IP – Internet Protocol (1)

Prof. José Gonçalves Pereira Filho
Departamento de Informática
zegonc@inf.ufes.br

Os Primeiros Ambientes Inter-Redes

- Cenário:
  - Os computadores só podiam se comunicar com outros computadores eles estivessem na mesma rede.
- Problema mais evidente nos anos 70, quando grandes organizações começaram a adquirir redes de diferentes fornecedores (SNA/IBM, DECnet/Digital, etc.).
- Cada rede individual formava uma “ilha isolada de dados” dentro das organizações.

Os Primeiros Ambientes Inter-Redes (cont.)

- Cada tarefa era executada somente na rede apropriada que a acomodava.
- Funcionários tinham acesso a múltiplos computadores, de múltiplas redes, gerando problemas de segurança e de administração.
- Funcionários eram obrigados a se mover de um computador a outro para o envio de dados em cada uma das redes.
- Este cenário contribuia para reduzir a produtividade e o grau de satisfação dos usuários com as redes.

Sistemas Abertos (“Open Systems)

- O cenário apresentado foi a motivação para a criação de um **serviço universal de comunicação de dados.**
- Tal serviço permitiria a qualquer usuário, independentemente do computador e da rede que estiver usando dentro da organização, enviar dados para qualquer outro computador, qualquer que seja a localização deste (i.e., em qualquer rede de qualquer organização do mundo).
- A existência de um serviço de comunicação universal forma a base do conceito de “Sistemas Abertos” (“**Open Systems**”).

Internet: O Ambiente Inter-Redes Global

- A interconexão de inúmeras redes físicas em escala global resultou no que hoje conhecemos como **Internet**.

- A Internet emprega elementos de hardware e de software e se baseia em dois elementos principais:
  - hardware: **roteador**
  - software: **protocolo IP**

- A Internet é, na verdade, uma imensa rede virtual, que provê a ilusão de uma rede única para os seus usuários e aplicações.

Internet: O Ambiente Inter-Redes Global
(cont.)

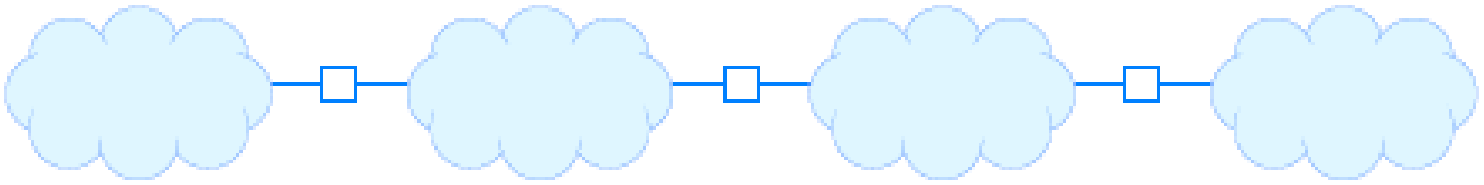
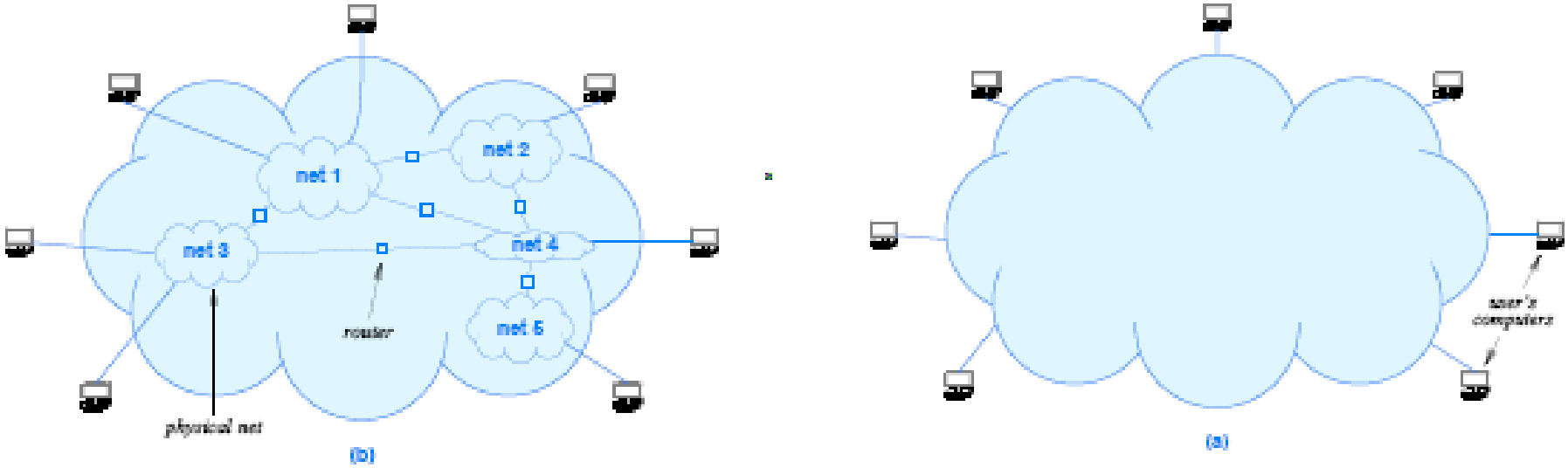


Figure 20.2 An internet formed by using three routers to interconnect four physical networks.

- Numa internet, as várias redes são interligadas por meio de roteadores, e usam o protocolo IP como software básico de interconexão lógica.

Internet: O Ambiente Inter-Redes Global
(cont.)

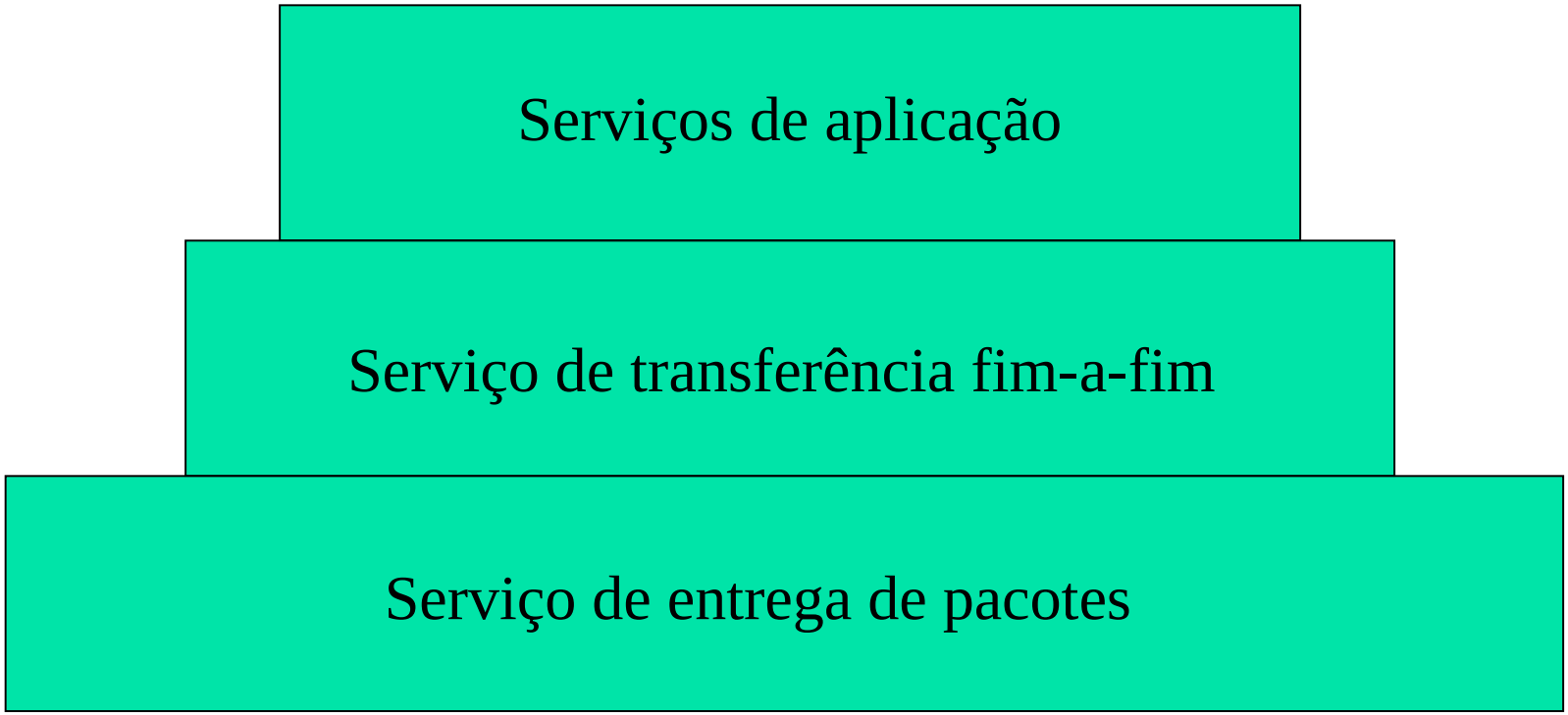


- O termo **Internet**, com a letra“i” maiúscula, refere-se à atual internet global e os seus protocolos associados.

Desafios de Inter-conexão na Internet

- Os elementos roteadores devem concordar sobre o repasse da informação entre as redes.
- Existem diferentes formatos de *frames* e esquemas de endereçamento nas várias rede interligadas ("*undelying networks*"). Isto torna a tarefa mais complexa.
- A camada de rede (ou, de "inter-redes") é a camada que torna possível a implementação de um serviço universal de comunicação.
- Duas principais pilhas de protocolos foram propostas para uso em ambiente internet: TCP/IP e OSI. A pilha TCP/IP o padrão atual de fato.

Tipos de Serviços na Internet

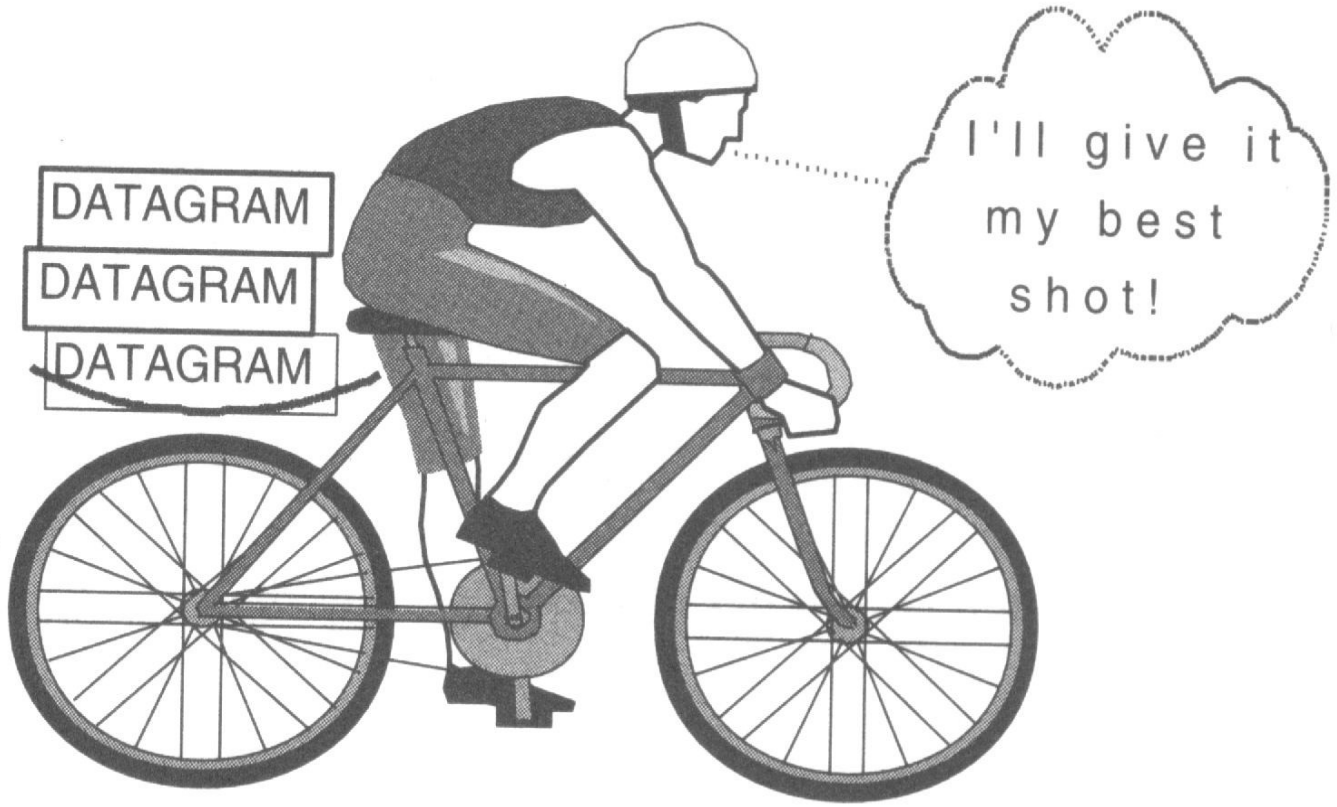


Serviço de Entrega de Pacotes

- Provê uma base sobre a qual todo o resto da pilha de protocolos de rede se apóia. Na arquitetura TCP/IP, esse serviço é fornecido pela Camada de Rede.

- É um serviço **não confiável** ("*unreliable*"):
  - a entrega de pacotes não é garantida. O pacote (datagrama) pode ser corrompido, perdido, duplicado, chegar atrasado ou entregue fora de ordem.
-
- É um serviço **não-orientado à conexão** ("*connectionless*"):
  - cada pacote é tratado independentemente dos outros. Sua transferência não tem qualquer relação com os que o antecederam ou com os que virão.
-
- É um serviço do tipo **maior esforço** ("best-effort"):
  - a entrega só não acontece quando os recursos estão exauridos ou a rede falha.

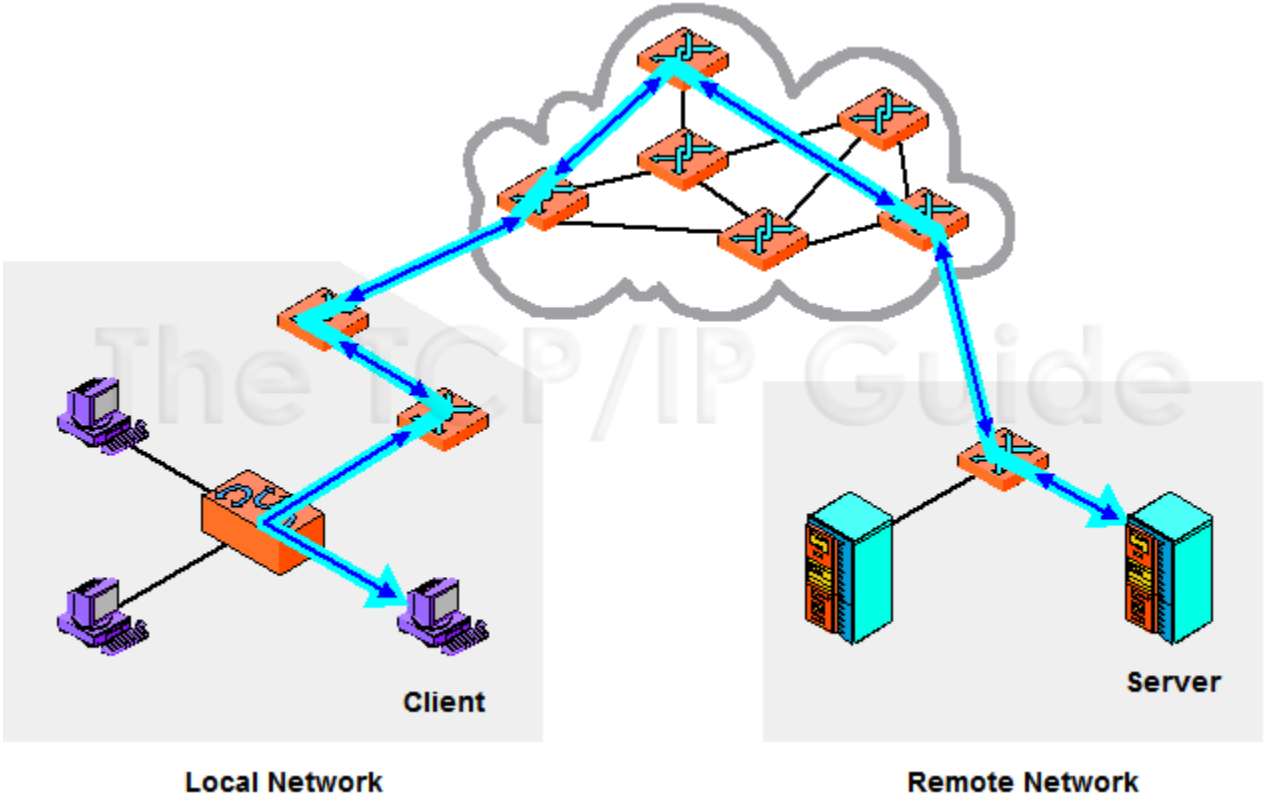
Serviço *Best-Effort*



IP – Internet Protocol

- O IP é o protocolo da camada de rede da arquitetura TCP/IP que implementa o serviço de entrega de pacotes não confiável, não-orientado a conexão, operando segundo o esquema de *best-effort*.
- O propósito fundamental do IP é rotear pacotes através de um conjunto de redes que estão inter-conectadas de maneira arbitrária.
- *Roteamento* refere-se ao processo de escolha de um caminho sobre o qual serão enviados os pacotes; *roteador* (*gateway*) refere-se ao elemento da rede que toma tal decisão.

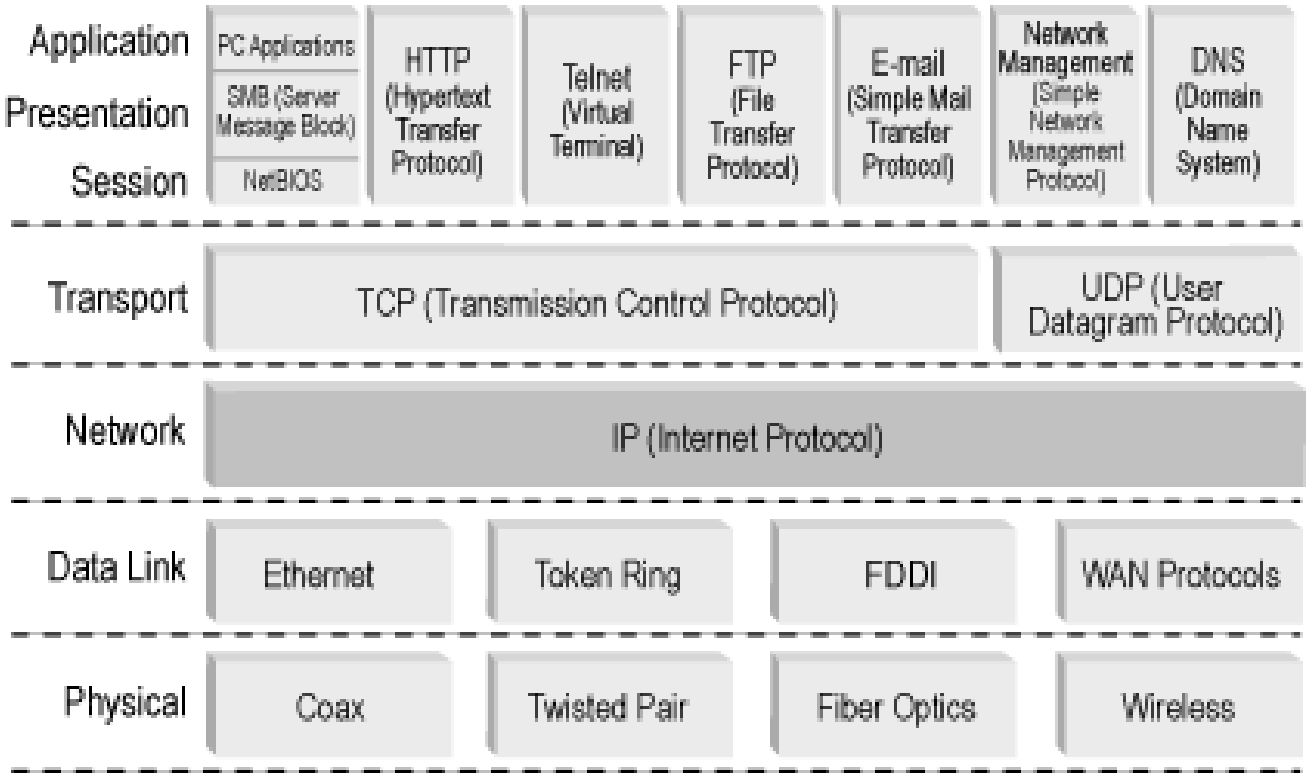
IP – Internet Protocol (cont.)



IP (“Internet Protocol”) (cont.)

- O pacote é roteado pelo IP com base no endereço destino que ele carrega no cabeçalho.
- Se um host A quer se comunicar com um host B localizado em uma rede remota, ele deve transmitir o pacote para um roteador diretamente conectado a sua rede local.
- O roteador encaminha então o pacote através do sistema interconectado de redes e roteadores, até que, eventualmente, o pacote chega a um roteador que está na mesma rede do *host* destino.
- Este roteador (chamado roteador final), entrega o pacote ao host B na rede destino.

O IP e o Modelo OSI



Características do IP

- Modelo de endereçamento:
  - É completamente independente da rede física.
- Encapsulamento:
  - Cria um cabeçalho com informações de controle, que é anexado aos dados vindo da camada de transporte (por exemplo, do TCP ou UDP) ou de usuários da própria camada de rede (por exemplo, ICMP ou IGMP).
- Não existe suporte para retransmissão de dados perdidos ou corrompidos.

Características do IP (cont.)

- Não existe reconhecimento de pacotes, isto é, não há *acknowledgment* de chegada de pacotes, seja fim-a-fim (*end-to-end*) seja roteador-a-roteador (*hop-by-hop*).
- Não existe nenhum mecanismo de controle de fluxo
- Não existe sequenciamento de pacotes (a entrega pode ser fora de ordem).

Características do IP (cont.)

- Realiza a fragmentação e a remontagem de pacotes.
  - Quebra os pacotes em pedaços menores para que eles possam atravessar redes com menor valor de MTU (*Maximum Transmission Unit*).
- O controle de erros exercido é mínimo:
  - Provê apenas um checksum de 16 bits no cabeçalho, que é usado pelas estações receptoras para validar os dados de controle.
- Permite às aplicações requisitarem diferentes tipos de níveis de desempenho para a entrega do pacote usando o campo TOS (Type of Service).

Questão Importante

- Por que então usar um serviço sem garantias de entrega e sem confirmação de recebimento de dados como base de toda a Internet?

- A resposta é simples:
  - Estabelecer conexões, garantir entrega, checagem de erros, controle de fluxo e outras funções tem um alto custo: **performance**.
  - Custa tempo, recursos do computador e largura de banda executar tais tarefas e elas não são necessárias para todas as aplicações.
  - Se uma certa característica de QoS é requerida pela aplicação, ela pode ser perfeitamente provida no nível de transporte ou mesmo no nível de aplicação, livrando as outras aplicações que não necessitam de tais características de terem de “usá-las”.

Grupos de Funções Básicas do IP

- Endereçamento
- Encapsulamento e formatação (packing) de dados
- Fragmentação e remontagem
- Roteamento / entrega indireta

Os campos do cabeçalho do protocolo IP contém informações que viabilizam a execução dessas funções.

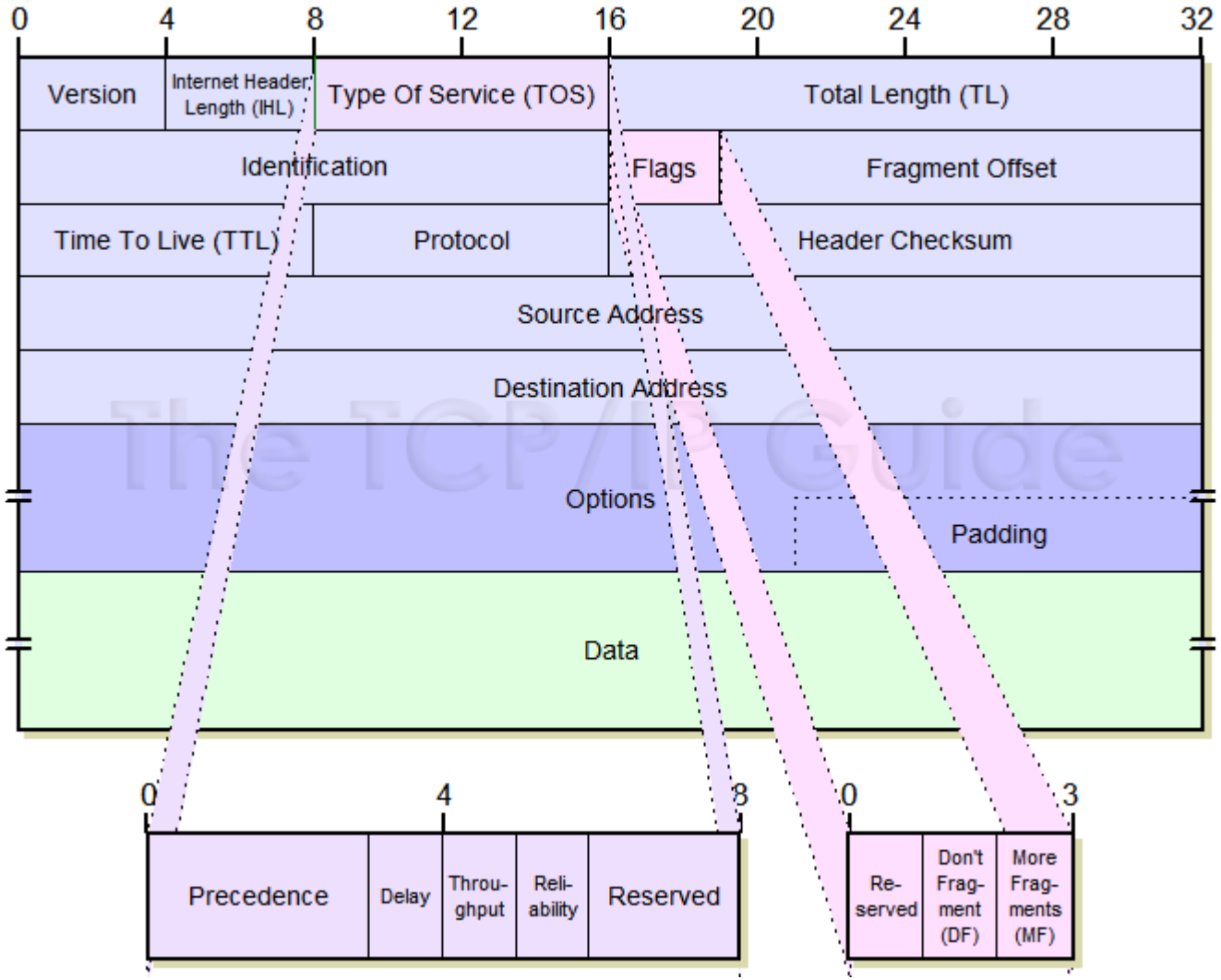
Formato do Datagrama

- O datagrama ou pacote IP é a unidade básica de dados do nível IP, ou nível de rede. Um pacote IP está dividido em duas áreas: cabeçalho e dados (payload).

- O cabeçalho contém toda a informação necessária para identificar o conteúdo do datagrama e tomar decisões de roteamento, dentre outras funções do IP.

- Na área de dados está encapsulado a unidade de dados do nível superior, ou seja um segmento TCP ou UdP ou da própria camada (pacote ICMP ou IGMP) .

Formato do Pacote IPv4



Versão (*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.

Tamanho do Cabeçalho (*Header Length*)

- Define o tamanho do *header* e é medido em número de palavras de 32 bits (4 bytes).
- O maior tamanho de um cabeçalho IP é de 15 palavras (60 bytes) mas a maioria possui o tamanho mínimo, que é de 5 palavras (20 bytes).
- No caso de um cabeçalho mínimo, o campo *Options* é vazio e não existe nenhum ajuste de tamanho de cabeçalho (*padding*).

Tamanho Total (*Total Length*)

- Define o tamanho do pacote, sendo expresso em número de bytes.
- A medida inclui cabeçalho + dados, logo:
  - Tamanho dos dados = Total Length – Header Length.
- Se o pacote for fragmentado, este campo indica o tamanho do fragmento e não do pacote original.

Tamanho Total (*Total Length*) (cont.)

- O tamanho do campo é de 16 bits, o que permite um tamanho total de até 65.536 bytes ou 64 KB.
- Todavia, este tamanho de pacote é impraticável para a maioria das redes conectadas à Internet.
- Por definição, o tamanho de datagrama IP que um host deve tratar é de 576 bytes (seja o original ou um fragmento).
- Um host somente pode enviar pacotes maiores do que 576 bytes caso tenha certeza que estes pacotes poderão ser encaminhados pela inter-rede (algoritmo *MTU Path Discovery*)

Endereço IP Origem e Destino (*Source and Destination IP Address*)

- Especificam os endereços IP de origem e de destino do datagrama: É com base no endereço de destino que o pacote é roteado pelo IP.
- Esses campos não são alterados no cabeçalho durante toda a transmissão, independentemente do número de roteadores pelos quais o pacote tenha passado ou se ele foi fragmentado.

Protocolo (*Protocol*)

- Número do protocolo que está sendo encapsulado no pacote IP.
- Permite entregar os dados incluídos no datagrama ao protocolo apropriado usuário do IP (UDP, TCP, ICMP, IGMP, etc.).
- Os números dos protocolos são atribuídos pela IANA e os valores mais comuns são:

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

Verificação de Soma (*Header Checksum*)

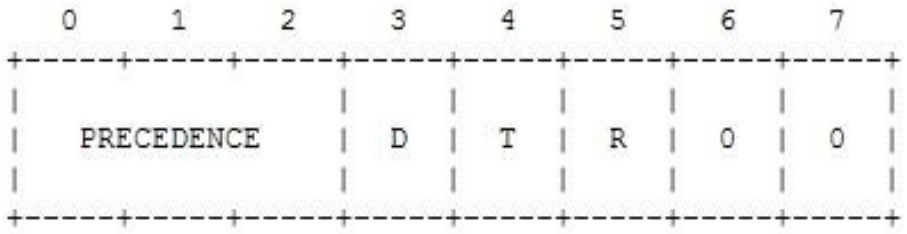
- Provê uma proteção básica contra a corrupção de dados durante a transmissão.
- É calculado apenas sobre o campo de cabeçalho, ou seja, não cobre o campo de dados, como é caso do TCP, UDP, ICMP e IGMP.
- O cálculo deste CRC é bem simples, ao contrário do código CRC de algumas tecnologias de enlace, como Ethernet.

TOS (Type of Service) (cont.)

- Contém informações de qualidade de serviço (QoS – Quality of Service) que podem afetar a maneira como o datagrama é roteado.
- No passado, a maioria dos roteadores ignorava o campo de TOS. Com a introdução das redes multimídia e a consequente necessidade de controle de QoS, este campo passou a ser valorizado.
- Originalmente definido na RFC 791, o campo TOS foi redefinido posteriormente pela RFC 1349 para uso da técnica de controle de QoS denominada “Serviço Diferenciado” (*DiffServ*).

TOS (Type of Service) (cont.)

- Conteúdo do campo ToS original:
  - 3 bits para indicar precedência
  - 3 bits para indicar retardo (D), vazão (T) e confiabilidade (R)
  - 2 bits para uso futuro, configurados como zero.



TOS (Type of Service) (cont.)

- Os bits de precedência:
  - São usados pelo transmissor para informar a importância relativa do datagrama.
  - Foram 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.:

TOS (Type of Service) (cont.)

- Os bits (D, T e R) foram definidos para serem usados para indicar uma combinação de retardo, vazão e confiabilidade.
- Na maioria das situações, essas indicações eram feitas aos pares ou até mesmo configurando três bits ao mesmo tempo.
- Por exemplo, um pacote que precisa de baixo retardo precisa normalmente de alta vazão e confiabilidade.

TOS (Type of Service) (cont.)

- Quando iguais a 0 estes bits “sinalizadores” indicam um valor normal do serviço e não especificam qualquer tratamento diferenciado ao pacote. Quando iguais a 1, indicam que o pacote deve procurar a saída que atenda ao sinalizador ativado.
- Por exemplo, caso o sinalizador de retardo esteja ativo, o pacote deverá ser encaminhado pela saída de menor retardo. No caso em que os sinalizadores de vazão e confiabilidade estão ativos o pacote deverá sair pela interface que mlelhor suportar essa combinação.

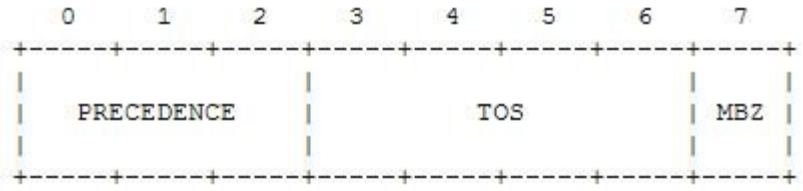
TOS (Type of Service) (cont.)

- Sensibilidade das aplicações aos parâmetros de QoS:

Tipo de Tráfego	Vazão	Perdas	Latência	Jitter
Voz	Muita Baixa	Média	Alta	Alta
Comércio Eletrônico	Baixa	Alta	Alta	Baixa
Transações	Baixa	Alta	Alta	Baixa
Correio Eletrônico	Baixa	Alta	Baixa	Baixo
Acesso Remoto (Telnet)	Baixa	Alta	Média	Baixa
Navegação Web Casual	Baixa	Média	Média	Baixa
Navegação Web Crítica	Média	Alta	Alta	Baixa
Transferência de Arquivos	Alta	Média	Baixa	Baixa
Videoconferências	Alta	Média	Alta	Alta
Multicast	Alta	Alta	Alta	Alta

TOS (Type of Service) (cont.)

- A RFC 1349 redefine as funções do campo ToS, bem como uma nova abordagem de seu uso. definindo-o do seguinte modo:
  - 3 bits para indicar precedência (nível de prioridade)
  - 4 bits para indicar ações de ToS
  - 1 bit reservado chamado de MBZ - Must be Zero



- As indicações de precedência (3 bits) permitem indicar até oito situações possíveis que podem ser configuradas em uma rede. A responsabilidade da decisão baseada na leitura deste campo, entretanto, é exclusiva da própria rede (não adianta configurar).

TOS (Type of Service) (cont.)

- Os sinalizadores D, T e R foram levemente modificados e substituídos pela indicação de ToS que ocupa quatro bits.
- Os três primeiros bits continuam com o mesmo significado; entretanto, o quarto bit indica o custo monetário do canal. Ou seja, o preço de usar ou não um determinado canal de saída pode ser decisivo.
- Configurando todos os campos como zero isso indica que o pacote utilizará a saída considerada normal. Quando todos os sinalizadores são configurados como zero o campo é chamado de ToS padrão (Default ToS).

TOS (Type of Service) (cont.)

- A decisões de roteamento são agora tomadas após uma análise lógica deste campo como um todo. O fato do bit de retardo estar ativo não define a saída com menor retardo como a melhor escolha. Antes é preciso avaliar também as combinações dos demais bits.
- Deste modo, um pacote não será descartado por ter este sinalizador ativo e o roteador acreditar que o retardo em questão é alto. As decisões são tomadas em função de minimizar ou maximizar uma característica em função de uma possível linha de saída e não como uma decisão arbitrária imposta pelo emissor do pacote.
- Em resumo, o campo de tipo de serviço (ToS) é usado como indicação para os routers e hosts para tratamento diferenciado na escolha da rota.

Time to Live (TTL)

- Tempo de vida do pacote. Representa o tempo máximo, em segundos, que o pacote pode circular na Internet (8 bits => 255 s = 4,25 min).
- O TTL é necessário devido a possibilidade do pacote circular indefinidamente na rede.
- Na prática, é implementado como um contador de *hops*, que é decrementado a cada roteador, ou seja, um salto ocorre sempre que o datagrama atravessa um roteador.
- Quando o valor do TTL chega a zero o datagrama é descartado e uma mensagem de controle ICMP é enviada para o *host* emissor do pacote.

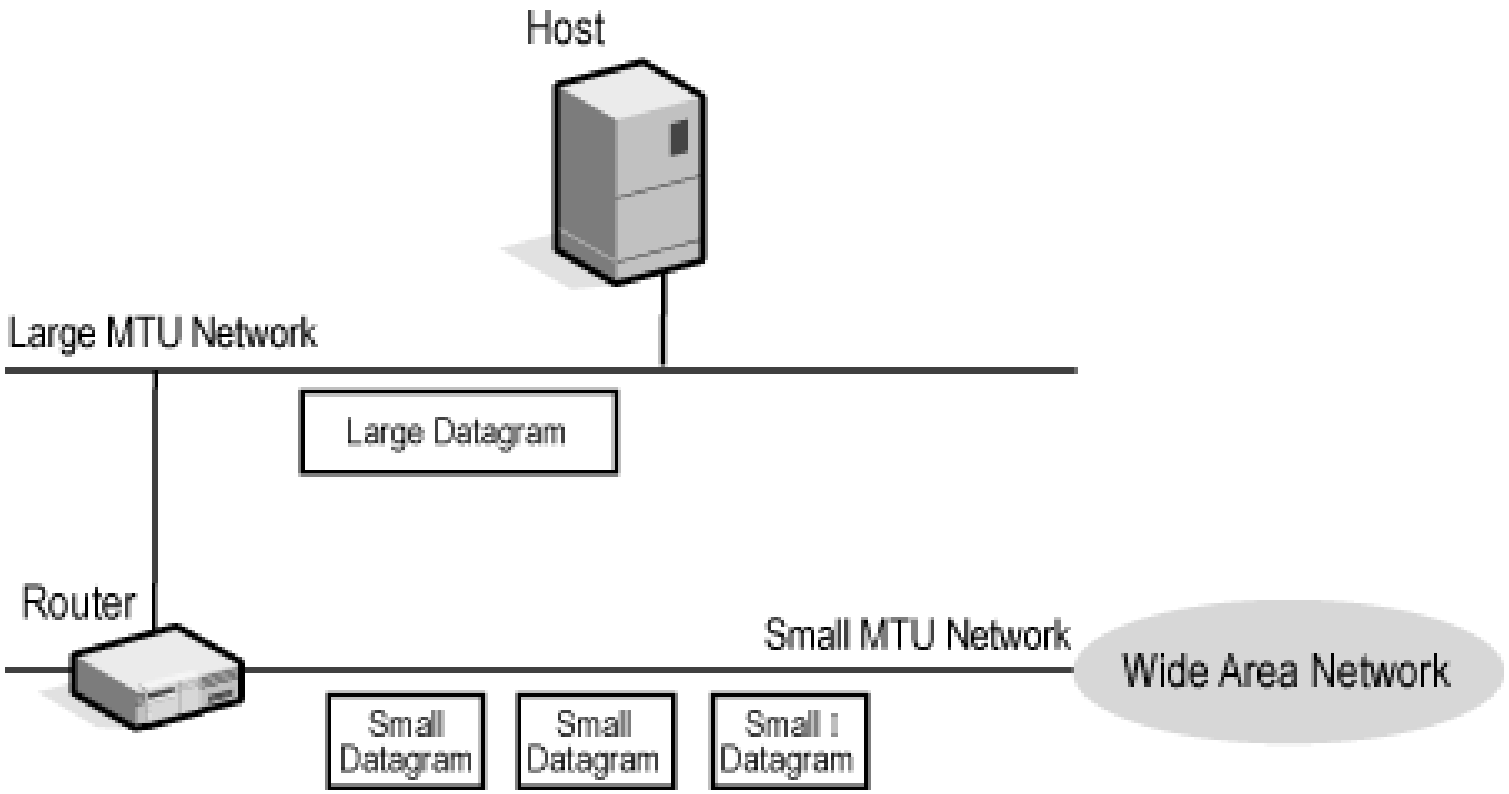
Time to Live (TTL) (cont.)

- O TTL representa, portanto, o número de saltos (*hops*) que o pacote poderá realizar antes de ser descartado.
- 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.
- O valor padrão sugerido atualmente para o campo TTL é 64.

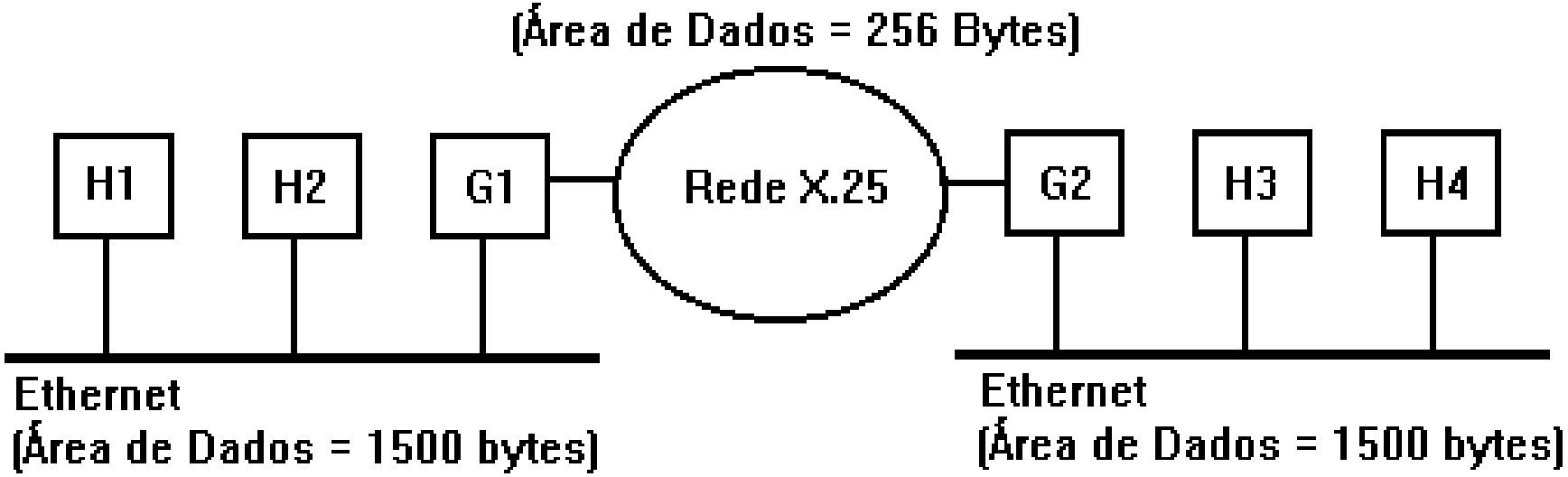
Fragmentação

- Cada padrão de tecnologia de rede possui um valor característico de MTU – Maximum Transmission Unit. Por exemplo:
  - Ethernet: 1500 bytes
  - ATM: 53 bytes,
  - FDDI: 4500 bytes
- Assim, para enviar datagramas maiores do que a MTU da rede, ele deve ser fragmentado.
- A remontagem deve ocorrer na estação destino, a partir de todos os fragmentos do datagrama.

Fragmentação (cont.)



Fragmentação (cont.)

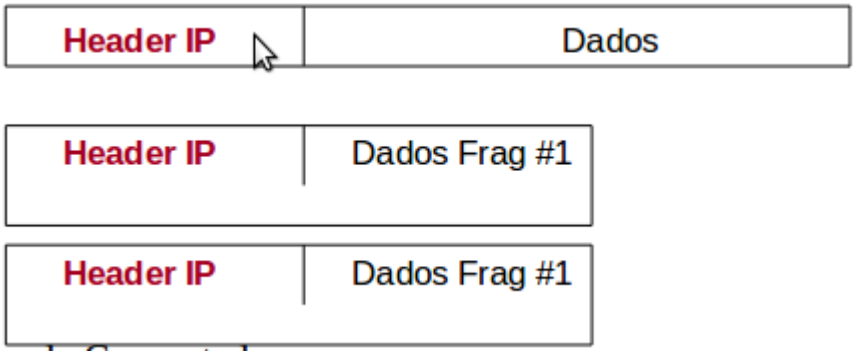


Fragmentação (cont.)

- Processo de divisão de um datagrama em unidades menores, denominados de *fragmentos*.
- Necessário quando o IP é forçado a transmitir um datagrama através de uma rede que opera com pacotes de menor tamanho (menor MTU).
- Os fragmentos atravessam a Internet separadamente até que chegam ao destino final. É responsabilidade da estação destino remontar os fragmentos da mensagem original.

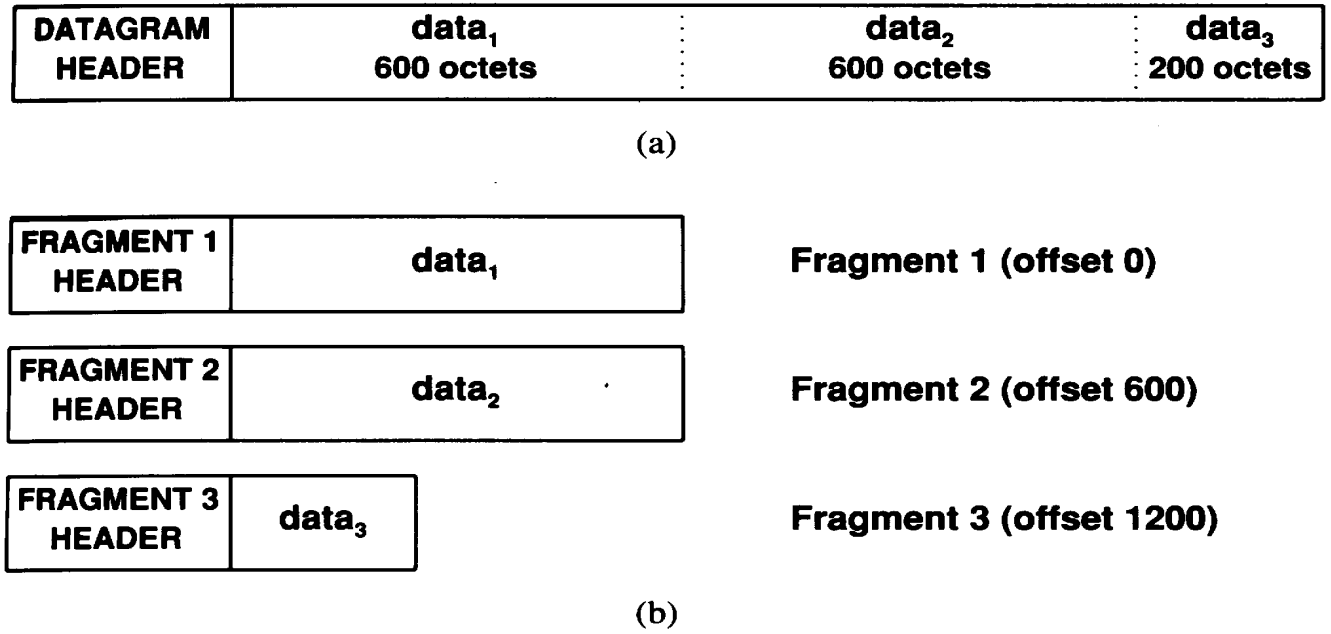
Fragmentação (cont.)

- Cada fragmento recebe uma cópia do cabeçalho IP do datagrama original e uma porção de dados



Fragmentação (cont.)

- Pacote IP = 1400 bytes de dados; cabeçalho = 20 bytes; MTU = 620 bytes



Fragmentação (cont.)

- Quatro campos do cabeçalho IP controlam a fragmentação e remontagem de datagramas:
  - *Identification*, *Total Length*, *Fragment Offset* e *Flags*.
- Esses quatro campos são alterados no header dos fragmentos (diferem do datagrama original)
- Eventualmente, um datagrama pode ser marcado como "*don't fragment*". Se ele for transmitido por uma rede com menor MTU, ele será descartado.
  - Ex: frame I do MPEG (codificador de vídeo)
- Observe que a perda de um fragmento implica na perda do pacote inteiro.

Identification

- É um rótulo de identificação para o datagrama. O transmissor atribui este valor ao datagrama original.
- Todos os fragmentos de um datagrama possuem o mesmo valor de identificação do datagrama original.
- Juntamente com o endereço IP origem (campo *Source IP Address*), identifica o pacote original ao qual os fragmentos pertencem.

Flags (cont.)

- Sinalizadores binários usados no processo de fragmentação:
  - Determinam se um datagrama pode ou não ser fragmentado.
  - Indicam se existem mais fragmentos ou se este é o último de uma série de fragmentos.
- Campo de 3 bits. Um deles é configurado como zero (reservado), sendo os outros dois:
  - bit DF = Don't Fragment
  - bit MF = More Fragments

Flags (cont.)

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

Flags (cont.)

- O bit DF informa ao roteador que ele não pode fragmentar o datagrama em nenhuma hipótese, pois o *host* de destino não saberá remontá-lo (ex: frame do tipo I do MPEG).
- Em algumas situações isso significa que o datagrama deve contornar essa rede. Caso não se possa contornar essa rede de menor MTU, então o datagrama será descartado.
- O descarte de datagramas acontece sempre que o encaminhamento não é viável sem que ocorra fragmentação.

Flags (cont.)

- Quando o sinalizador DF está desativado, e caso haja necessidade, o pacote poderá ser fragmentado na origem ou pela inter-rede.
- O bit MF quando ativo, indica que o pacote é um fragmento e que existem outros, de uma série de fragmentos, para chegar.
- O último fragmento de uma série dever ter o bit MF desativado (configurado em zero) indicando que não existem mais fragmentos.

Fragment Offset

- Indica o deslocamento do fragmento em relação ao datagrama original. Permite ao IP executar a remontagem dos fragmentos.
- É medido em unidades de 8 bytes (“*fragment blocks*”). O primeiro fragmento de uma série sempre possui *offset* zero e todos os outros um valor múltiplo de 8.
- É um campo de 13 bits; logo, os valores de offset podem variar de 0 a 8192, o que corresponde à faixa de 0 a 65.536 bytes.

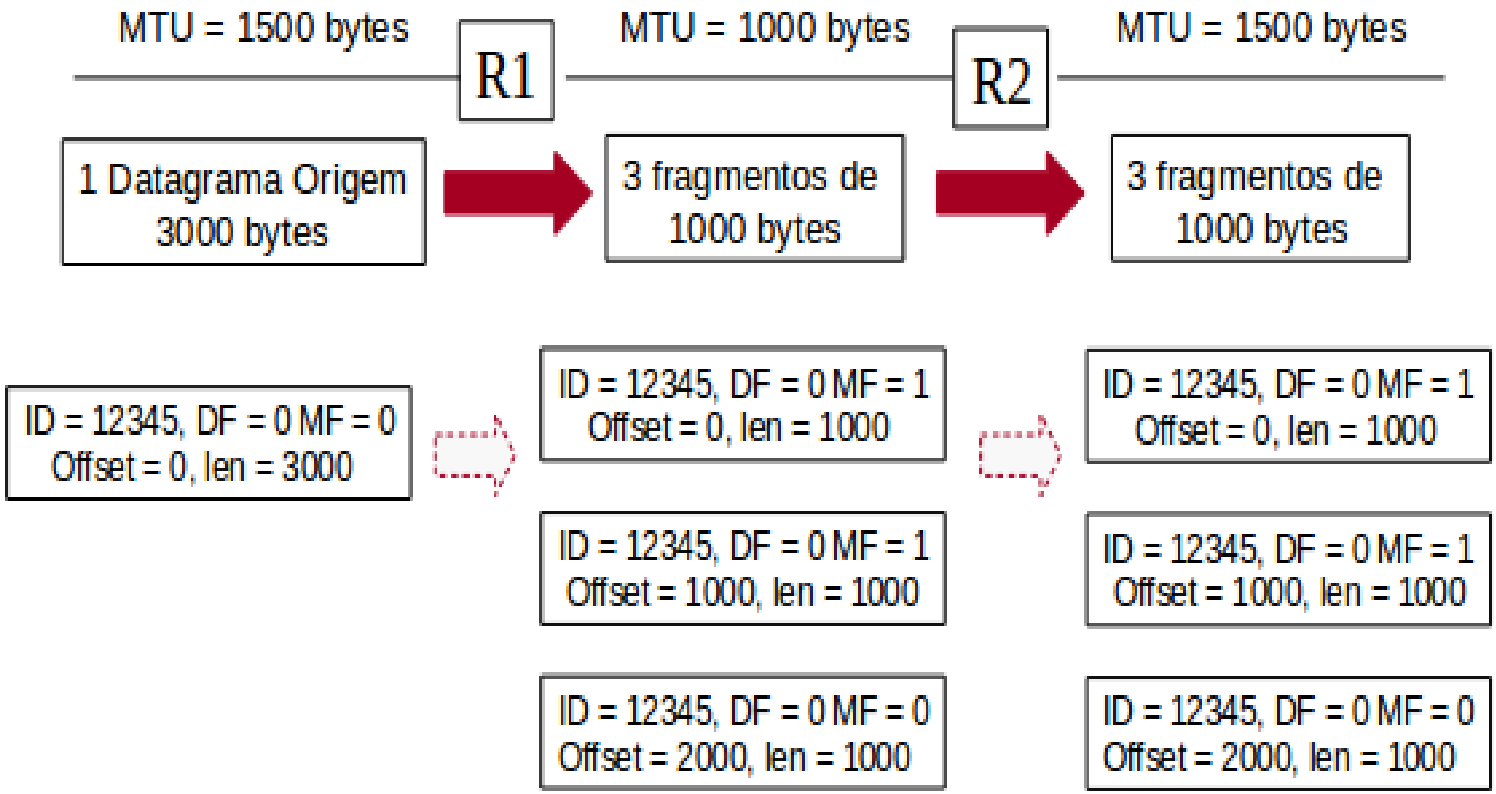
O Processo de Fragmentação

- Inicialmente é examinado o campo de *Flags*. Se bit DF = 1 e não há rota sem possibilidade de fragmentação, não há nada a fazer e o datagrama é descartado.
- Se o bit DF = 0, a porção de dados é quebrada em partes consistentes com o tamanho com o MTU do próximo *link*. Cada parte deve ter um tamanho múltiplo de 8 bytes ("*8-byte boundary*").
- A cada parte é atribuído um cabeçalho IP. Os seguintes campos serão iguais aos dos datagrama original:
  - *Destination Address*, *Source Address*, *Protocol* e *Identification*.

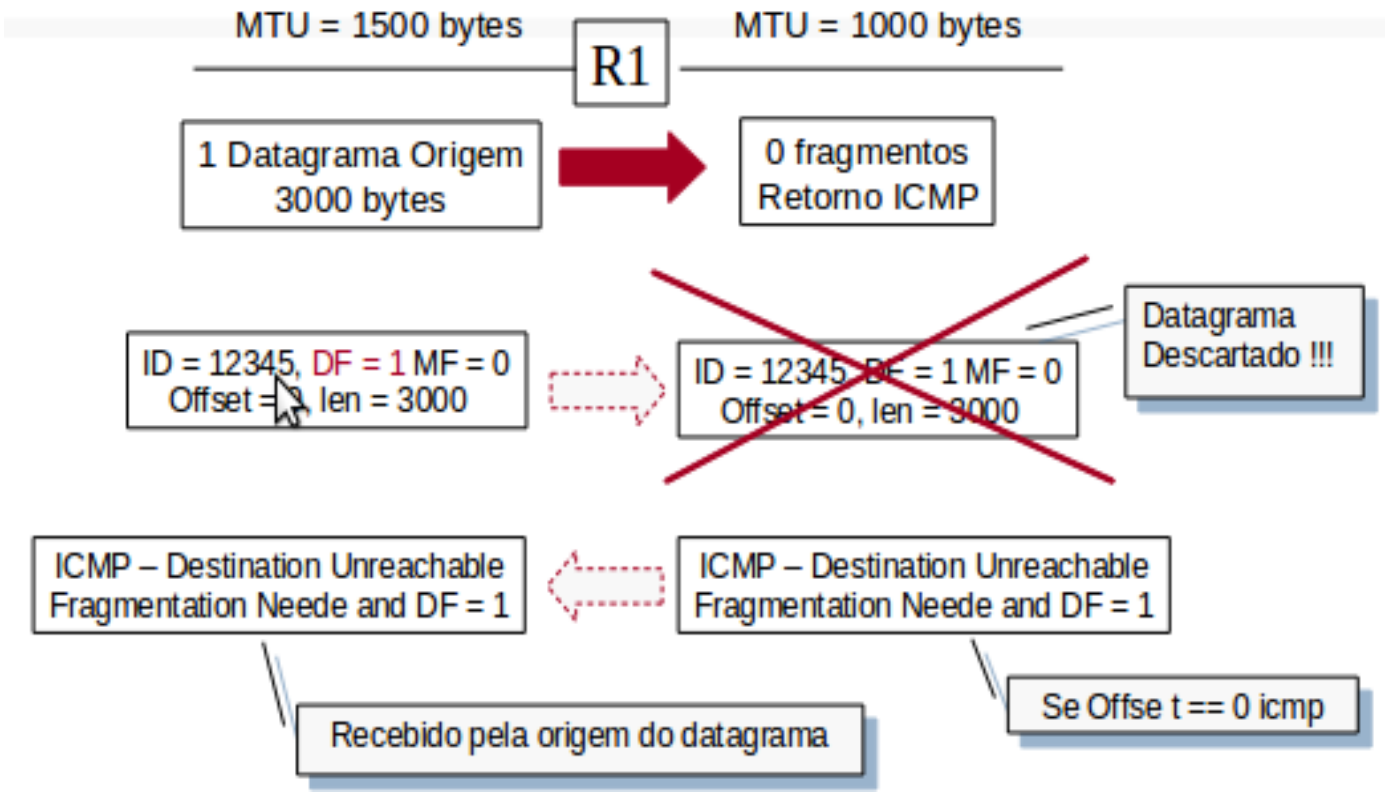
O Processo de Fragmentação (cont.)

- Os campos abaixo são atribuídos para cada fragmento, separadamente:
  - Total Lenght: tamanho total do fragmento levando em conta o pedaço corrente;
  - Bit MF do campo de Flags: deverá ter o valor 1 em todos os fragmentos, exceto o último, que terá o valor zero.
  - Fragment offset: deve indicar o deslocamento do fragmento em relação ao início do datagrama original. O valor atribuído a esse campo é o valor do deslocamento, em bytes, dividido por 8.
  - Checksum: é calculado para cada um dos fragmentos individuais.

Exemplo 1


MTU = 1500 bytes
R1
MTU = 1000 bytes
R2
MTU = 1500 bytes
1 Datagrama Origem
3000 bytes
3 fragmentos de
1000 bytes
3 fragmentos de
1000 bytes
ID = 12345, DF = 0 MF = 0
Offset = 0, len = 3000
ID = 12345, DF = 0 MF = 1
Offset = 0, len = 1000
ID = 12345, DF = 0 MF = 1
Offset = 1000, len = 1000
ID = 12345, DF = 0 MF = 0
Offset = 2000, len = 1000
ID = 12345, DF = 0 MF = 1
Offset = 0, len = 1000
ID = 12345, DF = 0 MF = 1
Offset = 1000, len = 1000
ID = 12345, DF = 0 MF = 0
Offset = 2000, len = 1000

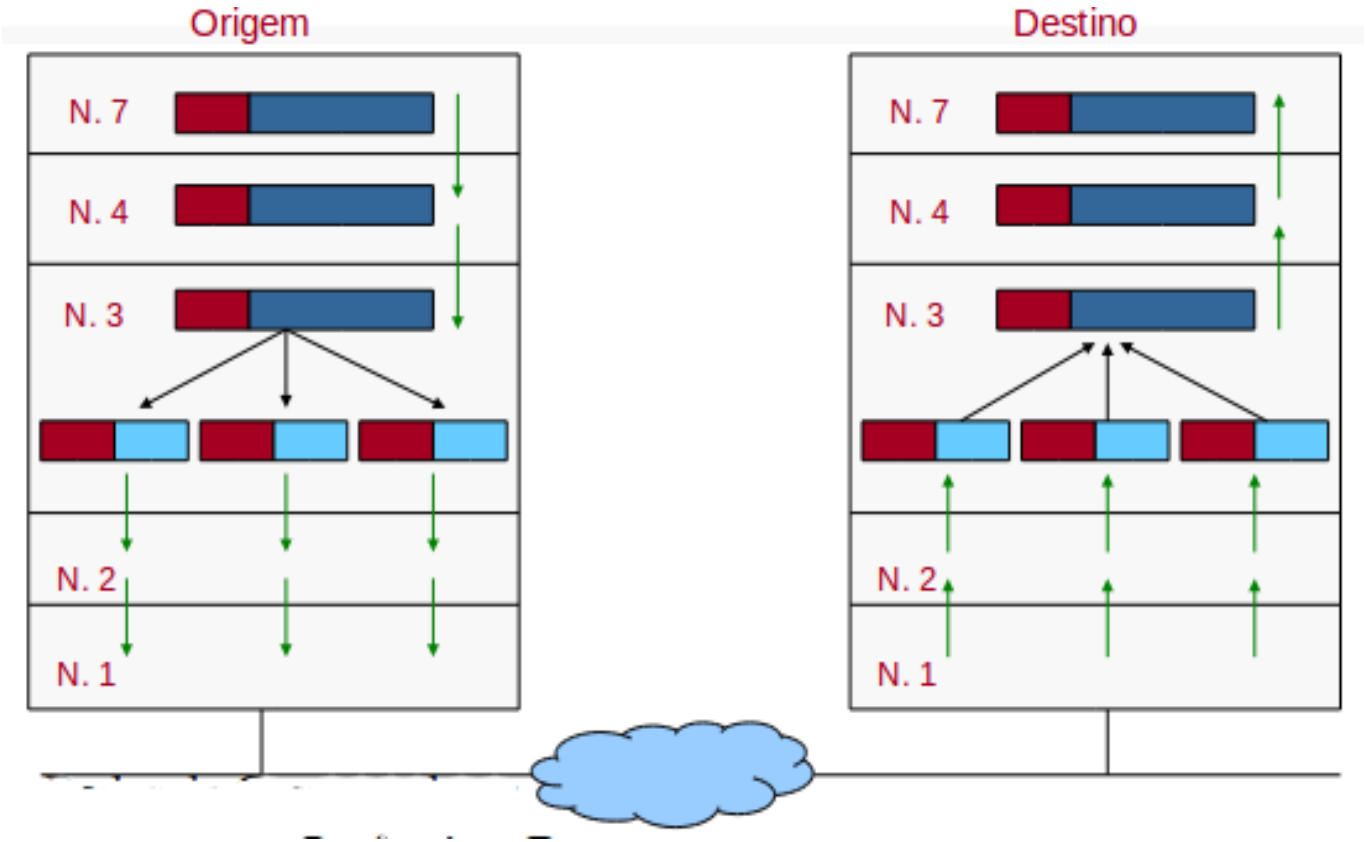
Exemplo 2


MTU = 1500 bytes
R1
MTU = 1000 bytes
1 Datagrama Origem
3000 bytes
0 fragmentos
Retorno ICMP
ID = 12345, DF = 1 MF = 0
Offset = 0, len = 3000
ID = 12345, DF = 1 MF = 0
Offset = 0, len = 3000
Datagrama
Descartado !!!
ICMP – Destination Unreachable
Fragmentation Neede and DF = 1
ICMP – Destination Unreachable
Fragmentation Neede and DF = 1
Recebido pela origem do datagrama
Se Offse t == 0 icmp

Remontagem de Datagramas (cont.)

- A remontagem ("*reassembly*") do pacote original só 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 MF também igual a zero.
- A estação receptora usa um "timeout" de remontagem. Se faltam fragmentos e o tempo se esgota, os fragmentos são descartados e a estação destino envia para origem uma mensagem ICMP de Time Exceeded.

Remontagem de Datagramas



Remontagem de Datagramas (cont.)

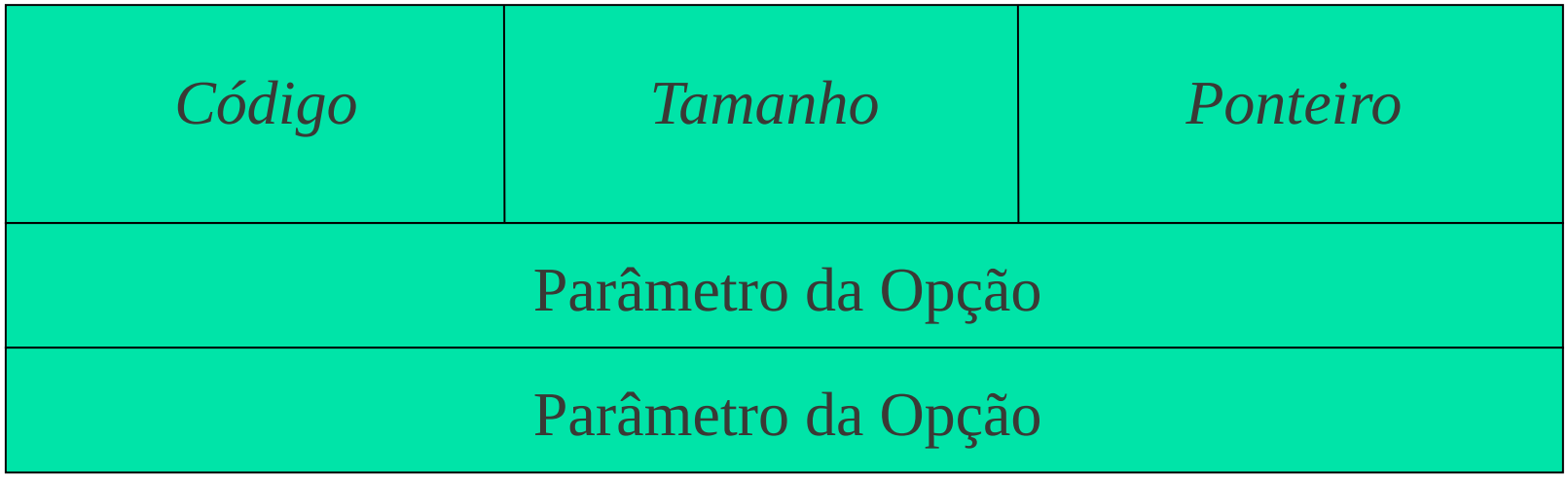
- O campo *Total Length* contém o tamanho total do fragmento e não o tamanho do datagrama original; logo, não tem como a estação destino prever com exatidão o tamanho do *buffer* necessário para acomodar os datagramas.
- Assim, existe uma omissão inconveniente no IP: a estação destino não tem como saber qual é o tamanho total do datagrama até que o último fragmento chegue.

O Campo “Options”

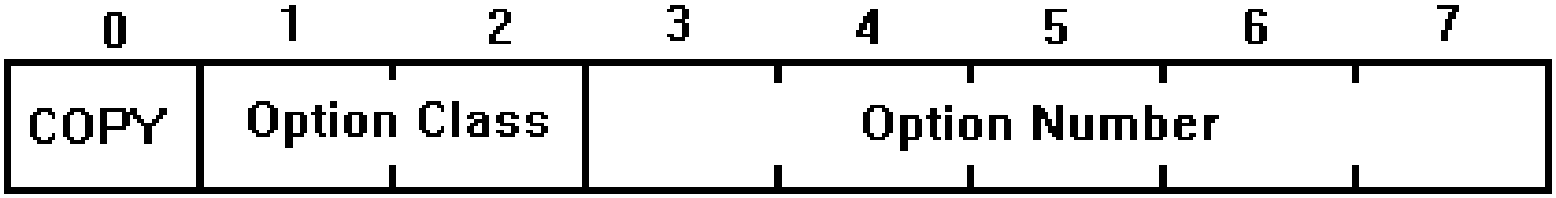
- O principal propósito do campo *Options* do cabeçalho do IP é prover para o administrador de rede ferramentas para testar e “depurar” a rede.
- Possui tamanho variável, sendo que até 40 bytes podem ser usados no cabeçalho para armazenar as *Options* do protocolo. Pode elevar, portanto, o tamanho do cabeçalho de 20 para até 60 bytes.
- O suporte às opções é obrigatório e deve estar presente em qualquer implementação do IP residente em *hosts* e roteadores.

O Campo “Options”

- 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.



Formato do Campo Código (“Code”)



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

O Sub-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 que identificam as 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)

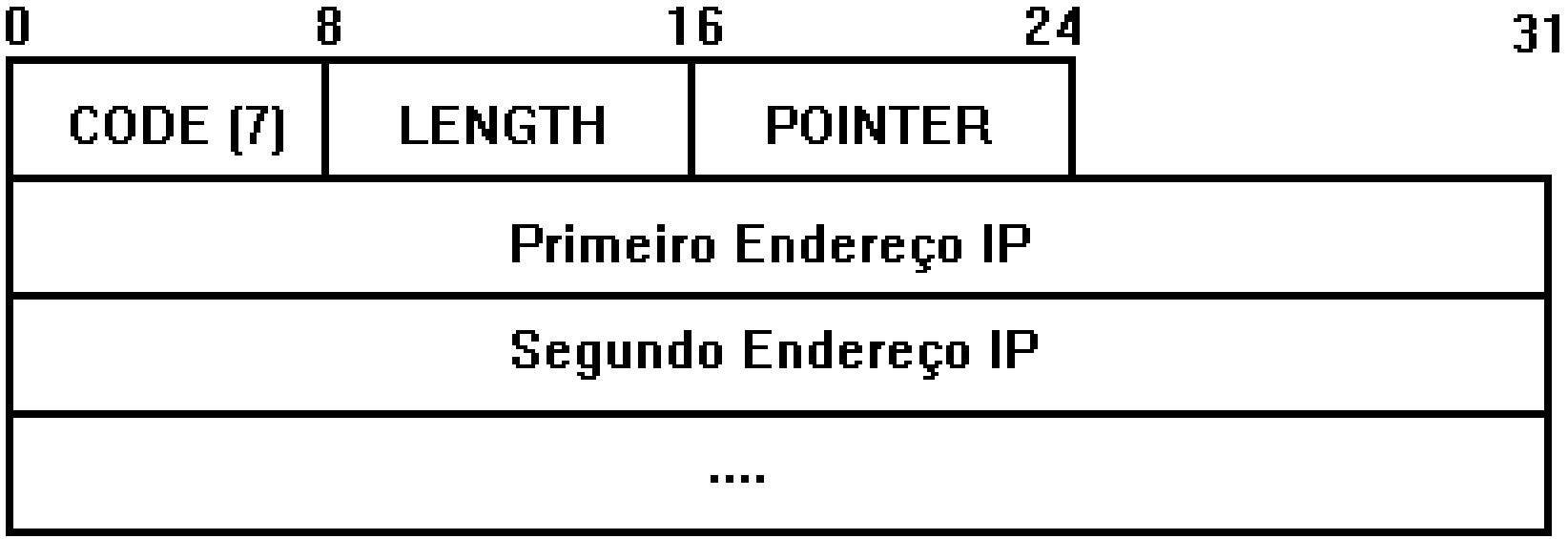
Valores do Campo *Code*

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

- Opção usada para monitorar o caminho que um pacote segue à medida que é roteado na Internet (ex: *ping –R*).
- A estação origem para reservar espaço no cabeçalho do datagrama para uma lista vazia de endereços IP.
- Cada roteador por onde o pacote passa adiciona o seu endereço IP à lista. Assim, ao chegar ao destino, o cabeçalho contém a lista dos roteadores visitados.
- Um máximo de 9 (nove) endereços IP podem ser armazenados no datagrama. Se a área já estiver cheia o datagrama é roteado sem que o endereço seja gravado no cabeçalho.

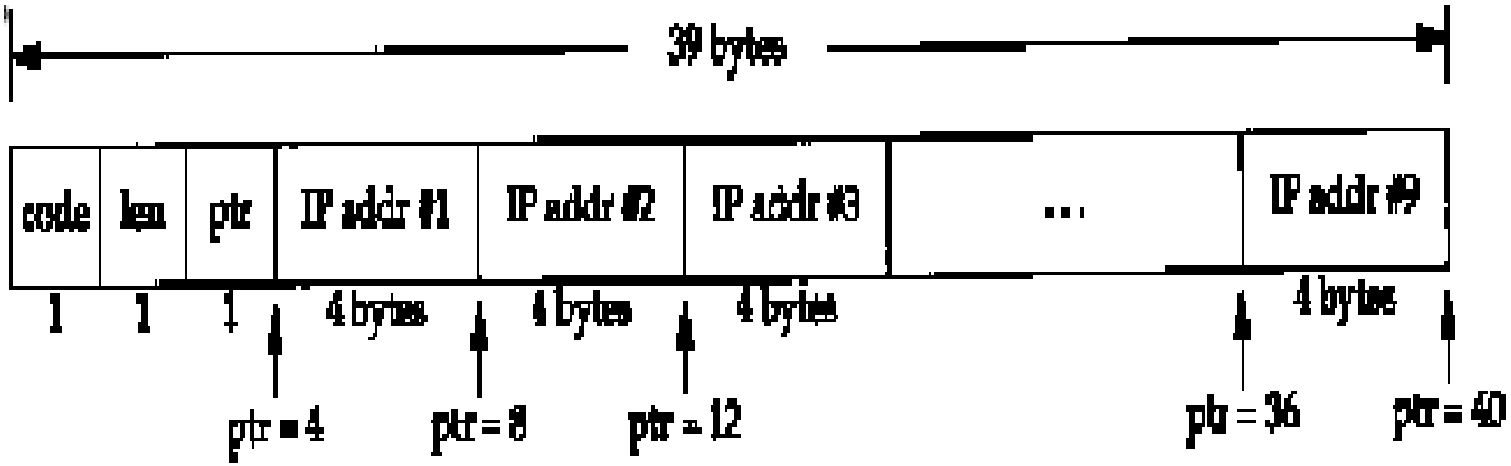
Record Route (cont.)



Record Route (cont.)

- Campos da opção:
  - *Code* define o código da opção (7);
  - *Length* define o número total de bytes;
  - *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.
- Após o nono endereço, o valor de *Pointer* é 40, indicando fim da lista.

Record Route (cont.)



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.

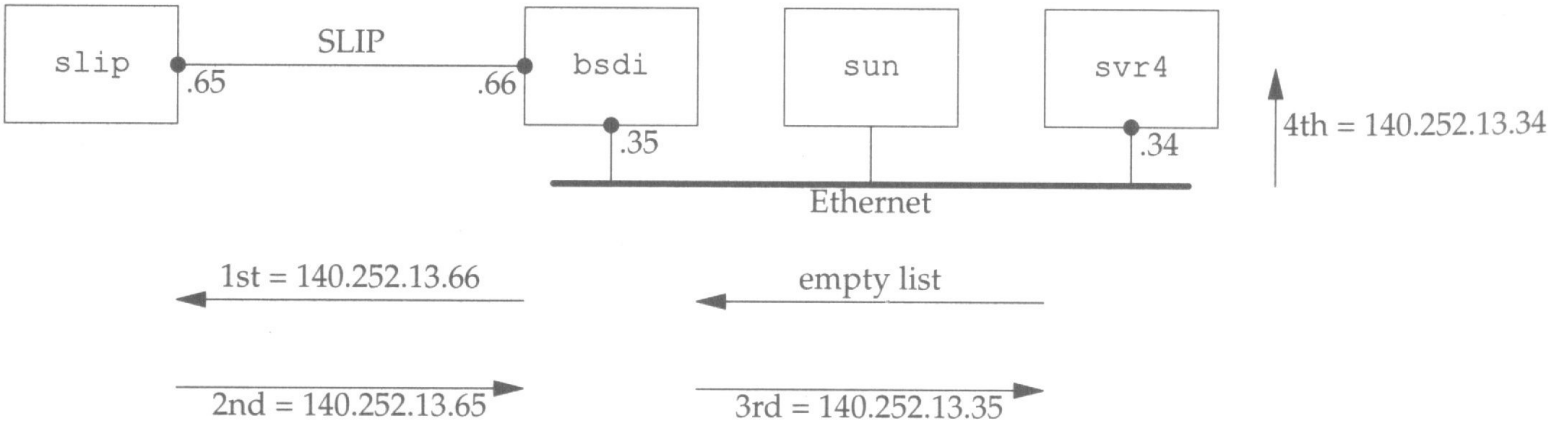
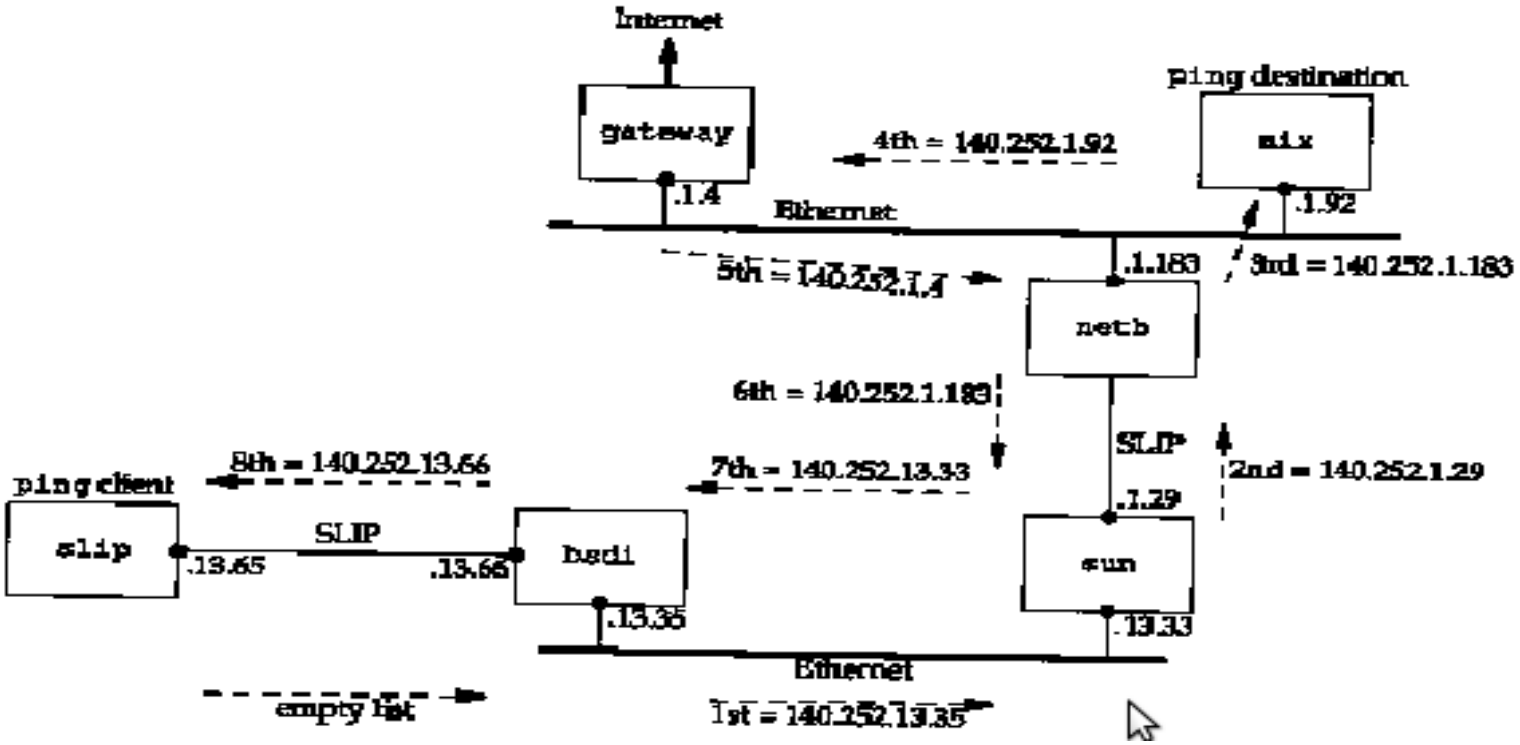


Figure 7.4 ping with record route option

Record Route (cont.)



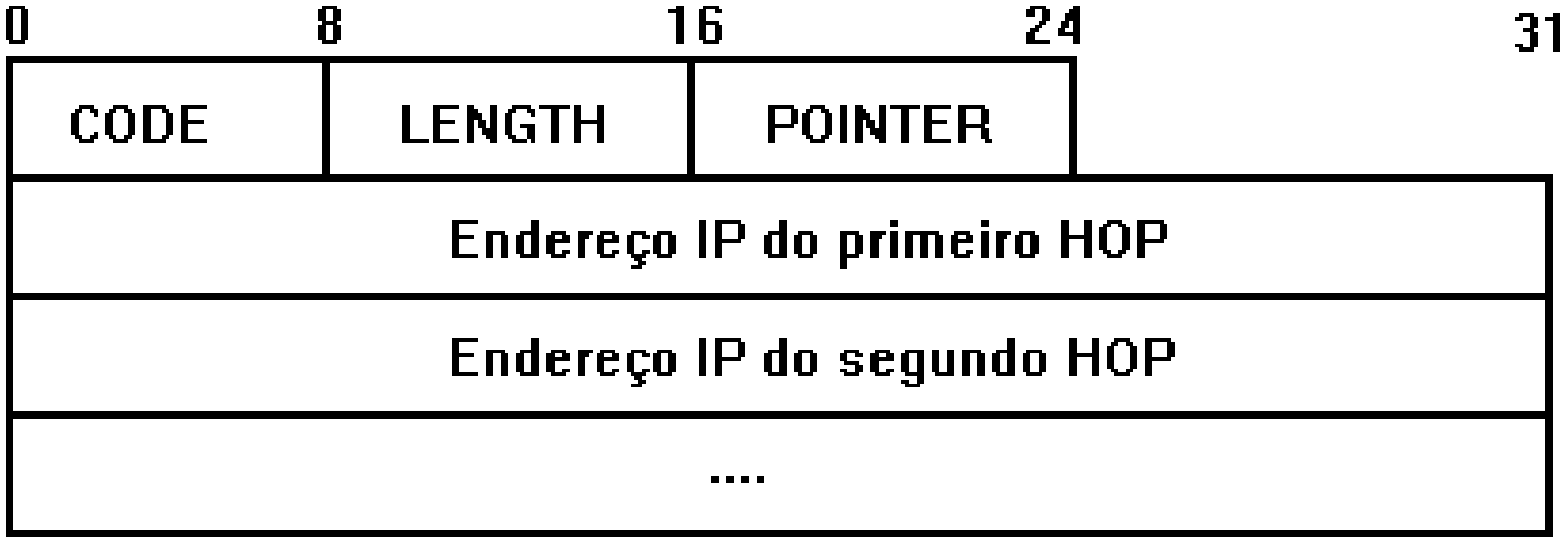
Record Route (cont.)

```
ctic-ufes@ctic-ufes:~$ ping -R ele.ufes.br
PING ele.ufes.br (200.137.67.20) 56(124) bytes of data.
64 bytes from 200.137.67.20: icmp_seq=1 ttl=62 time=3.52 ms
RR:     ctic-ufes.local (200.137.66.93)
        192.168.190.2
        200.137.67.126
        200.137.67.20
        200.137.67.20
        192.168.190.1
        router.inf.ufes.br (200.137.66.1)
        ctic-ufes.local (200.137.66.93)

64 bytes from 200.137.67.20: icmp_seq=2 ttl=62 time=4.45 ms     (same route)
64 bytes from 200.137.67.20: icmp_seq=3 ttl=62 time=4.24 ms     (same route)
64 bytes from 200.137.67.20: icmp_seq=4 ttl=62 time=3.45 ms     (same route)
64 bytes from 200.137.67.20: icmp_seq=5 ttl=62 time=4.25 ms     (same route)
64 bytes from 200.137.67.20: icmp_seq=6 ttl=62 time=3.88 ms     (same route)
64 bytes from 200.137.67.20: icmp_seq=7 ttl=62 time=3.75 ms     (same route)
64 bytes from 200.137.67.20: icmp_seq=8 ttl=62 time=4.41 ms     (same route)
64 bytes from 200.137.67.20: icmp_seq=9 ttl=62 time=4.49 ms     (same route)
64 bytes from 200.137.67.20: icmp_seq=10 ttl=62 time=4.28 ms    (same route)
^C64 bytes from 200.137.67.20: icmp_seq=11 ttl=62 time=3.97 ms  (same route)

--- ele.ufes.br ping statistics ---
```

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 (cont.)

- 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.
- Comando (Microsoft): `%ping -k routelist`

Strict Source Route (cont.)

- Opção faz parte do arsenal dos *hackers.*
- 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.

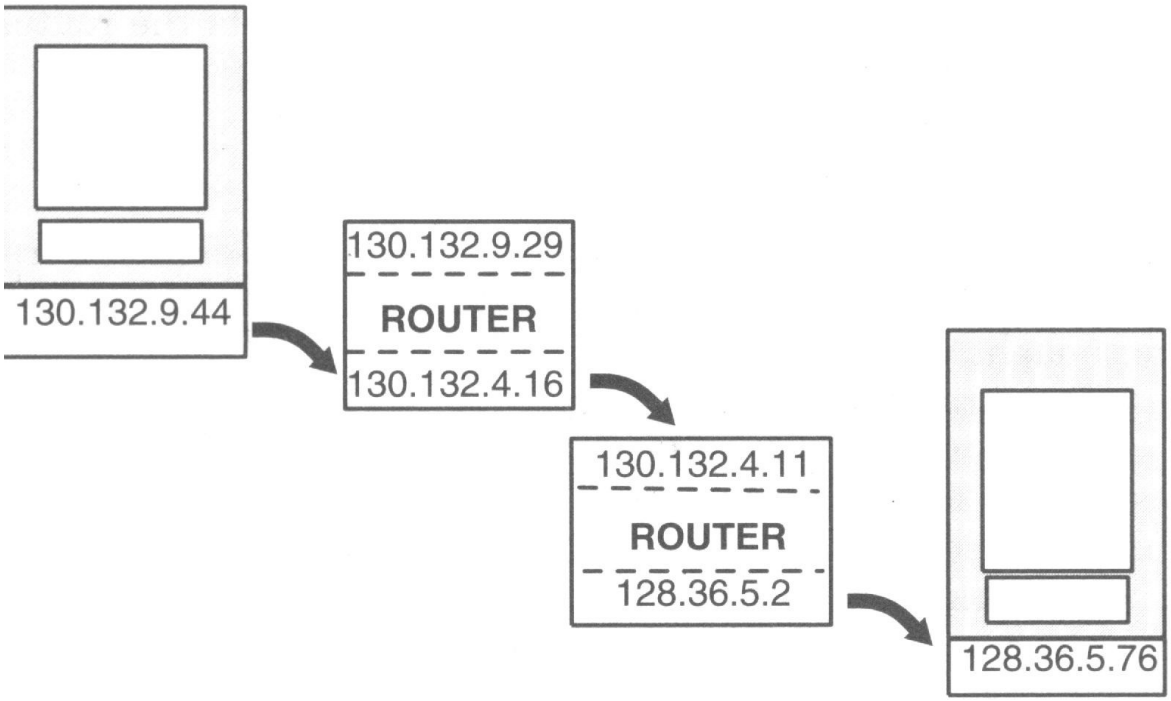
Loose Source Route

- Nesta 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).
- Comando (Microsoft): %ping -j *routelist*

Rota Reversa

- Quando a opção *SR/RR* 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 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.

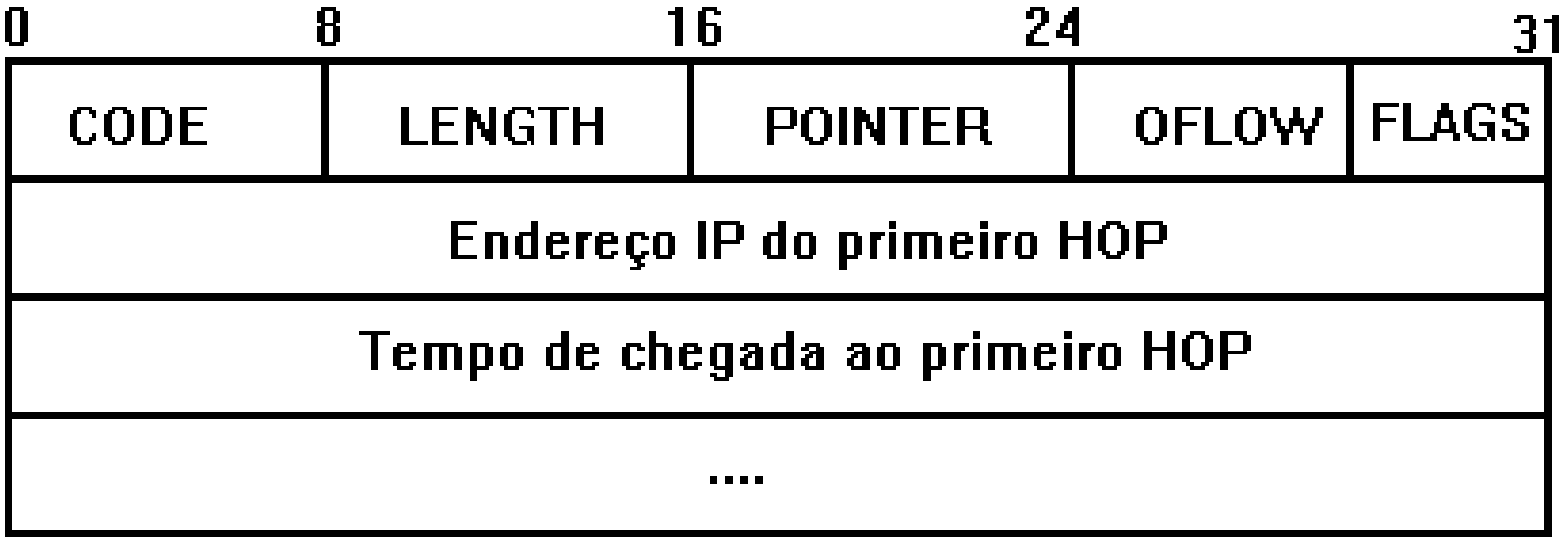
Time Stamp

- 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 por programas de administração para monitorar o desempenho da rede.

Time Stamp (cont.)

- 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 (cont.)

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

- 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.
- Descarta o pacote se o roteador não possui espaço (*buffer*) suficiente para processá-lo.
- Aplica rotinas de segurança pré-configuradas.

Processamento no Roteador (cont.)

- 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 Host Destino

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.