Camada de Inter-Redes

Luiz Arthur

A Internet, também conhecida como rede mundial de computadores tem a popularidade de hoje devido em grande parte a camada de Inter-Redes, que possibilita que diversos hosts estejam ligados entre si independentes de distância, quantidade de hosts e/ou redes, ou até mesmo tecnologia de rede (software/hardware). Pois o **modelo TCP/IP fornece com o protocolo IP uma forma simples e eficiente de endereçamento e roteamento para os pacotes da Internet.**

A forma que o modelo TCP/IP oferece comunicação inter-redes é tão eficiente que os usuários normalmente pensam na **Internet** como **uma única rede virtual** que interconecta todos os hosts através da qual a comunicação é possível; **sua arquitetura subjacente é escondida e irrelevante para os usuários finais.**

De certa forma, **inter-redes é uma abstração de redes físicas** porque, no nível mais inferior, tem a mesma funcionalidade: aceitar e entregar pacotes. Níveis mais altos de software de inter-redes acrescentam a maior parte da funcionalidade rica que os usuários percebem.

Então uma inter-rede TCP/IP provê três conjuntos de serviços, sendo o primeiro o serviço de entrega de pacotes sem conexão, o segundo um serviço de transporte confiável o qual as aplicações depender e o último nível são os serviços de aplicações. A Camada de Inter-rede trata no primeiro serviço, ou seja, entrega de pacotes sem conexão.

Sistema de entrega sem conexão

O serviço de inter-rede mais fundamental consistem em um sistema de entrega de pacotes. Tecnicamente, o serviço é definido como um sistema de entrega de pacotes **não-confiável**, de **melhor esforço** ou **sem conexão**. O serviço é denominado não-confiável por que a entrega não é garantida, assim o pacote pode ser: perdido, duplicado, adiado ou entregue fora de ordem, mas o serviço não detectará essas condições, nem informará ao emissor ou receptor.

O serviço também é determinado sem conexão porque cada pacote é tratado independentemente de todos os outros pacotes, cada **pacote** é tratado de uma forma **individual** e os pacotes podem seguir **caminhos diferentes** para chegar a um mesmo destino.

Finalmente, o serviço é considerado como entrega pelo melhor esforço porque o software de inter-rede faz a melhor tentativa de entregar os pacotes. Ou seja, a Rede não descarta pacotes por capricho; a não confiabilidade surge apenas quando os **recursos** são esgotados ou as redes sub-ajacentes **falham**.

Internet Protocol - IP

O protocolo que define o mecanismo de entrega não-confiável, sem conexão é o Internet Protocolo ou simplesmente IP. Como a versão atual do protocolo é a versão 4, normalmente ele é conhecido como **IPv4**. Mas chamaremos este de simplesmente IP.

Camada de Inter-Redes

Devido a idéia do IP ser sem conexão, não confiável e empregar o melhor esforço para entrega de pacotes, as informações a serem transmitidas via protocolo IP não levam normalmente o nome de pacote mas sim de **datagrama**.

O IP provê três definições importantes:

- Primeiro, o protocolo IP define a **unidade básica** de transferência de dados usada por toda rede TCP/IP. Assim, o protocolo IP especifica o formato exato de todos os dados à medida que eles passam pelas interligação em redes TCP/IP.

- Segundo, o IP realiza a função de encaminhamento (**roteamento**), escolhendo um caminho pelo qual um pacote será enviado ao seu destino, isto é feito normalmente passo a passo (roteador por roteador).

- Terceiro, além a especificação precisa e forma dos formatos de dados e roteamento, o IP inclui um conjunto de **regras** que incorporam a idéia da entrega não confiável. As regras caracterizam como os hosts e roteadores devem processar pacotes, como e quando as mensagens de erro devem ser geradas e as condição sob as quais os pacotes podem ser descartados.

O IP é uma parte tão importante do projeto de interligação redes que às vezes toda tecnologia que usa estes princípios é denominada simplesmente de tecnologia baseada em IP.

Datagrama do IPv4

Um datagrama IPv4, que também podem ser chamado de datagrama IP ou simplesmente datagrama é a unidade básica de transferência de dados na Camada de Inter-redes, tal datagrama é dividido: **cabeçalho**, área que contém informações sobre o pacote; e os dados, que é a **carga útil** do pacote, ou seja, as informações úteis do usuário final.

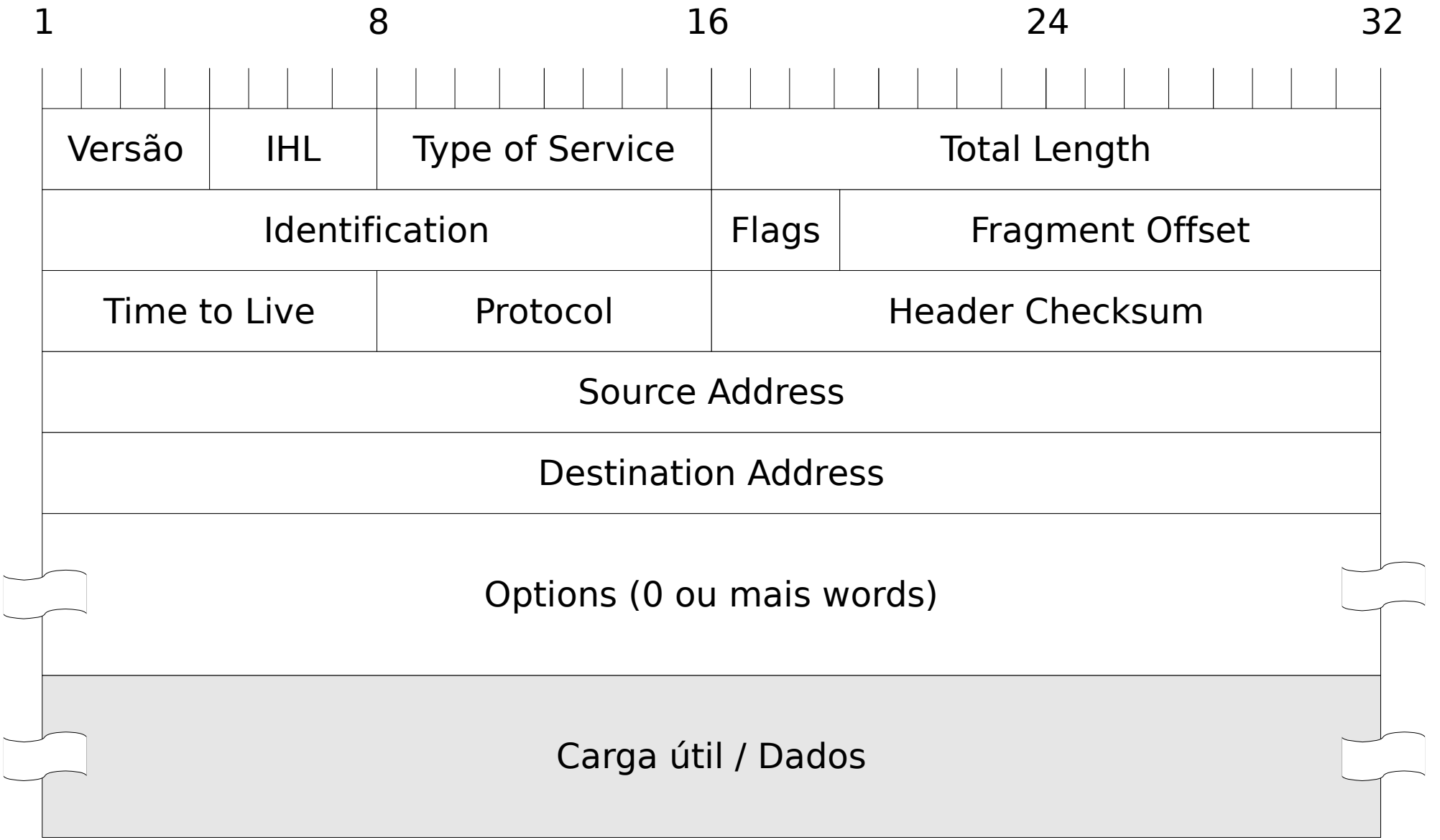
O protocolo IP especifica o formato do cabeçalho incluindo os **endereços IP** da **origem** e de **destino**, bem como outras informações pertinentes a transmissão do datagrama pelas várias redes que este pode vir a alcançar.

O IP não especifica o formato de uma área de dados, ele pode ser utilizado para o transporte de dados arbitrários. Normalmente a área de dados carrega informações sobre a Camada de Transporte, Camada de Aplicação e os dados a úteis do usuário.

O cabeçalho tem uma parte fixa de 20 bytes e uma parte opcional de tamanho variável. Um exemplo ilustrativo sobre cabeçalho e dados é visto na figura a seguir:

Camada de Inter-Redes

Formato do datagrama IP.



Para entender a tecnologia que movimenta a Internet é necessário endente como funciona cada campo do cabeçalho IP, que são:

Version:

Em português Versão, campo de quatro bits que contém a versão do protocolo IP utilizado para criar o datagrama IP. O protocolo IP normalmente utilizado é o IP **versão 4**, mas já é possível trabalhar com o IP **versão 6** ou IPv6. Os hosts transmissores e o receptores tem que primeiramente verificar tal campo e concordar sobre o uso da mesma versão de IP, caso contrário as máquinas podem rejeitar o datagrama. Todo o IP precisa verificar o campo versão antes de processar um datagrama, para garantir que ele combine com o formato que o software espera.

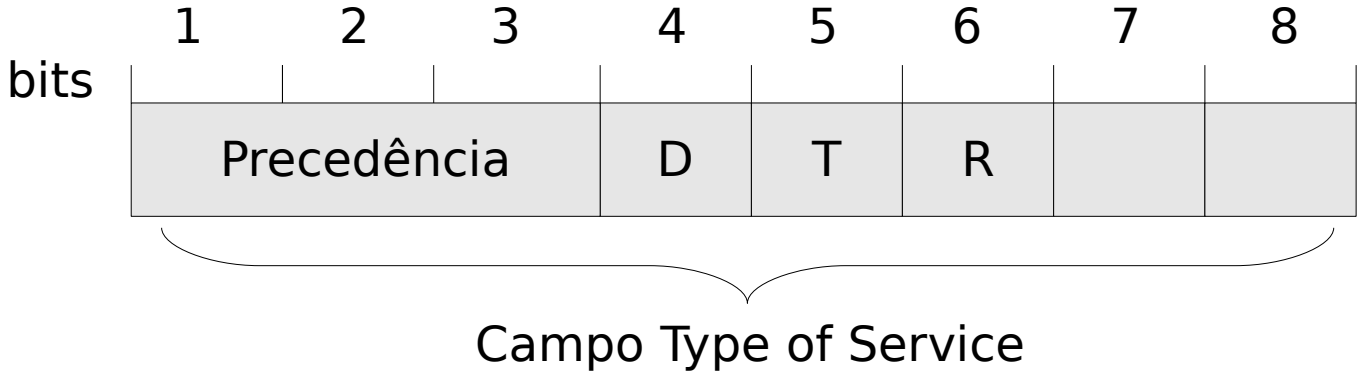
IHL:

Internet Header Length ou Tamanho do Cabeçalho IP, também com tamanho de 4 bits este campo **indica o comprimento do cabeçalho** do datagrama e é medido em palavras de 32 bits. A grande maioria dos campos do cabeçalho IP são fixos, mas o campo Options (Opções) e Padding (Enchimento) podem variar o que torna necessário o IHL. O tamanho mínimo do cabeçalho é de 5 words de 32 bits (20 bytes), sendo esse o valor mais comum, já que normalmente o campo Options e Pad não são usados. Já que o valor máximo desse campo de 4 bits é 15, o que limita o cabeçalho é 60 bytes e o campo de opções a 40 bytes.

Type of Service - TOS:

Em português Tipo de Serviço, este **campo informa a qualidade desejada** para a entrega do datagrama. Tal campo possui 8 bits e especificava inicialmente como o datagrama deveria ser tratado no envio a rede e é fracionado em cinco subcampos: Precedência, Delay, Throughput e Reliability.

Assim se expandirmos o campo Type of Service teremos os seguintes sub-campos:



Então através destes campos é possível controlar **atraso**, **velocidade** e **confiabilidade/segurança**. Já que em se tratando de pacotes de voz digitalizada, a entrega rápida vence a entrega segura, ou para a transferência de arquivos, uma transmissão sem erros é mais importante do que uma transmissão rápida.

Note que existem dois bits no final do campo TOS que não são usados, ou reservados para uso futuro.

Os três primeiros bits de **precedência** informavam a **prioridade** do datagrama, dependendo do tipo de dado que ele carregava as prioridades podem ser diferentes tal como uma conexão VoIP pode ter mais prioridade que um download via FTP.

Exemplo: 1 Indica datagrama do download via FTP; 6 Indica um pacote VoIP e 6 uma informação de controle de rede. Assim, o pacote com maior prioridade será tratado primeiro, neste caso o com maior prioridade são os pacotes de controle da rede, depois os pacotes de VoIP e por último ficam os pacotes de FTP. Os demais pacotes terão prioridade 0 que é o padrão, ou seja, a menor prioridade.

Os bits D, T e R significam, respectivamente **Delay (atraso), Throughput (velocidade) e Reliability (confiabilidade)**. O transmissor poderia ativar cada um desses bits (colocando em 1) quando necessitasse de baixo atraso, alta velocidade e/ou alta confiabilidade, dependendo da situação.

É claro que **não há como garantir essas características em uma rede grande tal como a Internet.**

Então a intensão do campo TOS é das melhores, porém essa idéia não deu certo. Apesar de o sistema de indicar prioridade ser interessante, ter indicadores demais acaba atrapalhando em vez de ajudar.

Para entender melhor por que o **campo TOS não deu certo** analise as seguintes perguntas:

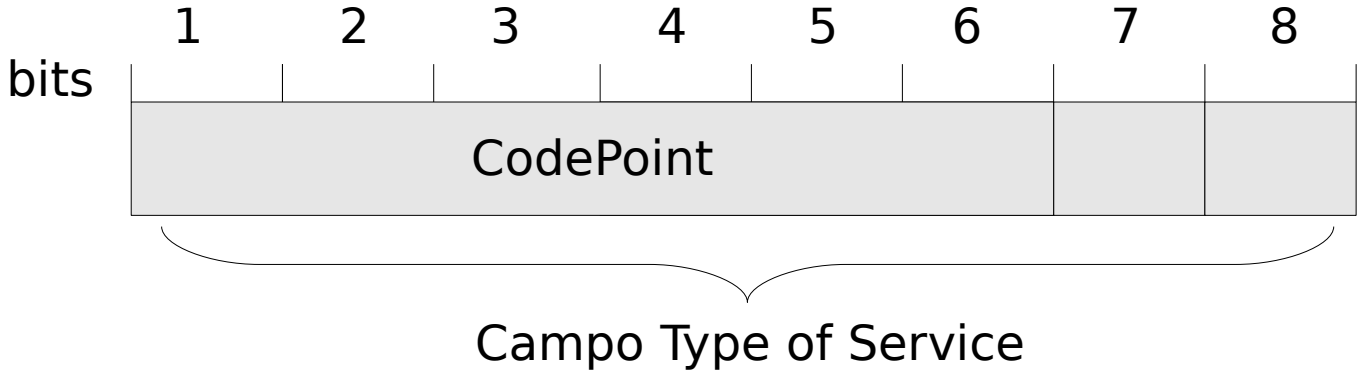
- Primeiro, atraso e velocidade são conceitos muito próximos, o que é uma conexão rápida e o que é uma sem atraso?

- Confiabilidade/segurança não é tão fácil de se medir: que parâmetros serão usados para se medir se uma rota é confiável ou não?

Depois, ter esses indicadores significa aumentar a complexidade dos roteadores, aumentando o seu custo, e é claro que isto inviabiliza o uso do campo TOS.

No final dos anos 90, o IETF redefiniu o significado do campo para acomodar um conjunto de serviços diferenciados (**DiffServ**).

Sob a interpretação dos serviços diferenciados, os seis primeiros bits do campo Type of Service são chamados de ponto de código (codepoint), que ás vezes é abreviado como DSCP, e os dois últimos bits não são usados.



O valor do **codepoint mapeia para uma definição de serviço** subjacente, normalmente por meio de um array de ponteiros. Embora seja possível definir 64 serviços separados, os projetistas sugerem que determinado roteador só precisará de alguns serviços, e vários codepoints mapearão cada serviço.

Desta forma o campo **TOS pode ser usado para dar prioridade de roteamento**, mas os roteadores que fazem isto devem tratar de forma diferente pacotes que usam o novo DiffServ dos pacotes que usam o estilo TOS antigo

Utilizando a interpretação do TOS original ou a interpretação dos serviços diferenciados revisados, é importante observar que o software de encaminhamento precisa escolher entre as tecnologias de rede física subjacentes à disposição e precisa aderir às políticas locais. Assim, especificar um nível de serviço em um datagrama **não garante que os roteadores ao longo do caminho combinarão para honrar a requisição**.

Consideremos a especificação do **TOS** como uma **dica** para o algoritmo de roteamento que o ajuda a escolher entre vários caminhos até um destino com base nas políticas locais e no seu conhecimento das tecnologias de hardware disponíveis nesses caminhos. Uma inter-rede **não garante** fornecer qualquer tipo de serviço em particular.

Assim, o **TOS fica restrito a redes locais** e normalmente é tratado por Firewalls, quando se quer dar mais prioridade a um serviço localmente.

Continuando com os campos do cabeçalho IP depois do Version, IHL e Type of Service vem o Total Length.

Total length:

o Total length ou melhor **Tamanho Total do Datagrama** indica o número total de bytes que compõem o datagrama, **incluindo cabeçalho e área de dados**. Esse campo possui 16 bits desta forma o datagrama só pode ter, no **máximo 65.535 bytes**, ou dois elevado a dezesseis (o número de bits).

Mas os datagramas normalmente utilizam tamanhos bem menores que 65.535 bytes, como por exemplo, 576 bytes. Com as futuras redes de Gigabits serão necessários datagramas maiores, o que é um problema para o IPv4.

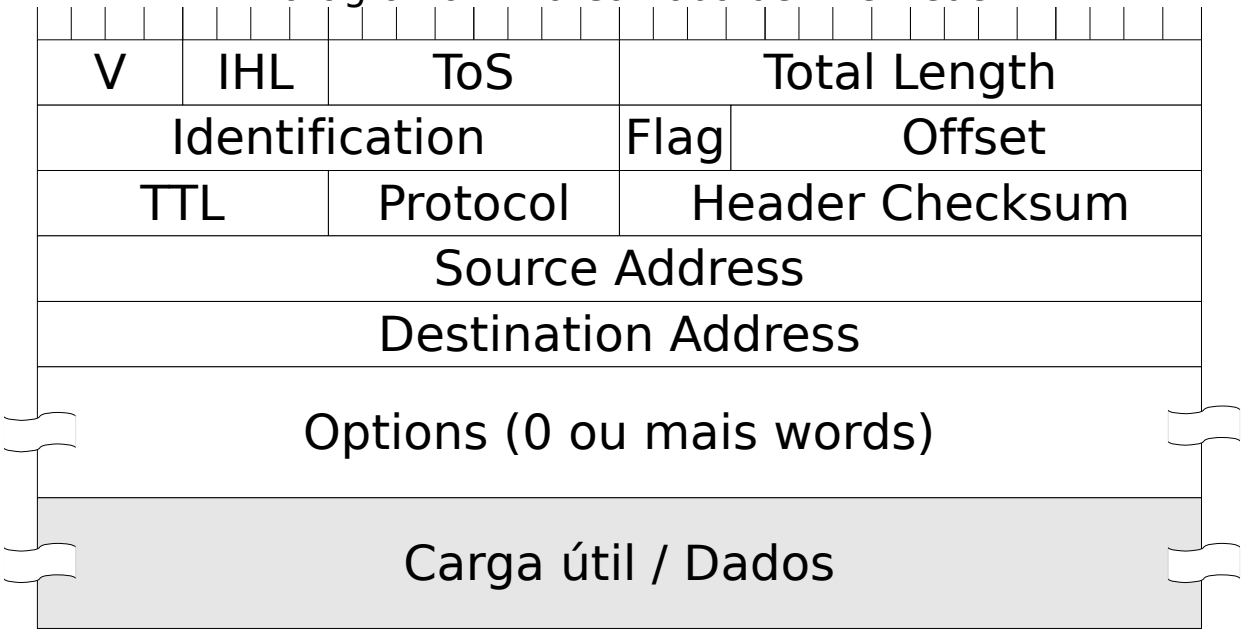
O tamanho da área de dados pode ser calculada subtraindo o comprimento do Total length pelo HLEN.

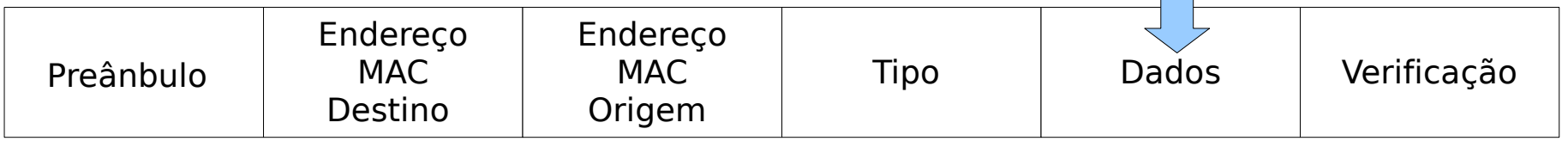
Na prática existem limites mais fundamentais no tamanho do datagrama. Visto que, quando os datagramas passam de uma máquina para outra, eles sempre precisam ser transportados pela rede física subjacente. Para tornar o transporte inter-rede eficiente, **é preciso garantir que cada datagrama trafegue em um quadro físico** (da Camada de Enlace) distinto. Ou seja, queremos que nossa abstração do pacote de rede física seja mapeada diretamente para um pacote real, se possível.

Camada de Inter-Redes

A idéia de transportar um datagrama em um quadro de rede da Camada de Enlace é chamada de **encapsulamento**. Para a rede subjacente, um datagrama é como qualquer outra mensagem enviada de uma máquina para outra. O hardware não reconhece o formato do datagrama, nem entende o endereço de destino IP. Assim, quando uma máquina envia um datagrama IP para outra máquina, o datagrama inteiro tráfega na parte de dados do quadro de rede na Camada de Enlace.

Datagrama IP na Camada de Inter-rede





Quadro de Rede na camada de Enlace

No caso ideal, o datagrama IP inteiro se encaixa em um quadro físico, tornando eficiente a transmissão pela rede física. Para alcançar essa eficiência, os projetistas do IP poderiam ter selecionado um tamanho de datagrama máximo, de modo que um datagrama sempre encaixa em um quadro. Mas isto é quase impossível em uma rede como a Internet.

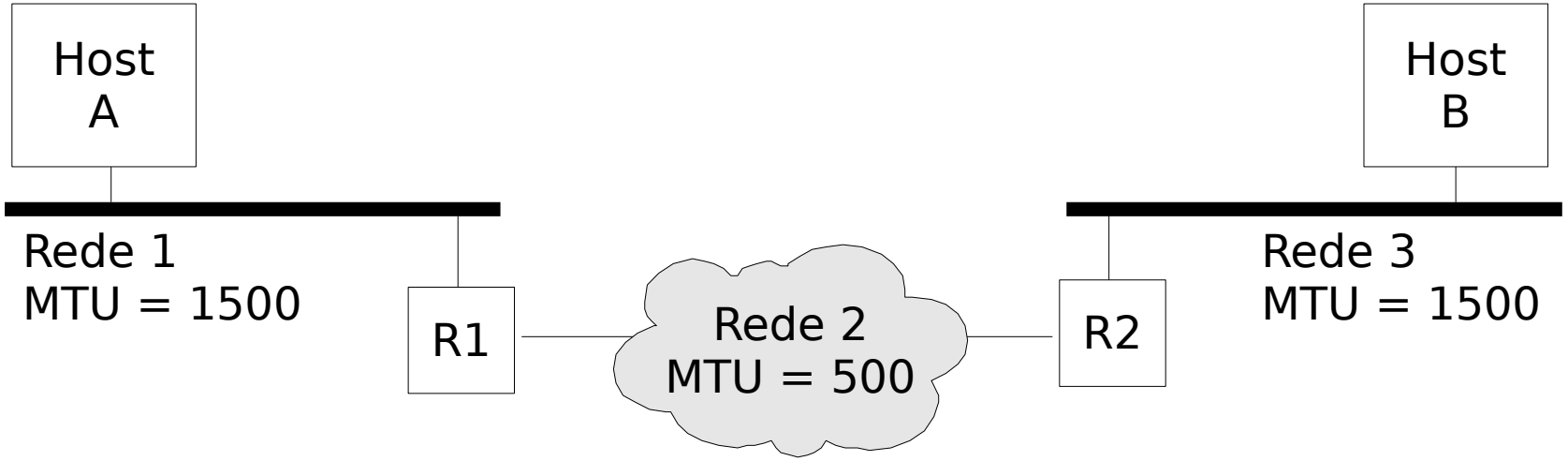
Porém, existe um grande problema quanto ao tamanho do datagrama, pois a principio **o datagrama é formado para caber dentro do quadro de rede de sua rede local**, mas no caminho ao seu destino ele passa por vários caminhos, ou seja, várias redes físicas (que exigem tamanhos de quadros diferentes) e para resolver este problema é que existe a segunda linha (word) do cabeçalho IP, com os campos, Identification, Flags e Fragment Offset, e esta linha cuida da fragmentação dos datagramas.

Fragmentação dos datagramas

Para entender o problema da fragmentação de datagramas IP, precisamos de um fato sobre **o hardware de rede**: cada tecnologia de comutação de pacotes que **impõe um limite superior fixo sobre a quantidade de dados que podem ser transferidos em um quadro físico**.

Por exemplo a Ethernet limita as transferências a 1.500 octetos (bytes) de dados, este limite máximo é chamado de **MTU** (Unidade Máxima de Transferência). O MTU pode ser **maior ou menor** que 1500 em outras redes.

Limitar os datagramas IP para caber a menor MTU possível na Internet torna as transferências **ineficazes** quando os datagramas passam por uma rede que pode transportar quadros de maior tamanho. Porém, **permitir que os datagramas sejam maiores que a MTU mínima de rede na Internet significa que um datagrama nem sempre poderá caber um único quadro de rede**.



Olhando a figura anterior, imagine que o host A tenha que enviar um datagrama para o host B. Observamos que inicialmente o MTU da Rede 1 é de 1500, bem como o da Rede 3. Porém, no meio do caminho está a Rede 2 que é caminho obrigatório para que o pacote passe da Rede 1 para a Rede 3. Neste caso o pacote de 1500 saindo da Rede 1 chegando em R1 deverá ser fragmentado (dividido) em vários datagramas (neste caso pode ser 3 datagramas de 500) de forma que caibam no quadro da Rede 2. Chegando o pacote em seu destino que é o Host B esses fragmentos deveram ser reagrupados para formarem o datagrama original. Noque que a fragmentação ocorreu na origem e não no R2, quando a rede voltou a ter o MTU original.

Remontagem de fragmentos

Um datagrama poderá ser remontado depois de passar por uma rede, mas o mais comum é a remontagem ser feita somente no destino final.

A remontagem de datagramas no destino final pode levar a ineficiência: mesmo que algumas das redes físicas encontradas após o ponto de fragmentação tenham grande capacidade de MTU, somente pequenos fragmentos as atravessam. Também, quaisquer fragmentos forem perdidos, o datagrama não poderá ser remontado.

A máquina receptora inicia um timer de remontagem quando recebe um fragmento inicial. Se o timer expirar antes que todos os fragmentos cheguem o destino descarta os fragmentos já recebidos.

Apesar da desvantagem, a realização da remontagem no destino final funciona bem, pois permite que cada fragmento seja encaminhado de forma independente, e não exige que os roteadores intermediários armazenem ou remontem fragmentos.

Então a **fragmentação de um datagrama significa dividi-lo em várias partes**.

Cada fragmento tem o mesmo formato do datagrama original que duplica a maior parte do cabeçalho, exceto por um bit no campo de Flags que diz respeito ao fragmento), seguido pelo máximo de dados que puderem ser transportados no fragmento enquanto o tamanho total menor que a MTU da rede na qual deverá atravessar.

Vamos olhar em detalhe agora os campos de controle de fragmentação na segunda linha do cabeçalho do datagrama IP:

Identification:

Identificação usado para identificar o datagrama. Quando o transmissor cria e envia um datagrama pela rede, é atribuído a ele um **número de identificação**. **Todos os fragmentos de um datagrama contêm o mesmo valor de Identification**. Esse número será usado para identificar o datagrama caso ele seja fragmentado no caminho até o destino. Sua finalidade principal é permitir que o destino saiba quais datagramas estão chegando e a que datagramas pertencem. Este campo contém 16 bits.

Em seguida, há um bit não utilizado ou para uso futuro.

Flags:

Esse campo é usado para **controlar a fragmentação** de datagramas dentro deste existem dois bits sendo eles chamados de:

- **DF** (Don’t Fragment) ou **não fragmente**: Trata-se de uma ordem para os roteadores não fragmentarem o datagrama porque a máquina de destino é incapaz de juntar os pedaços novamente. Isso ocorre em máquinas que serão inicializadas por boot remoto. É claro que quando este bit esta ativo (marcado com 1) este não pode ser fragmentado e isto implica é claro que o pacote não pode passar por redes com MTU menor do datagrama original.

- **MF** (More Fragments- **mais fragmentos**): Todos os fragmentos, exceto o último, têm esse bit ativo (marcado como 1), que é necessário para que se saiba quando todos os fragmentos de um datagrama chegaram.

Fragment offset

Offset do Fragmento: **Informa a que ponto do datagrama atual o fragmento pertence**. Especifica o **deslocamento**, **no datagrama original**, dos dados que estão sendo transportados no fragmento, medidos em unidades de oito octetos, iniciando em deslocamento zero. Assim é possível remontar os fragmentos no destino.

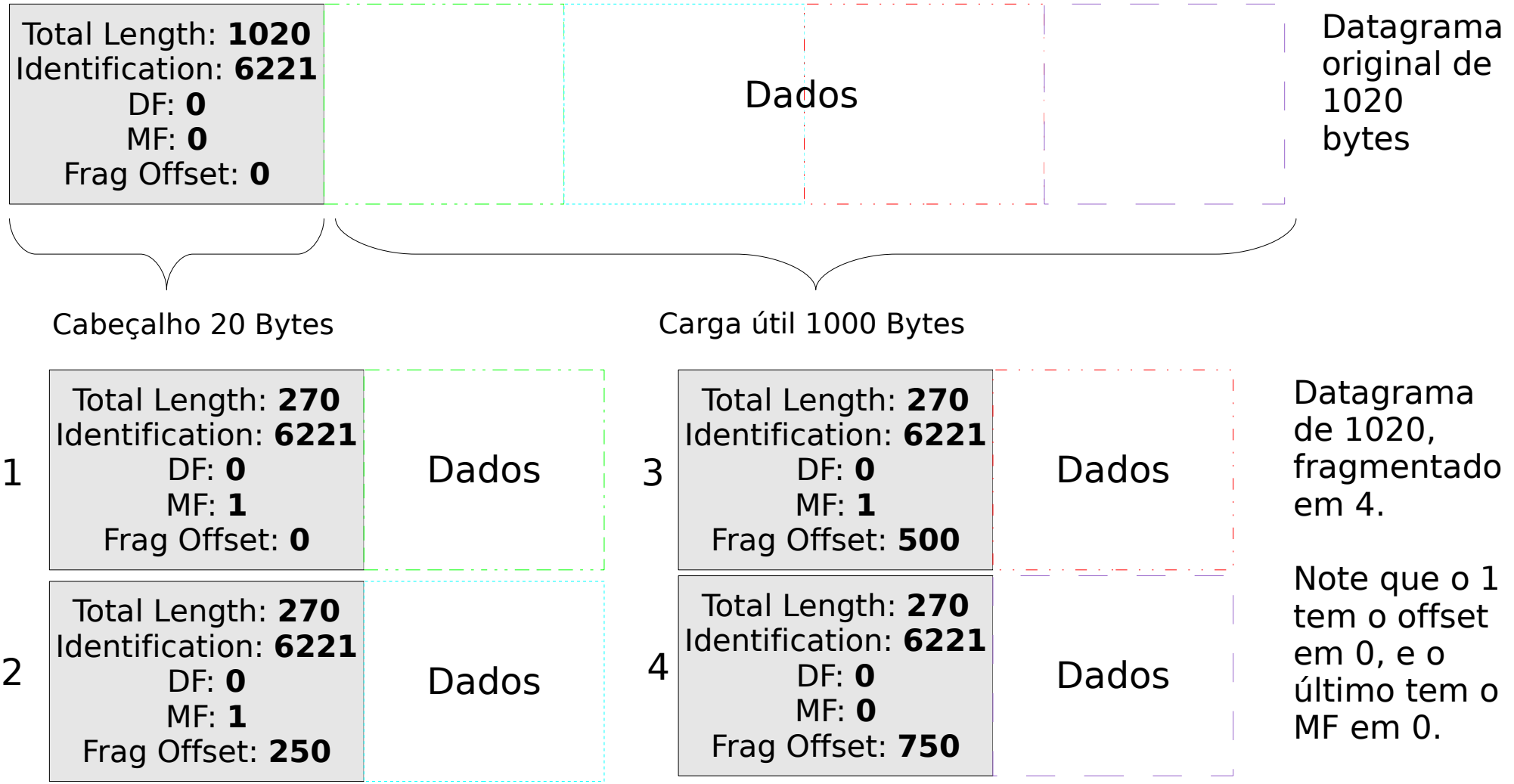
Na prática os campos que controlam a **fragmentação** e a **remontagem** de datagramas **funcionam da seguinte maneira**:

O campo **Identification** contém um número inteiro exclusivo que **identifica** o datagrama, e este campo é copiado exatamente igual em todos os fragmentos caso seja necessário fragmentar um datagrama então sua finalidade principal é permitir que o destino saiba quais **fragmentos** que chegam pertencem a quais datagramas, isto é feito juntamente com os endereços IP de origem.

Cada fragmento tem exatamente o mesmo formato de um datagrama completo. Para um fragmento, o campo **Fragment Offset** específica o offset (equiparar) que equipara a qual parte do datagrama faz parte o fragmento. **Para remontar o datagrama**, o destino precisa obter todos os fragmentos a partir do fragmento que possui o offset 0 até aquele com offset mais alto. Os fragmentos não necessariamente chegam na ordem.

A **flag DF específica que o datagrama não pode ser fragmentado**. Já a flag **MF específica se o fragmento contém dados do meio do datagrama original ou do final**. Para ver como o bit MF é importante, considere o IP no destino final tentando remontar um datagrama. Ele receberá fragmentos (possivelmente fora de ordem) e precisa saber quando recebeu todos os fragmentos de um datagrama. Quando um fragmento chega, o campo Total Length no cabeçalho refere-se ao tamanho total do fragmento e não ao tamanho total do datagrama original, de modo que o destino não pode usar o campo Total Length para dizer se ele coletou todos os fragmentos.

O bit MF soluciona este problema com facilidade: quando o destino recebe um fragmento com o bit MF desmarcado (em zero), ele sabe que esse fragmento transporta dados da ponta do datagrama original. Examinando o Fragment Offset e o Total Length de todos os fragmentos que chegaram, o destino final pode saber se os fragmentos disponíveis contêm todas as partes necessárias para remontar o datagrama original.



Seguindo os estudos dos campos do datagrama IP, depois dos campos Version, IHL, ToS, Total Length, Identification, Flags e Fragment Offset vem o campo Time to Live.

Time to Live:

Time to Live ou simplesmente TTL que em português significa **Tempo de Vida**, indica o tempo máximo de vida do datagrama que esteja vagando por um sistema inter-redes (passando por várias rotas). A principio o campo TTL **específica quanto tempo**, em segundos, **o datagrama** tem permissão para **permanece**r em um sistema inter-**redes**. É um contador usado par limitar a vida útil do pacote. Esse campo conta a principio o tempo em segundos, permitindo uma vida útil máxima de 255 segundos.

A idéia é simples e importante: sempre que um host injetar um datagrama em um sistema inter-redes (composto por vários roteadores), o host define um tempo máximo em que o datagrama deve "sobreviver" vagando pela inter-rede. Os **roteadores** e os hosts que precessam datagramas precisam **decrementar** o campo TTL à medida que o tempo passa e remover o datagrama da inter-rede quando seu tempo expirar (chegar em zero).

A idéia de controlar tempo é muito complexa, pois imagine como manter os relógios de todos os roteadores da Internet sincronizados! E qual é o horário que deveria ser utilizado (Brazil/São Paulo)? Por isso essa **idéia de segundos não é mais usada**.

Assim, na prática o campo **TTL atual é usado para contar "limite de salto" de rota**, em vez de uma estimativa de atraso. Cada roteador no caminho (ou seja, cada salto) decrementa o valor em 1.

Então cada vez que o datagrama passa por um roteador (por exemplo) esse número é decrementado do TTL. **Quando o TTL chega a zero, o datagrama é descartado, não atingindo o destino**.

No receptor, a **Camada de Transporte** caso esteja-se utilizando um protocolo orientado a conexão, este irá perceber que está faltando um datagrama e **pedirá uma retransmissão** do datagrama que está faltando, lembre-se o IP não oferece este tipo de recurso.

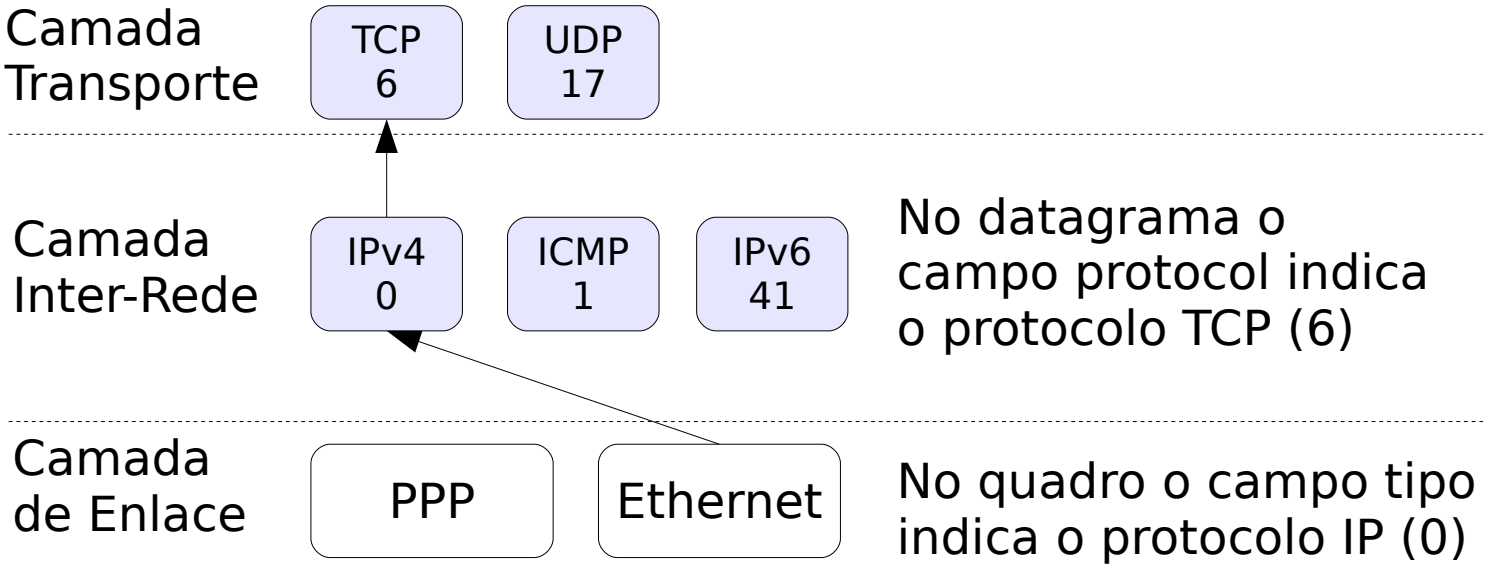
O objetivo do TTL é eliminar os datagramas que demorarem tempo demais para chegarem ao destino, o que pode ocorrer caso a rota escolhida seja muito longa ou mesmo errada, caso exista um roteador mal-configurado no meio do caminho. **Este recurso não existe** na Camada de Enlace **no protocolo Ethernet**, por isto acontece o problema de loops no switches, o TTL não permite que o mesmo ocorra com o IP na Camada de Inter-redes.

Assim, elimina-se datagramas que poderiam ficar vagando eternamente pela Internet à procura de seu destino caso eles encontrem problemas de rota, o que congestionaria a rede.

Protocol:

Quando um datagrama chegar por completo no destino final, a Camada de Inter-rede precisa saber **o que fazer com a informação transportada na área de dados** (normalmente cabeçalho de outros protocolos e os dados do usuário).

O campo Protocol informa a ao software IP o processo de transporte que deverá ser aplicado ao datagrama. Na verdade o valor do campo Protocol especifica o formato da área de dados. Esse campo **indica o protocolo de que pediu o envio do datagrama**, através de um código numérico. Por exemplo, o número 6 indica o TCP, o número 17 indica o UDP, o número 1 indica o ICMP e assim por diante. Desta maneira, no dispositivo receptor, a camada IP sabe para qual protocolo superior ela deverá entregar os dados presentes dentro do datagrama. Exemplo:



Header Checksum:

O protocolo IP adiciona um campo de checksum (soma de verificação) para os valores presentes nos cabeçalhos, isto **permite identificar se existem erros no cabeçalho do datagrama IP**.

Note que esse campo calcula o checksum **somente do cabeçalho** e, portanto não usa o campo de dados no cálculo, assim se a parte de dados estiver errada não é o IP quem vai descobrir isto.

A **vantagem** de se usar somente o cabeçalho é que a conta de **verificação** de erros fica menor e **mais rápida** de ser feita (já que o cabeçalho tem tipicamente 20 bytes). Os roteadores analisam esse campo e refazem o checksum para saber se o cabeçalho está ou não corrompido.

O algoritmo de checksum do protocolo IP tem como objetivo somar todas as meias palavras de 16 bits à medida que elas chegam, utilizando a aritmética de complemento de 1 e depois calculando o complemento de 1 do resultado. Para os propósitos desse algoritmo, supomos que o campo Header checksum seja zero no momento da chegada. O uso do algoritmo é mais eficaz do que uma soma normal.

Observe que Header **checksum deve ser recontado a cada hop**, porque pelo menos um campo sempre se altera (o campo Time to live).

Source address:

Como o próprio nome diz **endereço IP de origem**, neste campo há o endereço IP de onde está partindo o datagrama. Possui um tamanho igual a 32 bits. Este campo é utilizado normalmente para a resposta do host de destino, pois a grande maioria das conexões consistem de pacotes indo e voltando de um host para outro. Além do que existem algumas operações de redes que são feitas baseadas no endereço de origem, tal como roteamento pela origem, filtragem de pacotes por roteadores e firewalls, etc.

Exemplo de endereços IP de origem:

00000001.00000010.00000011.00000100 = 32 bits
1.2.3.4 = 32 bits, porém em decimal

Destination address:

Endereço IP de destino, ou seja, endereço IP de 32 bits (assim como o Source address) que representa um host de destino, tal endereço será utilizado pelo software IP primeiramente para identificar a rede a ser entregue, se for passado por roteadores estes irão submeter este campo as suas tabelas de roteamento até chegar ao destino final, que também será identificado por este campo. Normalmente o roteamento é feito pelo endereço de destino e não pelo endereço de origem.

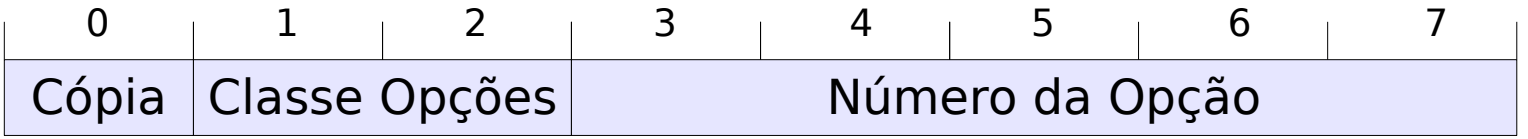
É importante ter ciência que **a máscara de sub-rede normalmente não é enviada com o datagrama IP,** não existe campo disponível para enviá-la no datagrama IP. Então as máscaras de redes são locais, ou seja, cada máquina aplica o IP de origem ou destino as suas máscaras locais.

Options:

O campo **Options é opcional**, assim, pode ou não fazer parte do cabeçalho IP de um dado datagrama. Esse campo foi projetado para permitir que versões posteriores do protocolo incluam informações inexistentes no projeto original, possibilitando a experimentação de novas idéias e evitando a alocação de bits de cabeçalho para informações raramente necessárias.

As opções **são incluídas principalmente par testes ou depuração da rede**. O tamanho do campo Options vária, dependendo de quais opções são selecionadas.

Cada opção consiste em um único código de opção de octeto, que pode ser seguido por um único tamanho de octeto e um conjunto de octetos de dados para essa opção. O octeto de **código de opção** é dividido em três campo:



O campo ***Cópia*** controla como os roteadores tratam das opções durante a **fragmentação**. Quando o bit de Cópia é definido como **1**, ele específica que a opção **deve ser copiada para todos os fragmentos**. Quando definido como **0**, o bit de Cópia significa que **a opção só deve ser copiada para o primeiro fragmento** e não para todos fragmentos, isto é importante.

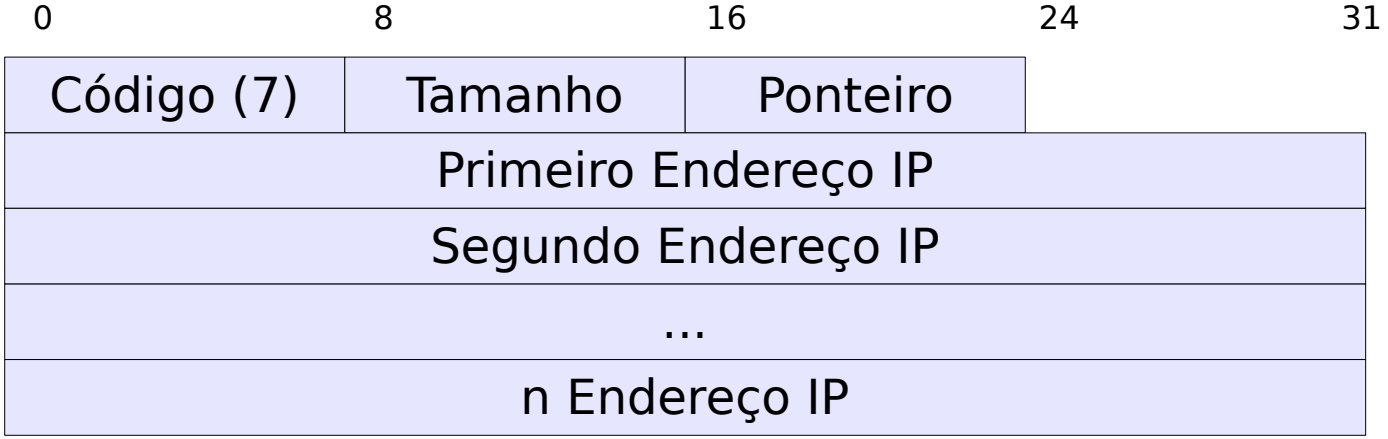
O campo ***Classe Opção*** e ***Número Opção*** especificam a classe geral da opção e uma opção específica nessa classe, é como uma **código e sub-código**. A Classe Opção é dividido em: 0 – Datagrama ou controle de rede; 1 – Reservado para uso futuro; 2 – Depuração e medição; 3 – Reservado para uso futuro.

Classe de Opção	Número de Opções	Tamanho	Descrição
0	0	-	Fim da lista de opções. Usado se as opções não terminarem no final do cabeçalho (ver explicação no campo de peenchimento de cabeçalho).
0	1	-	Nenhuma operação. Usado para alinhar octetos em uma lista de opções.
0	2	11	Seguranças e restrições de tratamento (para aplicações militares).
0	3	var	Rota de origem solta. Usado para requisitar roteamento que inclui os roteadores especificados.
0	7	var	Registrar rota. Usado para rastrear uma rota.
0	8	4	Identificador de fluxo. Usado para transportar um identificador de fluxo SATNET (obsoleto).
0	9	var	Rota de origem estrita. Usado para especificar um caminho exato pela inter-rede.
0	11	4	Sonda MTU. Usado para a descoberta de MTU do caminho.
0	12	4	Réplica de MTU. Usado para descoberta de MTU do caminho.
0	20	4	Alerta de roteador. O roteador deve examinar esse datagrama mesmo que não seja o destinatário.
2	4	var	Estampa de tempo de inter-rede. Usado para registrar estampas de tempo (time stamp) ao longo da rota
2	18	var	Traceroute. Usado pelo programa traceroute ou tracert para encontrar roteadores ao longo de um caminho de rede.

As Opções de rota e estampa de tempo são as mais interessantes, pois oferecem uma maneira de monitorar ou controlar como os roteadores encaminham datagramas.

Campo Option: Opção de registro de rota

A opção registrar rota permite que a origem crie uma lista vazia de endereços IP e arrume para que **cada roteador** que trata o datagrama **inclua seu endereço IP nessa lista**.



O campo ***Código*** contém a classe de opção e o número de opção (0 e 7 para registrar rota). O ***Tamanho*** especifica o **tamanho total da opção**, incluindo os três primeiros octetos. O campo ***Ponteiro*** especifica o deslocamento dentro da opção do próximo slot disponível.

Sempre que uma máquina trata de um datagrama que tem a opção registro de rota marcada, a **máquina acrescenta seu endereço à lista** de registrar rota (espaço suficiente precisa ser alocado na opção pela origem a fim de manter todas as entradas que serão necessárias).

Para acrescentar-se à lista, uma máquina primeiro compara os campos de ponteiro e tamanho.

Se o ponteiro for maior que o tamanho, a lista ficará cheia, de modo que a máquina encaminha o datagrama sem inserir sua entrada.

Se a lista estiver cheia, a máquina insere seu endereço IP de 4 octetos no posição específica pelo Ponteiro e incrementa o Ponteiro em quatro.

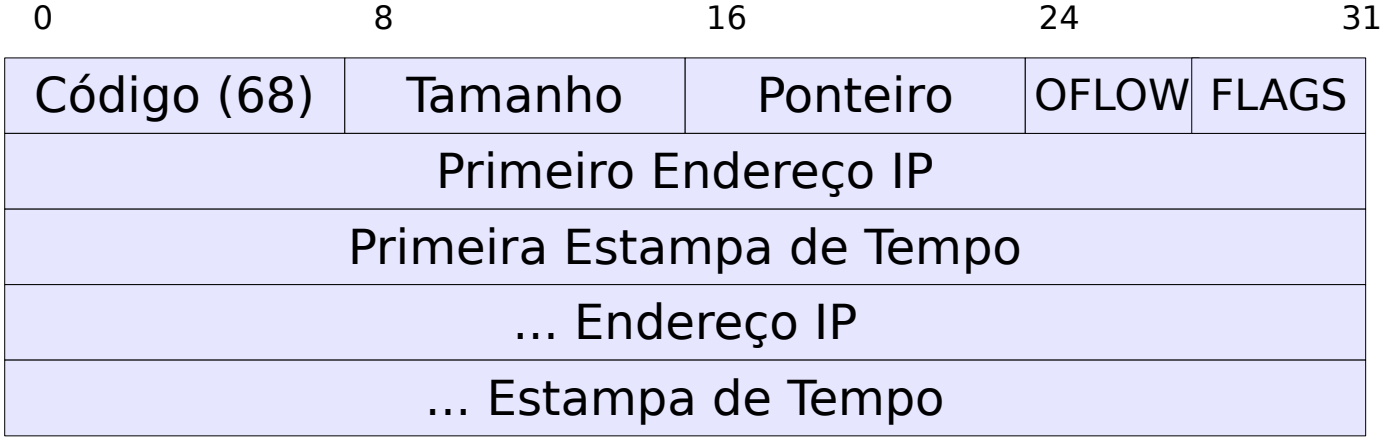
Esta opção ajuda a entender como esta trabalhando o roteamento de datagramas na Internet, por exemplo, e corrigir possíveis falhas.

Campo Option: Opções de rota de origem

A noção por trás do roteamento é que ele **oferece um modo de o emissor ditar um caminho pela Internet**. Por exemplo, para testar a vazão por uma rede, ou em caso de falhas em alguns roteadores, esta opção pode ser usada para ditar a rota do datagrama para corrigir o erro. Os campos são quase o mesmo da opção registro de rotas só que o código é 137 e são passados os Endereços de Saltos de rotas.

Campo Option: Opção estampa de tempo

Está opção funciona de forma similar ao registro de rota, mas **adiciona a idéia de que cada roteador irá adicional além de seu IP a estampa de tempo** (data e hora em milisegundos no horário universal) em que o datagrama passou pelo roteador.



Os campos *Código, Tamanho* e *Ponteiro* funcionam igual ao registro de rota.

O campo ***OFLOW*** contém um contador inteiro de roteadores que não poderiam fornecer uma estampa de tempo por que a opção era muito pequena.

O valor no campo ***FLAGS*** controla o formato exato da opção e diz como os roteadores devem fornecer estampas de tempo.

É claro que as estampas de tempo nem sempre são consistentes já que vem de máquinas independentes.

Carga útil / Dados

Depois de analisar todos os campos do datagrama IP chegamos ao último, que é a Carga útil, este **leva os dados dos outros protocolos e os dados úteis dos usuários**, tudo isto está encapsulado dentro do datagrama IP. O software IP a principio não tem conhecimento sobre o conteúdo ou formato da área de dados, exceto pelo fato do campo Protocol dizer a quem deve ser entregue os dados.

Para finalizar o datagrama IP é muito importante para a transmissão de informações em grandes redes, tal como a Internet.

O protocolo IP fornece:

- Uma forma de endereçar hosts;
- Rotear informações de uma origem para um destino;
- Permitir que os datagramas passem por redes de diferentes tecnologias (com MTUs diferentes).
- O protocolo IP é a base de toda a tecnologia TCP/IP.

O protocolo IP é a base de toda a tecnologia TCP/IP e principalmente da rede mundial de computadores, é provavelmente este procolo que vai ajudar a unificar diferentes tipos de redes, tal como rede telefônica, rede telefônica celular, rede de rádio e televisão. É bem provável que em um futuro não muito distante todos estes dados trafeguem por um único meio utilizando a tecnologia IP.

ARP (Address Resolution Protocol)

Bem pelo que já estudamos até agora sabemos como são transmitidos os dados pela Camada de Inter-redes através do datagrama IP e como os dados são transmitidos pelos quadros de rede na Camada de Enlace.

Sabemos também que a princípio **o host de origem comunica-se com o host de destino através do endereço IP**. Tal como, para que um micro acesse um servidor HTTP (Hypertext Transfer Protocol) o host de origem digita em seu browser o seguinte endereço http://204.152.191.37 de destino, por exemplo, ou usando um nome (exemplo, http://www.kernel.org) que vai ser traduzido por um endereço IP por um servidor DNS (Domain Name System).

Entretanto já devemos saber também que para os hosts se comunicarem é **obrigatório o uso de um endereço de hardware**, no quadro de rede a ser usado na Camada de Enlace. Que a princípio não é conhecido pelo host de origem!

Então como é resolvido este problema?

E o pior, toda a comunicação entre a Camada de Enlace só é feita em nível local, ou seja, como é possível que várias redes na Camada de Inter-redes troquem informações estando em Camadas de Enlace diferentes?

Embora na Internet cada máquina tenha um (ou mais) endereços IPs, na verdade eles não podem ser usados para o envio de pacotes, pois o hardware da **Camada de Enlace de dados não entende os endereços IPs**. E para resolver esse problema que foi inventado o protocolo ARP.

O protocolo ARP é responsável por fazer a conversão dentre os endereços IPs e os endereços físicos (MAC).

Pois, duas máquinas de determinada rede só podem comunicar se souberem o endereço de rede física uma da outra. Tal **mapeamento de endereço deve ser executado a cada passo** (em cada rede local), ao longo do trajeto, da origem até o destino final.

Então temos **dois cenários** de redes distintos:

- **Primeiro**, na última etapa da entrega de um pacote, esse deve ser enviado através de uma **rede física** até seu destino final. O computador que o está enviando deve mapear o endereço IP, do destino final até o endereço físico de destino.

- **Segundo**, em algum ponto do trajeto, da origem até o destino, que deve ser diferente da etapa final, **o pacote deve ser enviado a um roteador intermediário**. Assim o transmissor deve mapear o endereço IP do roteador intermediário até um endereço físico.

Contudo a **conversão de endereços é difícil** em algumas redes, por exemplo:

Na tecnologia Ethernet, cada interface de rede recebe um endereço físico de 48 bits quando o dispositivo é manufaturado. E como conseqüência, quando o hardware falha, e requer que a interface seja substituída, o endereço físico da maquina muda. Além disso, como o **endereço da Ethernet tem 48 bits** de comprimento, **não** há esperança de que **seja codificado em um endereço IP de 32 bits**.

Projetistas de protocolo do **TCP/IP** encontraram uma **solução** criativa o problema de conversão de endereço para **redes** como a Ethernet com capacidade de **difusão**. Essa solução **permite que novas máquinas sejam adicionadas à rede sem código de recompilação e não precisem da manutenção de um banco de dados centralizados.**

Para dispensar uma tabela de mapeamento, os projetistas optaram por usar um protocolo de ordem inferior para **ligar endereços de maneira dinâmica**. Denominado **Address Resolution Protocol (ARP)**, o protocolo fornece um mecanismo que é, ao mesmo tempo, razoavelmente eficiente e fácil de manter.

“O ARP permite que um host encontre o endereço físico de um host de destino na mesma rede física, apresentando somente o endereço IP de destino.”

ARP é um protocolo de baixo nível que esconde o endereçamento físico da rede subjacente, permitindo que alguém atribua um endereço IP qualquer para cada máquina.

Tecnologias que admitem mapeamento direto não precisam de ARP. Isto é possível quando a tecnologia de hardware permite que o administrador de rede atribua endereços de hardware manualmente. Assim, o administrador pode fazer uma relação direta (geralmente com o mesmo endereço) entre o endereço de hardware e o endereço IP. Porém, tecnologias como a Ethernet não permitem isto e o ARP é obrigatório. Lembre-se de que o padrão Ethernet está presente em aproximadamente 90% das redes locais no mundo.

Como funciona o ARP

O ARP funciona mandando primeiramente uma mensagem de broadcast para a rede perguntando, a todas as máquinas, qual responde pelo endereço IP para o qual pretende-se transmitir um pacote.

Então, **a máquina que corresponde a tal endereço responde identificando-se e informando o seu endereço físico** para que a transmissão de dados entre essas máquinas possa ser estabelecida.

Mas porque o host de origem simplesmente não envia por difusão o pacote a ser enviado?

A **difusão é muito cara** para ser usada cada vez que uma máquina precise transmitir um pacote para outra máquina. Isto por exemplo, o uso de difusão iria acabar com os benefícios trazidos pelo uso de switching.

Então para reduzir custos com comunicação, os computadores que usam **ARP mantém um cache com mapeamento ente endereços físicos e endereços IP**, recentemente adquirido, e assim não precisam frequentemente usar mensagens ARP em broadcast na rede.

Desta forma sempre quando a máquina for transmitir um pacote ele sempre procura em seu cache uma vinculação de endereços antes de enviar uma solicitação ARP.

Aprimoramentos de ARP

Primeiro, observe que se o host A está prestes a usar ARP porque precisa enviar a B, há grande probabilidade de que o host B vá precisar enviar ao A também. Para prever as necessidades de B e evitar tráfego extra de rede, A **inclui sua vinculação entre endereço físico e endereço IP quando envia uma solicitação** para B. B retira a vinculação de A da solicitação, armazena a vinculação em seu cache ARP e depois envia uma resposta para A.

Segundo, observe que, como A difunde sua solicitação inicial, **todas as máquinas da rede a recebem e podem retirar e armazenar em seu cache** a vinculação entre endereço físico e endereço IP de A. Como o ARP envia mensagens em broadcast (para todos os micros) se ele ficar enviando essas mensagens a todo momento iria congestionar a rede. Para evitar este problema o dispositivo transmissor armazena os endereços IPs recentemente acessados e seus endereços MAC correspondentes em uma tabela na memória, assim ele não precisará fazer um broadcast se precisar acessar um endereço IP já conhecido.

O ARP divide-se em duas partes:

A **primeira mapeia endereços IP para um endereço físico** quando envia um pacote; e a **segunda parte responde às solicitações de outras máquinas**.

A **difusão** de uma solicitação **ARP** para encontrar um mapeamento de endereço **pode ser complicada**. Pois, **máquinas** de **destino podem estar desativadas ou** simplesmente muito **ocupadas** para aceitar a solicitação, como a Ethernet é um sistema de entrega sem garantia, **a solicitação inicial de difusão ARP pode se perder**.

Enquanto isso, o host deve armazenar o pacote original de saída, de modo que o endereço possa ser enviado tão logo tenha sido convertido (se a demora for grande, o host pode optar por livrar-se dos pacotes que vão sair).

Finalmente **considere o caso em que uma máquina A obteve uma associação para a máquina B, mas o hardware B falha e é substituído**. Apesar do endereço físico de B ter mudado, a vinculação no cache de A não mudou. E como A usa um endereço de hardware inexistente, torna a recepção impossível. Esse caso mostra por que **é importante ter tabela de vinculação como um cache no qual removem-se entradas após um período fixo de tempo**. É claro que o **timer** para uma entrada na cache deve ser restaurado sempre que uma difusão ARP chegar contendo a associação (mas não é restaurado quando a entrada é usada para enviar um pacote).

Formato de protocolo ARP

Os dados dos pacotes **ARP não possuem um cabeçalho fixo**, isso é necessário para tornar o ARP útil para uma variedade de tecnologias de redes, a extensão dos campos que contem endereços depende do tipo de rede. No entanto, para tornar possível interpretar uma mensagem ARP arbitrária, o cabeçalho inclui campos fixos perto do início que especificam as extensões dos endereços encontrados em campos sucessivos.

Campos do protocolo ARP:

•Tipo de Hardware: especifica um **tipo de interface de hardware** para o qual o transmissor pede uma resposta, ele contém um valor para Ethernet.

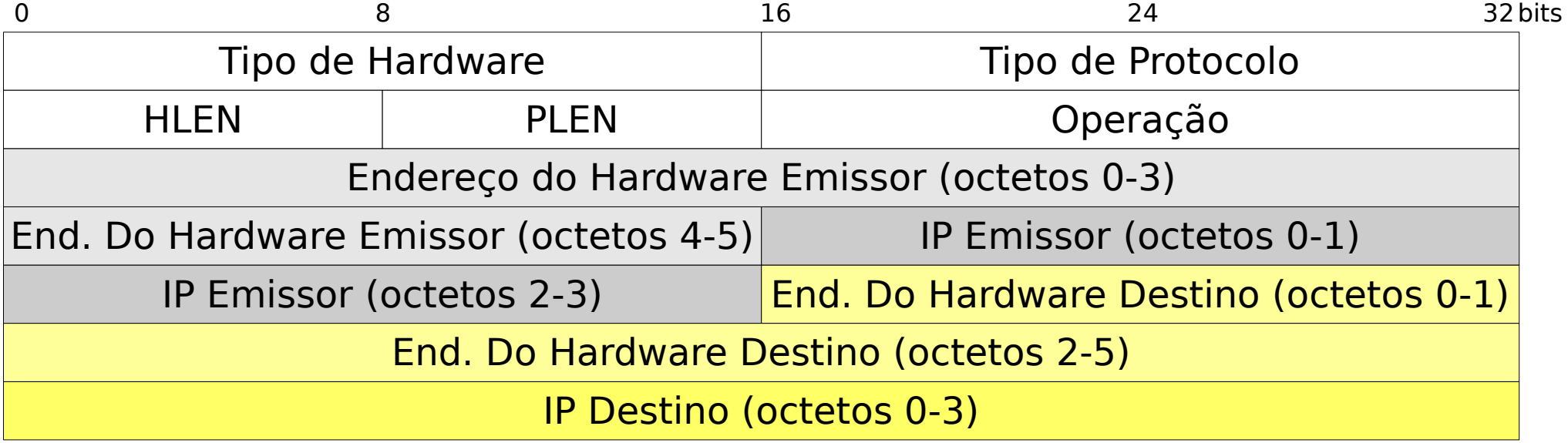
•Tipo Protocolo: especifica o **tipo de endereço de protocolo de alto nível** (contém 0800) para endereços IP.

HLEN e PLEN: permitem que ARP seja usado com redes arbitrarias porque **especificam a extensão do endereço de hardware e a extensão do endereço de protocolo de alto nível**.

Operação: **especifica** uma **solicitação** ARP (1), **resposta** ARP (2), solicitação RARP (3) ou resposta RARP (4).

Endereço Hardware Emissor e Endereço IP Emissor: Campos nos qual o **transmissor fornece seus endereços de hardware e endereço IP**. Ao fazer uma solicitação, o transmissor fornece também o endereço IP de destino (ARP), ou o endereço de hardware de destino (RARP), usando campos.

Endereço de Hardware Destino e Endereço IP de Destino: Antes que a máquina de destino responda, ela preenche os endereços vazios, troca os pares de destino e de envio e troca a operação para uma resposta.



RARP (Reverse Address Resolution Protocol)

O protocolo RARP permite que uma máquina descubra um endereço IP através de um endereço MAC, ou seja, ele faz o serviço inverso do ARP.

Uma máquina normalmente armazena seu endereço IP em alguma memória não volátil. E assim que o host for ligado ele irá nesta memória buscar o seu endereço IP. Entretanto existem algumas máquinas que não fazem uso de memórias para este fim, devido a qualquer motivo (máquinas de **boot remoto** - diskless), ou seja, estas máquinas não têm como saber o seu endereço IP, portanto não tem como iniciarem uma comunicação de rede usando o protocolo TCP/IP.

São nestes casos que entram o protocolo RARP funcionando como servidor, já que haverá uma máquina que armazenará uma tabela contendo os endereços MAC das placas de rede presentes na rede e os seus respectivos endereços.

Uma **desvantagem do RARP** é que ele utiliza um endereço de destino composto somente por valores 1 (difusão limitada) para chegar ao servidor RARP. Entretanto, essas difusões **não são encaminhadas pelos roteadores**; portanto, é necessário um servidor RARP em cada rede. Para resolver esse problema, foi criado um protocolo de inicialização alternativo, chamado **BOOTP**.

Diferente do RARP, **o BOOTP utiliza mensagens UDP** (na Camada de Transporte), **que são encaminhadas pelos roteadores**. O BOOTP também fornece informações adicionais a uma estação de trabalho sem disco, inclusive o endereço IP do servidor de arquivos que contém a imagem de memória, o endereço IP do roteador padrão e a máscara de sub-rede a ser usada.

Um problema sério com o BOOTP é que ele exige configuração manual de tabelas que mapeiam endereços IP para endereços Ethernet. Quando um novo host é adicionado a uma LAN, ele não pode usar o BOOTP enquanto um administrador não tiver atribuído a ele um endereço IP e inserido manualmente seu par (endereço Ethernet, endereço IP) nas tabelas de configuração do BOOTP.

Para eliminar essa etapa propensa a erros, o BOOTP foi ampliado e recebeu um novo nome, DHCP (Dynamic Host Configuration Protocol). O DHCP permite a atribuição manual e a atribuição automática de endereços IP. Na maioria dos sistemas, o DHCP substituiu em grande parte o RARP e o BOOTP.

O DHCP será estudado em detalhe na Camada de Aplicação, já que é lá que este reside apesar de prestar serviços na Camada de Inter-redes.

ICMP (Internet Control Message Protocol)

No **sistema sem conexão** utilizado pelo protocolo **IP**, cada roteador opera de forma autônoma, encaminhando e entregando **datagramas que chegam sem coordenar com o emissor original**.

O sistema funciona bem se todos os hosts e roteadores operarem corretamente e combinarem a respeito das rotas. Infelizmente, **nenhum sistema de comunicação grande funciona corretamente o tempo todo**.

Além de falhas das linhas de comunicação e processadores, o IP deixa de entregar datagramas quando a máquina de destino for temporariamente ou permanentemente desconectada da rede, quando o contador de tempo de vida expirar, ou quando roteadores intermediários ficarem tão congestionados que não possam processar o tráfego que chega.

A **depuração de erros**, principalmente em redes heterogêneas, torna-se extremamente **difícil**. O próprio protocolo IP não contém nada para ajudar o emissor a testar a conectividade ou descobrir falhas.

Para permitir que os roteadores em uma rede ou em várias informem erros ou ofereçam **informações** sobre circunstâncias inesperadas, é que foi inventado o **ICMP**.

Assim como outros tipos de tráfego, as mensagens ICMP atravessam a Rede na parte de dados dos datagramas IP. Então as **mensagens ICMP são encapsuladas dentro dos datagramas IP**. Porém, o destino final de uma mensagem ICMP não é um aplicativo ou usuário na máquina de destino, mas o software da Camada de Inter-redes nessa máquina.

Ou seja, quando chega uma mensagem de erro ICMP, o módulo de software ICMP trata dela. Naturalmente, se o ICMP estabelece que determinado protocolo ou aplicativo de nível superior causou um problema, ele informa o módulo apropriado.

O ICMP permite que os roteadores ou hosts enviem **mensagens de erro ou controle** para outros roteadores ou hosts; o ICMP oferece comunicação entre o software da Camada de Inter-rede em uma máquina e o software na Camada de Inter-rede em outra.

O ICMP é apenas um mecanismo de informação, não sendo função do ICMP corrigir o erro nem tão-pouco em verificar a integridade dos datagramas que circulam pela rede. Embora a especificação do protocolo esboce os usos intencionados do ICMP e sugira possíveis ações a serem tomadas em resposta aos relacionamentos de erro, o ICMP não específica totalmente a ação a ser tomada para cada erro possível.

Quando um datagrama causa um erro, o ICMP só pode relatar a condição de erro de volta à origem do datagrama; **a origem precisa relacionar o erro a um aplicativo individual ou tomar outra ação para corrigir o problema**.

Formato de mensagem ICMP

Embora **cada** mensagem **ICMP tenha seu próprio formato**, todas elas começam com os mesmos três campos:

- Um **campo Type** (Tipo) de mensagem, que tem 8 bits e é usado para identificar a mensagem;
- Um **campo Code** (Código) de 8 bits, que oferece mais informações sobre o tipo de mensagem;
- E um **campo Checksum** (verificação) de 16 bits usado para verificação de erros da mensagem ICMP.

Além disso, as **mensagens ICMP que informam erros** sempre **incluem o cabeçalho IP mais octetos adicionais** do datagrama que causou o problema (geralmente os cabeçalhos das camadas superiores).

O motivo para o ICMP retornar mais do que o cabeçalho do datagrama apenas é permitir que o receptor determine mais exatamente quais protocolos e quais programas foram responsáveis pelo datagrama. Já que a grande maioria dos protocolos superiores trazem informações cruciais nos primeiros 64 bits.

O **campo Type** do ICMP define o significado da mensagem, além de seu formato:

Campo de tipo	Tipo de mensagem ICMP
0	Resposta de eco (echo)
3	Destino inalcançável (host unreachable)
4	Extinção de origem
5	Redirecionamento (mudar uma rota)
6	Endereço de host alternativo
8	Requisição de echo (echo)
9	Anúncio de roteador
10	Solicitação de roteador
11	Tempo excedido para um datagrama
12	Problemas de parâmetro em um datagrama
13	Requisição de estampa de tempo
14	Resposta de estampa de tempo
15	Requisição de informações
16	Resposta de informações
17	Requisição de máscara de endereço
18	Resposta de máscara de endereço
30	Tracecoute
31	Erro de conversão de datagrama
32	Redirecionar host móvel
33	“Onde está você” do Ipv6
34	“Eu estou aqui” do Ipv6
35	Requisição de registro móvel
36	Resposta de registro móvel
37	Requisição de nome de domínio
38	Resposta de nome de domínio
39	SKIP
40	Photuris

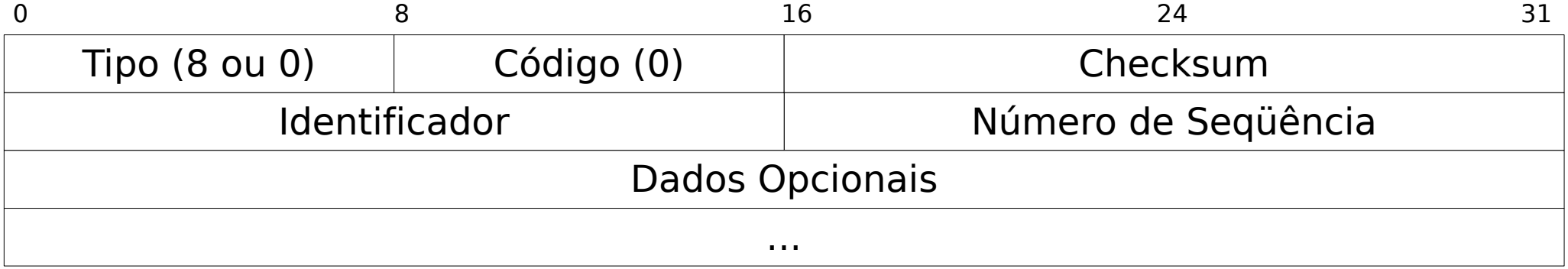
Teste de acesso e status do destino (ping)

Uma das ferramentas de depuração mais utilizadas envolve as mensagens de requisição de echo (**echo request**) e resposta de eco (**echo reply**) ICMP.

Um host ou um roteador envia uma mensagem de requisição de eco ICMP a um destino especifico. Qualquer máquina que receba uma solicitação de eco (echo request) envia um (echo reply) ao transmissor.

A requisição de eco e a resposta associada podem ser **usadas para testar se um destino é alcançável e está respondendo**. Como a requisição e a resposta associada trafegam em datagramas IP, o recebimento bem-sucedido de uma resposta verifica as principais partes do sistema de transporte IP (origem, roteadores e suas tabelas de roteamento, e o destino final).

Em muitos sistemas, o comando que os usuários invocam para enviar requisições de eco ICMP é chamado de **ping**.



O campo listado como DADOS OPCIONAIS é um campo de tamanho variável que contém dados a serem retornados ao emissor.

Uma resposta de eco sempre retorna exatamente os mesmos dados que foram recebidos na requisição. Os campos IDENTIFICADOR e NÚMERO DE SEQÜÊNCIA são usados pelo emissor para combinar com as respostas a requisições.

O valor do campo TIPO específica se a mensagem é uma requisição (8) ou uma resposta (0).

O comando ping usa mensagens ICMP de eco para verificar a comunicação entre as máquinas, porém **não confie cegamente no ping**, pois nas redes atuais muitos **administradores proibiram** mensagens ICMP de eco, o que significa que muitas máquinas mesmo estando ativas não iram responder a uma mensagem **ICMP** de eco simplesmente por que o **Firewall** não deixa.

Relatos de destinos inalcançáveis (hosts ou network unreachable)

Quando um roteador não pode encaminhar ou entregar um datagrama IP, ele envia uma mensagem ICMP de destino inalcançável (**unreacheble destination**) de volta à origem.

O **campo CÓDIGO em uma mensagem de destino inalcançável** contém um inteiro que descreve ainda mais o problema. Os valores para reste são:

Valor do código	Significado
0	Rede inalcançável
1	Host inalcançável
2	Protocolo inalcançável
3	Porta inalcançável
4	Fragmentação necessária e DF marcado
5	Rota de origem falhou
6	Rota de destino desconhecida
7	Host de destino desconhecido
8	Host de origem isolado
9	Comunicação com rede de destino proibida
10	Comunicação com host de destino proibida
11	Rede inalcançável para o tipo de serviço
12	Host inalcançável par o tipo de serviço
13	Comunicação proibida administrativamente
14	Violação de precedência de host
15	Corte de precedência em vigor

Os destinos podem ser inalcançáveis por que o hardware está temporariamente fora de serviço, ou porque o emissor especificou um destino inexistente, ou raramente porque o roteador não tem uma rota para a rede de destino.

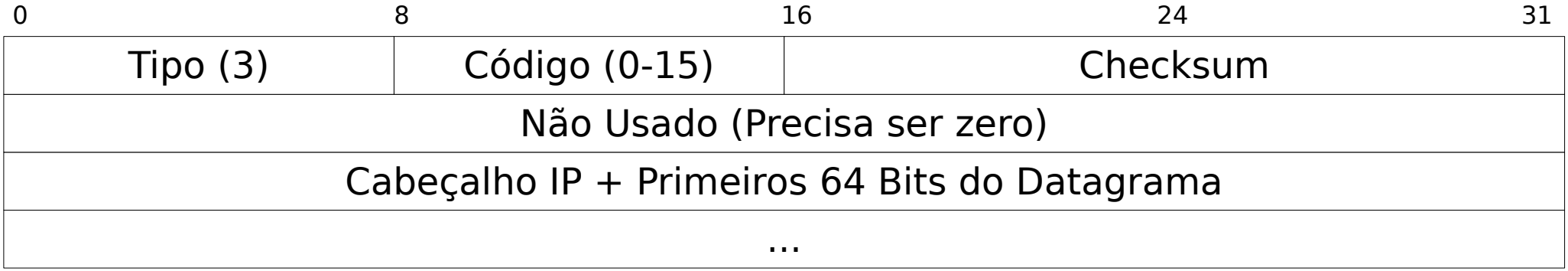
Embora um roteador envie uma mensagem de destino inalcançável quando encontra um datagrama que não pode ser encaminhada ou entregue, **um roteador não pode detectar todos esses erros.**

Se o datagrama tiver a opção de rota de origem com uma rota incorreta, ele pode disparar uma mensagem de **falha de rota de origem**.

Se um roteador precisar fragmentar um datagrama IP, mas o bit DF “não fragmentar” estiver marcado, o roteador enviará uma mensagem de **fragmentação necessária** de volta à origem.

O significado das mensagens de protocolo e porta inalcançável se tornará claro quando estudarmos as camadas superiores.

O formato da mensagem destino inalcançável, pode ser vista a seguir:



Controle de congestionamento e fluxo de datagramas

Como o **IP é sem conexão**, um **roteador não pode reservar recursos** de memória ou de comunicação antes do recebimento dos datagramas. Como resultado, os roteadores podem ser sobrecarregados de tráfego, uma condição conhecida como congestionamento.

É importante entender que o **congestionamento** pode surgir por **dois motivos** inteiramente diferentes:

Primeiro, um computador de **alta velocidade** pode ser capaz de gerar tráfego mais rapidamente do que uma rede pode transferi-lo.

Segundo, se **muitos computadores** precisarem simultaneamente enviar datagramas por um único roteador, o roteador pode sofre um congestionamento.

Quando os datagramas chegam muito rapidamente para um host ou roteador processar, ele os enfileira na memória temporariamente (buffer). Se essa **memória encher é necessário descartar os datagramas** que chegarem enquanto o buffer estiver cheio.

O **descarte de pacotes é um problema duplo**, pois além de **não receber** o **datagrama**, este **ocupa** uma vez os **recursos** da rede para chegar ao destino e encontrarem o buffer cheio e por uma possível segunda vez em sua retransmissão.

Uma máquina usa mensagens **ICMP de source quench** para informar o transmissor que há congestionamento. Então uma mensagem source quench é uma **solicitação para que o transmissor reduza seu presente nível de transmissão de datagramas.**

Normalmente roteadores **enviam mensagens source quench a cada datagrama que descartam**. Mas alguns roteadores tentam evitar o congestionamento como um todo, optando por enviar solicitação de source quench à medida que suas filas começam a ficar longas porém **antes que comecem a transbordar.**

Não há nenhuma mensagem ICMP que reverta o efeito de uma mensagem source quench. Então se um host recebe mensagens source quench, ele irá diminuir o índice de remessa dos datagramas, até que deixe de receber mensagens source quench; a partir daí, gradativamente o **índice cresce enquanto não recebe outra solicitação adicional source quench**.

Além dos tradicionais TIPO, CÓDIGO, SOMA DE VERIFICAÇÃO, e um campo de 32 bits não utilizado, as mensagens source quench possuem um campo que contém o **prefixo do datagrama**, que é o prefixo (64 bits do datagrama descartado) do datagrama que foi abandonado e que desencadeou a solicitação de source quench.

Atenção, um problema que deve ser levado em conta com o source quench é que ele não resolve o problema de congestionamento. Além, do que como pode um host que está congestionado enviar uma mensagem source quench, sendo que não existe recursos disponíveis? A resposta normalmente é que em alguns casos isto não vai ser possível. Por isto **o controle de congestionamento é tratado pelo TCP/IP na Camada de Transporte**.

Solicitação de mudança de rota por roteadores

Supõe-se que os roteadores conheçam as melhores rotas e os hosts (clientes dos roteadores) começam com um mínimo de informações sobre roteamento e aprendem novas rotas a partir dos roteadores.

Em situações especiais, **quando o roteador detecta um host que utiliza uma rota que não seja a melhor, o roteador envia ao host uma mensagem ICMP, conhecida como redirecionar, solicitando ao host que mude de rota**. O roteador também envia o datagrama original ao seu destino.

A vantagem do esquema de redirecionamento ICMP é sua simplicidade: já que permite que o host inicialize conhecendo o endereço de apenas um roteador da rede central. O primeiro roteador retorna mensagens ICMP de redirecionamento sempre que um host enviar um datagrama para o qual haja uma rota melhor. A tabela de roteamento do host permanece reduzida, mas ainda assim contém as melhores rotas para todos os destinos em uso.

Contudo as mensagens de redirecionamento não resolvem o problema de difusão de rotas em geral, porque são limitadas a interação entre um roteador e host em uma rede diretamente conectada.

Detectando rotas circulares ou excessivamente longas com ICMP.

Para evitar que os datagramas circulem para sempre em uma rede TCP/IP, cada datagrama IP contém um contador de tempo de vida chamado **TTL**. Sempre que processa algum datagrama, um roteador decrementa o contador de tempo de vida e descarta o datagrama quando o contador chega a zero.

Sempre que um host descartar um datagrama por que o TTL chegou a zero ou porque ocorreu um timeout enquanto esperava por fragmentos de um datagrama, um roteador envia uma mensagem de ICMP de tempo excedido de volta à origem do datagrama. Assim existem duas mensagens deste tipo a de Contador de tempo de vida excedido (código 0) e a de tempo de remontagem de fragmento excedido (código 1).

No caso do fragmento, existem esta mensagem por que quando um host de destino recebe o primeiro fragmento, este abre um timer para receber o restante, caso estes não cheguem antes do timer esgotar, o datagrama é considerado perdido.

Outras mensagens ICMP

Existem várias mensagens ICMP, que são tidas aqui neste máterial como menos importantes ou simplesmente estão obsoletas em todas as literaturas, são algumas:

Sincronismo de clock e estimativa de tempo de trânsito: O ICMP pode ser usado para obter a hora de outra máquina. Uma máquina solicitante enviar uma mensagem de solicitação de indicação de hora ICMP a outra máquina, pedindo que a segunda máquina retorne sua presente hora do dia. A máquina que recebe retorna uma resposta de solicitação de hora à máquina solicitante. Na verdade, a estimativa exata da duração de um retorno da transmissão pode ser difícil e restringe muito a utilidade das mensagens ICMP de indicação de hora em outras máquinas.

Mensagem de requisição e resposta de informações: Tinham por finalidade permitir que os hosts descobrissem seu endereço IP no boot do sistema, mas isto não é usado mais (obsoleto).

Mensagem de requisição e resposta de máscara de endereço: As mensagens de requisição e resposta de máscara de endereço do ICMP tinham por finalidade permitir que um host obtivesse a máscara de endereço usada na rede local.

Mensagens de requisição e anúncio de roteadores: usada para requisitar ou anunciar roteadores e permitir que um host descubram roteadores na rede.

O ICMP é uma parte obrigatória e integral do IP, usado para comunicação fora do normal. Na grande maioria dos casos, as mensagens de erro do ICMP originam de um roteador da Internet; as mensagens ICMP sempre voltam à origem do datagrama que causou o erro, e não fica restrito ao comando ping.

Este material é retirado dos seguintes livros:

TANENBAUM, Andrew S. **Redes de Computadores**. Editora Campus, 4 Edição. 2003.

COMER, Douglas E. **Interligação de Redes com TCP/IP, volume 1**. Editora Campus, 5 Edição. 2006.

TORRES, Gabriel. **Redes de Computadores: Curso Completo**. Editora Axcel Books. 1 Edição. 2001.

Todos os slides são apenas uma base para a disciplina e não dispensa a leitura dos próprios livros para compreensão do assunto como um todo.

fim