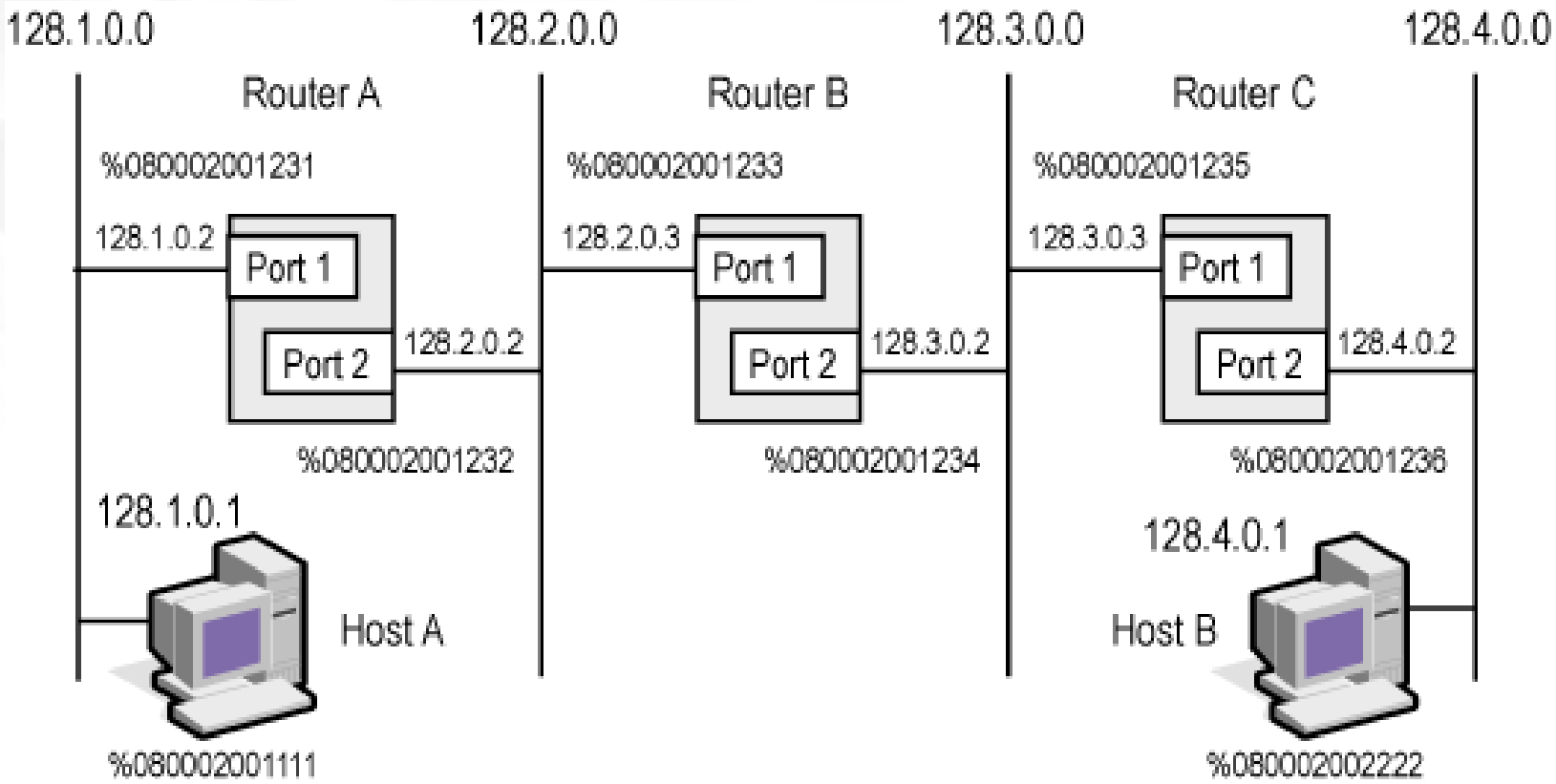
- As informações armazenadas da tabela de rotas podem ser inspecionadas através do comando *netstat*.

 # netstat –nr

Routing tables

Destination	Gateway	Flags	Refcnt	Use	Interf
127.0.0.1	127.0.0.1	UH	6	62806	lo0
default	128.121.50.50	UG	62	2999087	le0
128.121.54.0	128.121.50.2	UG	0	0	le0
128.121.50.0	128.121.50.145	U	33	1406799	le0

Exemplo


128.1.0.0
128.2.0.0
128.3.0.0
128.4.0.0
Router A
Router B
Router C
%080002001231
%080002001233
%080002001235
128.1.0.2
Port 1
128.2.0.3
Port 1
128.3.0.3
Port 1
Port 2
128.2.0.2
Port 2
128.3.0.2
Port 2
128.4.0.2
%080002001232
%080002001234
%080002001236
128.1.0.1
128.4.0.1
Host A
Host B
%080002001111
%080002002222

Exemplo (cont.)

Rot. A

Rede Destino	Roteador	Métrica
128.1.0.0	Direto(porta 1)	0
128.2.0.0	Direto(porta 2)	0
128.3.0.0	128.2.0.3	1
128.4.0.0	128.2.0.3	2

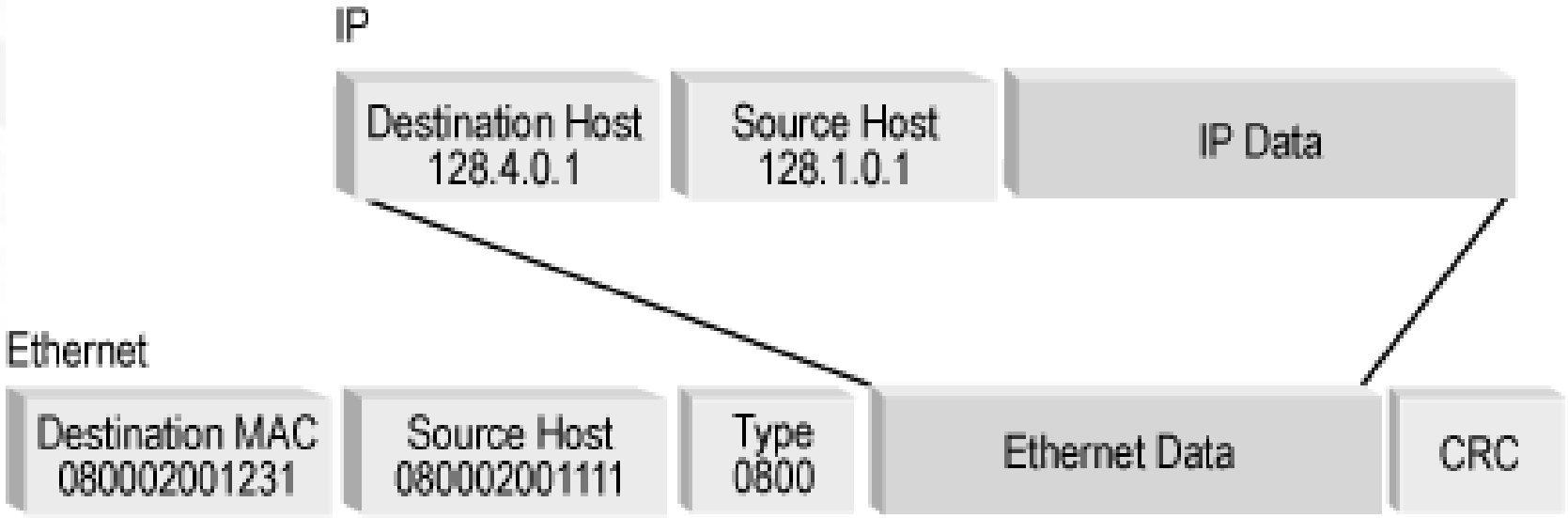
Rot. B

Rede Destino	Roteador	Métrica
128.1.0.0	128.2.0.2	1
128.2.0.0	Direto(porta 1)	0
128.3.0.0	Direto(porta 2)	0
128.4.0.0	128.2.0.3	1

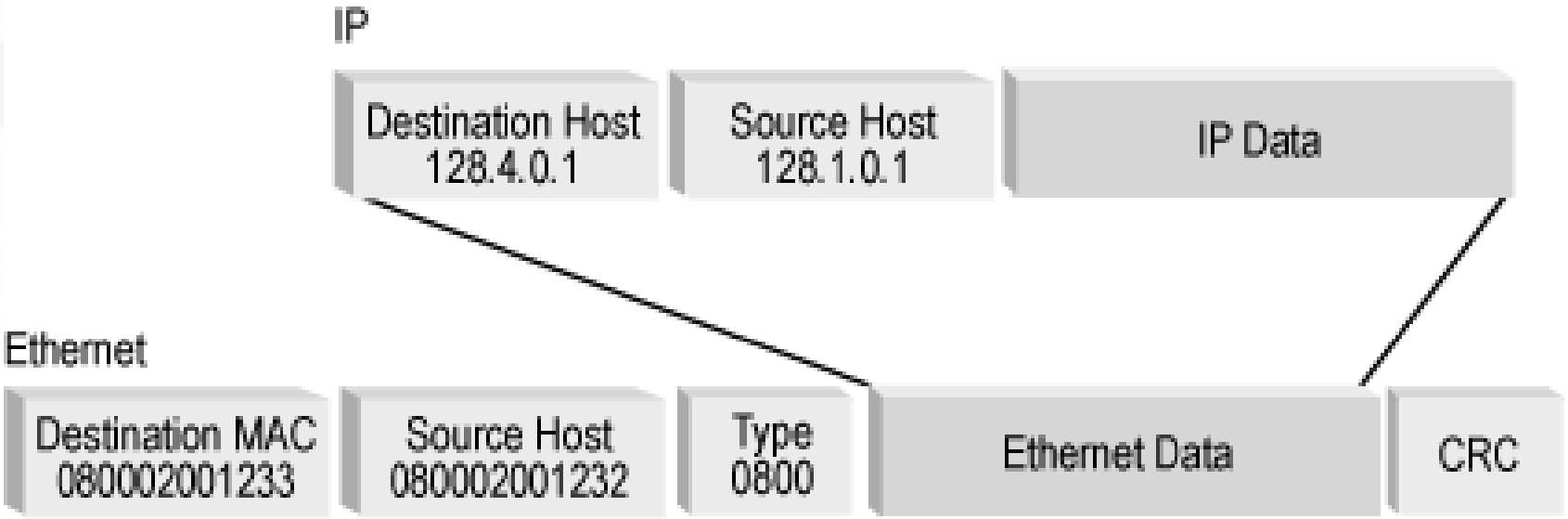
Rot. C

Rede Destino	Roteador	Métrica
128.1.0.0	128.3.0.2	2
128.2.0.0	128.3.0.2	1
128.3.0.0	Direto(porta 1)	0
128.4.0.0	Direto(porta 2)	0

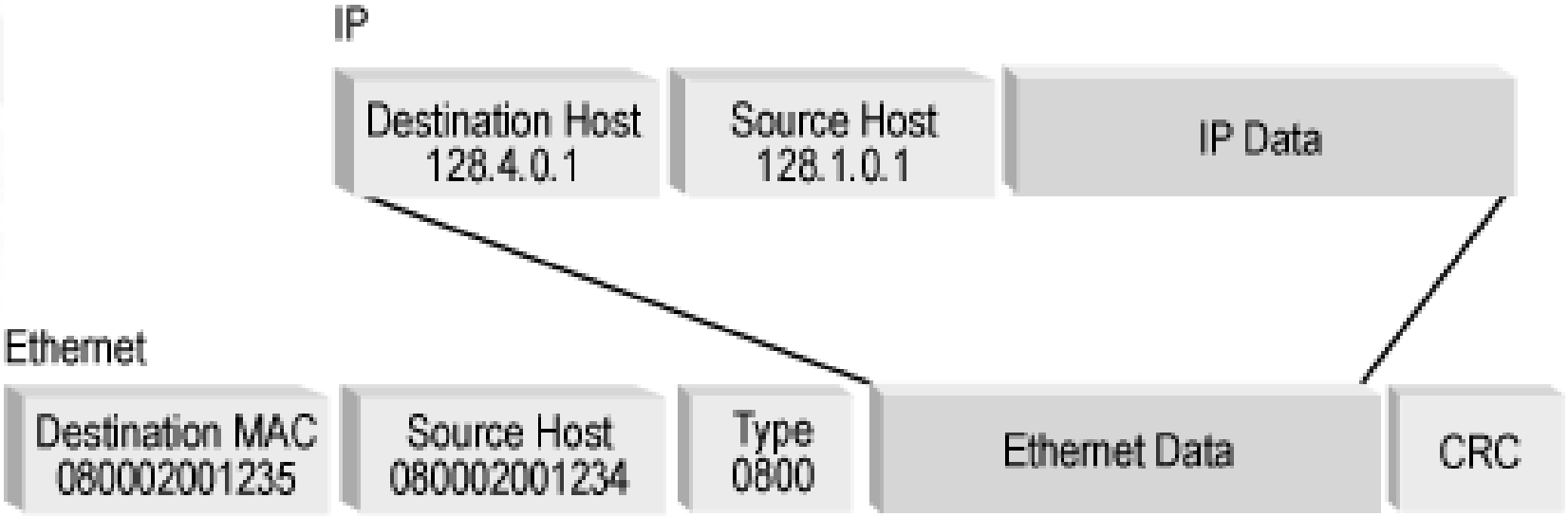
Pacote na Rede 128.1.0.0



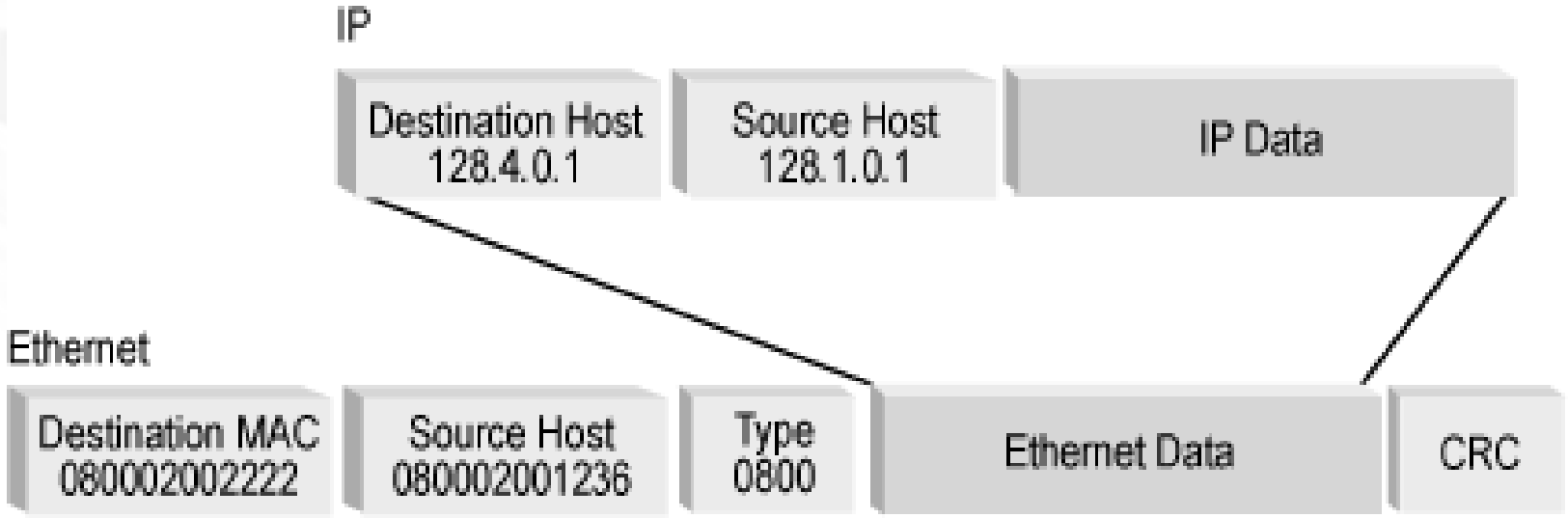
Pacote na Rede 128.2.0.0



Pacote na Rede 128.3.0.0

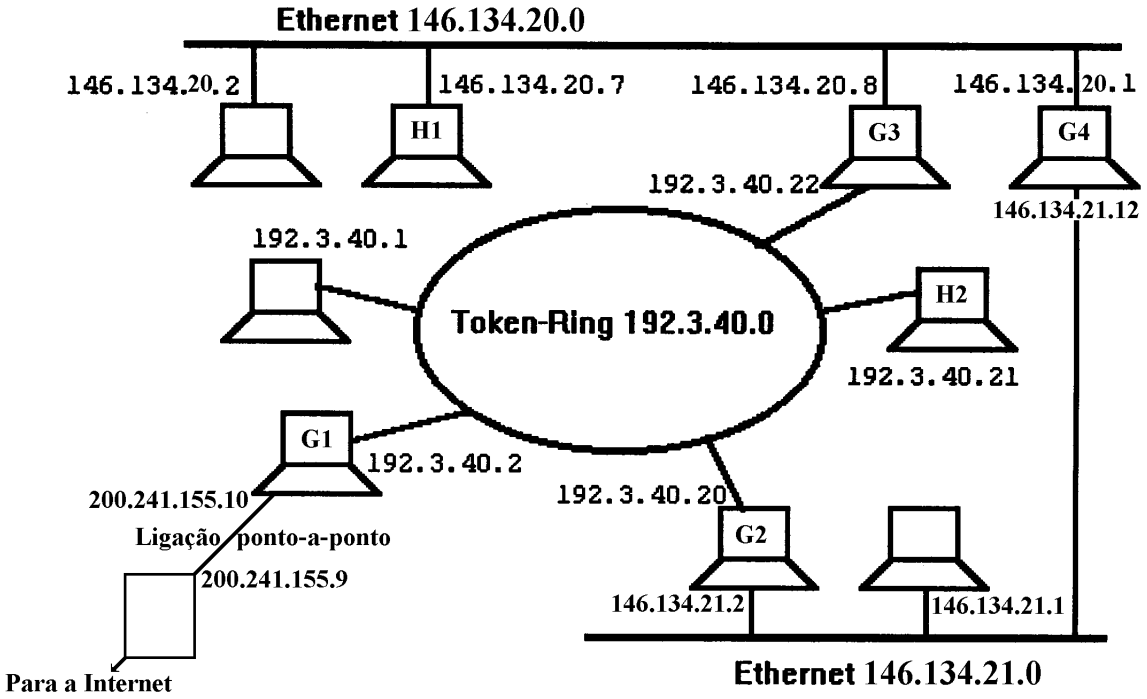


Pacote na Rede 128.4.0.0



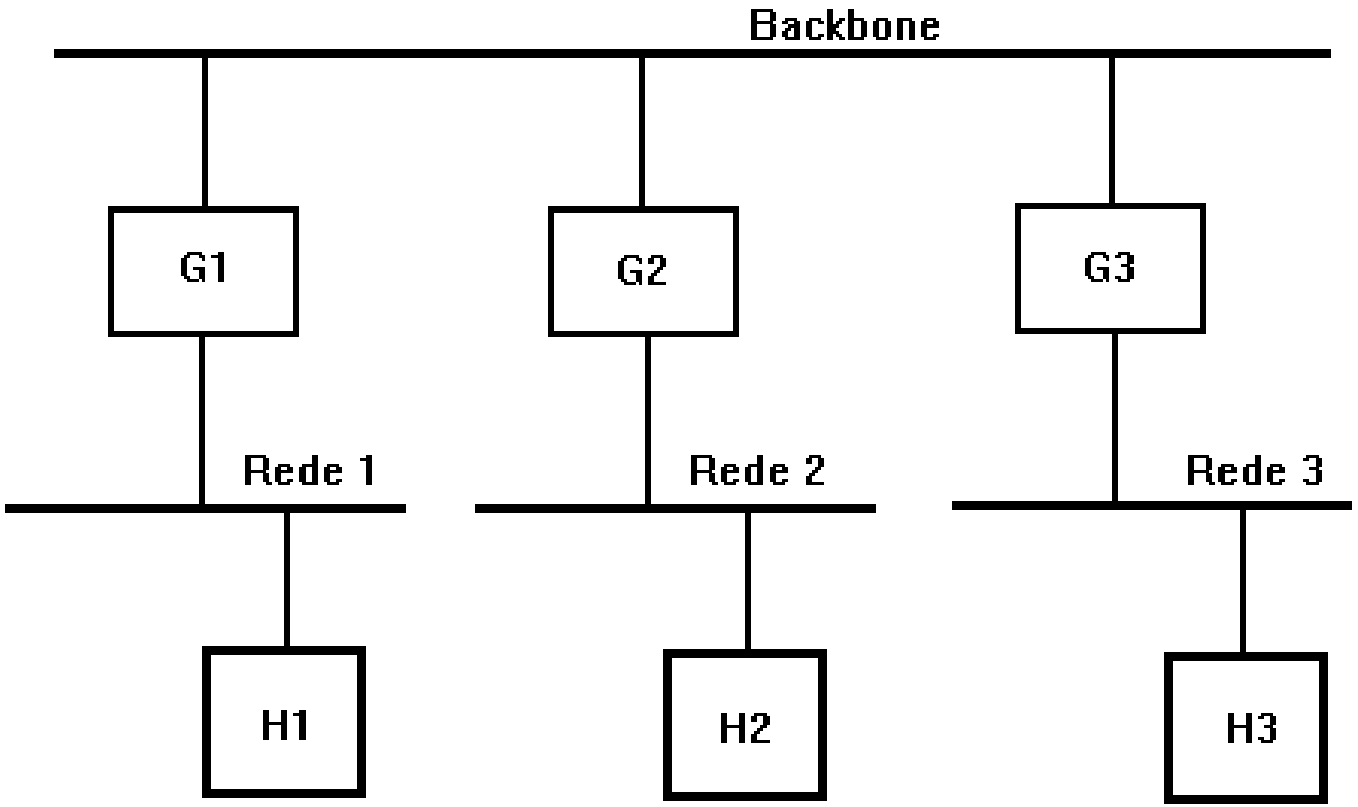
Exercício 1

- Estabelecer a tabela de rotas dos roteadores da figura.

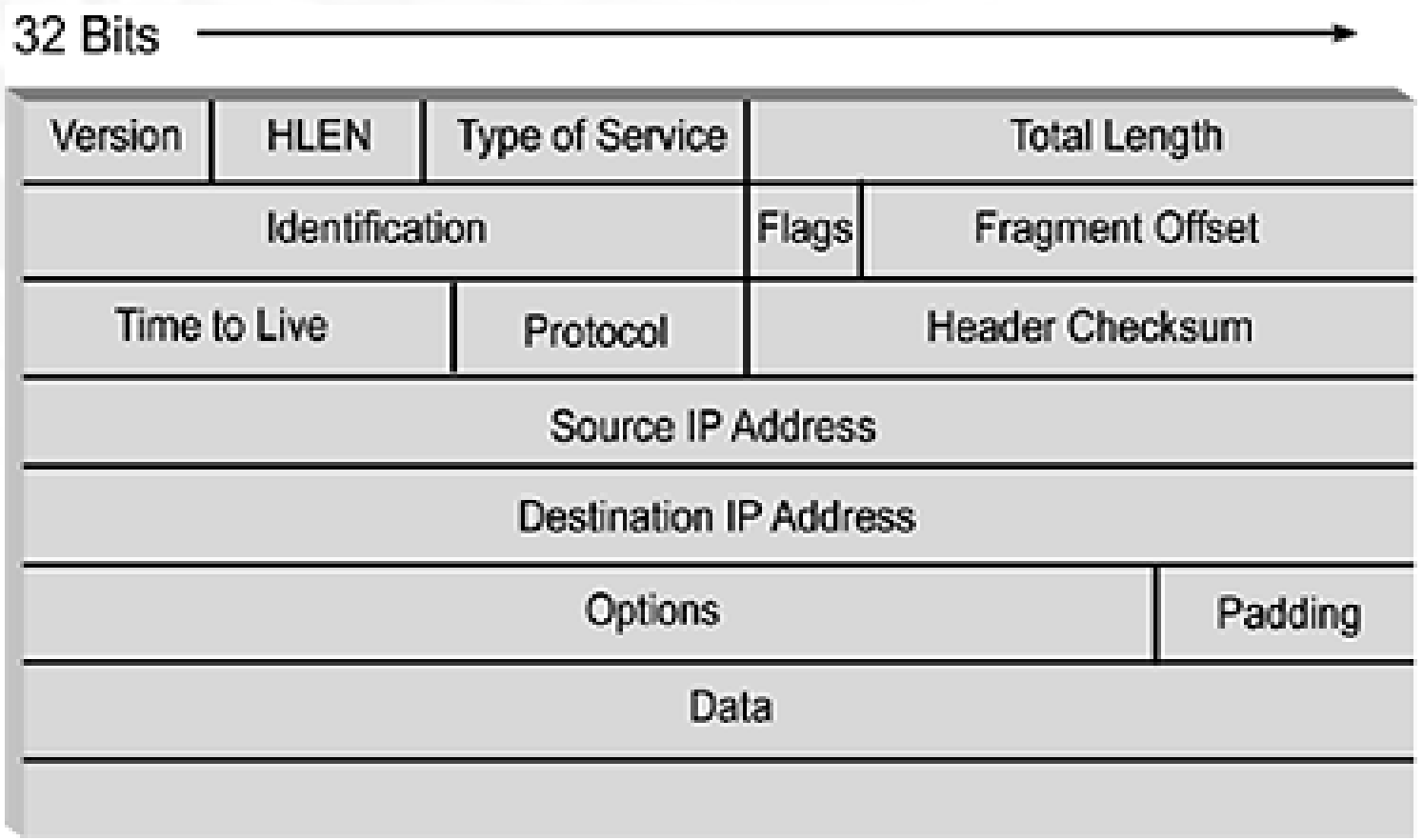


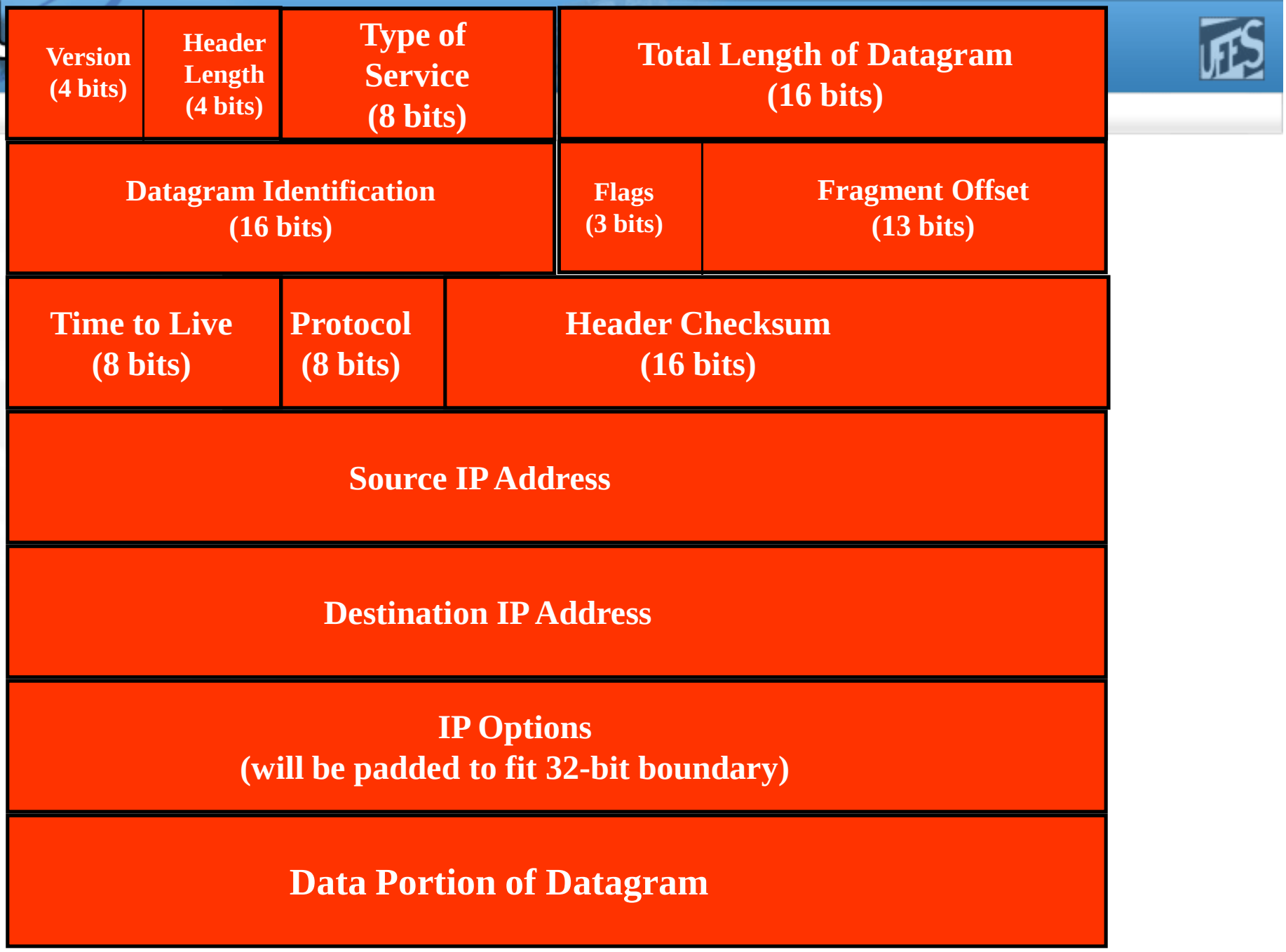
Exercício 2

- Estabelecer a tabela de rotas dos roteadores da figura.



Formato do Pacote IPv4




Version (4 bits)
Header Length (4 bits)
Type of Service (8 bits)
Total Length of Datagram (16 bits)
Datagram Identification (16 bits)
Flags (3 bits)
Fragment Offset (13 bits)
Time to Live (8 bits)
Protocol (8 bits)
Header Checksum (16 bits)
Source IP Address
Destination IP Address
IP Options (will be padded to fit 32-bit boundary)
Data Portion of Datagram

Version

- Indica a versão corrente do protocolo.
- Possui o valor 0100, em binário, correspondente à versão 4 do protocolo (IPv4).
- Campo usado para garantir que o transmissor, o receptor e os *gateways* intermediários estejam de acordo em relação ao formato do protocolo.

Header Length

- Define o tamanho do cabeçalho do datagrama.
- É medido em número de palavras de 32 bits (4 bytes).
- O maior tamanho de um cabeçalho é 15 palavras (60 bytes).
- Na prática, a maioria dos cabeçalhos possui o tamanho mínimo, de 5 palavras (20 bytes).
- Neste caso, o campo *Options* é vazio, bem como não existe nenhum ajuste de tamanho de cabeçalho (*padding*).

Total Lenght

- Define o tamanho do pacote, em bytes.
- A medida inclui o cabeçalho e a porção de dados
  - *Tamanho dos dados = Total Length – Header Length*.
- Se o pacote for fragmentado, este campo indica o tamanho do fragmento e não do pacote original.

Source and Destination IP Address

- Especificam os endereços IP do transmissor e receptor do datagrama .
- Esses campos não são alterados durante a transmissão, mesmo se o datagrama for fragmentado.
- O endereço destino é usado pelo IP para rotear o datagrama .

Protocol

- Número do protocolo que está sendo transportado pelo IP. Permite entregar o pacote ao protocolo apropriado.

Número	Protocolo
1	ICMP
2	IGMP
6	TCP
17	UDP
88	IGRP

Header Checksum

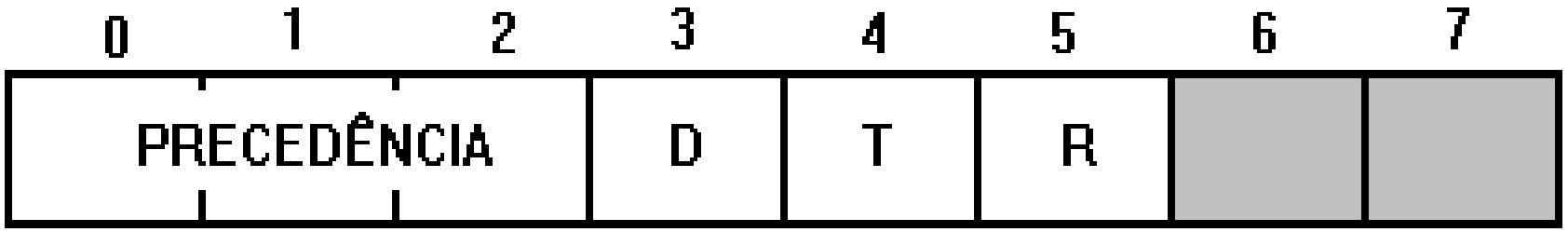
- É calculado apenas sobre o cabeçalho.
- Não cobre nenhum dado que segue o cabeçalho.
- Protocolos como TCP, UDP, ICMP e IGMP possuem um campo de *checksum* (dentro do cabeçalho) que cobre tanto o cabeçalho quanto os dados.

TOS (Type of Service)

- Contém informações de QoS que pode afetar como o datagrama é manipulado. Até recentemente, a maioria dos roteadores ignorava completamente esses bits.
  - Bit D (*Delay*): usado pelo transmissor para requerer baixo retardo (0).
  - Bit T (*Throughput*): idem, para requerer alta vazão (1).
  - Bit R (*Reliability*): idem, para requerer alta confiabilidade (1).
  - Bit C (Cost): idem, para requerer custo normal (0) ou alto (1).
- Exemplos:
  - Telnet e Ftp (sessões de controle): 1000
  - Ftp (sessões de dados): 0110
  - SNMP: 0010

Precedence Bits

- São usados pelo transmissor para informar a importância relativa do datagrama.
- Projetados para prover um mecanismo que permita ao roteador tratar certos datagramas como mais importantes do que outros.
- O padrão IP não especifica que ações devem ser tomadas conforme os valores desses bits.



Time to Live (TTL)

- Tempo de vida do pacote. Especifica o tempo máximo que o pacote pode circular na Internet.
- Campo necessário devido a possibilidade do pacote circular indefinidamente na rede.
- Geralmente implementado como um simples contador de *hops*, que é decrementado a cada roteador.
- Quando TTL = 0, o pacote é descartado.

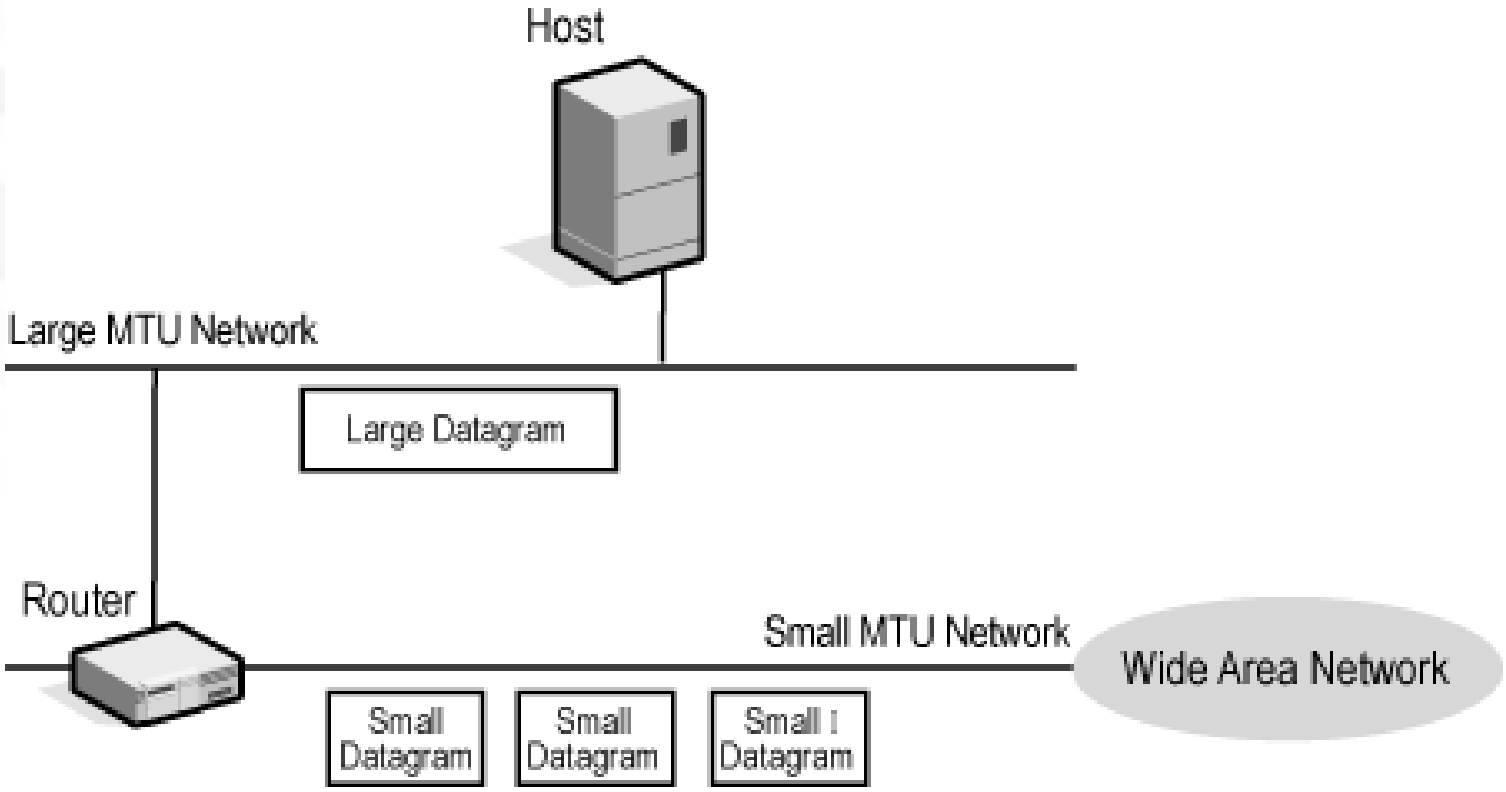
Time to Live (TTL) (cont.)

- Um decremento maior pode ser aplicado a um datagrama que acabou de passar por um enlace muito lento ou que tenha sido enfileirado para transmissão por um longo tempo.

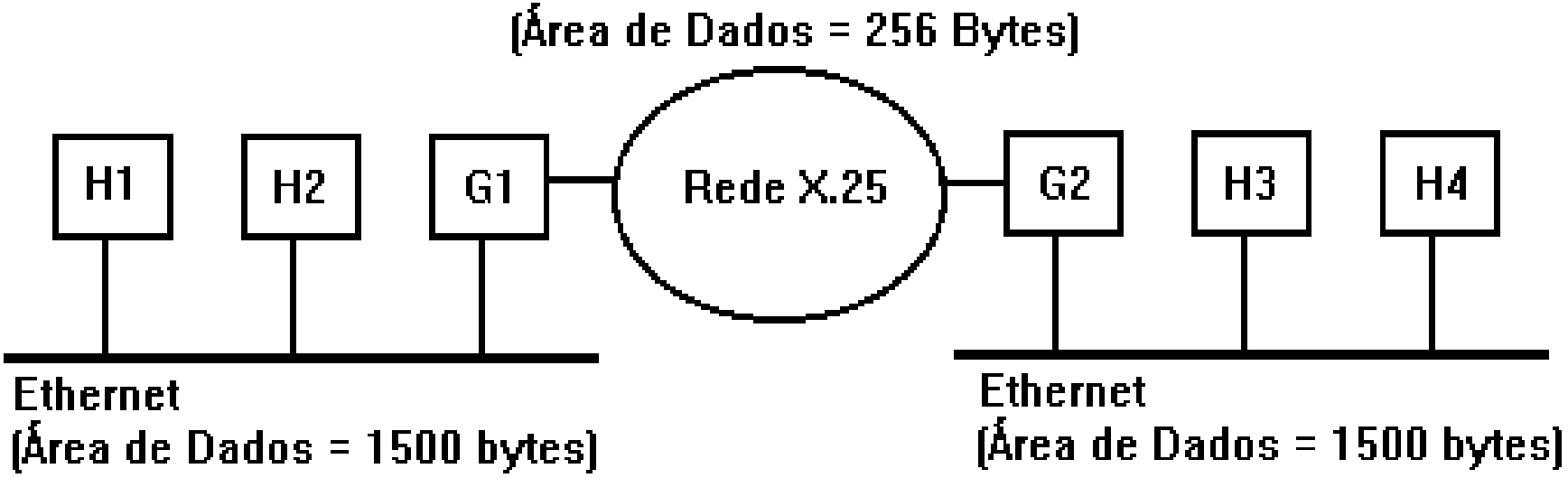
Fragmentação

- Processo de divisão de um datagrama em unidades menores, denominados de *fragmentos*.
- Necessário quando o IP é forçado a transmitir um pacote através de uma rede que opera com pacotes de menor tamanho (menor MTU).
- Campos que controlam a fragmentação e remontagem de datagramas:
  - *Identification*, *Total Length*, *Fragment Offset* e *Flags*.

Fragmentação (cont.)



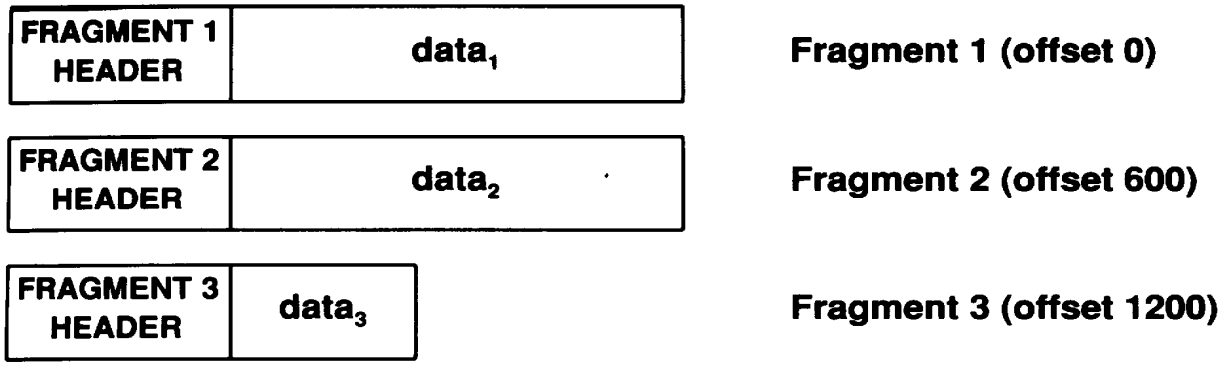
Fragmentação (cont.)



Exemplo

- Pacote IP = 1400 bytes de dados; cabeçalho = 20 bytes; MTU = 620 bytes
- O cabeçalho original é copiado nos fragmentos. Os cabeçalhos 1 e 2 terão o bit " *more fragments"* em 1.

(a)



(b)

Exemplo

Original Packet

2 Fragments

4 Fragments

Fragmentação (cont.)

- Um datagrama pode ser marcado como "*don't fragment*". Caso se torne necessário transmitir esse datagrama através de uma rede com menor MTU, ele será descartado.
  - Ex: frame I do MPEG (codificador de vídeo)
- A perda de um fragmento implica na perda do pacote inteiro.
- Os fragmentos atravessam a Internet separadamente até que chegam ao destino final. É responsabilidade da estação destino remontar os fragmentos da mensagem original.

Flags

- Determinam se um datagrama pode ou não ser fragmentado.
- Se o datagrama está fragmentado, é usado para indicar se existem mais fragmentos ou se este é o último de uma série de fragmentos.

Flags (cont.)

- O campo *flags* contém três bits:

Bit 0	Bit 1 (*Don't Fragment*)	Bit 2 (*More Fragments*)
0 = reservado	0 = pode fragmentar 1 = não fragmente	0 = último fragmento 1 = mais fragmentos

Identification

- Número único que identifica os fragmentos de um pacote. O transmissor atribui este valor ao pacote original.
- Caso o pacote seja fragmentado, este valor é copiado para o campo "*Identification*" de cada fragmento resultante.
- Este campo, junto com o endereço IP origem (*Source IP Address*), identifica o pacote original a que os fragmentos pertencem.

Fragment Offset

- Indica a posição do fragmento em relação ao datagrama original.
- É medido em unidades de 8 bytes ("*fragment blocks*"), a partir do início do datagrama original (o primeiro fragmento possui *offset* 0).
- Como é um campo de 13 bits os valores podem variar de 0 a 8192 blocos de fragmentos, correspondendo à faixa de 0 a 65.528 bytes no total.

O Processo de Fragmentação

- É examinado o campo de *Flags*. Se o bit "*don't fragment*" estiver ligado, não há nada a fazer e o datagrama é descartado.
- Se o bit "*don't fragment*" tiver o valor 0, a porção de dados é quebrada em pedaços consistentes com o tamanho com o MTU do próximo *link*. Cada pedaço deve ter um tamanho múltiplo de 8 bytes ("*8-byte boundary*").
- A cada pedaço é atribuído um cabeçalho IP semelhante ao datagrama original. Em particular, cada pedaço terá os mesmos campos de:
  - *Destination Address*, *Source Address*, *Protocol* e *Identification*.

O Processo de Fragmentação (cont.)

- Os seguintes campos serão atribuídos para cada fragmento, separadamente:
  - *Total Lenght*: tamanho total do pacote levando em conta o pedaço corrente;
  - Bit "*more data*" do campo de *flags*, que deve ter o valor 1 em todos os pedaços, exceto o último.
  - *Fragment offset* deve ser setado para indicar a posição do fragmento em relação ao início do datagrama original. O valor atribuído a esse campo é o valor real da posição dividido por 8.
  - *Checksums* separados devem ser calculados para cada um dos fragmentos.

Exemplo 1

- Datagrama Original:
  - Identificação = 12345
  - Tamanho total = 500 (20 IP header bytes + 480 data bytes)
  - Offset do fragmento = 0
  - Bit "*more data*" = 0

Exemplo 2

- Determinar o conteúdo dos campos *Identification*, *Flags* e *Fragment Offset* no seguinte cenário:
  - Datagrama Original:
    - Identificação = 348
    - Tamanho total = 3000 bytes
  - Fragmentos:
    - Três datagramas. Cada fragmento tem seu próprio cabeçalho e 1000 bytes (125 "*fragment blocks*") de dados.

Remontagem de Datagramas

- A remontagem (*reassembly*) estará completa quando existir um conjunto contíguo de dados no *buffer* da estação destino, iniciando com um campo de *fragment offset* igual a zero e terminando com dados de um fragmento com o bit "*more data*" também igual a zero.
- Existe uma omissão inconveniente no IP: a estação destino não tem como saber qual é o tamanho total do datagrama até que o último fragmento chegue.

Remontagem de Datagramas (cont.)

- O campo *Total Length* contém o tamanho total do fragmento e não o do datagrama original.
- Isso significa dizer que a estação destino não tem como prever com exatidão o tamanho do *buffer* necessário para acomodar os datagramas.
- A estação receptora necessita de um *reassembly timeout*.

O Campo *Options*

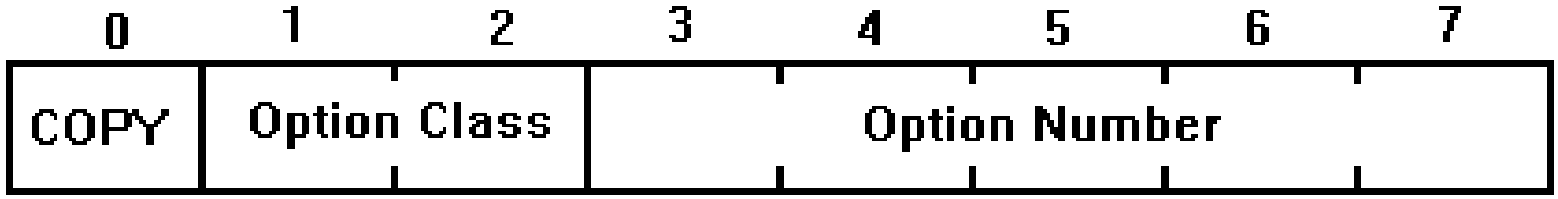
- O principal propósito do campo *Options* é prover para o administrador de rede ferramentas para testar e "depurar" a rede.
- As opções incluídas em um datagrama são escolhidas pela aplicação transmissora.
- Possui tamanho variável, sendo que até 40 bytes podem ser usados no cabeçalho para armazenar as *Options* do protocolo.
- O suporte às opções é obrigatório e deve estar presente em qualquer implementação do IP residente em *hosts* e roteadores.

Formato do Campo *Options*

Código	*Tamanho*	*Ponteiro*
Parâmetro da Opção		
Parâmetro da Opção		

- O campo consiste de um byte com o código da opção, que pode ser seguido por um byte de tamanho e por um conjunto de bytes referentes aos dados da opção escolhida.

O Campo *Code*



- É subdividido nos sub-campos *Copy*, *Option Class* e *Option Number*.
- O sub-campo *Copy* controla como os roteadores tratam as opções na presença de fragmentação. Se *Copy* = 1, então as opções devem ser copiadas em todos os fragmentos.

O Campo *Option Class*

- Os dois bits do sub-campo *Option Class* possuem o seguinte significado:

Option Class	Significado
0	Datagrama de controle
1	Reservado para uso futuro
2	Datagrama de *debug* e *measurement*
3	Reservado para uso futuro

O Sub-Campo *Option Number*

- Define os números das opções. São as seguintes as opções definidas no protocolo IP:
  - Record Route
  - Strict/Loose Source Route
  - Timestamp
  - Department of Defence Basic Security
  - Department of Defence Extended Security
  - No operation
  - End of listing (padding)

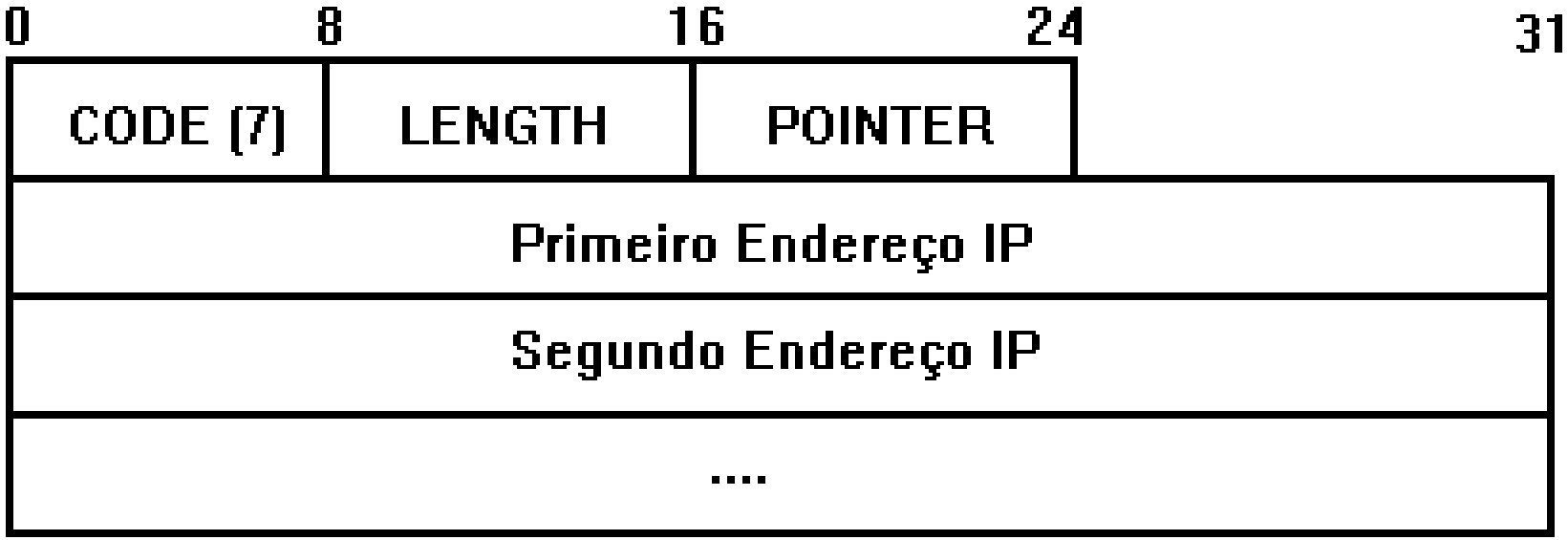
Valores do Campo *Code*

Code	Copy	Class	Number	Option
7	0	0	7	*Record Route*
137	1	0	9	*Strict Source Route*
131	1	0	3	*Loose Source Route*
68	1	0	2	*Timestamp*
130	1	0	2	*Security*
133	1	0	5	*Extended Security*
1	0	0	1	*No Operation*
0	0	0	0	*End of Option List*

Record Route

- Usada pela estação origem para reservar espaço no cabeçalho do datagrama para uma lista vazia de endereços IP.
- Cada roteador adiciona o seu endereço IP à lista. Assim, ao chegar ao destino, o cabeçalho contém a lista dos roteadores visitados pelo datagrama.
- Um máximo de 9 (nove) endereços IP podem ser armazenados no datagrama.
- Opção é útil para monitorar o caminho que um datagrama segue à medida que ele é roteado na Internet (ex: *ping –R*).

Record Route (cont.)



Record Route (cont.)

- *Code* define o código da opção. *Length* define o número total de bytes e *Pointer* indica em que posição armazenar o próximo endereço IP.
- O valor inicial é do campo *Pointer* é 4. A cada endereço IP adicionado à lista o seu valor muda para 8, 12, 16, etc., até 36.
- Um máximo de 9 endereços IP podem ser armazenados no datagrama.
- Após o nono endereço o valor de *Pointer* é 40, indicando fim da lista.

Record Route (cont.)

- Problema: quando um roteador adiciona o seu endereço IP à lista, que endereço de interface de rede ele armazena?
- A RFC 791 especifica que o roteador deve armazenar sempre o endereço da interface de saída.

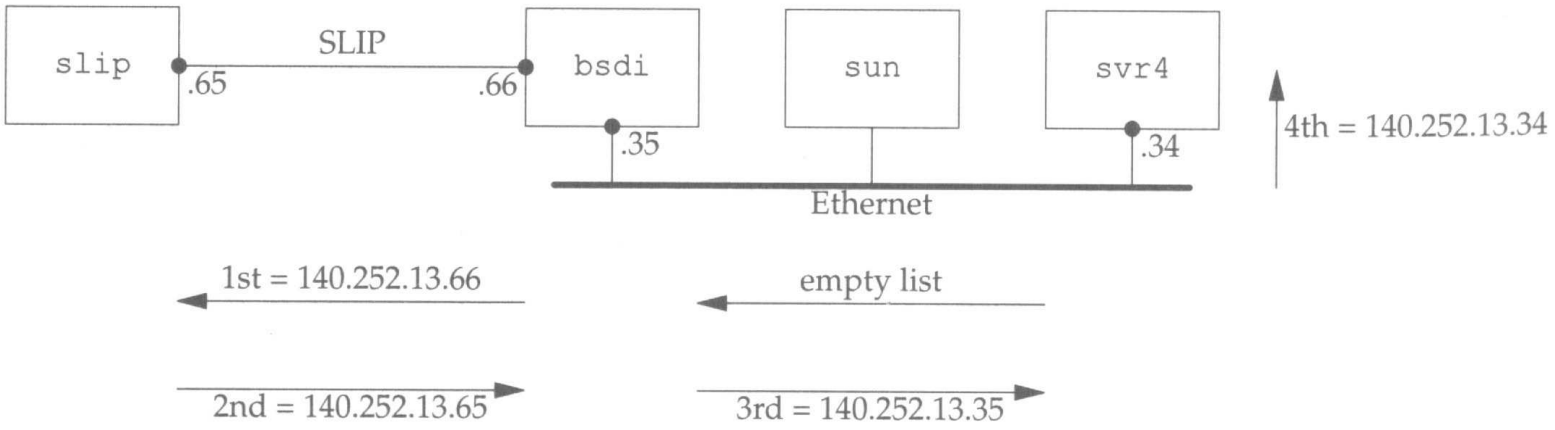
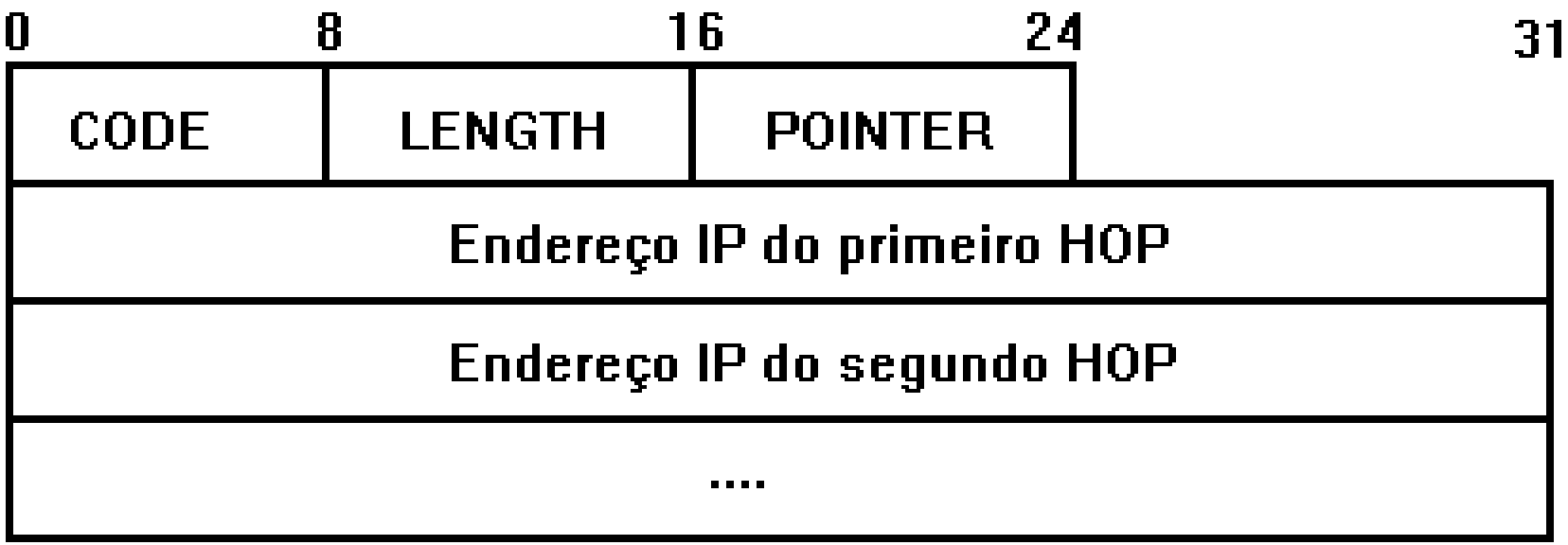


Figure 7.4 `ping` with record route option

Source Route

- Permite especificar a rota a ser seguida pelo datagrama até o computador destino.
- O formato da opção é semelhante à opção *Record Route* mas a lista de endereços IP deve ser definida antes de se enviar o datagrama.
- A rota especificada pode ser *strict* (code 137) ou *loose* (code 131).

Source Route (cont.)



Strict Source Route

- Nessa opção, apenas os roteadores listados podem ser visitados. O *host* origem especifica o caminho exato (a rota completa) que o datagrama deve seguir até o destino.
- Se um roteador encontra um *next hop* na rota que não está numa rede diretamente conectada uma mensagem ICMP "*source route failed*" é gerada.
- Usada com o intuito de aumentar a segurança dos dados, pré-definindo um caminho entre origem e destino.

Strict Source Route (cont.)

- Infelizmente, a opção também faz parte do arsenal dos *hackers* e pode ser usada como porta de entrada para seus ataques.
- Roteadores que filtram o tráfego que entra na organização devem ser configurados para ou descartar todos os pacotes "*source-routed*" ou examinar antes o campo *source route* verificando o real endereço destino do datagrama.

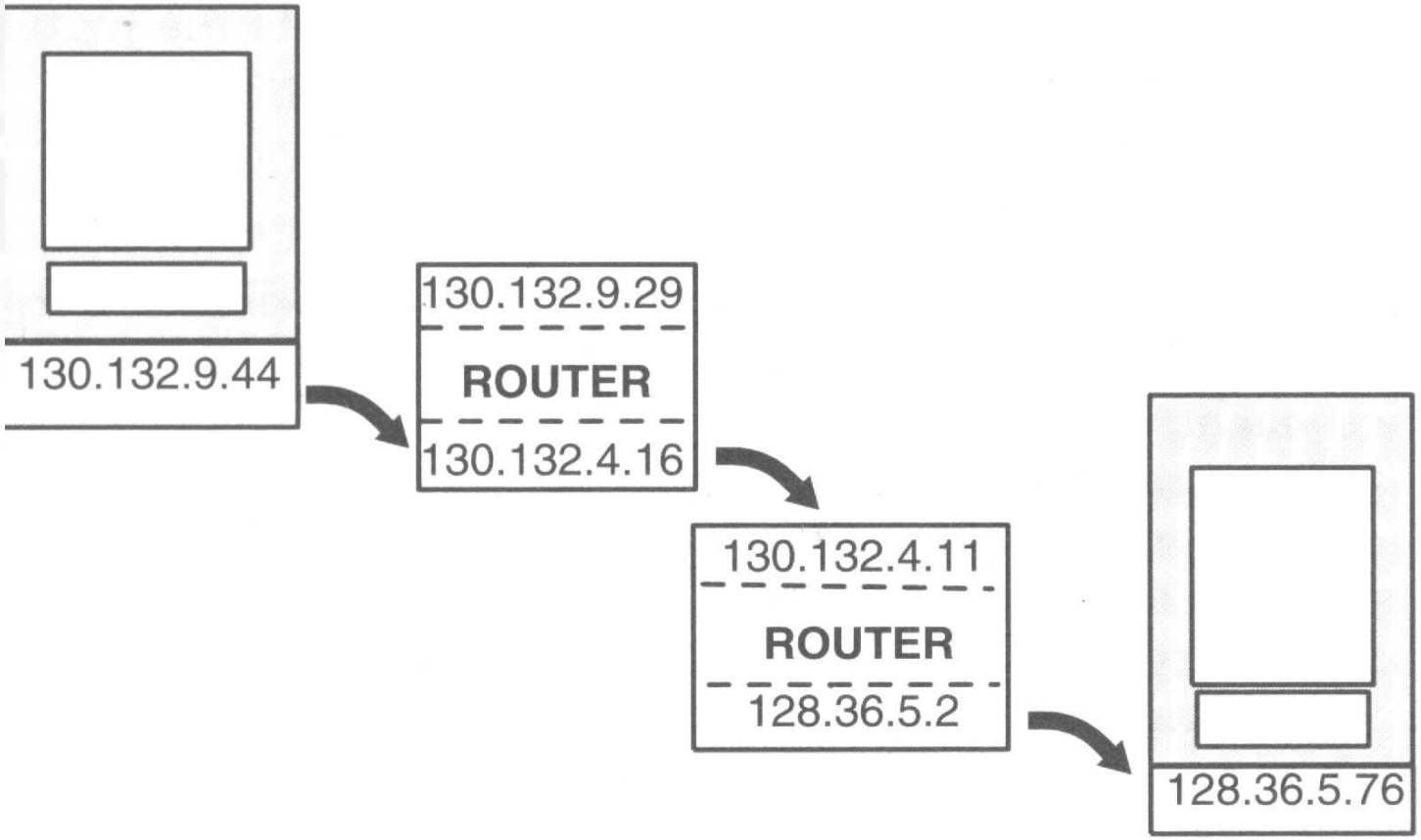
Loose Source Route

- Nessa opção, o pacote deve seguir a seqüência de endereços especificada na lista; entretanto, roteadores intermediários podem ser visitados.
- A opção especifica, na verdade, *milestones* ao longo do caminho. Qualquer rota pode ser seguida entre os *milestones*.
- Usado ocasionalmente com propósitos de teste da rede (ex: roteamento para locais muito distantes).

Rota Reversa

- Quando a opção *Source Route* é usada, o tráfego de volta deve seguir o mesmo caminho, isto é, o mesmo conjunto de roteadores deve ser visitado, mas na ordem inversa.
- Isso introduz um pequeno problema já que os endereços das interfaces do caminho de ida não são os mesmos do caminho de volta (os endereços de cada interface dos roteadores são diferentes porque as interfaces conectam diferentes sub-redes).

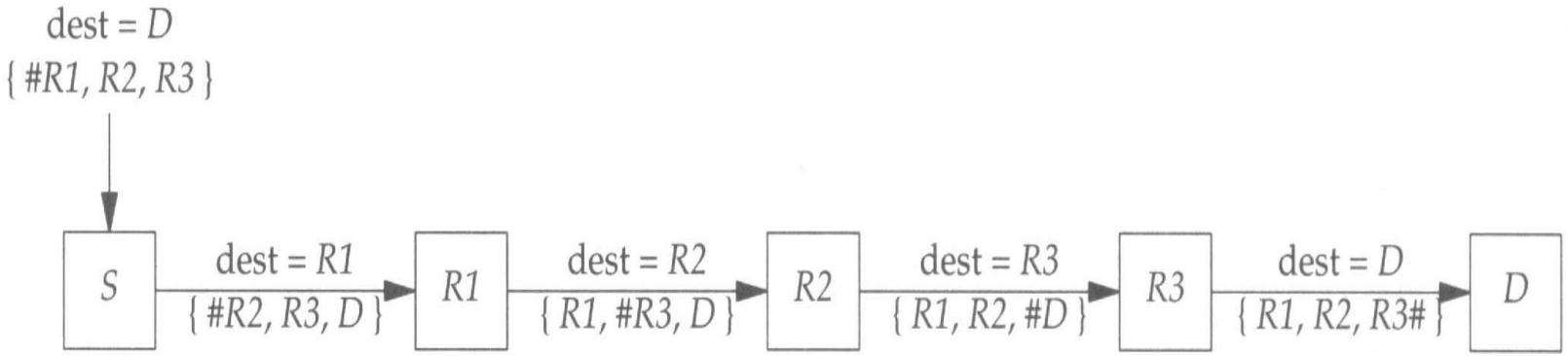
Rota Reversa (cont.)



Rota Reversa (cont.)

- Diferentes visões:
  - Máquina A: 130.132.9.29 e 130.132.4.11
  - Máquina B: 128.36.5.2 e 130.132.4.16
- Para resolver esse problema, a cada roteador visitado o endereço de entrada é substituído no campo "*source route*" pelo seu endereço de saída.

Rota Reversa - Exemplo



Rota Reversa - Exemplo

Passo	End Origem	End Destino	Source Route
1	**130.132.9.44**	**130.132.9.29**	130.132.4.11 **128.36.5.76**
2	130.132.9.44	130.132.4.11	130.132.4.16 128.36.5.76
3	130.132.9.44	128.36.5.76	130.132.4.16 128.36.5.2

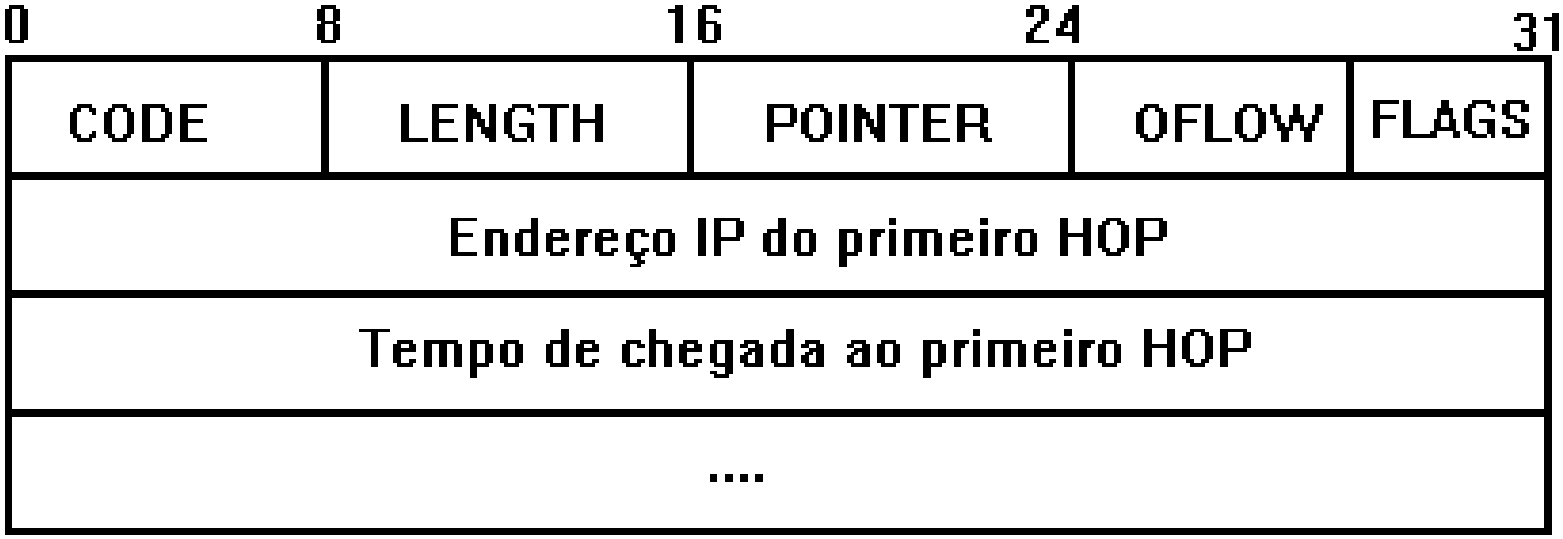
Time Stamp

- Opção é similar à opção *Record Route* exceto que a estação origem reserva espaço no cabeçalho do datagrama para uma lista vazia de *time stamps*.
- Cada roteador visitado deve estampar a hora e data em que ele processa o datagrama.
- Usada pelos administradores para monitorar o desempenho da rede.

Time Stamp

- Existem três formatos:
  - Uma lista de *time stamps* de 32-bits;
  - Uma lista de endereços IP e os correspondentes *time stamps*;
  - Uma lista de endereços IP pré-selecionados providos pela origem, cada qual seguido por um espaço para armazenar o seu *time stamp*.
- Na última opção, um nó armazena o *time stamp* apenas se o seu endereço é o próximo da lista.
- O espaço para armazenamento pode acabar logo quando o primeiro e segundo formatos são usados.

Time Stamp (cont.)



Time Stamp (cont.)

- *Code* é igual a 0x44 para a opção *time stamp*.
- *Overflow* indica o número de nós que não puderam guardar os seus *time stamps*.
- *Length* armazena o tamanho total da opção (normalmente 36 ou 40).
- *Pointer* (4 bits) é um ponteiro para a próxima entrada disponível (5, 9, 13, etc.).

Time Stamp

Flags	Descrição
0	Armazena apenas *time stamps*.
1	Armazena o endereço IP e *time stamp*.
2	O transmissor inicializa a lista de opções com até 4 pares de endereços IP e *time stamps*. O roteador armazena seu *time stamp* apenas se o próximo endereço da lista é igual ao seu próprio.

Time Stamp (cont.)

- Se um roteador não consegue adicionar o seu *timestamp* porque não há mais espaço ele simplesmente incrementa o campo *Overflow*.
- O valor preferido para os *timestamps* é o número de milisegundos após meia-noite, similar à mensagem ICMP *timestamp request* e *reply*.
- Opção não muito útil para aos administradores como medida acurada de tempo entre os roteadores devido às suas limitações de espaço e falta de controle sobre a "acuracidade" dos tempos estampados.

Processamento no Roteador

1. Verifica se o datagrama deve ser descartado:
   - Recomputa o *Header Checksum* e compara com o campo de *Checksum* do datagrama.
   - Examina os campos *Version*, *Header Length*, *Total Length* e *Protocol* em busca de alguma inconsistência.
   - Decrementa TTL e descarta se TTL = 0.
2. Descarta o pacote se o roteador não possui espaço (*buffer*) suficiente para processá-lo.
3. Aplica rotinas de segurança pré-configuradas.

Processamento no Roteador (cont.)

4. Executa os procedimentos de roteamento:
   - Verifica a presença de *strict* ou *loose route*.
   - Leva os bits TOS em consideração.
   - Verifica o bit *Don't Fragment*.
   - Fragmenta o datagrama se permitido e necessário.
   - Processa as *options*.
   - Roteia o datagrama para o *next-hop system*.

Processamento no *Host* Destino (cont.)

1. Recomputa o *Header Checksum* e compara com o campo de *Checksum* do datagrama.
2. Verifica se o endereço destino é válido.
3. Examina os campos *Version*, *Header Length*, *Total Length* e *Protocol.*
4. Verifica se o *host* não possui espaço (*buffer*) suficiente para processar o pacote.
5. Examina os campos que controlam a fragmentação. Usa o campo *Fragment Offset* para posicionar corretamente o datagrama.
6. Entrega o datagrama completo à camada superior.

Exemplos

- Apostila NetPrep – p. 3-16 a 3-19

COMANDOS!!!!!