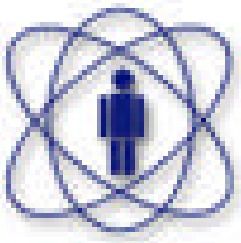
Tópicos sobre Protocolos de Comunicação



Anderson Alves de Albuquerque
Bruno Marques Amaral
Fadul da Costa Bento
Fernanda Santoro Jannuzzi
Fernando Spencer
Marcelo Eller de Amorim
Marcelo Ruela Felippe
Marita Maestrelli

Julho/2003

Prefácio

Esta nota técnica foi montada a partir de seminários apresentados pelos alunos da CAT, pertencentes ao grupo de segurança de sistemas e redes, envovendo o tema Protocolos de Comunicação em rede.

Os tópicos sobre protocolos devem ser utilizados como complemento dos livros sobre Redes de Computadores indicados no item Referências , e servem também como iniciação para os novos bolsistas do CBPF.

Apresentamos o modelo da arquitetura de rede TCP/IP, enfocando seus principais protocolos, como TCP, UDP, ICMP, IGMP, SNMP, entre outros.

Índice

1 – Introdução

A comunicação entre computadores tem vários aspectos que precisam ser tratados e por isso envolve diversos protocolos (regras, procedimentos).

A ISO (International Standards Organization), Organização Internacional para Padronização definiu um modelo de referência para a interconexão de sistemas (Open Systems Interconnection), mais conhecido como modelo OSI. Neste modelo a comunicação de dados é vista em sete camadas, desde o nível físico até o nível da aplicação utilizada pelo usuário. Em cada uma das camadas, há protocolos diferentes envolvidos, com tarefas específicas e concernentes à sua camada.

TCP/IP (Transmission Control Protocol/Internet Protocol), o tão famoso protocolo da Internet, na verdade se refere a toda uma família de protocolos voltados para o uso em redes de computadores.

Embora o TCP e o IP sejam apenas dois dos vários protocolos pertencentes a esta família, eles acabaram sendo usados para dar o nome a ela. Assim sendo, TCP/IP não é um protocolo, mas um conjunto (ou uma "suite") de protocolos.

2 – O Modelo de referência OSI / ISO

Open Systems Interconnection / International Standards Organization

O modelo OSI/ISO não especifica os serviços e protocolos que devem ser utilizados em cada camada , e sim o que cada camada deve fazer.

Camadas que compôem o modelo OSI:

- CAMADA 1: Física

  Essa camada está relacionada com a transmissão simples de bits através de um canal de comunicação.

- CAMADA 2: Enlace (link de dados)

  Estabelece meios funcionais e processuais para transmissão de dados através da rede.

- CAMADA 3: Rede

Especifica como são atribuídos endereços e como são encaminhados pacotes de um ponto da rede para outro.

- CAMADA 4: Transporte

Garante que um pacote chegue ao seu destino na transmissão em rede, isolando os níveis superiores.

- CAMADA 5: Sessão

Permite estruturar os circuitos oferecidos pelo nível de transporte.
Exemplo: login, transferência de arquivos

- CAMADA 6: Apresentação

Sua função é de realizar transformações adequadas, antes de seu envio ao nível de sessão.

- CAMADA 7: Aplicação

Define funções de gerenciamento e mecanismos genéricos que servem de suporte a aplicações distribuídas.

3 – O Modelo TCP/IP

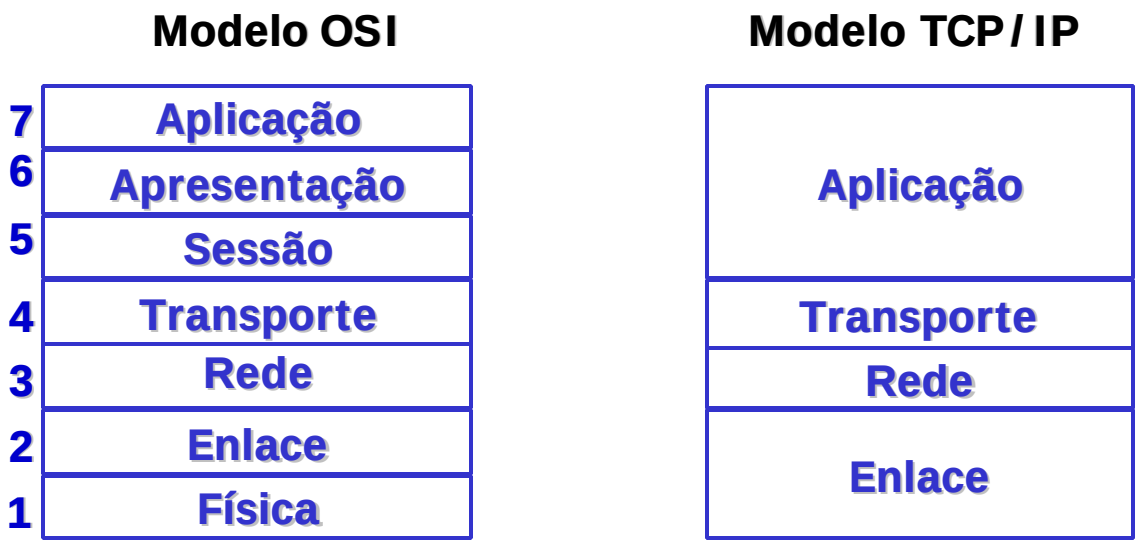
Transmission Control Protocol / Internet Protocol

O objetivo do TCP/IP é de conectar várias redes diferentes de maneira autônoma, com computadores heterogêneos ao mesmo tempo.

Trata-se de uma arquitetura extremamente flexível, capaz de se adaptar a aplicações muito diferentes.

O modelo TCP/IP quando comparado ao modelo OSI apresenta camadas que englobam mais de uma camada OSI. São a camada de aplicação e a camada de enlace(interface de rede).

Na figura a seguir vizualizamos a comparação entre os modelos OSI/ISO e TCP/IP:



Camadas TCP/IP:

- Camada de Aplicação

É formada pelos protocolos utilizados pelas diferentes aplicações do modelo TCP/IP. Esta camada não possui um padarão comum, pois o padrão é estabelecido por cada aplicação.

- Camada de Transporte

Camada fim-a-fim, isto é, uma entidade desta camada só se comunica com a sua entidade par do *host* destinatário. É nesta camada que se faz o controle da conversação entre as aplicações intercomunicadas da rede. Dois protocolos aqui são usados: o TCP e o UDP. O TCP é orientado à conexão e o UDP não. O acesso das aplicações à camada de transporte é feito através de portas que recebem um número inteiro para cada tipo de aplicação.

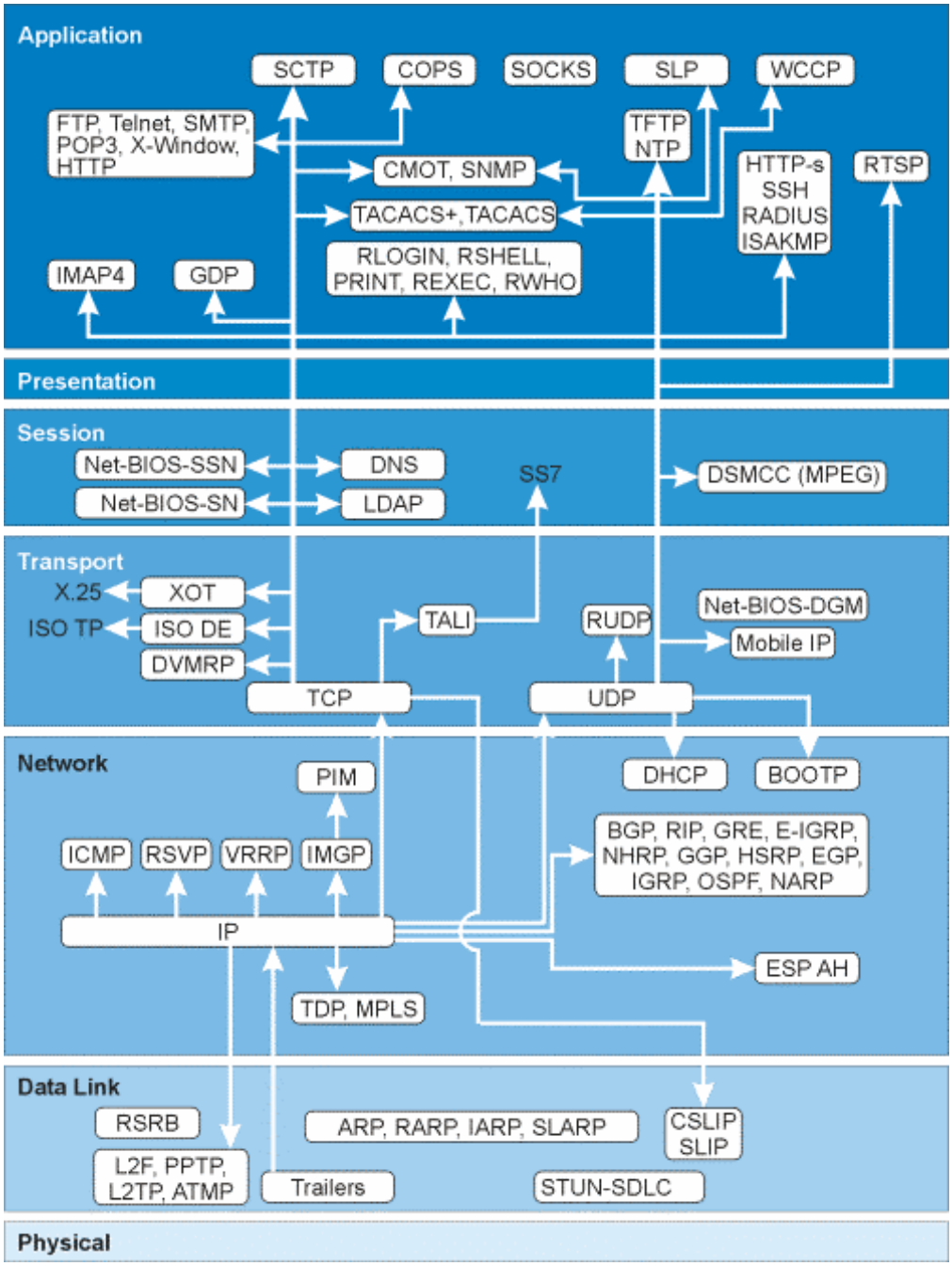
- Camada Rede (Internet)

Essa camada é a primeira normatizada do modelo. Também conhecida como camada Internet, é responsável pelo endereçamento, roteamento e controle de envio e recepção. Ela não é orientada à conexão, se comunica através de datagramas.

- Camada Enlace (host)

Camada de abstração de hardware, tem como principal função a interface do modelo TCP/IP com os diversos tipos de redes (X.25, ATM, FDDI, Ethernet, Token Ring, Frame Relay, PPP e SLIP). Por causa da grande variedade de tecnologias de rede, ela não é normatizada pelo modelo, o que provê a possibilidade de interconexão e interoperação de redes heterogêneas.

Na figura a seguir, o modelo TCP/IP em relação ao modelo OSI/ISSO com suas camadas, e os protocolos TCP/IP:



4 – Protocolo IP (Internet Protocol)

4.1 – Arquitetura e Filosofia da Interligação em Redes

Conceitualmente, uma interligação em redes TCP/IP oferece três grupos de serviços, conforme a figura 1; sua distribuição na figura sugere que há dependências entre elas. No nível mais baixo, um serviço de transmissão sem conexão oferece um fundamento sobre o qual repousa tudo mais. No nível seguinte, um serviço de transporte confiável oferece uma plataforma de nível mais alto da qual dependem os aplicativos.

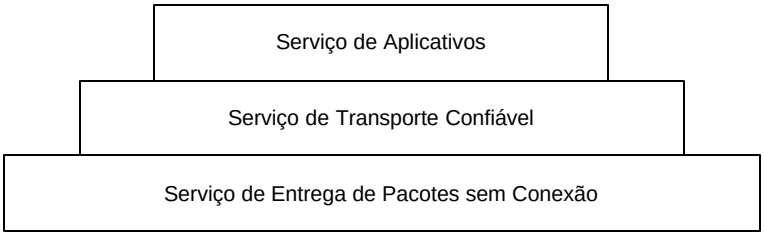


Figura 1

4.2 – O Conceito de Entrega Não-Confiável

Embora possamos associar softwares de protocolo a cada um dos serviços da figura 1, a razão de sua identificação como partes conceituais da interligação em redes é que elas claramente estabelecem as estruturas filosóficas do projeto. O ponto é:
O software da interligação em redes é projetado com base em três serviços de rede conceituais organizados hierarquicamente; grande parte de seu sucesso ocorreu porque essa arquitetura é surpreendentemente eficiente e adaptável.
Uma das vantagens mais significativas dessa divisão conceitual é que ela tona possível substituir um serviço sem prejudicar os demais. Assim, a pesquisa e o desenvolvimento podem progredir simultaneamente ao longo dos três serviços.

4.3 – Sistema de Transmissão sem Conexão

O serviço mais importante da interligação em redes consiste em um sistema de entrega de pacotes. Tecnicamente, o serviço é definido como um sistema de transmissão sem conexão, best-effort e não confiável; é análogo ao serviço oferecido por hardwares de redes que operam em um paradigma de transmissão best-effort. O serviço é conhecido como não-confiável porque a entrega não é garantida. O pacote pode ser perdido, reproduzido, atrasar-se ou ser entregue com problemas, mas o serviço não detectará tais condições, nem informará isso

ao transmissor nem ao receptor. Ele é denominado sem conexão porque cada pacote é independente dos outros. Uma seqüência de pacotes enviados de um computador a outro pode trafegar por caminhos diferentes, ou alguns podem ser perdidos enquanto outros são entregues. Finalmente, o serviço utiliza uma transmissão best-effort porque o software de interligação em redes faz uma série de tentativas para entregar os pacotes. Isso significa que a interligação em redes não rejeita pacotes por simples capricho; a não-confiabilidade surge quando os recursos esgotam-se ou as redes básicas falham.

4.4 – Finalidade do Protocolo de Interligação em Redes

O protocolo que define o mecanismo de transmissão sem conexão e não-confiável é conhecido como Internet Protocol, e é normalmente citado por suas iniciais IP. O IP oferece três definições importantes. Primeira, o protocolo IP define a unidade básica de transferência de dados utilizada através de uma interligação em redes TCP/IP. Assim, ela especifica o formato exato de todos os dados à medida que ela passa pela interligação em redes TCP/IP. Segunda, o software IP desempenha a função de roteamento, escolhendo um caminho por onde os dados serão enviados. Terceira, além da especificação formal e precisa de formatos de dados e de roteamento, o IP inclui um conjunto de regras que concentram a idéia da entrega não-confiável de pacotes. As regras definem como os hosts e os roteadores devem processar os pacotes, como e quando as mensagens de erro devem ser geradas e as condições segundo as quais os pacotes podem ser descartados. O IP é uma parte tão fundamental do projeto de uma interligação em redes TCP/IP às vezes é denominada uma tecnologia baseada em IP.

4.5 – O Datagrama de Interligação em Redes

A analogia entre uma rede física e uma interligação em redes TCP/IP é grande. Numa rede física, a unidade de transferência é um quadro que contém um cabeçalho e dados, onde o cabeçalho fornece informações como endereço de origem e de destino (físico). A interligação em redes denomina sua unidade básica de transferência de um datagrama de interligação em redes, às vezes citado como um datagrama IP, ou simplesmente um datagrama. Como um quadro típico de rede física, um datagrama é dividido em cabeçalho e áreas de dados. Também como um quadro, o cabeçalho de um datagrama contém os endereços de origem e destino e um tipo de campo que identifica o conteúdo do datagrama. Naturalmente, a diferença é que o cabeçalho do datagrama contém os endereços IP, enquanto o quadro contém os endereços físicos. A figura 2 mostra o formato geral de uma datagrama.

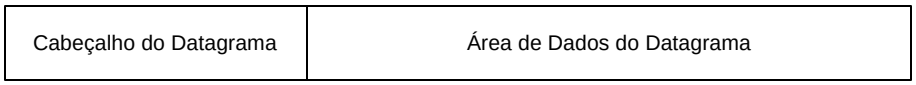


Figura 2

4.6 – Formato do Datagrama

Agora que já descrevemos o formato geral de um datagrama IP, podemos examinar o conteúdo com mais detalhes. A figura 3 mostra a organização dos campos em um datagrama.

Já que o processamento de datagramas se dá em softwares, o conteúdo e o formato não são restringidos por quaisquer hardwares. O primeiro campo de quatro bits de um datagrama (VERS), por exemplo, contém a versão do protocolo IP utilizada para criar o datagrama. Ele é utilizado para verificar se o transmissor, o receptor e quaisquer roteadores existentes entre ele concordam quanto ao formato do datagrama. Todo software IP precisa verificar o campo de versão antes de processar um datagrama, para assegurar-se de que ele se adapta ao formato que o software espera. Se os padrões mudarem, as máquinas rejeitarão datagramas com versões de protocolos diferentes dos seus, impedindo que eles deturpem o conteúdo da datagrama com um formato desatualizado. A versão atual do protocolo IP é a quatro.

O campo de comprimento do cabeçalho (HLEN), também de quatro bits, fornece o comprimento do cabeçalho do datagrama medido em palavras de 32 bits. Como veremos, todos os campos do cabeçalho contêm um comprimento fixo, exceto para OPÇÕES IP e os campos correnpondentes PADDING. O cabeçalho mais comum, que não contém qualquer opção e nenhum preenchimento, mede 20 octetos e o campo de comprimento de cabeçalho é cinco.

O campo COMPRIMENTO TOTAL fornece o comprimento do datagrama IP medido em octetos, incluindo octetos no cabeçalho e nos dados. O tamanho da área de dados pode ser calculando subtraindo-se de COMPRIMENTO TOTAL o comprimento do cabeçalho (HLEN). Já que o campo COMPRIMENTO TOTAL possui 16 bits de comprimento, o maior tamanho possível para um datagrama IP é 216 ou 65-535 octetos. Na maioria dos aplicativos, essa não é uma limitação rígida. No futuro pode tornar-se mais importante, se as redes de velocidade mais alta puderem transportar pacotes de dados maiores que 65.535 octetos.

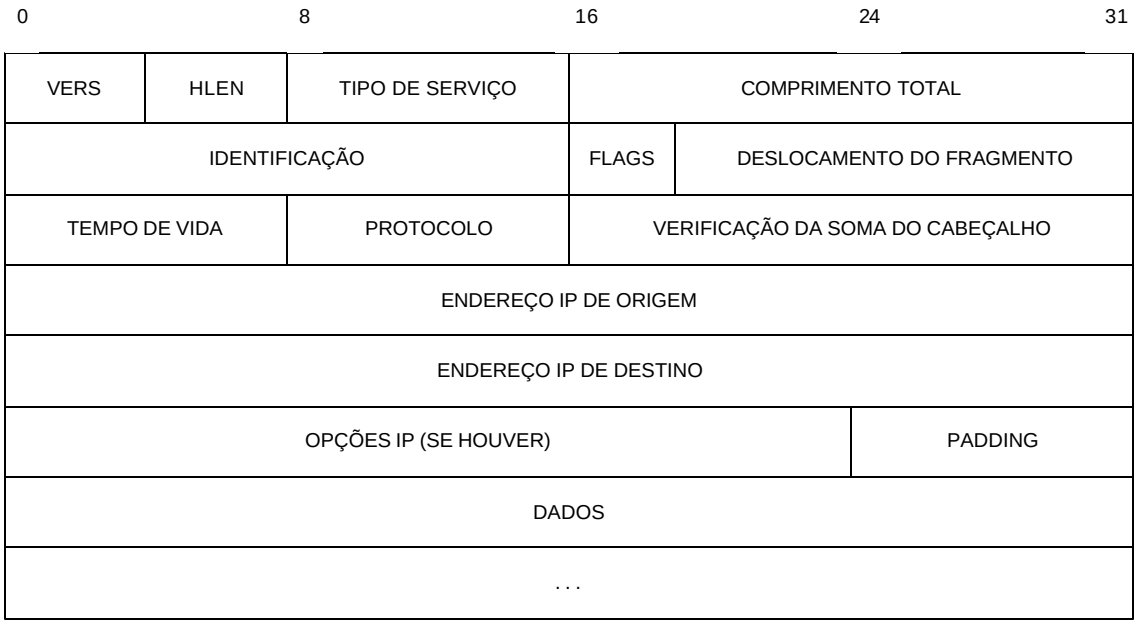


Figura 3

4.7 – Os Campos do Tipo de Serviço e Precedência

Denominado informalmente Type of Service (TOS), o campo TIPO DE SERVIÇO, de oito bits, especifica como o datagrama deve ser tratado e é fracionado em cinco subcampos, como mostra a figura 4.

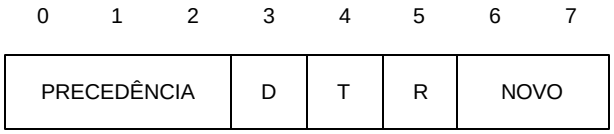


Figura 4

Três bits PRECEDÊNCIA especificam a precedência do datagrama com valores variando de zero (precedência normal) até sete (controle de rede), permitindo que os transmissores indiquem a importância de cada datagrama. Embora a maioria dos softwares que rodam e hosts ignorem o tipo de serviço, trata-se de um conceito importante, porque fornece um mecanismo que pode permitir que informações de controle tenham precedência sobre dados. Se, por exemplo, todos os hosts e roteadores reconhecem a precedência, é possível implementar algoritmos de controle de congestionamento que não sejam influenciados pelo congestionamento que estão tentando controlar.
Os bits D, T e R especificam o tipo de transporte que o datagrama deseja. Quando ajustado, o bit D solicita um intervalo baixo, o bit T solicita um throughput alto e o bit R solicita alta confiabilidade. É claro que não deve ser

possível que a interligação em redes garanta o tipo de transporte solicitado (ou seja, pode acontecer que nenhum caminho para o destino tenha a propriedade solicitada). Assim, consideramos, a solicitação de transporte como uma sugestão para os algoritmos de roteamento, e não uma exigência. Se um roteador realmente conhece mais de uma rota possível para determinado destino, ele pode utilizar o tipo de campo de transporte para selecionar aquelas cujas características mais se aproximem das desejadas. Suponha, por exemplo, que um roteador possa selecionar entre uma linha alugada, de baixa capacidade, e uma conexão de alta, de satélites de banda larga (mas de intervalo alto). O conjunto de bits D poderia solicitar aos datagramas que carregam toque no teclado de um usuário para um computador remoto que esses sejam entregues o mais rápido possível, enquanto um conjunto de bits T poderia solicitar aos datagramas correspondentes uma transferência de arquivos que trafeguem nos links de alta capacidade de satélites.
Também é importante entender que os algoritmos de roteamento precisam escolher entre tecnologias de redes físicas básicas as quais possuem, cada uma, características de intervalo, throughput e confiabilidade. De uma maneira geral, algumas tecnologias representam um compromisso entre duas características (por exemplo, um maior throughput em detrimento de maiores retardos). Assim, a idéia é apresentar uma sugestão ao algoritimo de roteamento sobre o que é mais importante, e raramente faz sentido especificar os três tipos de serviço. Para resumir:
Consideramos o tipo de especificação de transporte uma sugestão ao algoritmo de roteamento para ajudá-lo a escolher entre os vários caminhos para um destino, com base em seu conhecimento das tecnologias de hardware disponíveis nesses caminhos. Uma interligação em redes não garante o tipo de transporte solicitado.

4.8 – Encapsulamento de Datagramas

Antes que possamos compreender os próximos campos de um datagrama, é importante considerar como eles relacionam-se com os quadros de redes físicas. Começaremos com uma pergunta: “que tamanho um datagrama pode ter?” Ao contrário de quadros de redes físicas que precisam ser reconhecidos pelo hardware, os datagramas são tratados por software. Eles podem ter qualquer tamanho que os projetistas de protocolos escolherem. Já vimos que o atual formato de datagrama aloca somente 16 bits para o campo de comprimento total, limitando o datagrama a, no máximo, 65.535 octetos. Entretanto, esse limite poderia ser mudado em versões posteriores do protocolo. Os limites mais importantes para o tamanho de datagramas surgem na prática. Sabemos que, à medida que os datagramas movem-se de uma máquina para outra, eles precisam sempre ser transportados por uma rede física básica. Para tornar o transporte da interligação em redes eficiente, gostaríamos de assegurar que cada datagrama viaje em um quadro físico distinto. Isso significa que

desejamos que nossa abstração de um pacote de rede física mapeie diretamente para dentro de um pacote real, se possível.
A idéia de transportar um datagrama em um quadro de rede é denominada encapsulamento. Para a rede básica, um datagrama é como qualquer outra mensagem enviada de uma máquina a outra. O hardware não reconhece o formato do datagrama, nem entende o endereço de destino IP. Assim, conforme mostra a figura 5, quando uma máquina envia um datagrama IP a outra, todo o datagrama é transportado na porção de dados do quadro de rede.

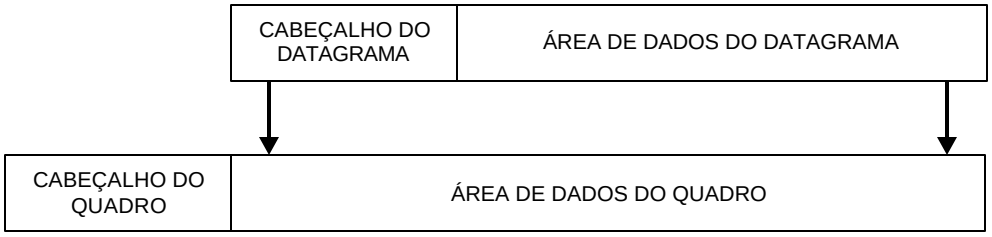


Figura 5

4.9 – Tamanho do Datagrama, MTU da Rede e Fragmentação

Na situação ideal, todo o datagrama IP encaixa-se em um quadro físico, tornando a transmissão na rede física eficiente. Para obter essa eficiência, os projetistas de IP devem ter selecionado um tamanho máximo de datagrama, de modo que ele sempre se encaixe em um quadro. Mas que tamanho de quadro deve ser escolhido? Acima de tudo, um datagrama pode trafegar em muitos tipos de redes físicas, à medida que move-se na interligação em redes para o seu destino final.
Para compreender o problema, precisamos entender um fato sobre o hardware de rede: cada tecnologia de comutação de pacotes coloca um limite superior, fixo, no total de dados que podem ser transferido em um quadro físico. A Ethernet, por exemplo, limita as transferências a 1.500 octetos de dados, enquanto a FDDI permite aproximadamente 4.470 octetos de dados por quadro. Referimo-nos a esses limites como MTU (maximum transfer unit). O tamanho da MTU pode ser bem pequeno: algumas tecnologias de hardware limitam a transferência para 128 octetos ou menos. Limitar os datagramas para encaixar a menor MTU possível na interligação em redes torna a transferência ineficaz quando aqueles datagramas trafegam em uma rede que pode transportar quadros de tamanho maiores. Enquanto, permitir que os datagramas sejam maiores que a MTU mínima da rede em uma interconexão significa que um datagrama nem sempre irá encaixar-se no quadro único de uma rede.
A escolha deve ser óbvia: o objetivo do projeto de interligação em redes é concentrar as tecnologias de rede básicas e facilitar a comunicação para o usuário. Assim, em vez de projetar datagramas que sigam as restrições de redes físicas, o software TCP/IP escolhe um tamanho inicial de datagrama conveniente e descobre uma forma de dividir os datagramas extensos em frações menores,

quando o datagrama precisa atravessar uma rede que tenha uma MTU pequena. As pequenas frações em que um datagrama é dividido são denominada fragmentos, e o processo de divisão de um datagrama é conhecido como fragmentação.
Conforme a figura 6, a fragmentação normalmente ocorre em um roteador situado em algum ponto ao longo do caminho entre a origem do datagrama e seu destino final. O roteador recebe um datagrama de uma rede com uma MTU grande, e precisa enviá-lo em uma rede para a qual a MTU seja menor do que o tamanho do datagrama.

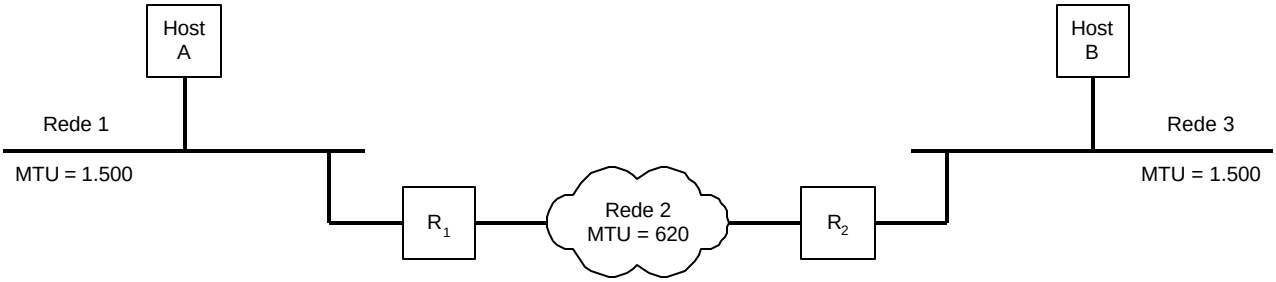


Figura 6

Na figura, ambos os hosts conectam-se diretamente às Ethernets que possuam uma MTU de 1.500 octetos. Assim, os dois hosts podem gerar e enviar datagramas de até 1.500 octetos de comprimento. O caminho entre eles, entretanto, inclui uma rede com uma MTU de 620. Se o host A envia ao host B um datagrama maior que 620 octetos, o roteador R1 fragmentará o datagrama. Da mesmo forma, se B envia um datagrama grande a A, o roteador R2 fragmentará o datagrama.
O tamanho do fragmento é escolhido de tal forma que cada fragmento possa ser transportado na rede básica em um quadro único. Além disso, já que o IP representa o deslocamento dos dados em múltiplos de oito octetos, o tamanho do fragmento precisa ser um múltiplo de oito. É claro que escolher o múltiplo de oito octetos mais próximo do MTU da rede nem sempre divide o datagrama em frações de igual tamanho; a última fração é normalmente menor que as outras. Os fragmentos devem ser remontados para produzir uma cópia completa do datagrama original, antes que ele possa ser processado no destino.
O protocolo IP não limita datagramas a um tamanho pequeno, nem garante que datagramas grandes serão entregues sem fragmentação. A origem pode escolher qualquer tamanho de datagrama que julgar apropriado; a fragmentação e remontagem ocorrem automaticamente, sem qualquer ação específica por parte da origem. A especificação do IP indica que os roteadores precisam aceitar datagramas até o máximo de MTUs de rede às quais se conectam. Além disso, um roteador precisa sempre tratar os datagramas de até 576 octetos. (Hosts também devem aceitar e remontar, se necessário, os datagramas de, no mínimo, 576 octetos.)

Fragmentar um datagrama significa dividi-lo em várias frações. Deve surpreendê-lo o fato de que cada fração tem o mesmo formato que o datagrama original. A figura 7 ilustra o resultado da fragmentação.
Cada fragmento contém um cabeçalho de datagrama que duplica a maior parte do cabeçalho do datagrama original (exceto para um bit no campo FLAGS que mostra que é um fragmento), seguido por tantos dados quantos puderem ser transportados no fragmento, enquanto mantém o comprimento total menor que a MTU da rede na qual precisa trafegar.

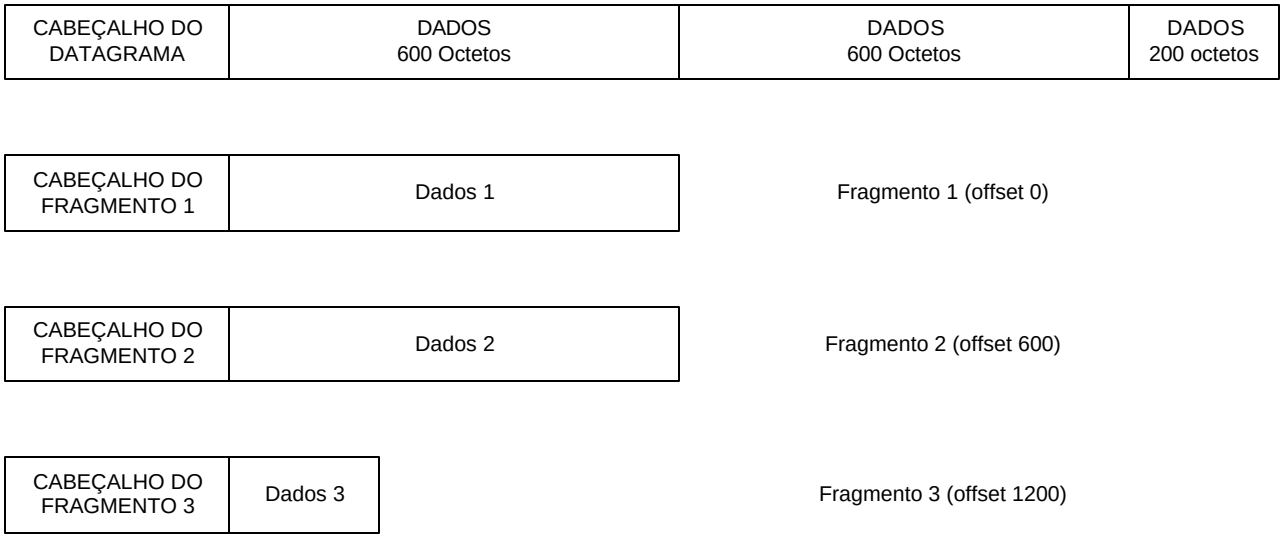


Figura 7

4.10 – Remontagem de Fragmentos

O datagrama deve ser remontado após passar em uma rede, ou os fragmentos devem ser transportados para o host final antes da remontagem? Em uma interligação de redes TCP/IP, quando um datagrama tiver sido fragmentado, os fragmentos trafegam como datagramas isolados ao longo do percurso até o último destino onde precisam ser remontados. Há duas desvantagens em preservar os fragmentos ao longo do percurso até o final. Primeira, porque datagramas não são remontados imediatamente após passarem por uma rede com uma MTU pequena, e os fragmentos pequenos precisam ser transportados do ponto de fragmentação até o destino final. A remontagem de datagramas no destino final pode levar à ineficiência: mesmo se algumas redes físicas encontradas após o pronto de fragmentação possuírem grande capacidade de MTU, apenas pequenos fragmentos atravessam-na. Segunda, se quaisquer fragmentos forem perdidos, o datagrama não pode ser remontado. A máquina receptora inicia um temporizador de remontagem quando recebe um fragmento inicial. Se o temporizador terminar antes que todos os fragmentos cheguem, a máquina receptora descarta os fragmentos remanescentes sem processar o datagrama. Assim, a probabilidade de perda de datagrama cresce quando a

fragmentação ocorre, porque a perda de um fragmento único resulta na perda do datagrama inteiro.
Apesar das pequenas desvantagens, a execução de remontagem no destino final funciona bem. Permite que cada fragmento seja roteado independentemente, e não exige que roteadores intermediários armazenem ou remontem fragmentos.

4.11 – Controle de Fragmentação

Três campos no cabeçalho do datagrama, IDENTIFICAÇÃO, FLAGS e OFF-SET DE FRAGMENTO, controlam a fragmentação e a remontagem de datagramas. O campo IDENTIFICAÇÃO contém um número inteiro único que identifica o datagrama. Lembre-se de que, quando um roteador fragmenta um datagrama, ele copia a maioria dos campos no cabeçalho do datagrama para cada fragmento. O campo IDENTIFICAÇÃO precisa ser copiado. Sua finalidade é permitir que o destino saiba quais datagramas estão chegando e a que datagramas pertencem. À medida que chega um fragmento, o destino utiliza o campo IDENTIFICAÇÃO juntamente com o endereço de origem do datagrama para que esse seja identificado. Os computadores que estão enviando os datagramas IP devem gera um valor único para o campo IDENTIFICAÇÃO, para cada datagrama. Uma técnica utilizada pelo software IP mantém uma contagem global em memória, incrementa-a a cada vez que um novo datagrama é criado e atribui o resultado como o campo IDENTIFICAÇÃO do datagrama.
Lembre-se de que cada fragmento possui exatamente o mesmo formato que um datagrama completo. Para um fragmento, o campo OFFSET DE FRAGMENTO especifica o deslocamento, no datagrama original, dos dados que estão sendo transportados no fragmento, medidos em unidades de oito octetos, iniciando em deslocamento zero. Para remontar o datagrama, o destino precisa obter todos os fragmentos que iniciam com o framento que possui deslocamento zero até o fragmento de maior deslocamento. Os fragmentos não chegam necessariamente em ordem, e não há comunicação entre o roteador que fragmento o datagrama e o destino que está tentando remontá-lo.
Os dois bits de baixa ordem, do campo FLAGS de três bits, controlam a fragmentação. Normalmente, o software aplicativo que utiliza TCP/IP não dá atenção à fragmentação, porque essa e a remontagem são procedimentos automáticos que ocorrem em um baixo nível do sistema operacional, invisível para usuários finais. Entretanto, para testar o software de interligação em redes ou depurar problemas operacionais, deve ser importante testar os tamanhos dos datagramas para os quais a fragmentação ocorre. O primeiro bit de controle auxilia nesse teste, especificando se o datagrama pode ser fragmentado. Ele é conhecido como bit não-fragmentar porque o seu ajuste em um especifica que o datagrama não deve ser fragmentado. Um aplicativo pode optar por não permitir uma fragmentação quando somente o datagrama inteiro é útil. Considere, por exemplo, um procedimento de inicialização de um computador, no qual uma máquina começa a executar um pequeno programa na ROM que utiliza a

interligação em redes para solicitar uma inicialização inicial, e uma outra máquina retorna uma imagem de memória. Se o software tiversido projetado de tal modo que a imagem de inicialização só tenha utilidade se obtida de uma única vez, o datagrama deve ter um conjunto de bits não-fragmentar. Toda vez que um roteador precisa fragmentar um datagrama que possui o conjunto de bits não-fragmentar, o roteador descarta o datagrama e retorna à origem uma mensagem de erro.
O bit de mais baixa ordem, no campo FLAGS, especifica se o fragmento contém a parte do meio ou do final dos dados do datagrama. Ele é conhecido como bit de mais fragmentos. Para verificar por que esse bit é necessário, considere o software IP no destino final tentando remontar um datagrama. Ele receberá fragmentos (possivelmente fora de ordem) e precisa saber quando recebeu todos os fragmentos para um datagrama. Quando um fragmento chega, o campo COMPRIMENTO TOTAL, do cabeçalho, aplica-se ao tamanho do fragmento e não ao tamanho do datagrama original; assim, o destino não pode utilizar o campo COMPRIMENTO TOTAL para inferir se reuniu todos os fragmentos. O bit de mais fragmentos resolve o problema facilmente: quando o destino recebe um fragmento com o bit mais fragmentos desativado, ele sabe que o esse fragmento transporta a parte final dos dados do datagrama original. Partindo dos campos de OFFSET DE FRAGMENTO e COMPRIMENTO TOTAL, ele pode calcular o comprimento do datagrama original. Examinando o OFFSET DE FRAGMENTO e o COMPRIMENTO TOTAL de todos os fragmentos que chegaram, um receptor pode dizer se os fragmentos sob controle contêm todos os dados necessários para remontar todo o datagrama original.

4.12 – TTL (Time To Live ou Tempo de Vida)

O campo TEMPO DE VIDA especifica quanto tempo, em segundos, o datagrama pode permanecer no sistema de interligação em redes. A idéia é simples e importante: toda vez que uma máquina injeta um datagrama na interligação em redes, ela estabelece um tempo máximo de vida para o datagrama. Os roteadores e os hosts que processam datagramas precisam decrementar o campo TEMPO DE VIDA à medida que o tempo passa e remover o datagrama da interligação em redes quando seu tempo expira.
Estimar o tempo exato é difícil porque os roteadores geralmente não sabem o tempo de trânsito para redes físicas. Poucas regras simplificam o processamento e facilitam o tratamento dos datagramas em o uso de relógios sincronizados. Primeiramente, cada roteador colocado ao longo do trajeto, da origem ao destino, precisa decrementar em um campo TEMPO DE VIDA quando ele processa o cabeçalho do datagrama. Além disso, para tratar as ocorrências de roteadores sobrecarregados implicam retardos longos, cada roteador registra o tempo local quando o datagrama chega e decrementa o TEMPO DE VIDA no número de segundos que o datagrama permaneceu dentro do roteador esperando serviço.

Sempre que um campo TEMPO DE VIDA alcança zero, o roteador descarta o datagrama e envia uma mensagem de erro de volta à origem. A idéia de manter um temporizador para datagramas é interessante, porque assegura que os datagramas não podem trafegar indefinidamente na interligação em redes, ainda que as tabelas de roteamento fiquem destruídas e os roteadores direcionem datagramas em círculo.

4.13 – Outros Campos do Cabeçalho de Datagramas

O campo PROTOCOLO é análogo ao campo de tipo em um quadro de rede. O valor no campo PROTOCOLO especifica qual protocolo de alto nível foi utilizado para criar a mensagem que está sendo transportada na área de DADOS do datagrama. Na verdade, o valor PROTOCOLO especifica o formato da área DADOS. O mapeamento entre um protocolo de alto nível e o valor de número inteiro no campo PROTOCOLO precisa ser administrado por uma autoridade central para garantir um consenso em toda a Internet.

O campo VERIFICAÇÃO DA SOMA DO CABEÇALHO assegura a integridade dos valores de cabeçalho. A verificação IP é formada com o tratamento do cabeçalho como uma seqüência de números inteiros de 16 bits (na ordem de bytes da rede), reunindo-os com uma aritmética complemento de um, e a seguir considerando o complemento de um como o resultado. Para a finalidade de calcular a soma de verificação, considera-se que o campo VERIFICAÇÃO DA SOMA DO CABEÇALHO contenha 0.

É importante observar que a soma de verificação somente se aplica a valores do cabeçalho IP, e não aos dados. Há vantagens e desvantagens em separar a soma de verificação para cabeçalhos e dados. Como o cabeçalho normalmente ocupa menos octetos que os dados, ter uma soma de verificação separada reduz o tempo de processamento nos roteadores que somente precisam calcular somas de verificação de cabeçalhos. A separação também permite que protocolos de mais alto nível escolham seu próprio esquema de soma de verificação para os dados. A principal desvantagem é que os protocolos de mais alto nível são forçados a acrescentar sua própria soma de verificação, ou arriscar que dados destruídos prossigam sem que sejam detectados.

Os campos ENDEREÇO IP DE ORIGEM e ENDEREÇO IP DE DESTINO contêm endereços IP de 32 bits do transmissor do datagrama e do receptor desejado. Embora o datagrama possa ser roteado através de muitos roteadores intermediários, os campos da origem e destino nunca mudam; eles especificam os endereços IP da origem e do último destino.

O campo denominado DADOS na figura 6 mostra o início da área de dados do datagrama. Seu comprimento depende, logicamente, do que está sendo enviado no datagrama. O campo OPÇÕES IP, abordado abaixo, é de comprimento variável. O campo denominado PADDING depende das opções selecionadas. Ele representa bits contendo zero e que podem ser necessários para garantir que o cabeçalho do datagrama se estenda até o múltiplo exato de 32 bits

(lembre-se de que o campo de comprimento do cabeçalho é especificado em unidades de palavra de 32 bits).

4.14 – Opções nos Datagramas de Interligação em Redes

O campo OPÇÕES IP que se segue ao endereço de destino não é necessário em todo datagrama, e as opções são incluídas principalmente para testes ou depuração da rede. Contundo, o processamento de opções é parte integrante do protocolo IP; assim, todas as implementações de padrões precisam incluí-lo.
O comprimento do campo OPÇÕES IP varia de acordo com as opções selecionadas. Algumas delas têm um octeto de comprimento e consistem em um único código de opção de octeto. Outras opções têm comprimento variável. Quando as opções são apresentadas em um datagrama, elas aparecem bem próximas, sem quaisquer separadores especiais. Cada opção consiste em um código de opção de octeto único que pode ser seguido por um comprimento de octeto único e um conjunto de octetos de dados para aquela opção. O octeto de código de opção é dividido em três campos, conforme a figura 8.
Os campos consistem em um flag CÓPIA de um bit, uma CLASSE DE OPÇÕES de dois bits e um NÚMERO DE OPÇÕES de cinco bits. O flag CÓPIA controla o modo como os roteadores tratam as opções durante a fragmentação. Quando o bit CÓPIA é ajustado em um, ele especifica que a opção deve ser copiada em todos os fragmentos. Quando ajustado em zero, o bit CÓPIA significa que a opção somente deve ser copiada no primeiro fragmento, e não em todos eles.

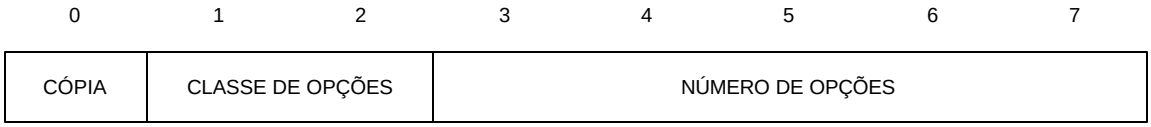


Figura 8

Os bits CLASSE DE OPÇÕES e NÚMERO DE OPÇÕES especificam a classe geral da opção e fornecem uma opção específica nessa classe.

A tabela da figura 9 mostra como as classes são atribuídas.

Classes de Opções	Significado
0	Controle de rede ou de datagrama
1	Reservado para utilização futura
2	Depuração e avaliação
3	Reservado para utilização futura

Figura 9

A tabela da figura 10 relaciona as opções possíveis que podem acompanhar um datagrama IP e fornece seus valores CLASSE DE OPÇÕES e NÚMERO DE OPÇÕES. Conforme mostra a lista, a maioria das opções é utilizada para fins de controle.

Classe de Opções	Número de Opções	Comprimento	Descrição
0	0	-	Fim da lista de opções. Será utilizado se opções terminarem no fim do cabeçalho (ver também campo padding do cabeçalho).
0	1	-	Nenhuma operação (utilizado para alinhar actetos em um alista de opções).
0	2	11	Restrições de segurança e tratamento (para aplicações militares).
0	3	Var	Roteamento de origem separado. Utilizado para rotear um datagrama ao longo de um caminho específico.
0	7	Var	Rota de registro. Utilizado para traçar uma rota.
0	8	4	Identificador de fluxo. Utilizado para transportar um identificador de fluxo SATNET (Obsoleto).
0	9	Var	Roteamento de origem restrito. Utilizado para rotear um datagrama ao longo de um caminho especificado.
2	4	Var	Indicação de hora da inter-rede. Utilizado para registrar a indicação de hora ao longo da hora.

Figura 10

4.15 – Opção de Armazenamento de Rota

As opções de indicação de hora (timestamp) e de roteamento são as mais interessantes porque oferecem uma forma para monitorar ou controlar como os roteadores de interligação em redes direcionam datagramas. A opção rota de

registro permite que a origem crie uma lista vazia de endereços IP e faz com que o endereço IP de cada roteador que processe o datagrama seja acrescentado à lista. A figura 11 mostra o formato da opção de rota de registro.

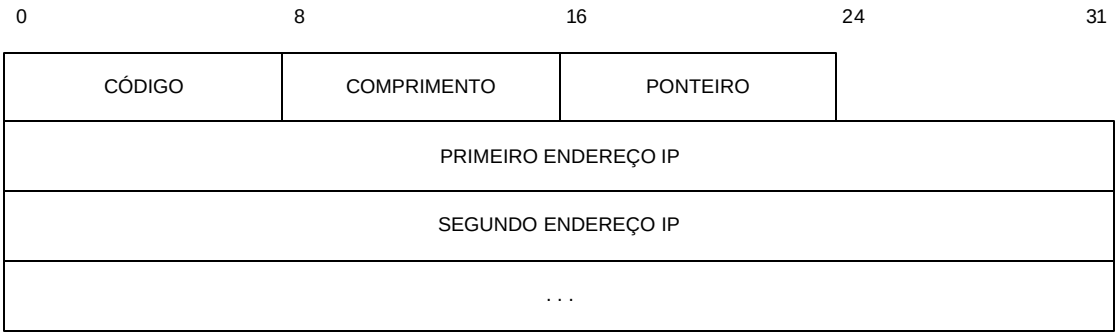


Figura 11

Conforme foi anteriormente descrito, o campo CÓDIGO contém a classe de opções e o número de opções (zero e sete para a rota de registro). O campo COMPRIMENTO especifica o comprimento total da opção conforme aparece no datagrama IP, inclusive os três primeiros octetos. Os campos que iniciam com o rótulo PRIMEIRO ENDEREÇO IP compreendem uma área reservada para a gravação de endereços da interligação em redes. O campo PONTEIRO especifica o deslocamento dentro da opção do próximo slot disponível.
Toda vez que uma máquina processa um datagrama. No qual é setada a opção de armazenamento da rota, ela acrescenta seu endereço à lista (espaço suficiente deve ser alocado pela origem para manter todas as entradas necessárias). Para acrescentar-se à lista, uma máquina primeiro compara os campos de ponteiro e comprimento. Se o ponteiro for maior que o comprimento, a lista estará completa. De modo que a máquina encaminhe o datagrama sem inserir sua entrada. Se a lista não estiver completa, a máquina insere seu endereço IP de quatro octetos na posição especificada pelo PONTEIRO e incrementa o ponteiro quatro posições.
Quando o datagrama chega, a máquina de destino pode extrair e processar a lista de endereços IP. Normalmente, um computador que recebe um datagrama ignora a rota armazenada. Utilizar a opção de rota de registro exige que duas máquinas concordem em cooperar; um computador não irá automaticamente receber rotas armazenadas em datagramas que saem. A origem precisa concordar em ativar a opção de rota de registro e do destino precisa concordar em processar a lista resultante.

4.16 – Opções de Rota de Origem

Uma outra idéia que outros projetista de rede acham interessante é a opção de rota de origem. A idéia implícita no roteamento de origem é que ele oferece uma maneira de o transmissor impor um caminho pela interligação em redes. Para

testar, por exemplo, o throughput em uma rede física especial, N, administradores do sistema podem utilizar o roteamento de origem para forçar datagramas IP a atravessar a rede N, mesmo se roteadores escolhessem normalmente um caminho que não a incluísse. A habilidade de executar esses testes é especialmente importante em um ambiente de produção, porque oferece ao gerente da rede a liberdade de rotear datagramas de usuários em redes conhecidas para operar corretamente, enquanto testem simultaneamente outras redes. É claro que esse roteamento é apenas útil para pessoas que compreendem a topologia de rede, pois o usuário médio não precisa conhecer este recurso ou utilizá-lo.
O protocolo IP aceita duas formas de roteamento de origem. Uma delas, denominada roteamento de origem, especifica um caminho de roteamento através da inclusão de uma seqüência de endereços IP na opção, como mostra a figura 12.

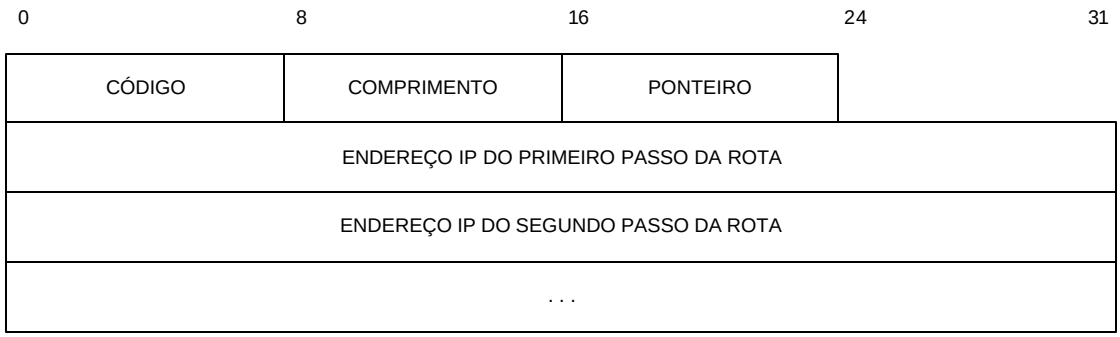


Figura 12

No roteamento restrito da origem os endereços especificam o caminho exato que o datagrama deverá seguir para chegar a seu destino. O caminho entre dois endereços consecutivos da lista deve ser composto de uma única rede física. Se um roteador não conseguir acompanhar uma rota de origem restrita, ocorrerá um erro. A outra forma, denominada roteamento flexível de origem, também contém uma seqüência de endereços IP. Ela determina que o datagrama deverá seguir a seqüência de endereços IP, mas permite a existência de vários passos de rota de rede entre endereços consecutivos na lista.
As duas opções de rota de origem exigem que haja roteadores ao longo do caminho para substituir itens da lista de endereços por seus endereços de rede locais. Dessa forma, quando o datagrama chega a seu destino, ele contém uma lista de todos os endereços percorridos, exatamente como a lista produzida pela opção de armazenamento de rota.
O formato de uma opção de rota de origem assemelha-se ao da opção de armazenamento de rota, mostrado anteriormente. Cada roteador analisa os campos PONTEIRO e COMPRIMENTO para verificar se a lista foi esgotada. Se for esse o caso, o ponteiro será maior que o comprimento e o roteador direcionará o datagrama até seu destino, como de costume. Se a lista não tiver sido esgotada, o roteador seguirá o ponteiro, selecionará o endereço IP, o

substituirá pelo endereço do roteador e roteará o datagrama utilizando o endereço que obteve na lista.

4.17 – Opções de Indicação de Hora

A opção de indicação de hora funciona de forma semelhante à opção de armazenamento de rota, porque a opção de indicação de hora contém uma lista inicialmente vazia e cada roteador do caminho, da origem até o destino, preenche um item da lista. Cada entrada da lista contém dois itens de 32 bits: o endereço IP do roteador que forneceu a entrada e uma indicação do número inteiro de 32 bits que indica a hora em que o datagrama foi processado. A figura 12 mostra o formato da opção de indicação de hora.
Na figura, os campos COMPRIMENTO e PONTEIRO são usados para especificar o comprimento do espaço reservado para a opção e a localização do próximo slot não utilizado (exatamente como na opção de armazenamento de rota). O campo SOBRECARGA de quatro bits contém um contador (número inteiro) de roteadores que não poderiam fornecer uma indicação de hora, porque a opção era muito pequena.
O valor no campo FLAGS de quatro bits controla o formato exato da opção e diz como os roteadores devem fornecer indicações de hora. A tabela da figura 14 mostra os valores aceitos.
As indicações de hora fornecem a hora e a data em que um roteador trata o datagrama, expressas em milissegundos deste a meia-noite, Hora Universal. Se a representação padrão para hora não estiver disponível, o roteador poderá utilizar qualquer representação de hora local, dede que ative o bit de mais alta ordem no campo de indicação de hora. Naturalmente, as indicações de hora transmitidas por computadores autônomos nem sempre são consistentes, mesmo se representadas em hora universal; cada máquina informa a hora de acordo com seu relógio local e os relógios podem ser diferentes.

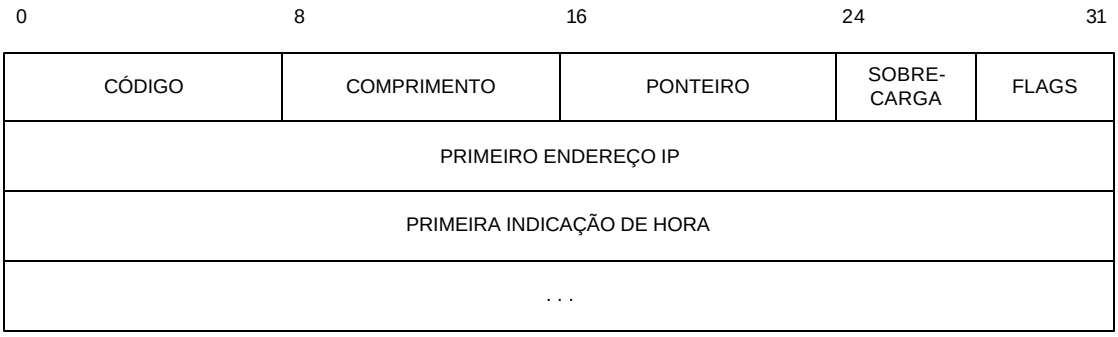


Figura 13

Pode parecer estranho que a opção de indicação de hora inclua um mecanismo para que os roteadores registrem seus endereços IP juntamente com indicações de hora, porque a opção de armazenamento de rota já oferece esse recurso.

Entretanto, o registro de endereço IP com indicações de hora elimina ambigüidade. Gravar a rota juntamente com indicações de hora também é útil, porque permite que o receptor saiba exatamente o caminho seguido pelo datagrama.

Valor de Flags	Significado
0	Somente indicações de hora de registro; omitem endereços IP
1	Precede cada indicação de hora por um endereço IP (este é o formato mostrado na figura 4.17).
3	Endereços IP são especificados pelo emissor; um roteador somente registra uma indicação de hora se o próximo endereço IP da lista coincidir com o endereço IP do roteador.

Figura 14

4.18 – Como Processar Opções Durante a Fragmentação

A idéia implícita no bit CÓPIA no campo de opção CÓDIGO agora deve estar clara. Quando fragmenta um datagrama, um roteador reproduz algumas opções IP em todos os fragmentos, enquanto coloca outros em apenas um fragmento. Considere, por exemplo, a opção utilizada para registrar a rota do datagrama. Dissemos que cada fragmento deverá ser tratado como um datagrama independente; assim, não há nenhuma garantia de que todos os fragmentos sigam o mesmo caminho até o destino. Se todos os fragmentos estivessem contidos na opção de rota de registro, o destino poderia receber uma lista diferente de rotas de cada fragmento. Não poderia ser produzida uma única lista significativa de rotas para o datagrama remontado. Assim, o padrão IP especifica que a opção de armazenamento da rota dever ser copiada somente para um dos fragmentos.

Nem todas as opções IP podem ser restritas a um fragmento. Considere a opção de rota de origem, por exemplo, que especifica como um datagrama deve ser transportado através da interligação em redes. As informações de roteamento de origem precisam ser reproduzidas em todos os cabeçalhos dos fragmentos, ou estes não seguirão a rota especificada. Assim, o campo de código para a rota de origem especifica que a opção precisa ser copiada para todos os fragmentos.

5 - Protocolo UDP

User Datagram Protocol

5 -1. Características Gerais do Protocolo UDP

O protocolo UDP é um protocolo de camada 4 (de transporte) no modelo OSI, que se caracteriza por ser mais simples que o TCP o outro protocolo da camada 4. Enquanto o TCP se preocupa com a conexão e a chegada correta dos dados no destino, o UDP por ser mais simples não tem a mesma preocupação, portanto, ele não verifica o recebimento dos dados pelo destino (também não possui o serviço de reenvio), não ordena as mensagens, ou seja, elas vão sendo agrupadas conforme vão chegando, não controla o fluxo de informações e não verifica a integridade dos dados para o destino. As possibilidades de o destino não receber os dados são várias como por exemplo: perder os dados, duplicar os dados ou agrupar de forma errada.

Essa simplicidade do UDP pode parecer, à primeira vista, um pouco estranha e provavelmente um leigo diria: se existe um protocolo como o TCP, que garante a chegada correta dos dados no destino, para que usarmos o UDP? A resposta é fácil, por que se o protocolo é simples, ele também é menor, então devemos ver isto como ganho de velocidade na transmissão e recepção de dados, algo que nos dias atuais se torna cada vez mais importante. É claro que em muitas das vezes o pacote pode não chegar ao destino, mas também devemos avaliar que só valerá a pena enviar pacotes utilizando o UDP quando este for pequeno, neste caso menor que 516 Bytes. Ou seja, não será em uma transferência de arquivos, como em um download usual, que será recomendado a utilização do UDP como “meio de transporte”.

Uma outra característica importante do UDP e neste ponto, semelhante ao TCP é que ele se baseia em portas para a troca de informações, desta forma, é atribuída uma porta ao destino e uma porta a origem.

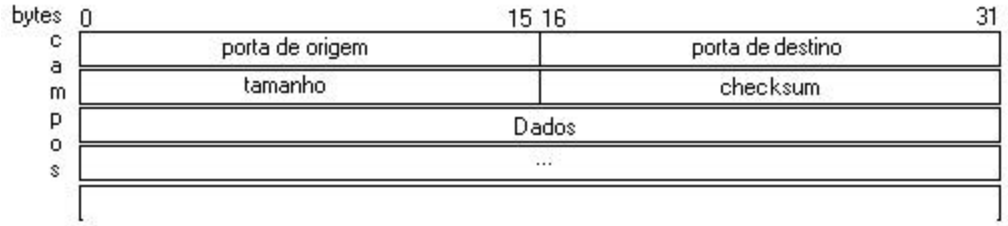


Figura 1 – Formato do Datagrama UDP

O formato do UDP divide o pacote em campos como mostrado na figura1.

- Os campos porta de origem e porta de destino, especificam que portas que serão utilizadas na comunicação.
- O campo tamanho descreve quantos bytes terá o pacote completo.

- O campo checksum é opcional e faz uma soma verificadora para garantir que os dados estarão livres de erros.

5 -2. Protocolos que Utilizam o UDP

Como dito anteriormente, o protocolo UDP é simples se comparado ao TCP, então somente alguns protocolos utilizam o UDP para transporte de dados que são: o TFTP (Trivial File Transfer Protocol), SNMP (Simple Network Management Protocol), DHCP (Dynamic Host Control Protocol), DNS (Domain Name Service).

5-2-1. TFTP

Este protocolo é semelhante ao FTP porém sem confirmação de recebimento pelo destino ou reenvio. É comumente usado por administradores de rede ao se fazer o download do IOS (Internetwork Operational System) de um roteador ou do arquivo de inicialização.

5-2-2. SNMP

É utilizado para configurar dispositivos como switches ou roteadores e permite que estes enviem o seu status. O problema é que os hackers, utilizam este protocolo para obter informações sobre o sistema, como as tabelas de roteamento. As últimas versões do SNMP podem fazer criptografia md5, porém a maioria ainda usa versões antigas que passa o password em formato de texto.

5-2-3. DHCP

É utilizado em redes que sofrem constantes alterações na topologia e o administrador não pode verificar o IP (Internet Protocol) de cada máquina devido a enorme quantidade, então o roteador distribui IPs automaticamente para as estações. Como esta atribuição é feita com a utilização do UDP, caso haja algum problema o usuário terá que pedir o reenvio ou reiniciar a máquina. O único problema técnico deste protocolo é que como os IPs são atribuídos aleatoriamente, fica mais difícil para o administrador ter controle sobre o que cada host está fazendo.

5-2-4. DNS

Um tradutor dos nomes na rede, na qual cada IP pode ser correspondido com um nome. Neste caso, imaginemos que um usuário esteja acessando a internet e deseja ir para outra página. Ele digita o endereço no campo apropriado e entra. Se a página, por acaso, não abrir por não ter reconhecido o endereço, o problema poderá ter sido no envio ou resposta do servidor de nomes utilizando o UDP, e então o usuário tentará de novo acessar a página e provavelmente conseguirá. Agora, imagine que isto fosse feito com o TCP, provavelmente esta falha não ocorreria, porém o tempo gasto para o computador saber qual IP se refere àquele nome seria inimaginável para as necessidades atuais.

5 -3. Vulnerabilidades do Protocolo UDP

O protocolo UDP assim como o TCP utiliza portas para a comunicação. Um dos primeiros passos que um hacker pode explorar é fazer a verificação das portas que estão abertas para comunicação. Caso, o administrador da rede queira impedir a invasão no sistema ele poderá fechar as portas mais isso significa ficar sem comunicação, então esta não é a melhor solução.

5-3-1 Alguns Tipos de Ataques

IP Spoofing: Um dos ataques utilizados por hackers, chamado IP Spoofing consiste em trocar o IP do host de origem por um outro qualquer. Isto normalmente é utilizado como proteção da identidade do hacker. Portanto, a máquina poderá achar que está recebendo dados de uma máquina, mas na verdade estará recebendo dados de outra.

SYN Flood: Para o UDP, ataque do tipo SYN Flood que junta IP Spoofing com Denial of Service (negação de serviço), pode ser um problema, pois o hacker pode lançar algum processo que sobrecarregue a máquina servidora.

Fraggle: O FRAGGLE é um ataque que utiliza o UDP como protocolo de transporte. Ele causa uma chuva ao enviar um pacote UDP para uma lista de endereços de broadcast.

New Teardrop: O New Teardrop diminui a parte de dados do pacote e o tamanho total do protocolo UDP é falsificado. Este ataque provoca o travamento das máquinas invadidas.

Worm W32.SQLExp.: O worm W32.SQLExp. é um worm que atinge os sistemas que estiverem executando o Microsoft SQL Servidor 2000 e também o Microsoft Desktop Engine (MSDE) 2000. O worm envia 376 bytes para a porta UDP, a porta de SQL Server Resolution Service.
O worm tem o efeito não pretendido de realizar um ataque de Negação de Serviço devido ao grande número de pacotes enviados. Para resolver o problema deste ataque é recomendável fazer a configuração de dispositivos de perímetro a fim de bloquear a entrada de trânsito UDP para a porta 1434 de hosts não confiáveis e o bloqueio da entrada de trânsito UDP de uma rede para a porta destino 1434. Quando o worm W32.SQLExp.Worm ataca um sistema vulnerável, ele faz o seguinte: envia uma cópia de si mesmo para o SQL Server Resolution Service, o qual monitora a porta 1434 do UDP. Explora uma vulnerabilidade de transbordo de buffer que permite que uma porção da memória do sistema seja sobrescrita. Ao fazer isso, ele é executado no mesmo contexto de segurança do que o serviço do Servidor SQL. Chama a função API do Windows, GetTickCount, e utiliza o resultado como fonte para gerar endereços IP aleatórios. Abre um soquete no computador infectado e tenta

enviar uma cópia de si mesmo repetidamente à porta 1434 do UDP nos endereços IP que gerou, usando uma porta de fonte temporária. Como o worm não ataca os hosts seletivamente na sub-rede, o resultado é uma grande quantidade de trânsito.

5-3-2 Fragilidades

O serviço NFS (Network File System – faz o compartilhamento de discos remotos), por exemplo, utiliza o UDP para a solicitação de criação, remoção e escrita de arquivos. Logo, um invasor, com IP falso, poderia ter acesso ao servidor, ler dados ou até mesmo alterá-los. O único modo de não permitir isso seria colocando um firewall na rede.
Outro tipo de ataque com UDP se dá ao protocolo RIP (Routing Information Protocol), utilizado por rotedores para detectar o melhor caminho na rede. Os pacotes RIP são pacotes UDP enviados para a porta 513. Portanto, o problema principal, o qual é aproveitado pelos hackers, é que as informações não são checadas. Um atacante pode, então, enviar informações falsas e maliciosas de roteamento para os outros roteadores, de tal forma que possa fingir que é uma outra máquina, pretensamente confiável, ou até mesmo para gravar todos os pacotes enviados para uma determinada máquina.

5-4. Conclusão

Portanto, vimos que o protocolo UDP apesar de simples e não confiável é bastante útil em pequenos serviços como: TFTP, SNMP, DHCP, DNS.
Outro ponto importante, é suas vulnerabilidades em que o invasor normalmente utiliza o recurso de falsificar o IP, para não ser reconhecido. Ele também pode utilizar o UDP, para que não haja a conexão entre computadores, e invadir computadores em que esteja sendo usado serviços que necessitem deste protocolo.

6 – Protocolo IGMP

Internet Group Management Protocol

6 –1 - Introdução

A entrega *Multicast* IP é seletiva: apenas estações interessadas podem receber tráfego dirigido a um dado grupo. Almejando implementar essas árvores de distribuição seletiva, que apenas atingem os membros do grupo, torna-se básico que os membros devem informar os *routers* onde estão, e que grupo(s) lhe(s) interessa(m).
Como é sabido, os grupos *multicast* são dinâmicos, a constituição dos grupos é variável, o *status* de cada grupo deve ser conhecido por quem tem de entregar os pacotes *multicast*: os *routers*.

O IGMP - *Internet Group Management Protocol*, permite às estações agregarem-se e abandonar grupos *multicast*. Enviando um relatório de associação ou parceria (*membership report*) ao *router* de vizinhança imediata, uma estação informa o *router* que deseja fazer parte de um grupo *multicast*. Os *routers* transmitem periodicamente mensagens com interrogações de parceria (*membership query*) para determinar quais os "host groups" que têm membros nas suas redes diretamente conectadas.
Um *host* responde com um *membership report* para cada grupo ao qual pertence. Para limitar o número *membership reports*, cada estação inicia uma espera de tempo aleatório depois de ter recebido o *membership query*.
As estações "vasculham" o meio tomando conhecimento dos relatórios de parceria enviados ao *router*; se um relatório é submetido para o grupo ao qual a estação pertence o seu tempo de espera expira, e cancela o seu relatório para o grupo. Este mecanismo assegura apenas um *membership report* é gerado por cada grupo. Baseado nas informações das constituições dos grupos fornecidas através do IGMP, os *routers* estão capacitados para determinar que tráfego *multicast* (se houver algum) se deve encaminhar para as redes interligadas.

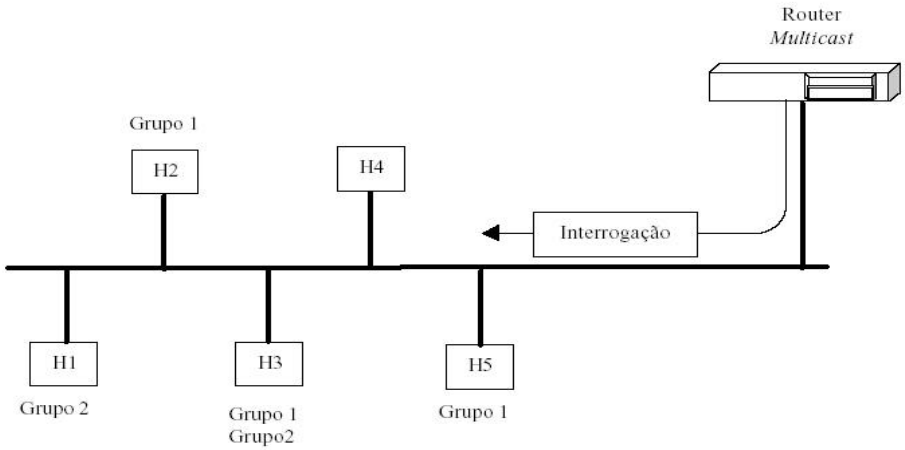


Figura 1 - *Internet Group Management Protocol* - Mensagem de interrogação.

Quando o software aplicacional pede ao software de rede da estação para esta se juntar a um grupo *multicast*, uma mensagem IGMP é enviada ao *router* mais próximo (se o *host* não for já um membro do grupo). Ao mesmo tempo, o endereço *multicast* de classe D do grupo ao qual se junta é mapeado como um endereço de baixo nível e a interface da rede é programada para aceitar pacotes para esse endereço.
Por exemplo, se uma estação passa a integrar um grupo num interface *Ethernet*, os 23 bits mais baixos do endereço de classe D são mapeados aos 23 bits mais baixo do endereço *Ethernet*. Devido a esta filtragem de endereços *multicast* por hardware, um *router* não necessita manter uma lista detalhada das estações que pertencem a cada endereço de grupo, mas apenas esse membro, pelo menos, do grupo, está presente na sub-rede à qual se encontra vinculado.

IGMP v1

Uma das fraquezas da primeira versão do IGMP era a latência elevada associada com o término de sessões *multicast*. Depois do último membro de um grupo *multicast* numa sub-rede ter abandonado o grupo, os outros *routers* não são imediatamente notificados para deter a propagação de tráfego para o grupo. Esta demora era causada pelo IGMP esperando até que várias interrogações indicassem que não restavam membros na sub-rede, de um grupo em particular. No entanto, indesejavelmente, tráfego desnecessário seria encaminhado para a sub-rede. O custo deste envio inútil podia ser elevado, particularmente num segmento da Internet com largura de banda constrangida.

IGMP v2

A versão 2 do IGMP, apresenta alguns refinamentos que ajudaram a reduzir o *overhead* do protocolo. As mensagens de interrogação dirigidas a grupos específicos (*Group Specific Query Message*) permitem ao *router* interrogar

grupos específicos nas redes onde estão diretamente vinculados em vez de serem forçados a interrogar todos os grupos indiscriminadamente. Começando com a versão 2, o término de uma sessão *multicast* já não é feito de forma passiva. O último *host* de uma sub-rede a deixar o grupo *multicast*, transmite uma mensagem de saída de grupo (*Leave Group*) ao *router* na qual é indicado qual o grupo abandonado. Depois de verificar a partida com uma mensagem de interrogação dirigida a esse grupo específico, o *router* notifica outros *routers* para cessarem o encaminhamento de tráfego para a sub-rede dirigido ao grupo.

IGMP v3

A versão 3 do IGMP vai mais longe na redução do *overhead*. A largura de banda será conservada pela mensagem *Group-Source Report* que permitirá às estações receber tráfego de fontes específicas de um grupo *multicast*. Em versões prévias do IGMP, o tráfego de todas as fontes tinha de ser encaminhado para uma sub-rede mesmo se as estações estivessem apenas interessadas em receber tráfego de fontes específicas. As mensagens *Leave-Group* apresentadas em primeira instância pela versão 2 foram também aperfeiçoadas para permitir às estações largar um grupo inteiro ou para especificar a fonte a que queriam renunciar.
Levando-se em conta que as versões recentes do IGMP podem reduzir o tráfego desnecessário, otimizando a utilização deste protocolo, deve ser favorecida a sua utilização em detrimento das anteriores.
Pelos métodos acima mencionados, os *routers multicast* estão habilitados a manter, por interface, uma tabela atualizada contendo os grupos cujo tráfego tem interesse para a sub-rede pela qual, após a recepção de pacotes *multicast*, os *mrouters* sabem para que interfaces os pacotes devem ser encaminhados.

6 – 2 - Possibilidades de erros

- Como não existe nenhum mecanismo de alocação de endereço multicast pode ocorrer de existirem dois grupos distintos utilizando o mesmo endereço. A única solução para este fato é confiar na probabilidade baixíssima de que dois grupos sejam formados no mesmo instante, com o mesmo endereço em locais próximos.
- Como um host vai descobrir o endereço para onde estão sendo enviados os pacotes de um grupo, já que os endereços são alocados dinamicamente? A solução é implementar um mecanismo onde as sessões multicast sejam anunciadas e esta informação seja espalhada pela Internet.
- Nem todos os roteadores IP dão suporte a multicast Com isso, se um roteador que não suporta multicast receber um pacote com o endereço destino da classe D, ele não encontrará um caminho para enviar este pacote e então será descartado.

6 – 3 - Observações importantes

- Observar que quando um pacote destinado a um endereço multicast chega em uma LAN, o tratamento dado a ele é o mesmo que no caso de um pacote com endereço broadcast. Porém, ao invés de todos os usuários lerem aquele único pacote que está circulando na rede, somente as estações que desejarem recebe-lo o armazenarão (figura 2). Com isso È possÌvel observar que a economia não é realizada somente na subrede de comunicação, mas sim também na rede local porque se não existisse este mecanismo, vários pacotes com a mesma informação deveriam ser replicados para cada estação que desejasse recebe-lo.

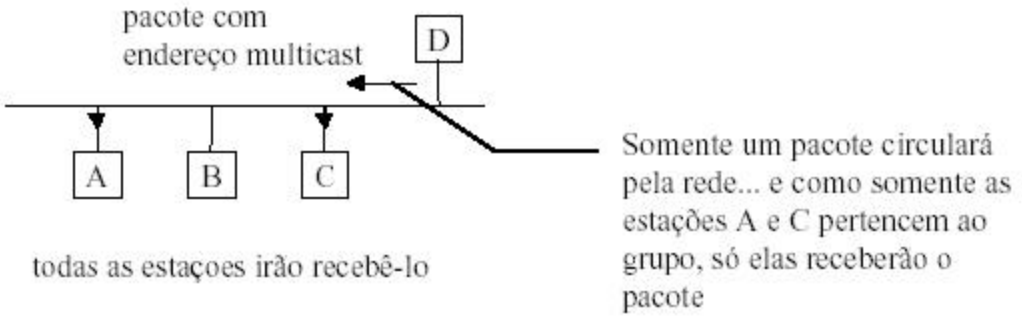


Figura 2 – Como é a circulação de um pacote multicast pela rede local

6 – 4 - Algumas redes que oferecem suporte a igmp (multicast)

Mundo Unix

Atualmente a maioria das máquinas Unix já saem de fábrica com este suporte (o IGMP é incorporado ao Kernel). Caso à máquina seja antiga, é provável que não exista este suporte, portanto será necessário sua requisição diretamente ao representante ou a aquisição deste arquivo em qualquer site FTP que ofereça este programa.

Windows / Dos

Também existe suporte para multicast disponível para sistemas Windows da Microsoft. Os sistemas Windows 95 e Windows NT já vêm com suporte para IGMP.
Informações adicionais podem ser obtidas no site da Microsoft.

6 – 5 - Segurança

Uma técnica para configuração de roteadores para admitir somente pacotes úteis a rede é chamada de filtragem. Estes filtros são utilizados pelos chamados firewalls que ficam entre sua rede e a Internet global.
Com os filtros, serão aceitos somente pacotes vindo de determinados destinos (um conjunto de determinados endereços IP).
No nosso caso, serão aceitos somente IGMP para pacotes de atualização de roteamento ou de participação em grupos.
Esses pacotes IGMP para consulta (query) e relatório (report), como foi descrito anteriormente não saem da rede local. Porém as mensagens de atualização de roteamento IGMP DVMRP(Distance Vector Multicast Routing Protocol) são trocadas por túneis vizinhos. Mensagens trocadas por mrouters através de redes que oferecem suporte para multicast são enviados para o endereço 224.0.0.4, enquanto que entre vizinhos conectados por túnel, o pacote trocado é um pacote normal (unicast). As atualizações DVMRP são enviados com o TTL igual a 1, garantindo que o pacote não sairá da rede local. Isto faz com que somente as mensagens IGMP DVMRP passem pela fronteira do firewall.

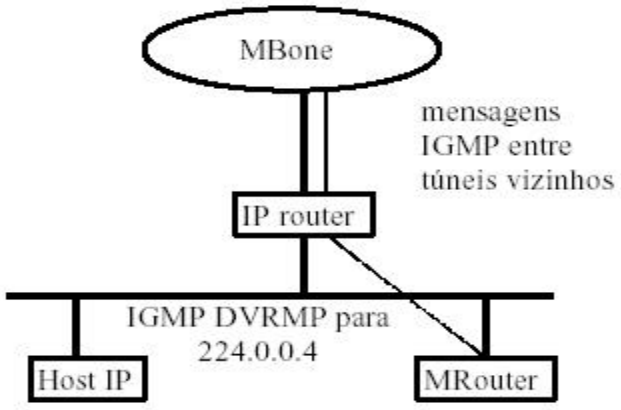


Figura 3 – circulação de mensagem Mbone

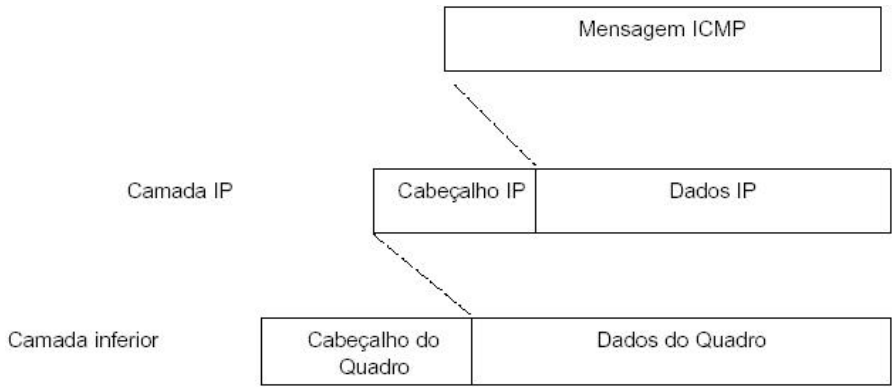
- Para monitorarmos o tráfego IGMP na rede, bastaria digitar:
  %tcpdump ip proto igmp
- As estatísticas IGMP podem ser obtidas pelo comando “netstat –p igmp”.

7– Protocolo ICMP

Internet Control Message Protocol

O protocolo ICMP é usado no transporte de mensagens de erro e de controle entre gateways e hosts.

Na figura abaixo vemos o cabeçalho ICMP dentro do pacote

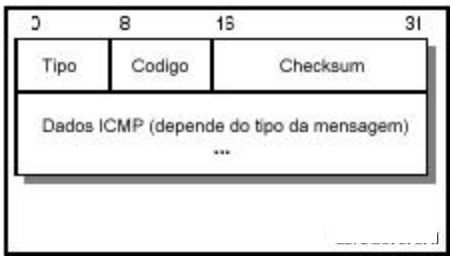


O protocolo ICMP possui as seguintes características:

- ICMP usa o IP como se ele fosse um protocolo de maior nível (isto é, mensagens ICMP são encapsuladas em datagramas IP). Todavia, ICMP é uma parte integral do IP e deve ser implementado por cada módulo IP.
- ICMP é usado para informar erros e não para tornar o IP um protocolo confiável. Datagramas continuarão sendo perdidos sem gerar qualquer aviso.
- ICMP não pode ser usado para alertar erros de mensagens ICMP. Isto evita repetições infinitas.
- Para datagramas fragmentados, mensagens ICMP são enviadas para relatar erros do primeiro fragmento.
- Mensagens ICMP nunca são enviadas em resposta a datagramas com endereço de destino broadcast ou multicast.
- Mensagens ICMP nunca são enviadas em resposta a datagramas que não possuam um endereço que representa um único host (Isto é, o endereço de origem não pode ser um endereço loopback, broadcast ou multicast).

Mensagens ICMP

Mensagens ICMP são enviadas em datagramas IP. O cabeçalho IP possui o número do protocolo igual a 1 (ICMP) e o tipo de serviço igual a zero. O campo de dados do datagrama IP contém a mensagem ICMP como é exibido na figura abaixo.



É o campo **tipo** que identifica uma mensagem ICMP particular, conforme veremos a seguir.

O campo **código** é usado na especificação dos parâmetros da mensagem.

O campo **checksum** (*soma de verificação*) corresponde ao verificador de erro, calculado a partir da mensagem ICMP completa.

O campo **Dados ICMP** tipicamente contém uma parte da mensagem IP que deu origem à mensagem ICMP.

Nos casos em que a mensagem ICMP se referir a um datagrama anteriormente enviado, ela transporta também o cabeçalho completo daquele datagrama IP, para facilitar ao emissor a identificação do datagrama problemático.

A mensagem ICMP pode ser dos seguintes tipos:

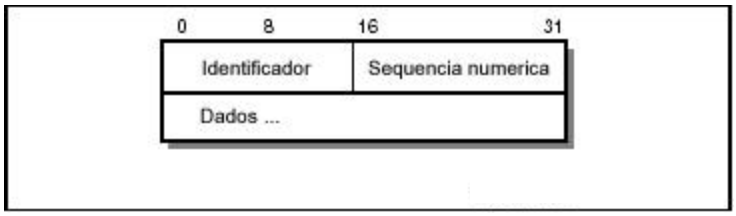
- 0: Resposta ao eco
- 3: Destino inacessível
- 4: Ajuste de fonte
- 5: Redireção
- 8: Eco
- 9: Aviso do roteador
- 10: Solicitação do roteador
- 11: Tempo excedido
- 12: Problema de parâmetro
- 13: Solicitação de marca de tempo
- 14: Resposta à marca de tempo
- 15: Solicitação de informações
- 16: Resposta de informações
- 17: Solicitação de máscara de endereço
- 18: Resposta de máscara de endereço
- 30: Traceroute
- 31: Erro na conversão de datagrama
- 32: redireção de host móvel
- 33: Ipv6 Where-Are-You (Cadê você?)
- 34: Ipv6 I-Am-Here (Eu estou aqui)

- 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

Alguns tipos estão descritos abaixo:

Eco (8) e Resposta ao eco (0)

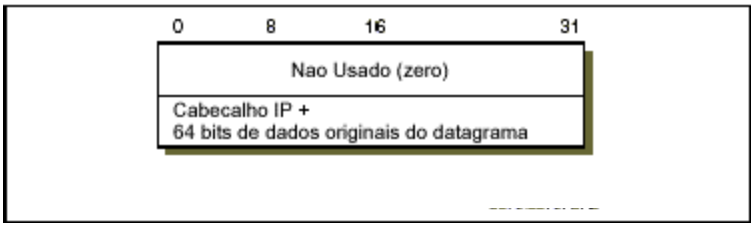
Eco é usado para detectar se um host está ativo na rede. Ele é usado pelo comando ping. O emissor inicializa os campos identificador, seqüência numérica e dados. O datagrama é então enviado ao host de destino. O receptor desta mensagem muda o tipo da mesma para resposta ao eco (0) e envia a mensagem de volta para o emissor.



Destino inacessível (3)

Esta mensagem é enviada por um roteador intermediário, informando a quem enviou a mensagem que o endereço IP de destino está inacessível.

Se esta mensagem for enviada por um host, significa que o protocolo especificado no campo número do protocolo do datagrama de origem não está ativo ou a porta não está ativa.



O campo código do cabeçalho ICMP terá um dos seguintes valores:

- 0: Rede inacessível

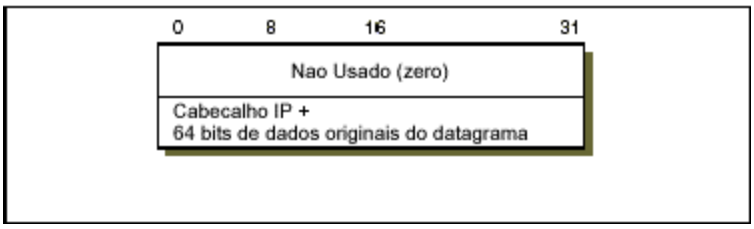
- 1: Host inacessível
- 2: Protocolo inacessível
- 3: Porta inacessível
- 4: Fragmentação necessária mas o bit não fragmente foi ativado
- 5: Rota fonte falhou
- 6: Rede de destino desconhecida
- 7: Host de destino desconhecido
- 8: Host fonte isolado (obsoleto)
- 9: Rede de destino administrativamente proibida
- 10: Host de destino administrativamente proibido
- 11: Rede inacessível para este tipo de serviço
- 12: Host inacessível para este tipo de serviço
- 13: Comunicação administrativamente proibida para este tipo de serviço
- 14: Violação de precedência de host
- 15: Preferência efetivamente retirada

Ajuste de fonte (4)

Se esta mensagem for enviada por um roteador intermediário, significa que o roteador não possui buffer suficiente para armazenar o datagrama.

Se esta mensagem for enviada por um host, significa que os datagramas estão chegando rápido demais para ele processar.

O campo código do cabeçalho ICMP é sempre zero

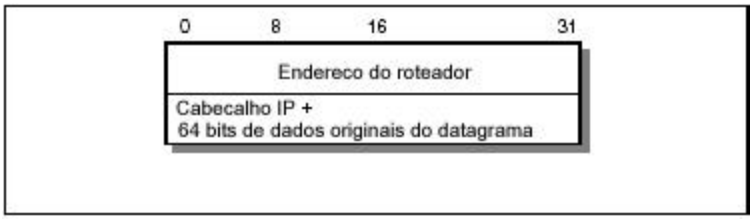


Redireção (5)

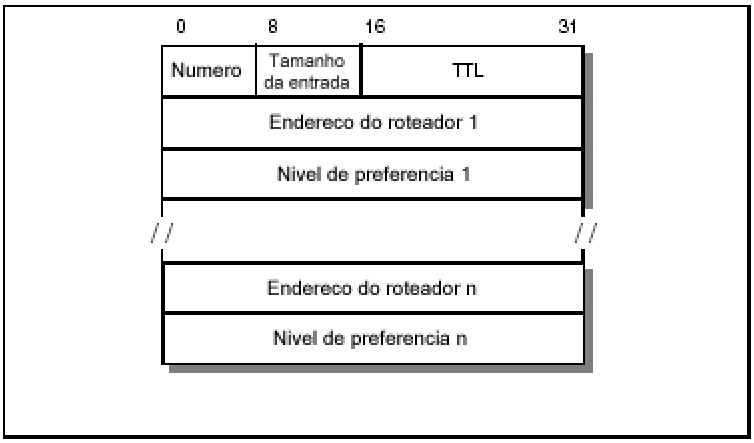
Se esta mensagem for enviada por um roteador intermediário, significa que o host origem deve enviar os próximos datagramas para a rede do roteador cujo endereço IP é o especificado na mensagem ICMP. Este roteador preferencial estará sempre na mesma sub-rede do host que enviou o datagrama e do roteador que retornou o datagrama IP. O roteador enviará o datagrama para seu próximo nó. Esta mensagem não será enviada se o datagrama IP possuir uma rota fonte.

O campo código do cabeçalho ICMP terá um dos seguintes valores:

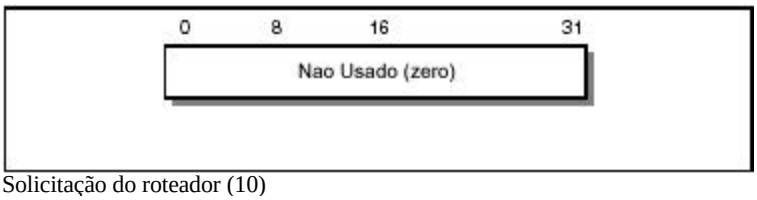
- 0: Redireção de rede
- 1: Redireção de host
- 2: Redireção de rede para este tipo de serviço
- 3: Redireção de host para este tipo de serviço



Aviso do roteador (9) e Solicitação do roteador (10)



Aviso do roteador (9)



Solicitação do roteador (10)

Onde:

- Número: O número de entradas da mensagem
- Tamanho da entrada: O tamanho da entrada em unidades de 32 bits. Esta é 2 (32 bits para o endereço IP e 32 bits para o valor da preferência).
- TTL: O número de segundos que uma entrada será considerada válida.
- Endereço do roteador: Um dos endereços IP enviados
- Nível de preferência: É um número de 32 bits indicando a preferência a ser atribuída a um roteador quando for selecionar o roteador padrão. Cada roteador na sub-rede é responsável por avisar seu nível de

preferência. A preferência cresce com este número e decresce com o mesmo. O valor padrão é zero, que está no meio da faixa possível. O valor X'80000000' (-231) indica que o roteador nunca deve ser usado como padrão.

O campo código do cabeçalho ICMP é zero para ambas as mensagens.

Estas duas mensagens são usadas se um host ou um roteador suporta o protocolo de descoberta de roteador. Roteadores periodicamente avisam seu IP naquelas sub-redes onde eles foram configurados para tal. Avisos são feitos no endereço all-systems multicast (224.0.0.1) ou o endereço broadcast (255.255.255.255). O comportamento padrão é enviar avisos a cada 10 minutos com o valor TTL igual a 1800 (30 minutos). Roteadores também respondem a mensagens de solicitação que eles recebem. Eles podem responder diretamente para o host que enviou a mensagem ou esperarem por um pequeno intervalo e responder com um multicast.

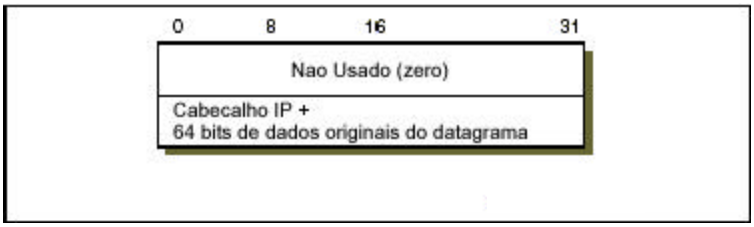
Hosts podem enviar mensagens de solicitação. Estas mensagens são enviadas usando o endereço all-routers multicast (224.0.0.2) ou o endereço broadcast (255.255.255.255). Tipicamente 3 mensagens de solicitação são enviadas em intervalos de 3 segundos. Alternativamente um host pode esperar por avisos periódicos. Cada vez que um host recebe um aviso com um nível de preferência maior, ele atualiza seu roteador padrão. O host também ajusta o marcador de tempo (TTL) da nova entrada com o valor que veio no aviso. Quando o host recebe um novo aviso do seu atual roteador padrão, ele muda o valor do TTL para o contido no novo aviso.

Este processo também proporciona um mecanismo para os roteadores se declararem não disponíveis, bastando somente enviar um aviso com o valor do TTL igual a zero.

Tempo Excedido (11)

Se esta mensagem for envidada por um roteador intermediário, isto significa que o campo TTL do datagrama IP expirou.
Se esta mensagem foi envidada pelo host de destino, significada que o tempo de remontagem de fragmentos IP expirou enquanto o host estava esperando por um fragmento do datagrama. O campo código do cabeçalho ICMP pode conter um dos seguintes valores:

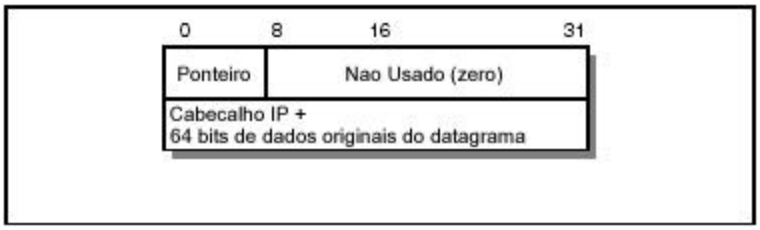
- 0: TTL excedido no transito
- 1: TTL excedido na remontagem



Problema de parâmetro (12)

Esta mensagem indica que um problema foi encontrado durante o processamento dos parâmetros do cabeçalho IP. O campo ponteiro indica o octeto do datagrama IP original onde o problema foi encontrado. O campo código do cabeçalho ICMP pode conter um dos seguintes valores:

- 0: Erro não especificado
- 1: Opção requerida ausente

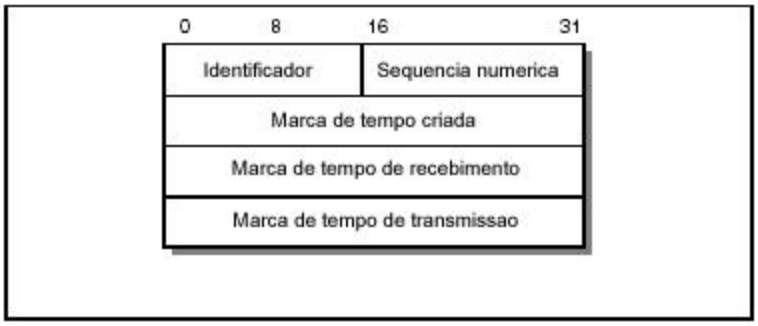


Solicitação de marca de tempo (13) e Resposta à marca de tempo (14)

Estas duas mensagens são usadas na solução de alguns problemas e na medição de performance.

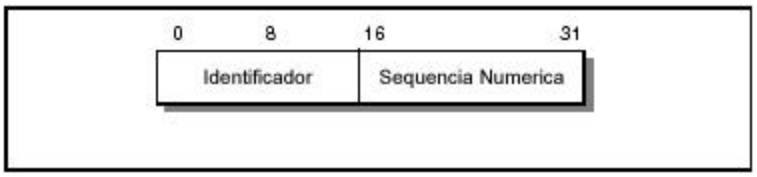
O emissor inicializa o identificador e a seqüência numérica (que é usada quando múltiplas solicitações de marcas de tempo são feitas), cria uma marca de tempo e envia a mensagem (tipo 13) ao host de destino, o host de destino por sua vez preenche as marcas de tempo de recebimento e transito, muda o tipo da mensagem para (tipo 14) e retorna a mensagem para o host de origem. O datagrama possui duas marcas de tempo se há um intervalo perceptível entre o tempo de recebimento e o tempo de transmissão. Na prática a maioria das implementações executa as 2 operações (recebimento e resposta) numa única operação. Neste caso ajustam-se as 2 marcas de tempo para o mesmo valor.

OBS.: Marca de tempo é o tempo decorrido desde a meia-noite UT (GMT)



Solicitação de informações (15) e Resposta de informações (16)

Uma solicitação de informação é emitida por um host para obter um endereço IP de uma rede anexa. O emissor coloca na solicitação o endereço IP de destino no cabeçalho IP ajustado com zero (significando esta rede) e espera por uma resposta de um servidor autorizado a assinar endereços IP para outros hosts. O campo código do cabeçalho ICMP é zero. A resposta contém endereços de rede IP em ambos os campos fonte e destino do cabeçalho IP. Atualmente este mecanismo é obsoleto.



Solicitação de máscara de endereço (17) e Resposta de máscara de endereço (18)

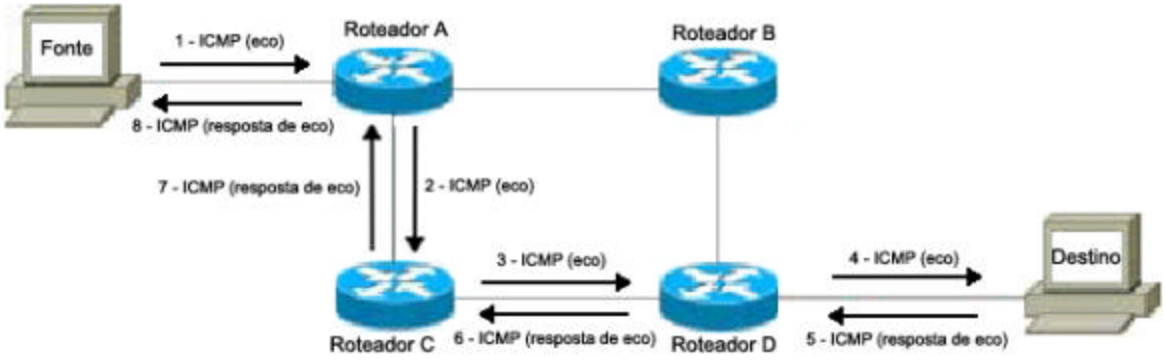
Uma solicitação de máscara de endereço é usada por um host para determinar a máscara de sub-rede usada em uma rede anexa. A maioria dos hosts são configurados com sua(s) máscara(s) de sub-rede. Entretanto algumas (como por exemplo, workstations sem disco) devem obter esta informação de um servidor. Para obter uma máscara de sub-rede, o host manda uma mensagem (tipo 17) usando endereço broadcast. Qualquer host na rede que foi configurado para responder a mensagem (tipo 17), irá colocar na mensagem a máscara de sub-rede, converter o pacote para mensagem (tipo 18) e retornará a mesma para o emissor. O campo código do cabeçalho ICMP é zero.

Aplicações ICMP

Existem dois aplicativos muitos conhecidos que são baseados no protocolo ICMP, o Ping e o Traceroute.

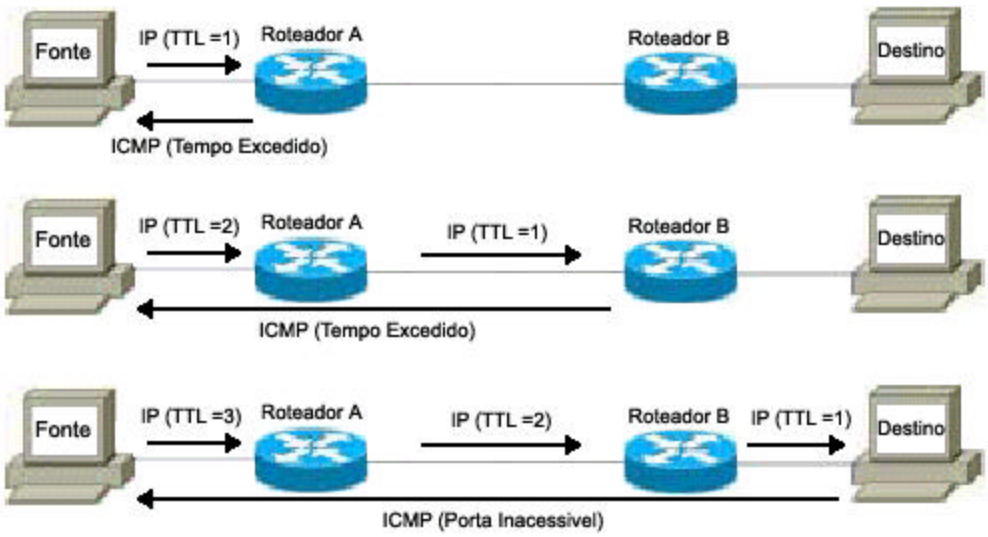
Ping

O ping usa as mensagens ICMP Eco (8) e Resposta de eco (0) para verificar a conectividade entre dois hosts em uma rede. Além disso, o ping mede o tempo de atraso entre o pacote ICMP enviado e o recebido, nos dando uma idéia de como a velocidade da rede está entre o computador local e o remoto.
Para fazer isso, ele envia a mensagem ICMP eco (8) para o host em que se deseja testar. Este host recebendo a mensagem, troca o tipo da mesma para (0) e envia de volta ao emissor.



Traceroute

O traceroute é usado para determinar a rota que os datagramas IP seguem para atingir um destino. Ele usa o ICMP e o UDP para isso.
Ele envia 3 pacotes UDP com a porta de destino fora da faixa padrão, inicialmente com TTL igual a um (1). Quando passar pelo primeiro roteador, torna-se zero e uma mensagem ICMP tempo excedido (tipo 11) retornará. Com isso teremos informações sobre o primeiro roteador no meio do caminho. Depois o TTL é aumentado para dois (2) e novamente são enviados 3 pacotes UDP, porém a mensagem ICMP ocorrerá somente no segundo roteador. O processo se repete até alçar o destino onde não mais retornará a mensagem ICMP de tempo excedido e sim uma mensagem ICMP porta inacessível (tipo 3 código 3).



8 – Protocolo TCP

Transmission Control Protocol

O protocolo especifica o formato dos dados e das confirmações que os dois computadores trocam para oferecer uma transferência confiável e, também, os procedimentos de que se valem os computadores para assegurar que os dados cheguem corretamente.

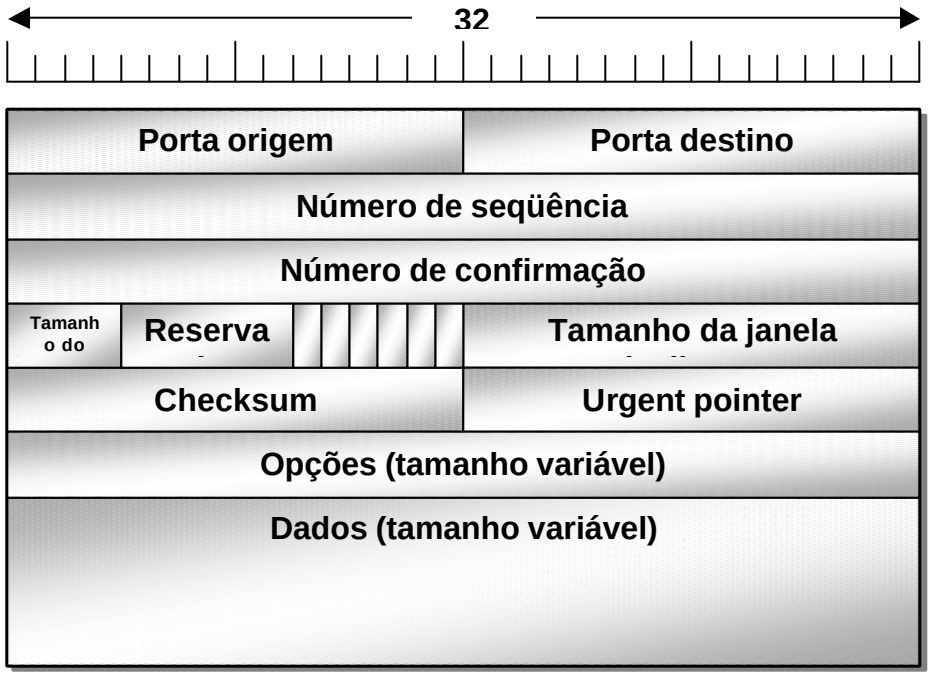
No caso do TCP ele apresenta como principais características transferência de dados confiável fim-a-fim (todo pacote transmitido requer um Ack que é um bit de reconhecimento, há recuperação de dados perdidos, descarte de dados duplicados e reorganização dos dados recebidos fora de ordem), comunicação bidirecional (full-duplex) entre cliente servidor, o sequenciamento: bytes de segmentos são numerados, de forma a garantir a entrega em ordem e a deteção e eliminação de duplicatas e é voltado para atuar sobre redes heterogêneas com tamanhos máximos de pacotes variáveis, faixas de passagem variáveis topologias distintas. O ponto fraco atual deste padrão é a adaptação a taxas de erros grandes, comum em comunicação sem fio (wireless).

Este protocolo divide o processo de comunicação em três fases o início onde o cliente envia segmento tipo SYN (pedido de conexão, com número inicial da seqüência de numeração de bytes no sentido cliente - servidor), o servidor reconhece pedido de conexão enviando segmento tipo SYN com bit de

reconhecimento (ACK) ligado e com número inicial de seqüência de numeração no sentido servidor - cliente.e o destino envia segmento ACK reconhecendo SYN do servidor, a troca de dados onde efetivamente ocorre a transferência de dados e o encerramento da conexão que pode ser iniciada tanto pelo cliente como pelo servidor e origem envia segmento FIN, o destino envia reconhecimento: ACK e algum tempo depois a destino envia FIN (sinalizando fim da conexão) e por fim origem envia reconhecimento

Quando cada segmento for transmitido é adicionado um *checksum* e quando estes são recebidos eles são verificados, se danificados os pacotes são descartados, como se tivessem se perdido pela rede e retransmitidos pela origem.

O cabeçalho TCP é representado desta maneira:



Onde:

Porta origem e porta destino –Identificam os pontos terminais da conexão: processos ou threads

Número de seqüência –Identifica a posição deste segmento no fluxo de dados e cada conexão possui um fluxo de dados particular.

Número de confirmação –Utilizado para confirmar o recebimento de segmentos enviados anteriormente e especifica o próximo segmento aguardado

Tamanho do cabeçalho –Tamanho do cabeçalho TCP (números de palavras de 32 bits)

URG –Seu valor é igual a 1 se houver informação no campo Urgent Pointer

ACK –Se seu valor for 1: indica que o segmento é parte de uma conversação e que o valor do campo Acknoledgement number é válido, se seu valor for 0 e o flag SYN for 1: indica que o segmento é uma solicitação de conexão

PSH –Campo usado pelo remetente do segmento para indicar ao receptor que o segmento em questão deve ser entregue imediatamente ao nível superior

RST –Utilizado para reiniciar uma conexão que tenha ficado confusa devido a uma falha na estação ou por qualquer outra razão

SYN –Usado em conjunto com o ACK para solicitar ou aceitar uma conexão

–SYN=1 ACK=0: requisição de conexão
–SYN=1 ACK=1: conexão aceita
–SYN=0 ACK=1: “confirmação do recebimento”

FIN –Usado para encerrar uma conexão e indica que o transmissor não tem mais dados para enviar

Tamanho da janela deslizante - Indica o tamanho (disponível) do buffer do receptor e é usado pelo receptor para indicar ao transmissor que diminua o fluxo de transmissão de dados

Checksum –Verificação de erros

Urgent pointer –Usado pela origem para indicar onde se encontra algum dado urgente dentro do segmento

Opções –Campo para configuração de opções

Dados –Dados das aplicações

O protocolo tcp é fundamental para as comunicações da internet desde os primórdios da rede, pois a maior parte dos protocolos de aplicação necessita de transmissões confiáveis. Como conseqüência, a maioria dos protocolos de aplicação são implementados sobre TCP e não UDP e alguns poucos protocolos são implementados diretamente sobre IP.

Sua importância pode ser constatada através de fatos como um sistema de “peso” como o Unix já vir com ele implementado no núcleo do SO (assim como UDP).

9 – Protocolo SNMP

Simple Network Management Protocol

Visão geral:

O SNMP é um protocolo desenvolvido para gerência de rede, através de objetos ocorrências podem ser contabilizadas.

A contabilização do SNMP pode ser realizada de forma direta ou indireta.

A forma direta é quando o sistema operacional ou software manipula diretamente a variável, a forma indireta ocorre quando a variável é manipulada indiretamente através de uma outra referência.

Para exemplificar o modo direto podemos imaginar a contagem de frames ou datagramas que entram ou saem via interface do equipamento em estudo.

Esse exemplo foi retirado do Xinu[1] o arquivo net_in.c:

```
#include <conf.h>
#include <kernel.h>
#include <network.h>
#include <ospf.h>
int arp_in(struct netif *, struct ep *);
int rarp_in(struct netif *, struct ep *);
int ip_in(struct netif *, struct ep *);
 /*---------------------------------------------------------------------
* ni_in - network interface input function
*--------------------------------------------------------------------- */
int ni_in(struct netif *pni, struct ep *pep, unsigned len) {
  int rv;
  pep-&gt;
  ep_ifn = pni - &amp;
  nif[0];
  /* record originating intf # */
 pni-&gt;ni_ioctets += len;
 if (!memcmp(pni-&gt;ni_hwa.ha_addr, pep-&gt;
    ep_dst, EP_ALEN)) pni-&gt;
    ni_iucast++;
 else pni-&gt;
    ni_inucast++;
switch (pep-&gt;ep_type) {
    case EPT_ARP: rv = arp_in(pni, pep);
     break;
case EPT_RARP: rv = rarp_in(pni, pep);
     break;
case EPT_IP: rv = ip_in(pni, pep);
     break;
default: pni-&gt;
    ni_iunkproto++;
freebuf(pep); rv = OK;
}
   return rv;
}
```

Nesse caso as variáveis “ni_iunkproto” e “ni_iucast” contam quantos frames chegam destinados para protocolos desconhecidos e de unicast respectivamente.

Para acompanhar melhor, logo abaixo estão as variáveis usadas pelo driver da placa de rede no Xinux[1] na struct “netif” na sessão “MIB”:

```
/* Interface MIB */
        char        *ni_descr;      /* text description of hardware */
        int         ni_mtype;       /* MIB interface type           */
        long        ni_speed;       /* bits per second              */
        char        ni_admstate;    /* administrative status (NIS_*)*/
        long        ni_lastchange;  /* last state change (1/100 sec)*/
        long        ni_ioctets;     /* # of octets received         */
        long        ni_iucast;      /* # of unicast received        */
        long        ni_inucast;     /* # of non-unicast received    */
        long        ni_idiscard;    /* # dropped - output queue full*/
        long        ni_ierrors;     /* # input packet errors        */
        long        ni_iunkproto;   /* # in packets for unk. protos */
        long        ni_ooctets;     /* # of octets sent             */
        long        ni_oucast;      /* # of unicast sent            */
        long        ni_onucast;     /* # of non-unicast sent        */
        long        ni_odiscard;    /* # output packets discarded   */
        long        ni_oerrors;     /* # output packet errors       */
        long        ni_oqlen;       /* output queue length          */
        long        ni_maxreasm;    /* max datagram can reassemble  */
```

Para exemplificar a forma indireta podemos imaginar quando temos que informar o tempo que o sistema está ativo, ele subtrai a data e horário atual daquela que foi armazenado quando o sistema foi iniciado.

Para ilustrar melhor foi coletado um exemplo no Linux no arquivo “/proc/net/snmp”, nesse arquivo são colocadas informações sobre os frames, datagramas e pacotes que são tratados pelo kernel.

```
bash# cat /proc/net/snmp

Ip: Forwarding DefaultTTL InReceives InHdrErrors InAddrErrors ForwDatagrams
InUnknownProtos InDiscards InDelivers OutRequests OutDiscards OutNoRoutes
ReasmTimeout ReasmReqds ReasmOKs ReasmFails FragOKs FragFails
FragCreates
Ip: 1 64  38  0 0 0 0 0  20 38 0 0 0 0 0 0 0 0
Icmp: InMsgs InErrors InDestUnreachs InTimeExcds InParmProbs InSrcQuenchs
InRedirects InEchos InEchoReps InTimestamps InTimestampReps InAddrMasks
InAddrMaskReps OutMsgs OutErrors OutDestUnreachs OutTimeExcds
OutParmProbs OutSrcQuenchs OutRedirects OutEchos OutEchoReps
OutTimestamps OutTimestampReps OutAddrMasks OutAddrMaskReps
Icmp: 20 0 18 0 0 0 0  1 1  0 0 0 0  19 0 18 0 0 0 0 0  1  0 0 0 0 0
Tcp: RtoAlgorithm RtoMin RtoMax MaxConn ActiveOpens PassiveOpens
AttemptFails EstabResets CurrEstab InSegs OutSegs RetransSegs InErrs OutRsts
Tcp: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Udp: InDatagrams NoPorts InErrors OutDatagrams
Udp: 0 18 0 18
```

9.0 – Principais características do SNMP:

- Gerenciamento de redes;
- Controlar equipamentos de rede;
- Detectar problemas e erros;
- Usa o paradigma busca/armazenamento;
- Porta 161/UDP.

9.1 - Versões do SNMP:

- **SNMP versão 1:**

A recomendação existente para a versão um do SNMP é a RFC 1157, suas principais vantagens são a simplicidade, flexibilidade e popularidade.

Mas, a versão um possui algumas desvantagens como pouca preocupação com a segurança, com a mensagem SNMP é enviada a senha sem criptografia no campo “community”.

Para o acesso de busca costuma -se ter a senha “public” e para acesso de armazenamento costuma -se usar a senha “private”.

- **SNMP versão 2:**

Nessa versão existe maior preocupação com segurança na transmissão de senhas.

Foram adicionadas mensagens como: “GetBulkRequest” e “InformRequest/Response”.

A recomendação existente para a versão 1 do SNMP é a RFC 1448.

- **SNMP versão 3:**

Esse trouxe maior avanço em segurança.

O desenvolvimento dessa versão suporta algoritmos criptográficos na de senhas.

O único problema é que muitos equipamentos não suportam essa versão, sendo necessário realizar atualizações.

A recomendação existente para a versão 1 do SNMP é a RFC 3414.

9.2 - Vocabulário utilizado:

- **Gerente:**
  Aquele que guarda as informações e realiza pedidos no agente.

- **Agente:**
  Aquele que recebe a requisição do gerente e coleta os dados na "MIB".

- **MIB (Management Information Base):**
  É a coletânea dos objetos que se quer gerenciar de forma estruturada em uma árvore hierárquica.

- **ASN.1 (Abstract Syntax Natition):**
  Determina formas como as variáveis serão representadas.
  Garante interoperabilidade entre arquiteturas diferentes, evitando erros de representações.

- **SMI (Structure Management information):**
  Determina regras de como a MIB será estruturada (RFC 1155, RFC 1212, RFC 1215). Define o tipo de dados e regras e atributos de nomes.
  Para exemplificar melhor, logo abaixo está o arquivo "asn1.h" do Xinux[1]:

```
/* asn1.h - A1_SIGNED */

/* constants for parsing an SNMP packet, according to ASN.1          */
/* ASN.1 object types */
#define ASN1_SEQ  0x30      /* sequence object              */
#define ASN1_INT    0x02    /* integer                      */
#define ASN1_OCTSTR         0x04    /* octet string                     */
#define ASN1_NULL           0x05    /* null                             */
#define ASN1_OBJID          0x06    /* object identifier          */
#define ASN1_IPADDR         0x40    /* ip address                       */
#define ASN1_COUNTER        0x41    /* counter                    */
#define ASN1_GAUGE          0x42    /* gauge                      */
#define ASN1_TIMETICKS      0x43    /* time ticks                       */
/* Protocol Data Unit types -- SNMP specific                            */
#define PDU_GET             0xA0    /* get request                      */
#define PDU_GETN  0xA1      /* get-next request             */
#define PDU_RESP  0xA2      /* response                           */
#define PDU_SET             0xA3    /* set request                      */
#define PDU_TRAP  0xA4      /* trap message                 */
/* Constants used for conversion of objects to/from ASN.1 notation    */
#define CHAR_BITS 8                 /* number of bits per char    */
#define CHAR_HIBIT          0x80            /* octet with the high bit set      */
#define BYTE2_HIBIT         0x8000          /* 2 bytes with high bit set        */
#define BYTE3_HIBIT         0x800000 /* 3 bytes with high bit set       */
#define BYTE4_HIBIT         0x80000000      /* 4 bytes with high bit set        */
#define     A1_SIGNED(x)    ((x) == ASN1_INT)
/* the standard MIB prefix - 1.3.6.1.2.1                              */
extern char MIB_PREFIX[];
/* the standard MIB prefix is encoded by ASN.1 into 5 octets          */
#define MIB_PREF_SZ         5
```

Podemos ver os códigos usados para representar tipos de dados (ASN1_INT, ASN1_IPADDR e ASN1_COUNTER) e comandos (PDU_GET, PDU_SET e PDU_TRAP).

- **Objetos:**

  É a abstração daquilo que se quer monitorar. Pode ser na forma de variáveis ou estruturas.

- **CMIP (Common Management Information Protocol):**
  Esse protocolo segue o padrão ISO-OSI e trabalha na camada de transporte.
  Muito usado em telecomunicações principalmente em telefonia, ele é muito complexo comparado ao SNMP.

9.3 - Solicitações enviadas:

O SNMP versão 1 trabalha com os comandos:

- "Get": Para realizar buscas nas MIBs.
- "Set": Para armazenamento nas MIBs.
- "Trap": Para informar erros ou alguma anomalia.

Os comandos usados podem ser visto no arquivo asn1.h do xinu[1] mostrado logo acima na sessão "vocabulário usado" no item "SMI".

9.4 - Tipos de MIB:

- **MIB I:**
  Foi a primeira MIB desenvolvida (RFC1066), simples .

- **MIB II:**
  Uma evolução da MIB I (RFC 1213).
  Nessa foram adicionados diversos objetos.

- **MIB Experimental:**
  Define as MIBs que ainda estão em desenvolvimento.

- **MIB Privada:**
  Define a MIB desenvolvida por fabricantes que atendem a características particulares dos equipamentos.

9.5 - Aspecto da MIB:

Arquivo "mib.h" do Xinu[1]:

```
/* mib information structure: 1 per mib entry;  Initialized in snmib.c  */
struct mib_info {
        char    *mi_name;       /* name of mib variable in English       */
        char    *mi_prefix;     /* prefix in English (e.g., "tcp.")      */
        struct  oid mi_objid;   /* object identifier                     */
        int     mi_vartype;     /* type: integer, aggregate, octet str   */
        Bool    mi_writable;    /* is this variable writable?            */
        Bool    mi_varleaf;     /* is this a leaf with a single value    */
        int     (*mi_func)(struct snbentry *, struct mib_info *, int);
                                /* function to implement get/set/getnext*/
        char    *mi_param;      /* parameter used with function          */
        struct  mib_info *mi_next;  /* pointer to next var. in lexi-order*/
};
```

Essa estrutura usado no XINU[1] mostra como a MIB é manipulada.

- **mi_name:** Temos o nome do objeto.

- **mi_objid:** Temos a localização desse objeto na árvore, tendo como referência a posição da MIB.

Segue o padrão ASN.1, codifica os nós em vez de nome por números. Exemplo: “osi.org.dod” será “1.3.6” .

- **mi_vartype:** Tipo da variável definida em “asn1.h” pelo padrão ASN.1 (mib.h do XINU[1]).

- **mi_writable:** Define permissão de escrita (mib.h do XINU[1]).

- **mi_varleaf:** Define se esse ponto é um folha da árvore, ou seja, se esse tem um valor ou tem um filho (mib.h do XINU[1]).

- **mi_func:** Função que irá tratar esse objeto ou manipular dados realizando conversões.

- **mi_next:** Ponteiro para o próximo objeto lexiográfico, lembrando que a árvore é organizada por números e pontos, pontos esses separando os níveis para facilitar a busca lexiográfica. O ponteiro “mi_next” é usado pelo comando “Get Next” para localizar o próximo item.

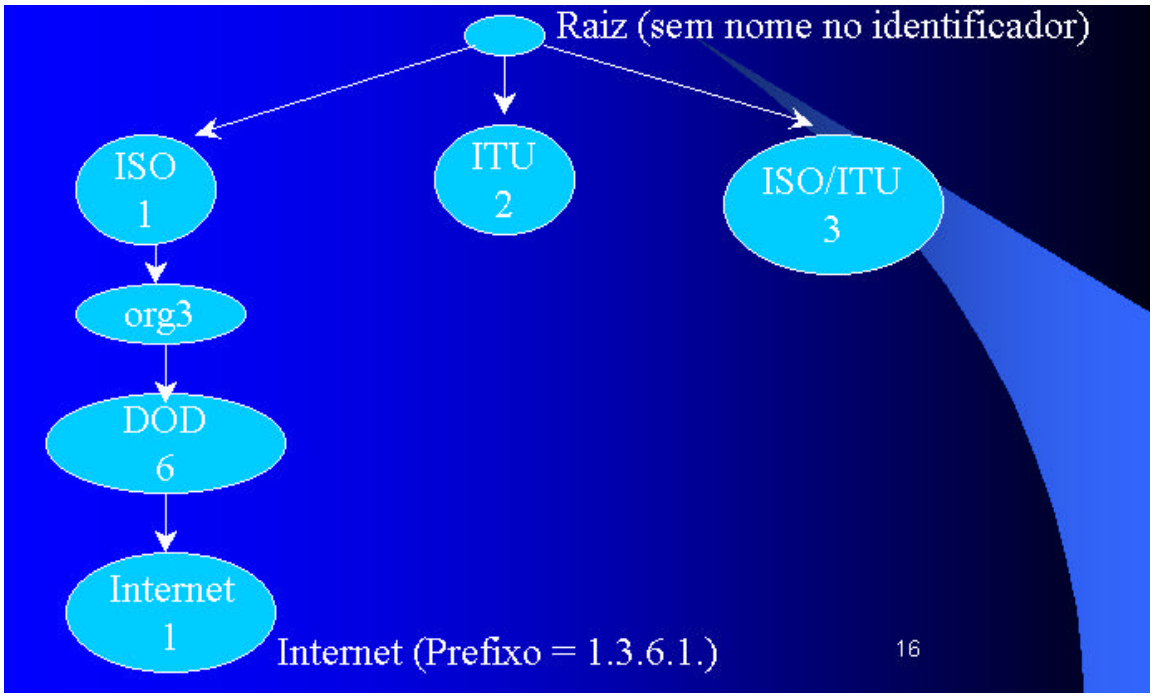
9.6 - MIB:

A MIB é organizada em uma árvore, cada nó recebe um nome hierárquico como identificador.

Quando a estrutura interna manipula a árvore são dados números para

facilitar a busca dos objetos.

No exemplo abaixo o nó “osi.org.dod” pode usar o prefixo “1.3.6”, lembrar que a raiz não é nomeada.

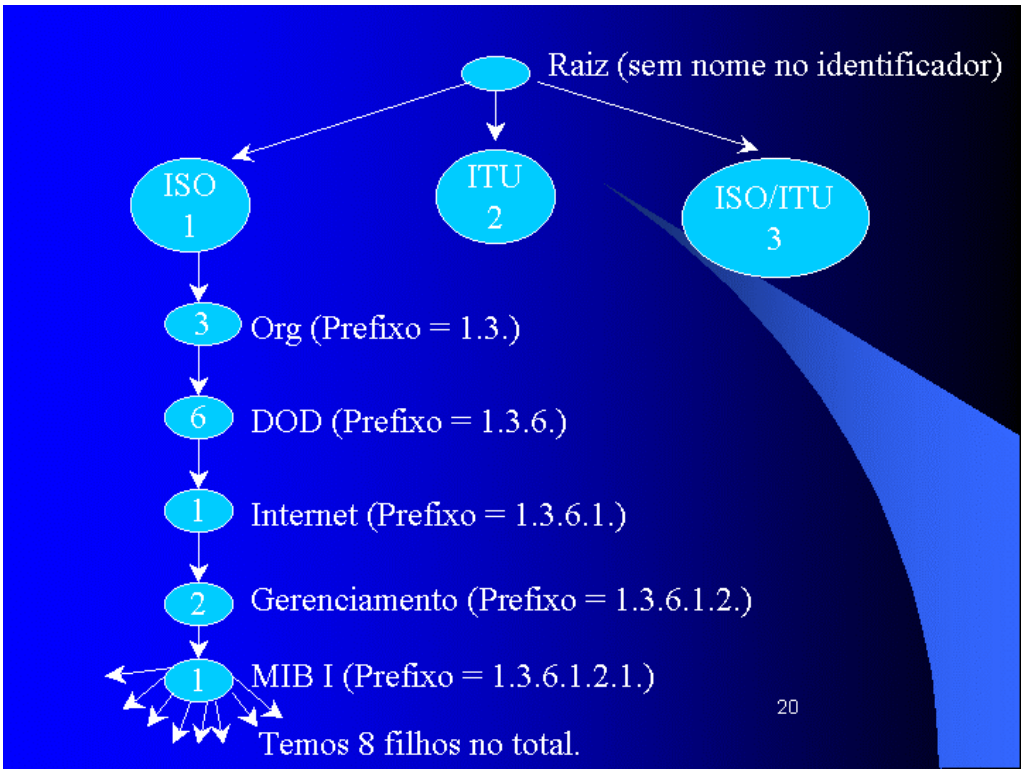


Os identificadores usados podem ser visto no arquivo “snmvars.h” do XINU[1]:

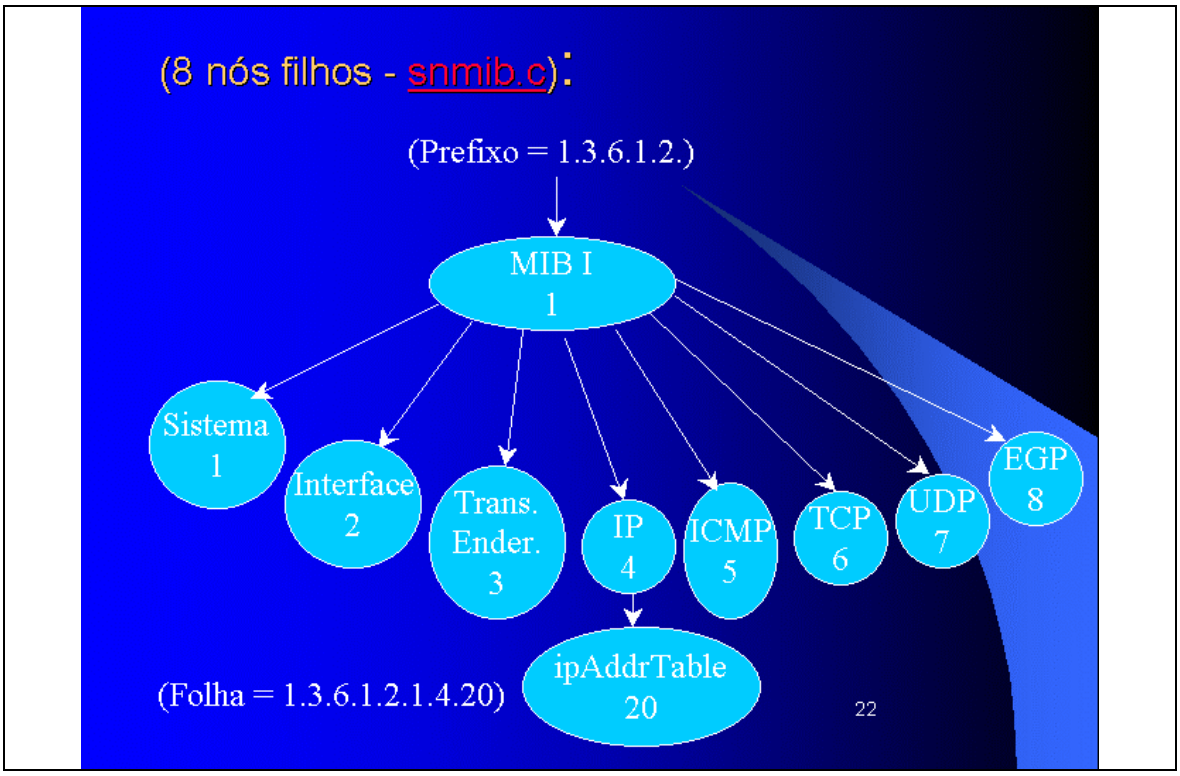
```
/* snmpvars.h */
/* System & Interface MIB */
extern char             SysDescr[], SysContact[], SysName[], SysLocation[];
extern unsigned
            SysUpTime, SysServices, IfNumber;
/* IP MIB */
extern unsigned
            IpForwarding, IpDefaultTTL, IpInReceives, IpInHdrErrors,
            IpInAddrErrors, IpForwDatagrams, IpInUnknownProtos, IpInDiscards,
            IpInDelivers, IpOutRequests, IpOutDiscards, IpOutNoRoutes,
            IpReasmTimeout, IpReasmReqds, IpReasmOKs, IpReasmFails, IpFragOKs,
            IpFragFails, IpFragCreates, IpRoutingDiscards;
/* ICMP MIB */
extern unsigned
            IcmpInMsgs, IcmpInErrors, IcmpInDestUnreachs, IcmpInTimeExcds,
            IcmpInParmProbs, IcmpInSrcQuenchs, IcmpInRedirects, IcmpInEchos,
            IcmpInEchoReps, IcmpInTimestamps, IcmpInTimestampReps,
            IcmpInAddrMasks, IcmpInAddrMaskReps, IcmpOutMsgs, IcmpOutErrors,
            IcmpOutDestUnreachs, IcmpOutTimeExcds, IcmpOutParmProbs,
            IcmpOutSrcQuenchs, IcmpOutRedirects, IcmpOutEchos,
            IcmpOutEchoReps, IcmpOutTimestamps, IcmpOutTimestampReps,
            IcmpOutAddrMasks, IcmpOutAddrMaskReps;
/* UDP MIB */
extern unsigned
```

```
            UdpInDatagrams, UdpNoPorts, UdpInErrors, UdpOutDatagrams;
/* TCP MIB */
extern unsigned
            TcpRtoAlgorithm, TcpRtoMin, TcpRtoMax, TcpMaxConn, TcpActiveOpens,
            TcpPassiveOpens, TcpAttemptFails, TcpEstabResets, TcpCurrEstab,
            TcpInSegs, TcpOutSegs, TcpRetransSegs, TcpInErrs, TcpOutRsts;
/* SNMP MIB */
extern unsigned
      snmpInPkts, snmpOutPkts, snmpInBadVersions,
      snmpInBadCommunityNames, snmpInBadCommunityUses,
      snmpInASNParseErrs, snmpInTooBigs, snmpInNoSuchNames,
      snmpInBadValues, snmpInReadOnlys, snmpInGenErrs,
      snmpInTotalReqVars, snmpInTotalSetVars, snmpInGetRequests,
      snmpInGetNexts, snmpInSetRequests, snmpInGetResponses,
      snmpInTraps, snmpOutTooBigs, snmpOutNoSuchNames,
      snmpOutBadValues, snmpOutGenErrs, snmpOutGetRequests,
      snmpOutGetNexts, snmpOutSetRequests, snmpOutGetResponses,
      snmpOutTraps, snmpEnableAuthenTraps;
```

Um exemplo da MIB até o nó “MIB” alvo do nosso estudo:



Já abaixo do nó “MIB” temos 8 nós filhos (sistema, Interface, Transformação de Endereço/Valor, IP, ICMP, TCP, UDP e EGP):



Podemos ver nesse exemplo a folha ipAddrTable que tem como identificador “1.3.6.1.2.1.4.20”,

9.7 - ASN.1:

Uma das principais funções do ASN.1 é a representação numérica da árvore da MIB.

Os números são escritos em 7 bits, o oitavo bit mais significativo é usado para representar continuidade da representação no octeto seguinte ou término.

Ou seja, números maiores que 127 o bit mais significativo tem valor 1 e números menores e iguais à 127 o oitavo bit tem valor zero.

Abaixo temos um exemplo retirado do livro “Interligação em rede com TCP/IP – Douglas E. Comer – Volume 1 – Capítulo 21”.

Nesse exemplo o número "909904" maior que 127 é separado em grupos de 7 bits, o oitavo bit é colocado como um para indicar continuidade.

Os demais números menores e iguais a 127 o oitavo bit é colocado como zero, pois, esses não precisão de outro octeto para representa-los.

Se essa representação não fosse usada um nó somente poderia ter 255 nós filhos usando a representação de números.

9.8 - Softwares SNMP:

- Proprietários:

  SunNet Manager, OpenView e MibBrowser.

- GPL:

  Scotty, Snmp-Perl, CMU-snmp e MRTG, Net-SNMP.

- Cracker:

  ADMsnmp

9.9 - Usando o software Net-SNMP e ADMsnmp:

Para exemplificar o uso do SNMP usaremos o Net-SNMP (http://www.net-snmp.org/).

O Net-SNMP é composto de diversos programas, mostraremos os principais.

Para localizar um equipamento com SNMP ativo basta usar o programa de varredura de portas "nmap":

```
$ nmap –p 161,162  -sU  10.10.10.0/24
```

9.1 – O "snmpwalk" realiza busca com o comando Get-Next em equipamentos com suporte a SNMP:

```
$ snmpwalk  XyZ.xxx.yyy.zzz -c public

SNMPv2-MIB::sysDescr.0 = STRING: Bridge in 7600 Card Rev 1.0: ZipChip Rev 2.0, i960, Nucleus OS, Software Rev Ver.  4.62
SNMPv2-MIB::sysObjectID.0 = OID: SNMPv2-SMI::enterprises.43.1.8.16
SNMPv2-MIB::sysUpTime.0 = Timeticks: (8039557) 22:19:55.57
SNMPv2-MIB::sysContact.0 = STRING:
SNMPv2-MIB::sysName.0 = STRING:
SNMPv2-MIB::sysLocation.0 = STRING:
SNMPv2-MIB::sysServices.0 = INTEGER: 2
IF-MIB::ifNumber.0 = INTEGER: 5
IF-MIB::ifIndex.1 = INTEGER: 1
IF-MIB::ifIndex.2 = INTEGER: 2
IF-MIB::ifIndex.3 = INTEGER: 3
IF-MIB::ifIndex.4 = INTEGER: 4
IF-MIB::ifIndex.5 = INTEGER: 5
IF-MIB::ifDescr.1 = STRING: Fast Ethernet port # 3.3
IF-MIB::ifDescr.2 = STRING: Fast Ethernet port # 3.4
IF-MIB::ifDescr.3 = STRING: Fast Ethernet port # 3.11
IF-MIB::ifDescr.4 = STRING: Fast Ethernet port # 3.12
IF-MIB::ifDescr.5 = STRING: 7600 Card Rev 1.0 Emulated LAN port
IF-MIB::ifType.1 = INTEGER: ethernetCsmacd(6)
IF-MIB::ifType.2 = INTEGER: ethernetCsmacd(6)
IF-MIB::ifType.3 = INTEGER: ethernetCsmacd(6)
IF-MIB::ifType.4 = INTEGER: ethernetCsmacd(6)
..
.. <continuando>
..
```

Como podemos ver a listagem são os nós da árvore da MIB do equipamento alvo.

9.2 - O "snmpnetstat" pode ser usado para obter várias informações

como rotas, portas e serviço ativos, contabilidade de protocolos e outras.

No exemplo abaixo temos a listagem das rotas.

```
$ snmpnetstat -r Xyz.xxx.yyy.www  public|more
Warning: positional community parameter is deprecated. Use -c
Routing tables
Destination             Gateway              Flags   Interface
default                  gateway1.xxx.com     UG      if0
x.w.23.10/32            gateway1.xxx.com     UG      le0
yy.xx.yx.203/32          gateway1.xxx.com     UG      le0
zz.wx.172.51/32          gateway1.xxx.com     UG      le0
```

No exemplo abaixo temos a listagem das portas e serviços ativos.

```
$ snmpnetstat -a Xyz.yyy.zzz.www  public
Warning: positional community parameter is deprecated. Use -c
Active Internet (tcp) Connections (including servers)
Proto Local Address              Foreign Address            (state)
tcp    *.*                      *.*                      CLOSED
tcp    *.ssh                     *.*                      LISTEN
tcp    *.smtp                    *.*                      LISTEN
tcp    *.time                    *.*                      LISTEN
tcp    *.sunrpc                  *.*                      LISTEN
tcp    *.uucp                    *.*                      LISTEN
tcp    *.submissi                *.*                      LISTEN
tcp    *.4045                    *.*                      LISTEN
tcp    *.5987                    *.*                      LISTEN
tcp    *.x11                     *.*                      LISTEN
tcp    *.32771                   *.*                      LISTEN
tcp    *.32777                   *.*                      LISTEN
tcp    *.32778                   *.*                      LISTEN
tcp    *.32779                   *.*                      LISTEN
```

9.3 - O “snmpset” pode alterar os parâmetros na MIB caso tenhamos acesso para armazenamento.

Inicialmente temos que fazer a busca para escolher um nó alvo na MIB:

```
$ snmpwalk  x.y.z.w -c  public

SNMPv2-MIB::sysDescr.0 = STRING: Bridge in 7600 Card Rev 1.0: ZipChip Rev 2.0, i960, Nucleus OS, Software Rev Ver.  4.62
SNMPv2-MIB::sysObjectID.0 = OID: SNMPv2-SMI::enterprises.43.1.8.16
SNMPv2-MIB::sysUpTime.0 = Timeticks: (8136406) 22:36:04.06
SNMPv2-MIB::sysContact.0 = STRING:
SNMPv2-MIB::sysName.0 = STRING:
SNMPv2-MIB::sysLocation.0 = STRING:
SNMPv2-MIB::sysServices.0 = INTEGER: 2
IF-MIB::ifNumber.0 = INTEGER: 5
IF-MIB::ifIndex.1 = INTEGER: 1
```

```
IF-MIB::ifIndex.2 = INTEGER: 2
IF-MIB::ifIndex.3 = INTEGER: 3
IF-MIB::ifIndex.4 = INTEGER: 4
IF-MIB::ifIndex.5 = INTEGER: 5
IF-MIB::ifDescr.1 = STRING: Fast Ethernet port # 3.1
IF-MIB::ifDescr.2 = STRING: Fast Ethernet port # 3.2
IF-MIB::ifDescr.3 = STRING: Fast Ethernet port # 3.9
IF-MIB::ifDescr.4 = STRING: Fast Ethernet port # 3.10
..
.. <continua>
..
```

Nesse caso o alvo será o nó "sysName.0".
Usando o comando “snmpset”:

```
$ snmpset -c private -v 1 x.y.z.w  system.sysName.0 s "teste123"

SNMPv2-MIB::sysName.0 = STRING: teste123
```

Após a alteração podemos ver "sysName.0" modificado:

```
$ snmpwalk  x.y.z.w -c  public
SNMPv2-MIB::sysDescr.0 = STRING: Bridge in 7600 Card Rev 1.0: ZipChip Rev 2.0, i960, Nucleus OS, Software Rev Ver.  4.62
SNMPv2-MIB::sysObjectID.0 = OID: SNMPv2-SMI::enterprises.43.1.8.16
SNMPv2-MIB::sysUpTime.0 = Timeticks: (8136406) 22:36:04.06
SNMPv2-MIB::sysContact.0 = STRING:
SNMPv2-MIB::sysName.0 = STRING: teste123
SNMPv2-MIB::sysLocation.0 = STRING:
SNMPv2-MIB::sysServices.0 = INTEGER: 2
IF-MIB::ifNumber.0 = INTEGER: 5
IF-MIB::ifIndex.1 = INTEGER: 1
IF-MIB::ifIndex.2 = INTEGER: 2
IF-MIB::ifIndex.3 = INTEGER: 3
IF-MIB::ifIndex.4 = INTEGER: 4
..
.. <continua>
..
```

Um exemplo do uso do comando "getnext" pode ser mostrado com o "snmpgetnext", podemos ver que ele mostra o nó seguinte do “sysName.0” :

```
$ snmpgetnext  x.y.z.w -c public   sysName.0
 SNMPv2-MIB::sysLocation.0 = STRING:
```

9.4 - Um outro software muito usado para quebra de senhas é o ADMsnmp (ftp://ADM.isp.at/ADM/), ele funciona com auxílio de um arquivo de senhas.

Exemplo: Com acesso a MIB com a senha "private".

```
$ ./snmp x.y.z.w -wordfile snmp.passwd
ADMsnmp vbeta 0.1 (c) The ADM crew
ftp://ADM.isp.at/ADM/
greets: !ADM, el8.org, ansia
>>>>>>>>>>> get req name=router  id = 2 >>>>>>>>>>>
>>>>>>>>>>> get req name=cisco  id = 5 >>>>>>>>>>>
>>>>>>>>>>> get req name=public  id = 8 >>>>>>>>>>>
>>>>>>>>>>> get req name=private  id = 11 >>>>>>>>>>>
<<<<<<<<<<< recv snmpd paket id = 12 name = private ret =0 <<<<<<<<<<
>>>>>>>>>>>> send setrequest id = 12 name = private >>>>>>>>
>>>>>>>>>>> get req name=admin  id = 14 >>>>>>>>>>>
<<<<<<<<<<< recv snmpd paket id = 13 name = private ret =0 <<<<<<<<<<
>>>>>>>>>>> get req name=proxy  id = 17 >>>>>>>>>>>
<<<<<<<<<<< recv snmpd paket id = 140 name = private ret =0 <<<<<<<<<<
>>>>>>>>>>> get req name=write  id = 20 >>>>>>>>>>>
<<<<<<<<<<< recv snmpd paket id = 140 name = private ret =0 <<<<<<<<<<
>>>>>>>>>>> get req name=access  id = 23 >>>>>>>>>>>
>>>>>>>>>>> get req name=root  id = 26 >>>>>>>>>>>
>>>>>>>>>>> get req name=enable  id = 29 >>>>>>>>>>>
>>>>>>>>>>> get req name=all private  id = 32 >>>>>>>>>>>
>>>>>>>>>>> get req name= private  id = 35 >>>>>>>>>>>
>>>>>>>>>>> get req name=test  id = 38 >>>>>>>>>>>
>>>>>>>>>>> get req name=guest  id = 41 >>>>>>>>>>>
<!ADM!>          snmp check on x.y.z.w              <!ADM!>
sys.sysName.0:ADMsnmp
name = private _[1write access
```

9.10 - Conclusão:

Com o crescimento na utilização de equipamentos móveis, redes sem infra-estrutura e equipamentos domésticos em rede os protocolos de gerência precisaram sofrer simplificações em função da carência de processamento e memória RAM disponíveis nesses equipamentos.

Quando os equipamentos domésticos estiverem sendo usados será necessária uma preocupação com a segurança no que tange: acesso, autenticação e ações de controle nesses equipamentos.

10 - Referências:

1 - http://www.protocols.com/pbook/tcpip1.htm

2 - Redes de Computadores e a Internet - Uma Nova Abordagem, James F. Kurose e Keith W. Ross 2003

3 - Computer Networks, 4th Edition - (August 9, 2002) Andrew S. Tanenbaum

4 - Redes de Computadores e Internet, Douglas E. Comer (2001)

5 - Claudio G. Mello, Gustavo H. M. B. Carneiro, Paulo R. Lira Gondim – Implantação de um nó mbone do IME

6 - Sérgio Alipi Domingues Deusdado – Integração adaptativa de aplicações multicast para conferencia multimídia

7 - http://mesonpi.cat.cbpf.br/mcast

8 – www.redbooks.ibm.com; TCP/IP Tutorial and Technical Overview.

9 - www.symantec.com.br

10 – Sitema Operacional Xinu comentado pelos livros [2] e [3].

11 - Livro: "Interligação de rede com TCP/IP Volume 2"
Autor: Douglas E. Commer

12 - Livro: "Interligação de rede com TCP/IP Volume 1"
Autor: Douglas E. Commer

13 – Site: GTA/UFRJ – http://www.ufrj.br/

14 - TCP/IP Tutorial and Technical Overview
(http://www.redbooks.ibm.com/redbooks/pdfs/gg243376.pdf)
(http://www.redbooks.ibm.com/)

15 – Introdução ao ping e traceroute
(ftp://ftp2.biblioteca.cbpf.br/pub/apub/2002/nt/nt_zip/nt01002.pdf)

16 – Aspectos da segurança no protocolo IP
(http://www.modulo.com.br/pdf/mo7e001p.pdf)