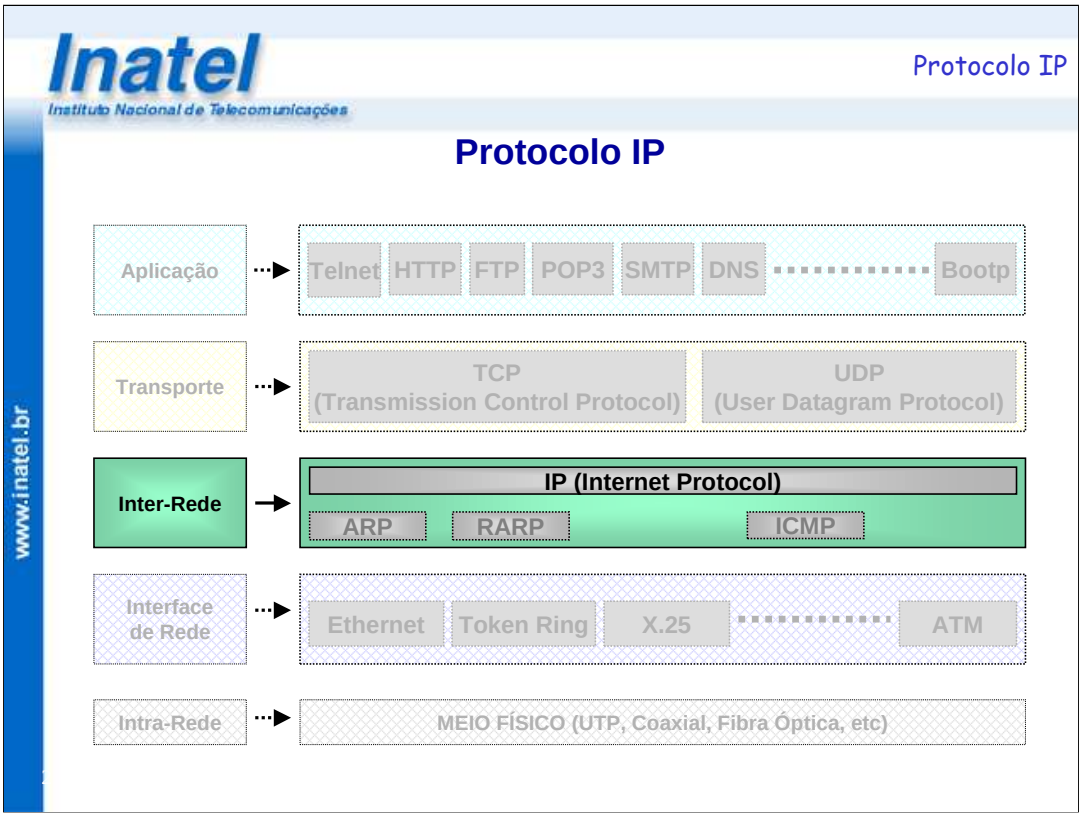
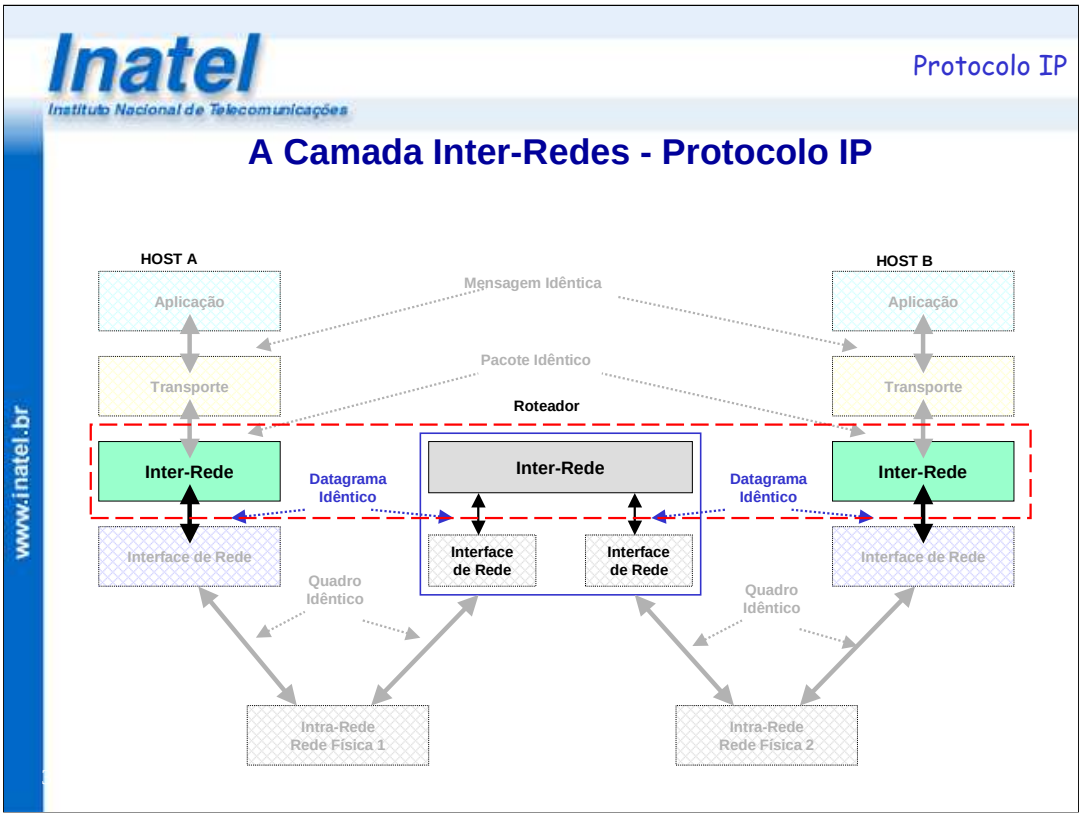
Neste capítulo será mostrado o funcionamento do Protocolo IP (Internet Protocol) que é o principal protocolo na camada de Inter-Rede. As características deste protocolo assim como seu datagrama, que é a sua unidade de dados também serão estudados. Estes datagramas são enviados para a camada de Interface de Rede e encapsulados como sendo um quadro físico de rede para transmissão. Os datagramas podem ainda sofrer um processo de fragmentação quando transmitidos em uma rede que tenha uma unidade de transmissão menor que o datagrama.

E, finalmente, será mostrado algumas características da nova versão do protocolo IP que o chamado IP versão 6 ou IP v6 ou simplesmente IPng.



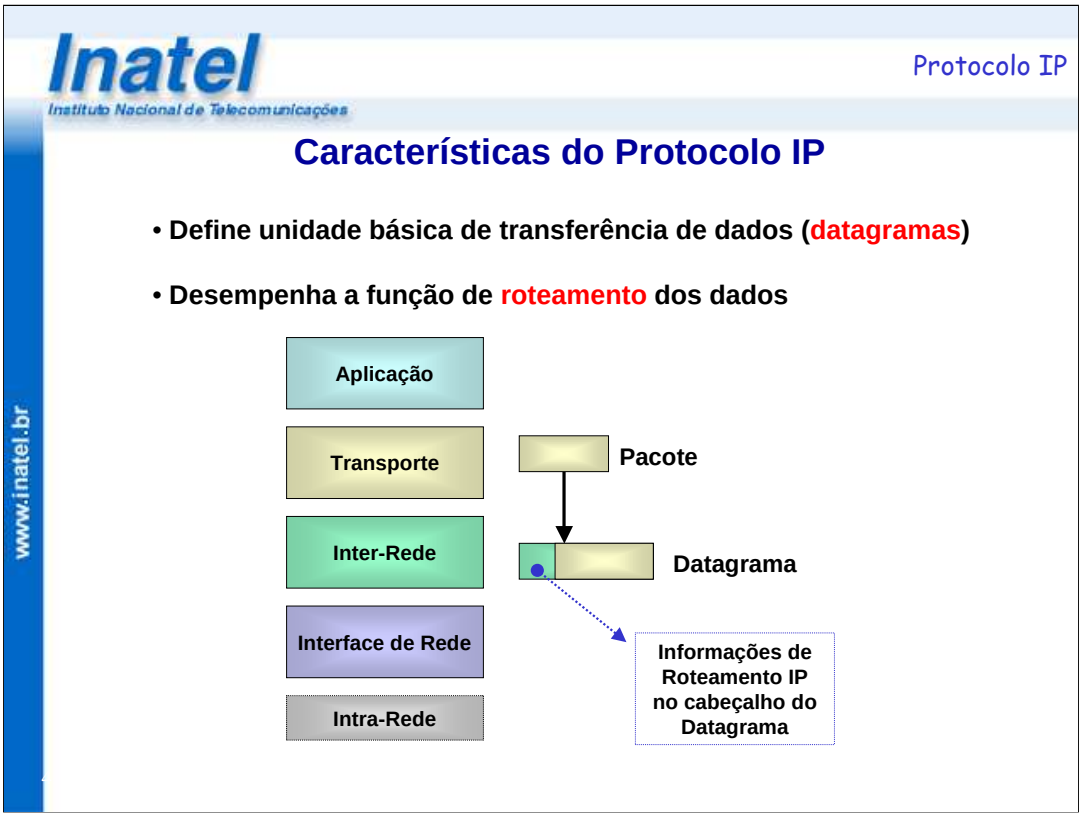
O protocolo IP é um dos principais protocolos da pilha de protocolo TCP / IP. Ele é o responsável pelo roteamento dos pacotes dentro de uma rede TCP / IP.

O protocolo IP está localizado na Camada Inter-Redes da pilha de camadas do protocolo TCP / IP.



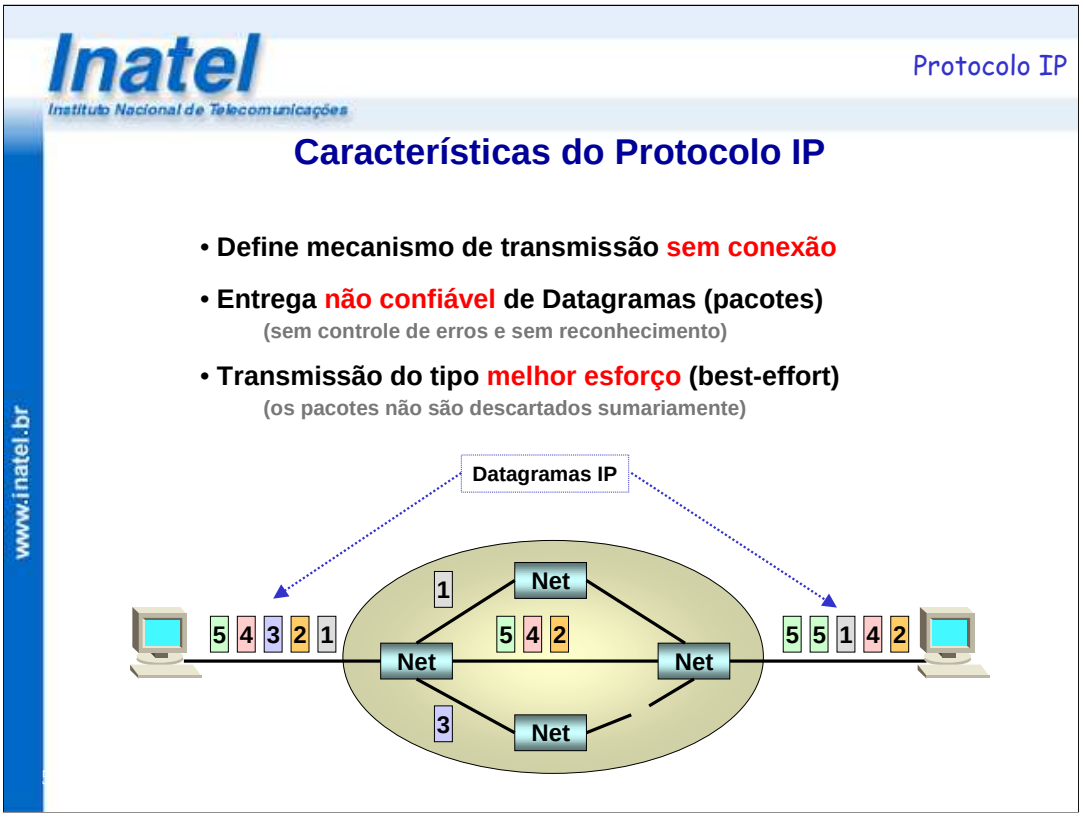
O IP é o protocolo que oculta a rede física subjacente, criando uma visão virtual da rede. É um protocolo de entrega de pacotes não-confiável, de melhor esforço e sem conexão. O IP é um dos principais protocolos utilizados em interligação de redes. O IP é o responsável por receber os pacotes da camada de transporte, encapsular estes pacotes em um datagrama IP (que contém informações para o roteamento do datagrama na rede), e enviá-lo para a camada de interface de rede para transmissão na rede física. Na recepção, o IP recebe um quadro físico de rede (Frame), retira o datagrama e o entrega para a camada de transporte.

Repare que o protocolo IP é a camada de interligação da rede física com a rede lógica (internetworking).



A unidade básica de transferência de um pacote de dados na rede TCP / IP chama-se Datagrama. Ele é formado por um cabeçalho contendo informações para o IP (roteamento) e dados que são relevantes apenas para os protocolos de nível mais alto.

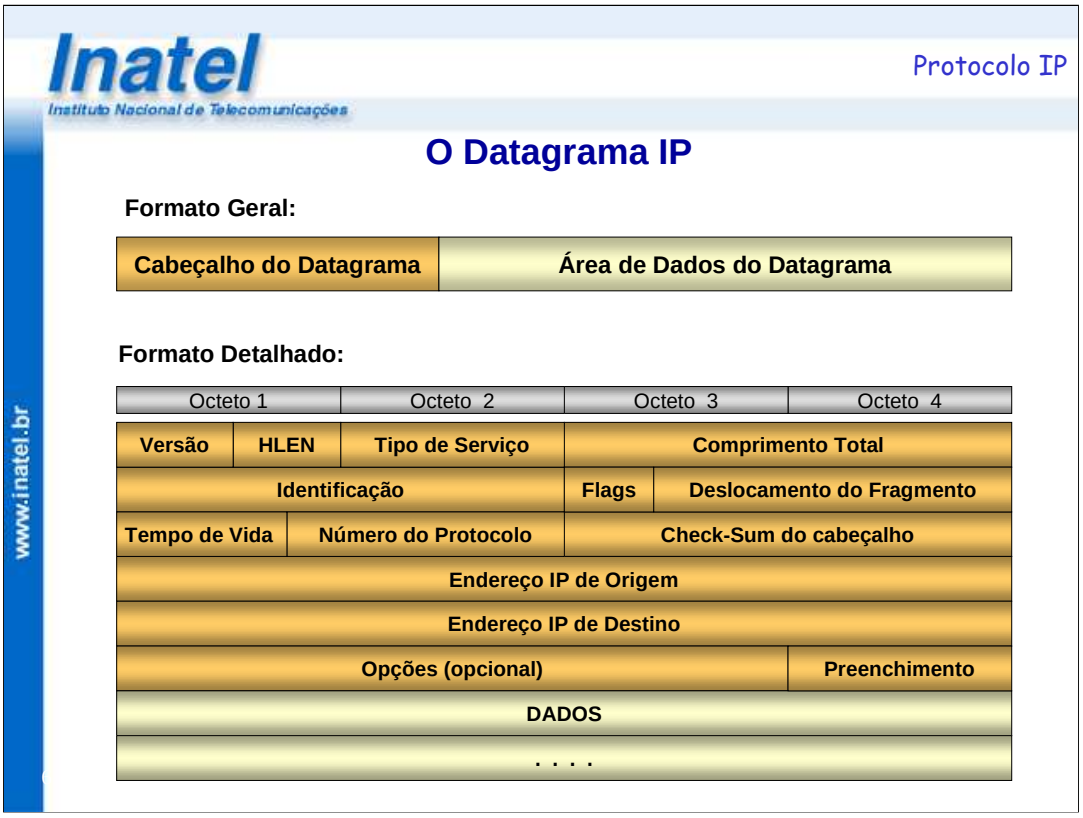
Um pacote vindo da camada de transporte é encapsulado em um datagrama IP, que armazena em seu cabeçalho os endereços IP de origem e destino do datagrama, e estas informações são utilizadas para roteamento dos datagramas pela rede, através dos roteadores.



O serviço de entrega de datagramas prestado pelo protocolo IP, é definido, tecnicamente, como um serviço de transmissão **sem conexão, com melhor esforço e não confiável**.

O serviço é conhecido como não confiável porque a entrega dos datagramas não é garantida. O datagrama pode ser perdido, duplicado, atrasar-se ou ser entregue com problemas, e nem informará isso ao transmissor nem ao receptor. O IP não se responsabilizará por estas situações. Tratar destas situações fica a cargo dos protocolos de camada mais alta.

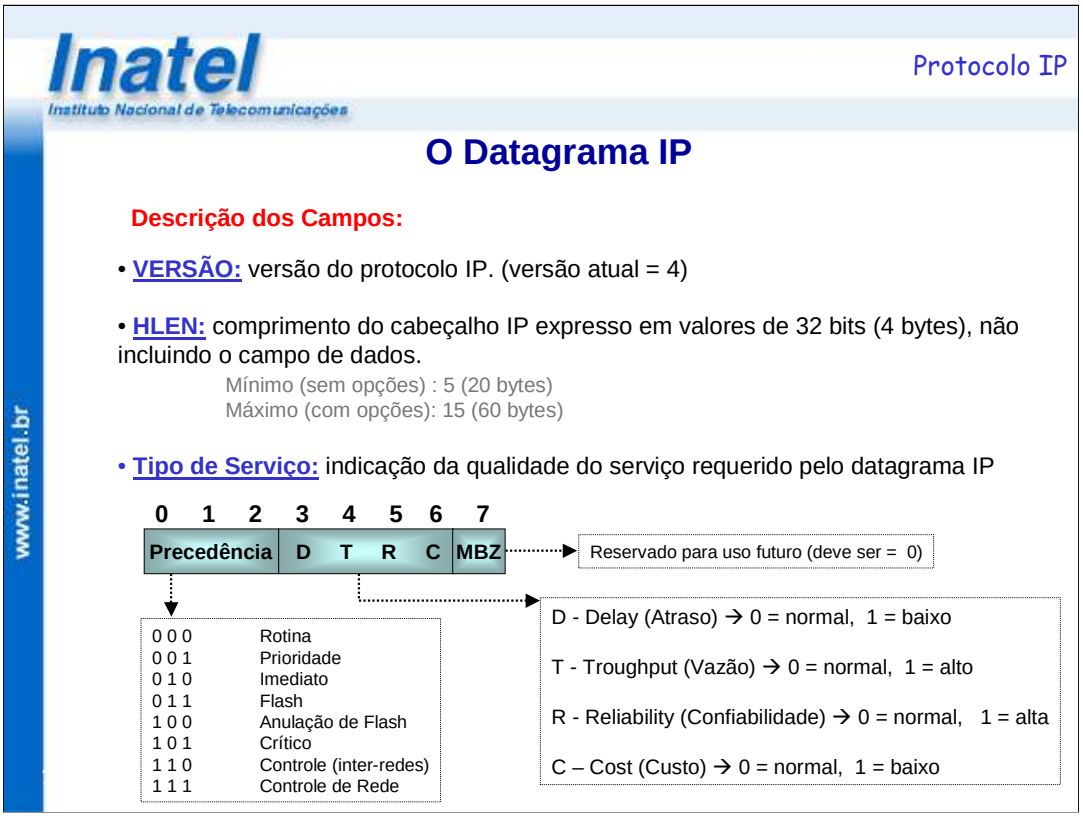
Ele é denominado sem conexão porque cada datagrama é independente dos outros. Uma seqüência de datagramas enviados de um computador a outro pode trafegar por caminhos diferentes, ou alguns podem ser perdidos enquanto outros são entregues. Uma das razões para o uso do protocolo de rede sem conexão foi a de minimizar a dependência de centros de computação específicos que usavam redes hierárquicas orientadas a conexão. A proposta do TCP / IP é o funcionamento em uma rede que ainda fosse operacional mesmo que partes desta rede não funcionem por algum motivo.



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 (frame) que contém um cabeçalho e dados, onde o cabeçalho fornece informações como endereço físico (MAC) de origem e de destino. A interligação em redes denomina sua unidade básica de transferência de um datagrama IP, ou simplesmente datagrama. Como um quadro de rede física, o datagrama é dividido em cabeçalho e área de dados. Também como um quadro, o cabeçalho de um datagrama contém os endereços de origem e de 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.

O comprimento máximo de um datagrama IP é de 65.535 bytes (ou octetos). Há também uma exigência para que todos os hosts e roteadores TCP / IP suportem datagramas IP com até 576 bytes sem fragmentação, acima deste valor, dependendo da rede um datagrama pode ser fragmentado em vários pedaços.

O cabeçalho de um datagrama IP, tem no mínimo 20 bytes (octetos) de comprimento e possui os seguintes campos descritos a seguir.



VERS: Informa a versão do protocolo IP sendo carregado. Atualmente a versão de IP é 4.

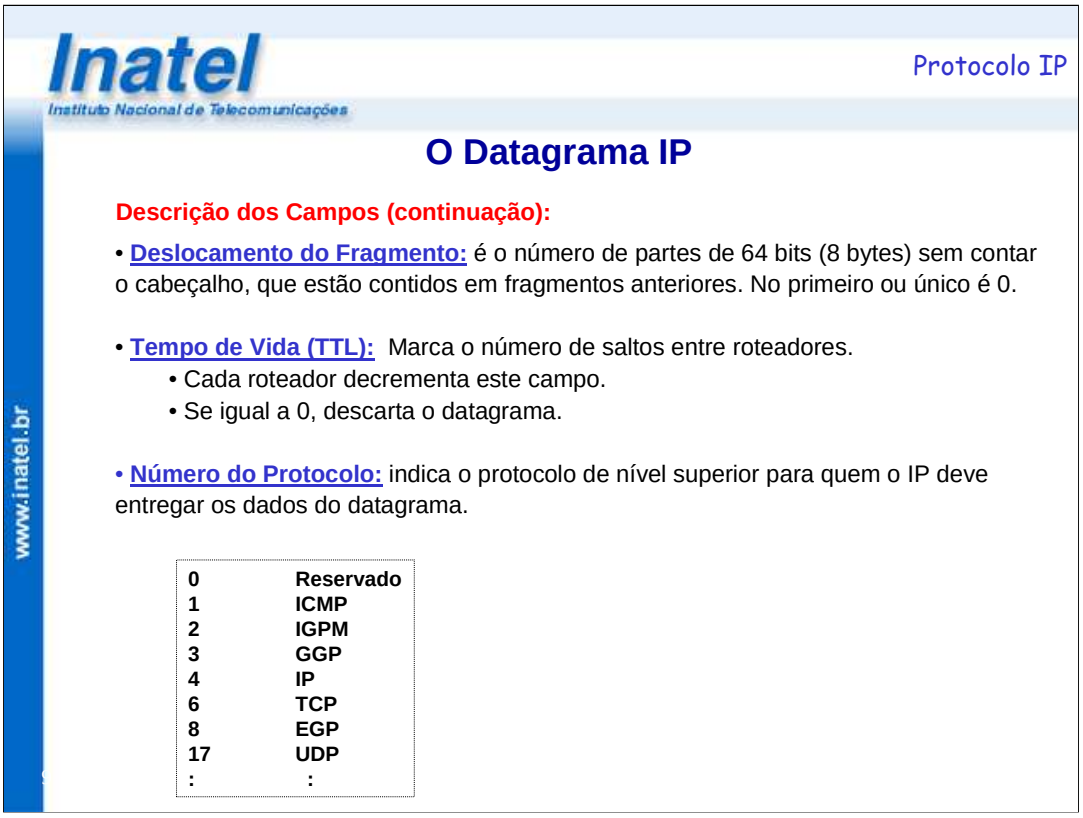
HLEN: Informa o tamanho do cabeçalho IP em grupos de 4 bytes (32 bits). Não inclui o campo de dados.

TIPO DE SERVIÇO: Informa como o pacote deve ser tratado, de acordo com sua prioridade e o tipo de serviço desejado como Baixo Retardo, Alta Capacidade de Banda ou Alta Confiabilidade. É uma indicação da qualidade de serviço requerido por este datagrama.

COMPRIMENTO TOTAL: Especifica o comprimento total do datagrama (cabeçalho e dados), especificado em bytes.

IDENTIFICAÇÃO: Identifica o pacote IP unicamente entre os outros transmitidos pela máquina. Este campo é usado para identificar o pacote IP no caso de haver fragmentação em múltiplos datagramas, onde todos os fragmentos terão o mesmo número de identificação.

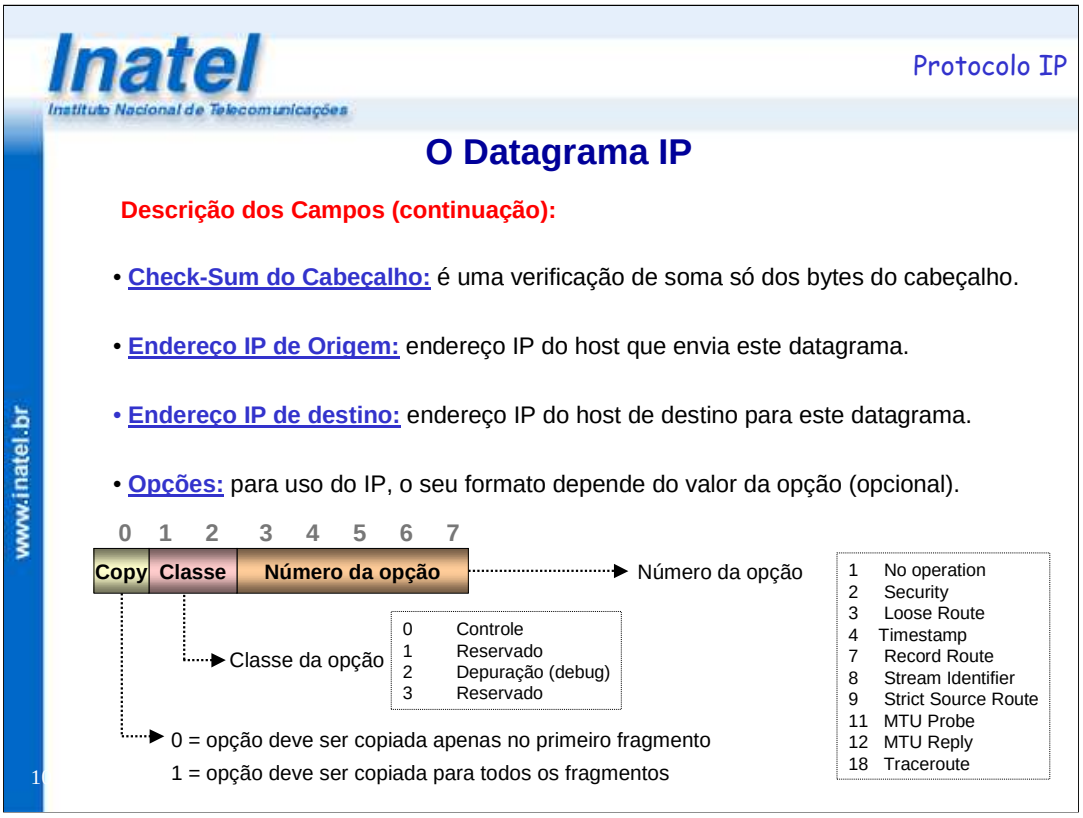
FLAGS (3 bits): um bit (MF - More Fragments) identifica se este datagrama é o último fragmento de um pacote IP ou se existem mais. Outro bit (DNF - Do Not Fragment) informa aos roteadores no caminho se a aplicação exige que os pacotes não sejam fragmentados. O outro bit é reservado e deve ser sempre 0 (zero).



DESLOCAMENTO DO FRAGMENTO: usado com datagramas fragmentados, para ajudar no reagrupamento completo do datagrama. O valor é o número de partes de 64 bits (8 bytes), onde os bytes do cabeçalho não são contados, que estão contidos em fragmentos anteriores. No primeiro fragmento ou se o datagrama não está fragmentado, este valor é sempre zero.

TEMPO DE VIDA (TTL - Time To Live): Este valor é decrementado a cada 1 segundo que o pacote passa na rede e a cada roteador pelo qual ele passa. Serve para limitar a duração do pacote IP e evitar que um pacote seja roteado eternamente na rede como resultado de um loop de roteamento. O TTL torna-se uma medida de "número de saltos" (número de roteadores pelo qual o datagrama passa) em vez de ser uma medida de tempo. Quando o valor chega a zero, assume-se que este datagrama tenha viajado em círculos e ele é então descartado. O valor inicial deve ser configurado pelo protocolo de nível superior que cria o datagrama.

NÚMERO DO PROTOCOLO: Indica o protocolo de nível superior para quem o IP deve entregar os dados contidos neste datagrama.



CHECK-SUM DO CABEÇALHO: Valor que ajuda a garantir a integridade do cabeçalho do pacote IP. Se o check-sum do cabeçalho não for igual ao conteúdo, o datagrama é descartado porque pelo menos 1 bit no cabeçalho está danificado e o datagrama pode até mesmo ter chegado no destino errado.

ENDEREÇO IP DE ORIGEM: Endereço IP do host de origem do datagrama IP.

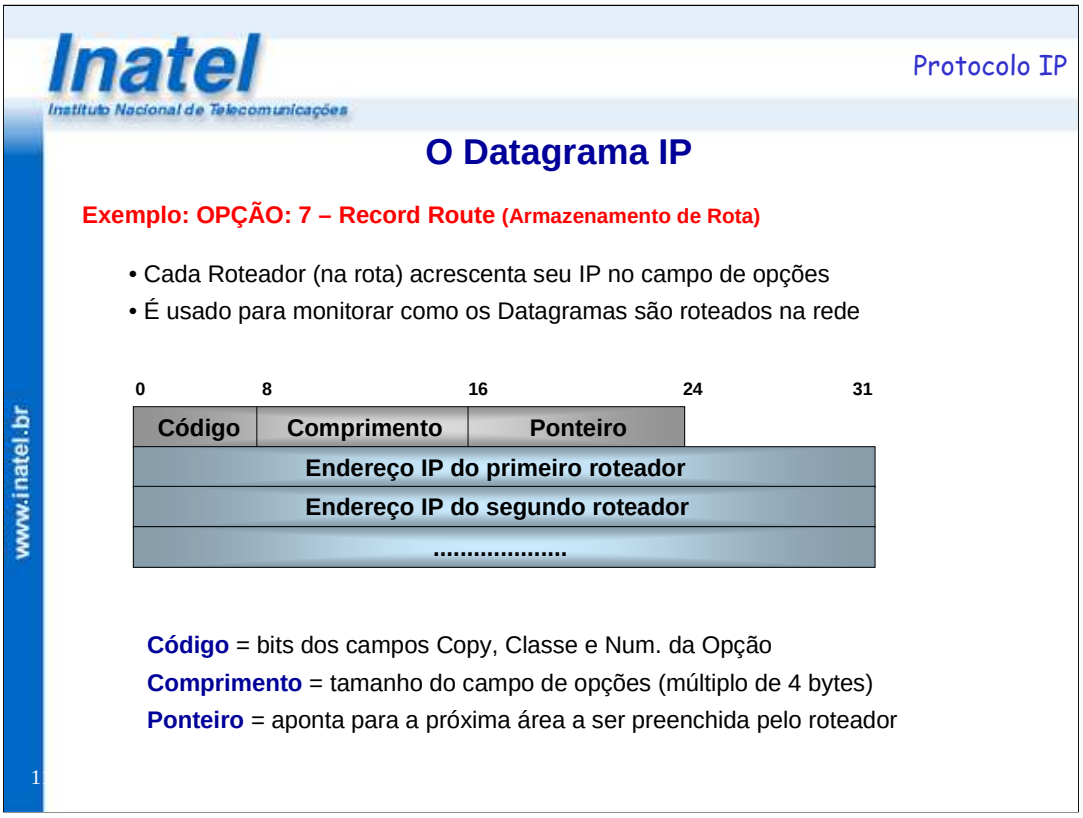
ENDEREÇO IP DE DESTINO: Endereço IP do host de destino do datagrama IP.

OPÇÕES: Opções com informações adicionais para o protocolo IP. Consiste de um byte com a identificação da opção e uma quantidade de bytes variável com as informações específicas. Um pacote IP pode transportar várias opções simultaneamente.

Se o valor de 'Copy' for 0 significa que este campo de opção deve ser copiado apenas para o primeiro fragmento do datagrama. Se for 1 significa que o campo de opção deve ser copiado para todos os fragmentos do datagrama.

A classe pode ser: Controle de Redes e Datagramas, Reservado para uso futuro, Medição e depuração na rede, Reservado para uso futuro

As opções podem ser: 1 No operation, 2 Security, 3 Loose Route, 4 Timestamp, 7 Recorde Route, 8 Stream Identifier, 9 Strict Source Route, 11 MTU Probe, 12 MTU Reply, 18 Traceroute

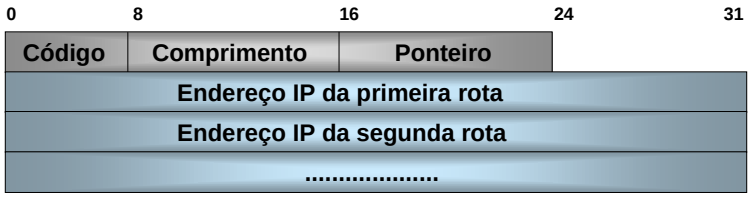


As opções IP são utilizadas basicamente como forma de verificação e monitoração de uma rede IP. As opções que especificam a rota até o destino não são utilizadas normalmente pois o IP é baseado na técnica de Next-Hop routing. Ainda assim, estes mecanismos são pouco utilizados como ferramenta de testes e verificação, sendo raros os programas que os implementam

O Datagrama IP

Exemplos: OPÇÃO: 9 – Strict Source Route (Roteamento Restrito da Origem)
OPÇÃO: 3 – Loose Source Route (Roteamento Flexível da Origem)

- Strict Source Route – Rota exata a ser seguida pelo datagrama
- Loose Source Route – o datagrama deve passar pelo menos em uma das rotas

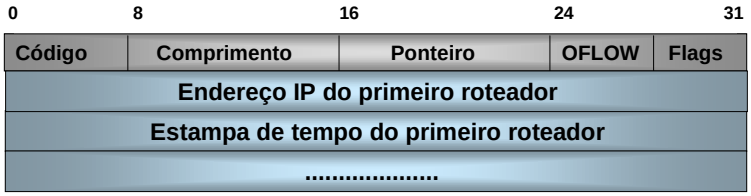


Código = bits dos campos Copy, Classe e Num. da Opção
Comprimento = tamanho do campo de opções (múltiplo de 4 bytes)
Ponteiro = aponta para a próxima área a ser preenchida pelo roteador

O Datagrama IP

Exemplo: OPÇÃO: 4 – Timestamp Route (Indicação de Hora do Roteamento)

- Inicialmente contém uma lista vazia de roteadores e tempos
- Cada roteador acrescenta seus dados (IP e tempo)
- Cada entrada na lista contém IP (32 bits) e tempo (32 bits)



Código = bits dos campos Copy, Classe e Num. da Opção

Comprimento = tamanho do campo de opções (múltiplo de 4 bytes)

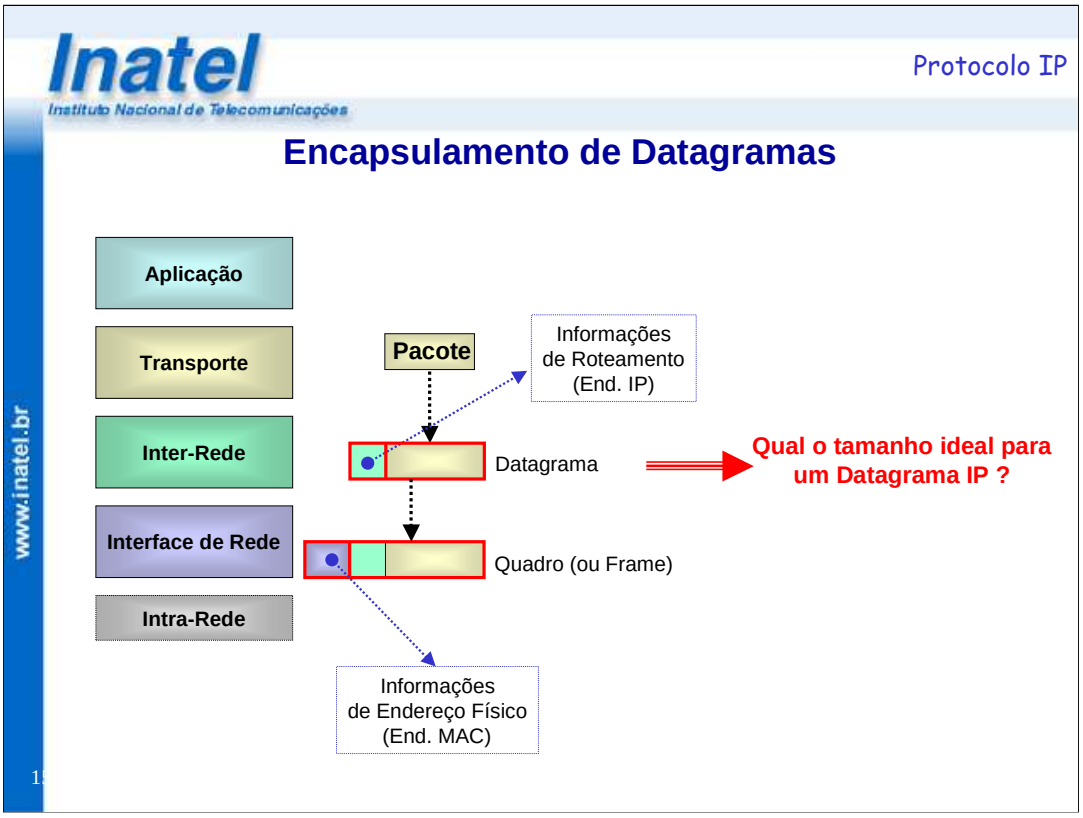
Ponteiro = aponta para a próxima área a ser preenchida pelo roteador

OFLOW = contador do número de roteadores que não conseguiram gravar o tempo

Flags = controla o formato das informações (0 = grava só o tempo, 1 = grava IP + tempo)

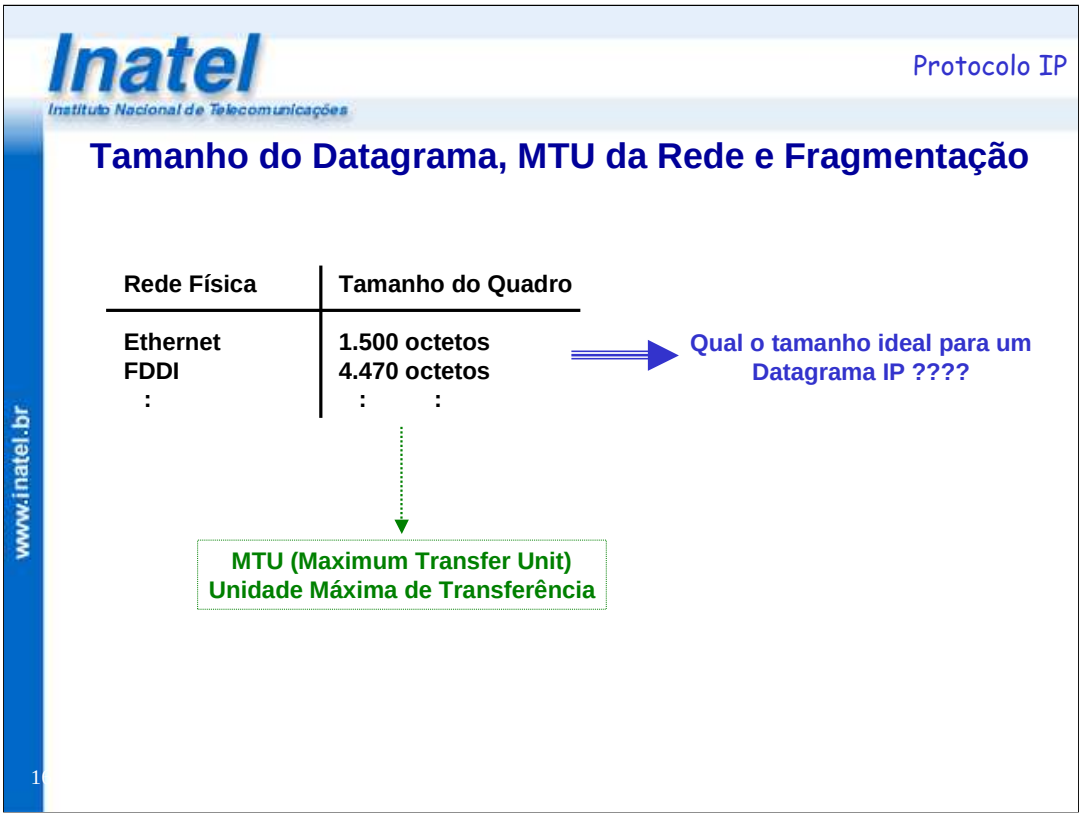
PREENCHIMENTO: Se uma OPÇÃO for usada, o datagrama é preenchido com zeros até a próxima palavra de 32 bits.

DADOS: Os dados contidos no datagrama são passados para um protocolo de nível superior, especificado no campo PROTOCOLO, ou em sentido contrário.



Sabemos que, à medida que os datagramas se movem de uma máquina para outra, eles precisam sempre ser transportados por uma rede física básica (por ex. Ethernet). Para tornar o transporte da interligação em redes eficiente, temos que 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 e nem entende o endereço de destino IP. Assim, conforme mostrado, quando uma máquina envia um datagrama IP a outra, todo o datagrama é transportado na parte de dados do quadro de rede.



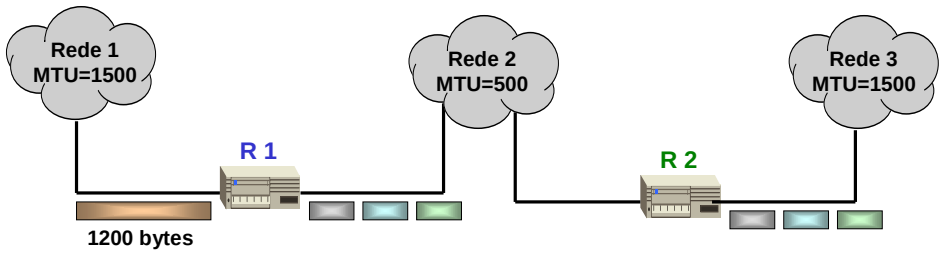
É importante considerar como os datagramas relacionam-se com os quadros de redes físicas. Começaremos com uma pergunta: “que tamanho um datagrama pode Ter ?”. Ao contrário dos quadros de redes físicas que precisam ser reconhecidos pelo hardware, os datagramas são tratados por softwares. Eles podem Ter 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 (ou bytes). Entretanto, esse limite poderá ser mudado em versões posteriores do protocolo.

Os limites mais importantes para o tamanho dos datagramas surgem na prática. Na situação ideal, todo o datagrama IP encaixa-se em um quadro físico, tornando a transmissão na rede física eficiente. Mas que tamanho de quadro deve ser escolhido ? Um datagrama pode trafegar em muitos tipos de redes físicas diferentes, à medida que move-se na interligação em redes para o seu destino final.

Cada tecnologia de comutação por pacotes coloca um limite superior, fixo, para o total de dados que podem ser transferidos em um quadro físico. A Ethernet, por exemplo, limita as transferências a 1.500 octetos de dados por quadro, enquanto que na FDDI este valor é de aproximadamente 4.470 octetos.

Inatel
Instituto Nacional de Telecomunicações

Tamanho do Datagrama, MTU da Rede e Fragmentação



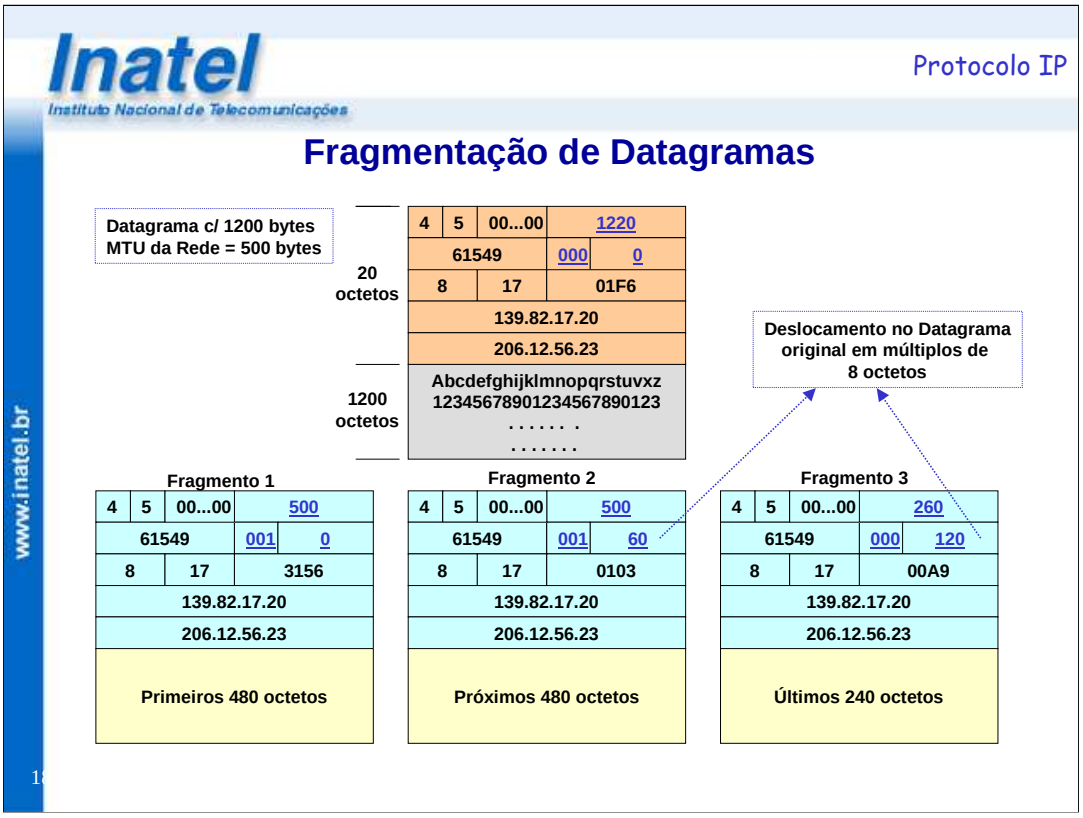
Observações:

- A Fragmentação é feita na Camada Inter-Rede pelo protocolo IP
- Um Datagrama IP pode ter até 64 Kbytes (cabeçalho + dados)
- Fragmentos são remontados somente no host de destino
- MTU mínimo para os roteadores = 576 bytes

Assim, em vez de projetar datagramas que sigam as restrições das redes físicas, o TCP / IP escolhe um tamanho inicial de datagrama conveniente e descobre uma forma de dividir os datagramas extensos em frações menores, quando o datagrama precisar atravessar uma rede que tenha um MTU pequeno.

As pequenas frações em que um datagrama é dividido são denominadas de fragmentos, e o processo de divisão de um datagrama é conhecido como fragmentação.

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 um MTU grande, e precisa enviá-lo em uma rede para a qual o MTU seja menor do que o tamanho do datagrama. Então o roteador fragmenta o datagrama em fragmentos de tamanho igual ao menor MTU.

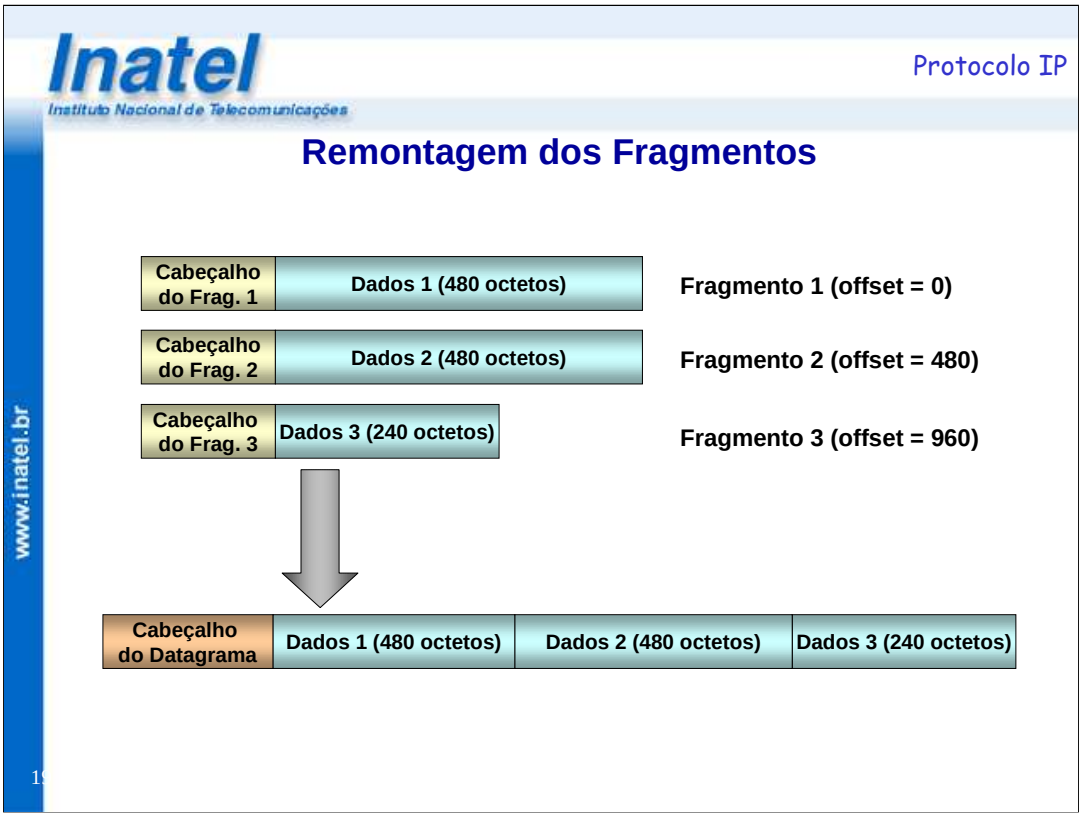


Cada fragmento contém um cabeçalho de datagrama que duplica a maior parte do cabeçalho do datagrama original, seguido por tantos dados quantos puderem ser transportados no fragmento, enquanto mantém o comprimento total menor que a MTU da rede na qual precisa trafegar.

O tamanho do fragmento é escolhido de tal forma que cada fragmento possa ser transportado na rede física 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.

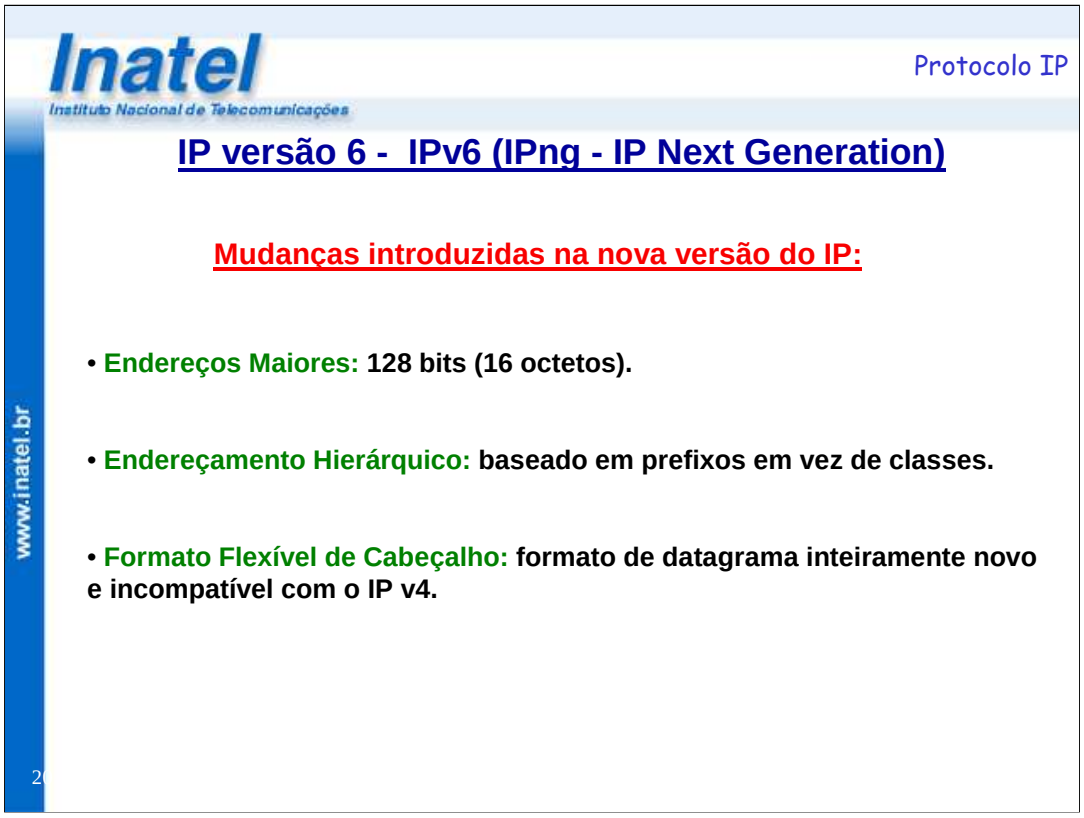
No exemplo é ilustrado um datagrama de 1.200 octetos sendo transportados por uma rede com MTU igual a 500 octetos. O datagrama original será fragmentado em 3 fragmentos, onde em cada fragmento é colocado no campo DESLOCAMENTO DO FRAGMENTO do cabeçalho do datagrama, o valor (múltiplo de oito) de octetos que já foram transportados em outros fragmentos anteriores.

No host de destino, os dados tem que ser reagrupados em um datagrama. O host emissor escolhe um número único para o campo de identificação do datagrama. Como a fragmentação não altera este campo, os fragmentos que chegam no host receptor podem ser identificados por este campo.



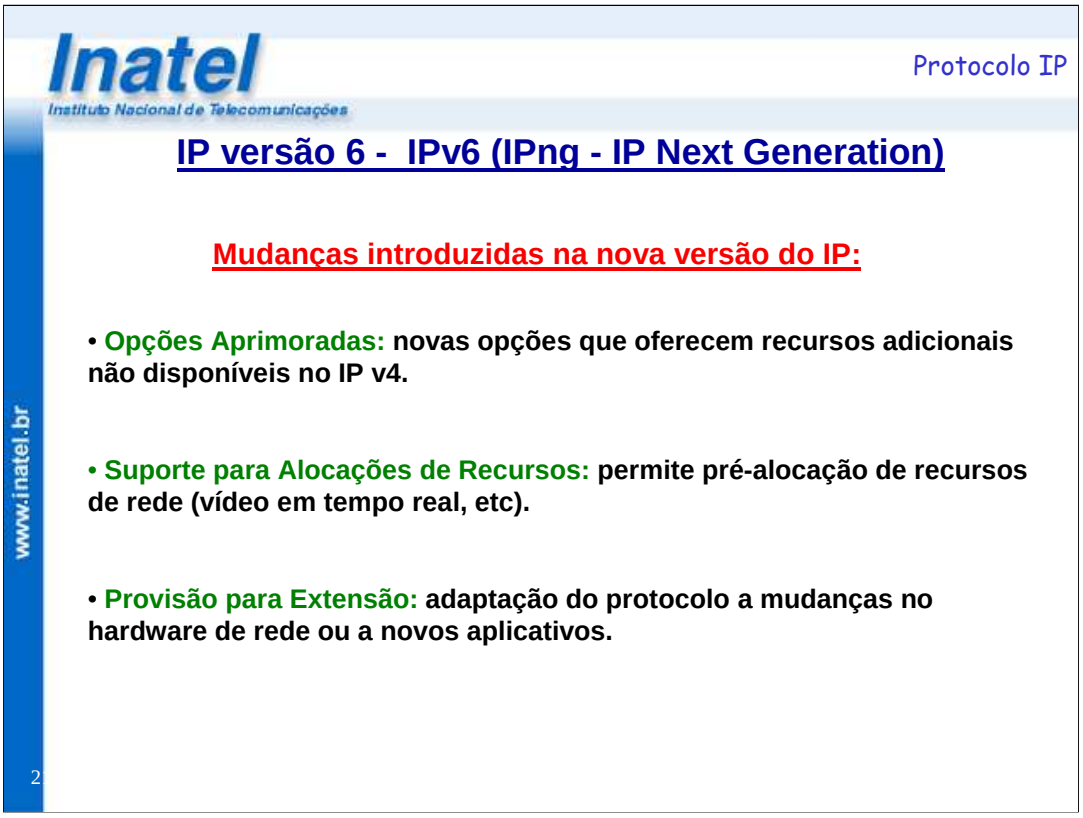
Para reagrupar os fragmentos, o host receptor reserva um buffer na memória assim que o primeiro fragmento chegar. É então iniciada uma rotina que marca o tempo. Quando o tempo se esgota sem que todos os fragmentos tenham chegado, o datagrama é descartado. O valor inicial deste cronômetro é chamado de TTL do datagrama IP. Quando os fragmentos subseqüentes chegarem antes que o tempo se esgote, os dados são simplesmente copiados no buffer, na localização indicada pelo campo de DESLOCAMENTO DO FRAGMENTO. Assim que todos os fragmentos chegarem, o datagrama original completo é restaurado.

A fragmentação e o reagrupamento são realizados na camada de interface de rede de maneira transparente ao protocolo IP.

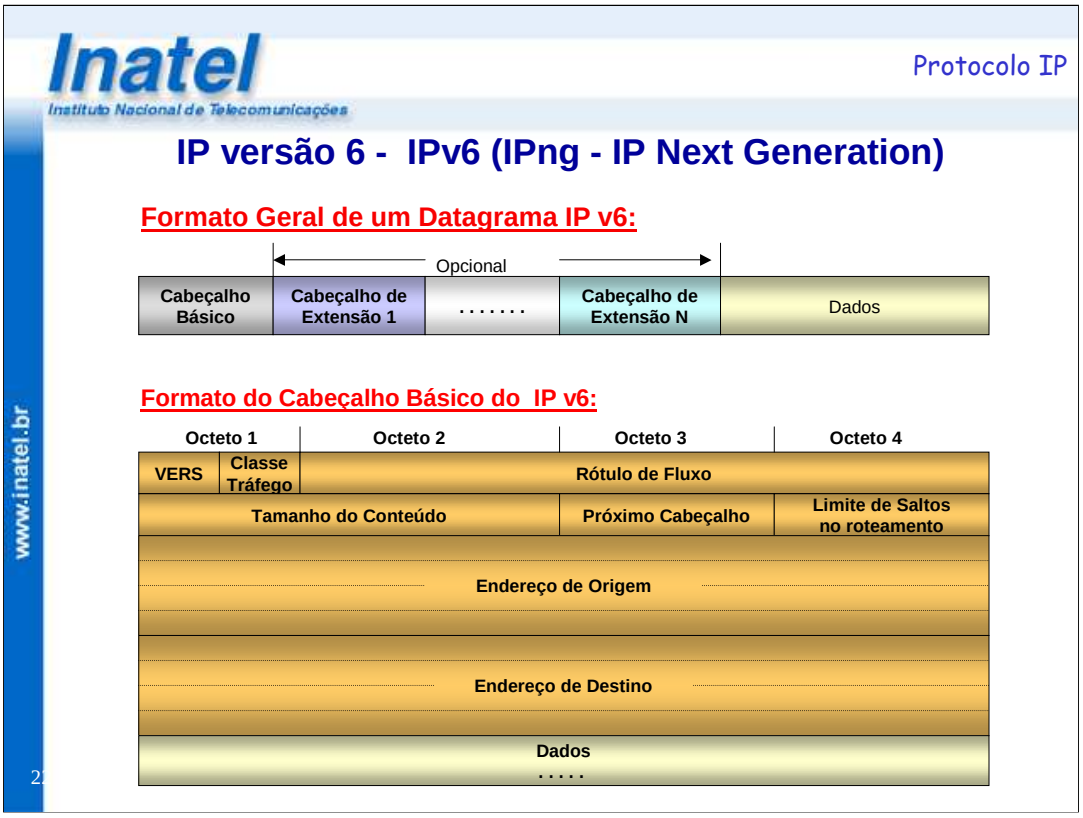


A versão 4 do Internet Protocol (IP v4) fornece o mecanismo básico de comunicação da pilha TCP / IP e da Internet. Essa versão permaneceu quase inalterada desde o seu início no final da década de 70. A longevidade do IP v4 mostra que o projeto é flexível e poderoso. Desde quando foi projetado, o desempenho do processador mais de dez vezes, os tamanhos típicos de memória aumentaram 32 vezes, a largura de banda da rede do backbone da Internet cresceu 800 vezes, tecnologias de rede local afloraram e o número de hosts na Internet cresceu de apenas algumas centenas para 4 milhões. Além disso, as mudanças não ocorreram simultaneamente; o IP conciliou mudanças em uma tecnologia, diante das mudanças em outras.

Apesar de seu projeto sólido, o IP v4 deve ser logo substituído. A principal motivação para a atualização do IP é o esgotamento iminente do espaço de endereço. Embora a necessidade de um espaço maior de endereço esteja forçando uma mudança imediata no IP, outros fatores estão também contribuindo para o projeto. Por exemplo, como áudio e vídeo em tempo real precisam de limites garantidos em retardo de transmissão, uma nova versão do IP deve fornecer um mecanismo que torne possível associar um datagrama a uma reserva de recursos atribuída previamente. Além disso, já que muitos dos novos aplicativos da internet requerem comunicações seguras, uma nova versão do IP deverá incluir recursos que tornem possível validar o transmissor.



Esta nova versão do IP é o IP versão 6 (IP v6 ou IPng – Next Generation). Esta nova versão mantém muitas das características que contribuíram para o sucesso do IP v4.O IP v6 possui as mesmas características do IP v4, com algumas modificações. Por exemplo, o IP v6 permite que o transmissor escolha o tamanho do datagrama e especifique o número máximo de passos da rota que um datagrama pode fazer antes de ser concluído, usa endereços maiores (128 bits) e acrescenta algumas características novas de endereçamento, revisa completamente o formato do datagrama, simplificando o cabeçalho de tamanho variável do IP v4 por uma série de cabeçalhos de formato fixo e muitas outras melhorias.



Um datagrama IP v6 tem um cabeçalho básico de tamanho fixo seguido de nenhum ou mais cabeçalhos de extensão seguidos de dados.

Embora deva acomodar endereços maiores, um cabeçalho IP v6 contém menos informações do que um cabeçalho de datagrama IP v4. As opções e alguns campos fixos foram removidos no cabeçalho IP v6. Em geral, as mudanças no cabeçalho de datagrama refletem mudanças no protocolo:

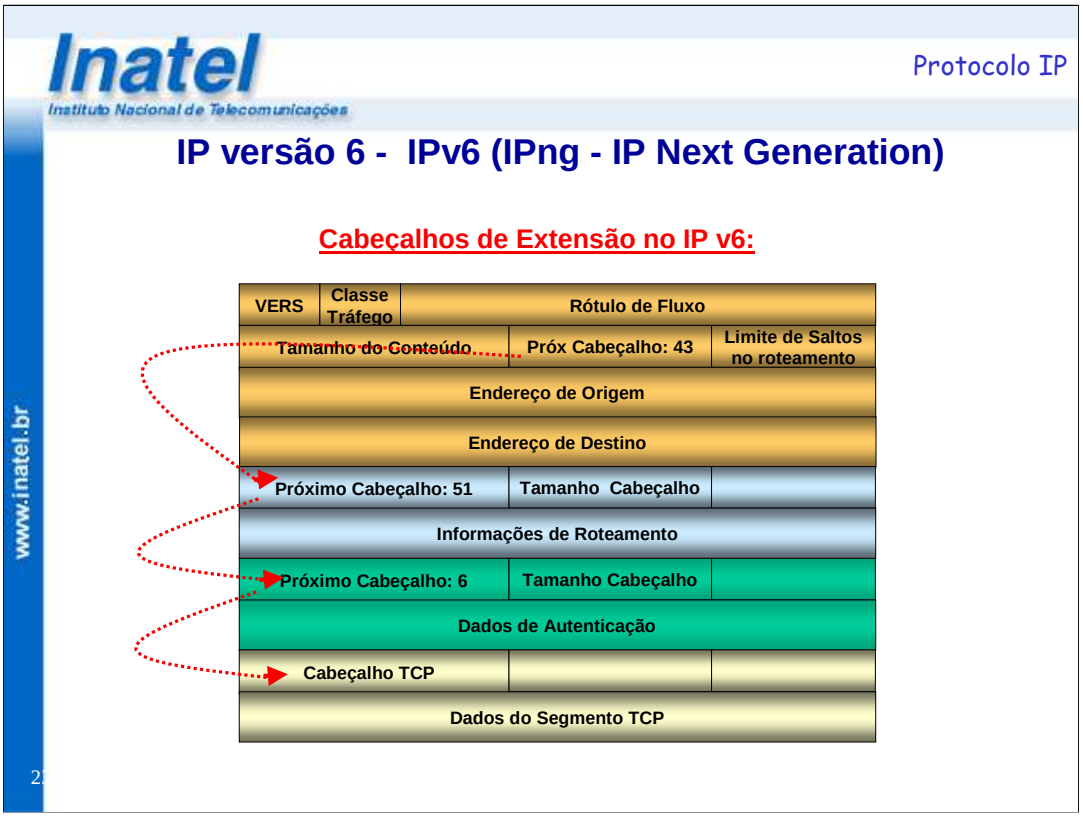
• O alinhamento foi mudado de múltiplos de 32 bits para múltiplos de 64 bits.

• O campo de comprimento do cabeçalho foi eliminado e o campo de comprimento de datagrama foi substituído por um campo COMPRIMENTO DO PAYLOAD.

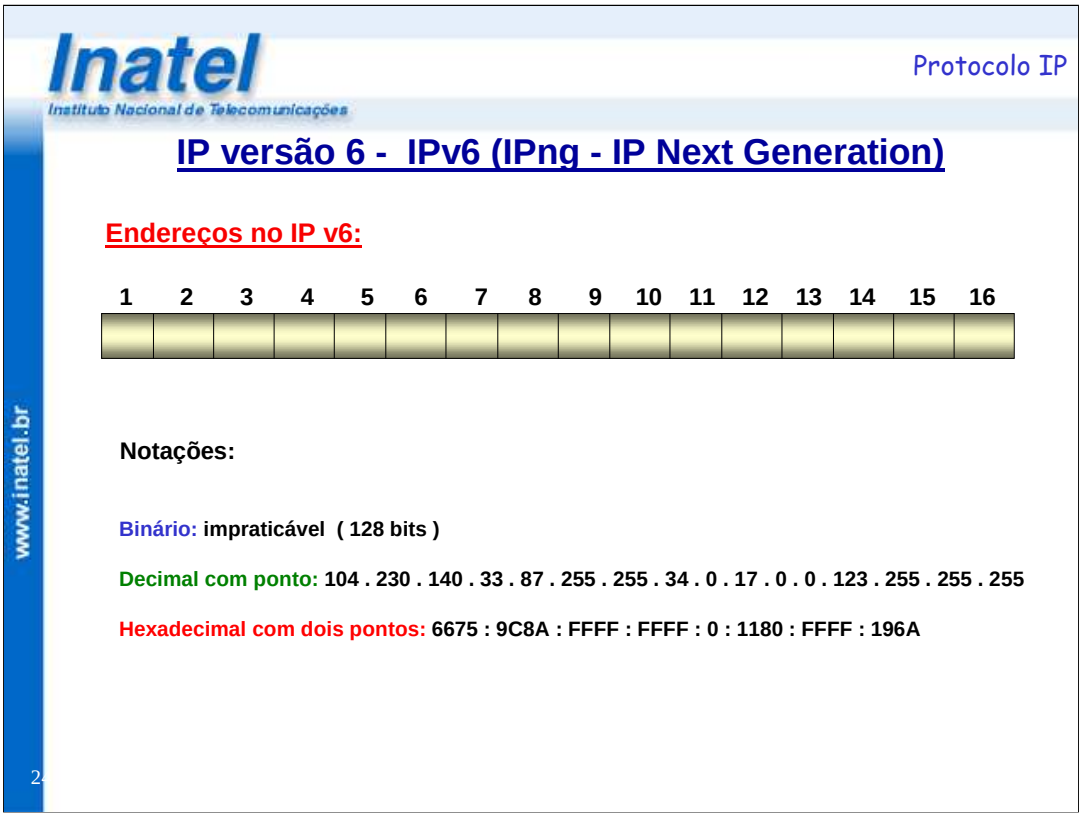
• O tamanho dos campos de endereço de origem e de destino foi aumentado para 16 octetos (128 bits) cada.

• As informações de fragmentação foram retiradas de campos fixos do cabeçalho básico, para um cabeçalho de expansão.

• O campo TEMPO DE VIDA foi substituído por um campo LIMITE DE PASSOS DE ROTA.

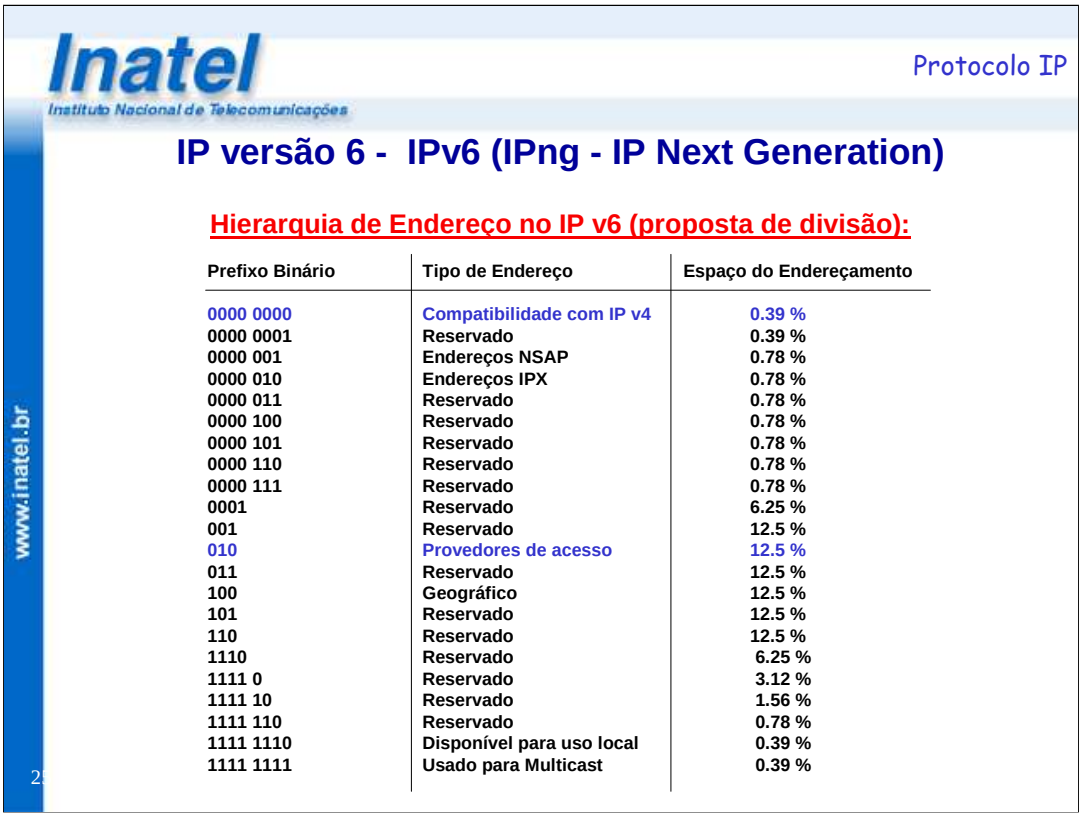


O paradigma de um cabeçalho fixo, seguido de um conjunto de cabeçalhos de extensão opcionais, foi escolhido como uma acomodação entre a generalidade e a eficiência. Para ser totalmente geral, o IP v6 precisa incluir mecanismos a fim de aceitar funções como fragmentação, roteamento de origem e autenticação. Entretanto, a opção por alocar campos fixos no cabeçalho do datagrama para todos os mecanismos não é eficaz, porque a maioria dos datagramas não utilizam todos os mecanismos. No IP v6 um transmissor pode optar por escolher quais cabeçalhos de extensão incluir em determinado datagrama e quais omitir. Assim, os cabeçalhos de extensão fornecem flexibilidade máxima ao IP v6.



No IP v6, cada endereço ocupa 16 octetos, quatro vezes o tamanho de um endereço IP v4. É difícil compreender o tamanho do espaço de endereço do IP v6. Um modo de examiná-lo consiste em relacionar a magnitude ao tamanho da população do planeta, ou seja, cada pessoa pode ter endereços suficientes para ter sua própria interligação em redes tão grande quanto a Internet atual.

Embora solucione os problemas de capacidade insuficiente, o tamanho grande do endereço IP v6 cria um novo problema. A notação usada para representar esse número de 16 octetos (128 bits). A notação binária seria impraticável (128 bits). A notação decimal com ponto seriam 16 números separados por pontos. Uma maneira mais compacta proposta foi a notação hexadecimal de dois pontos, ou seja, seriam 8 números hexadecimais (de 4 dígitos) separados por dois pontos (:). Esta parece ser a maneira ideal de representação do endereço no IP v6.

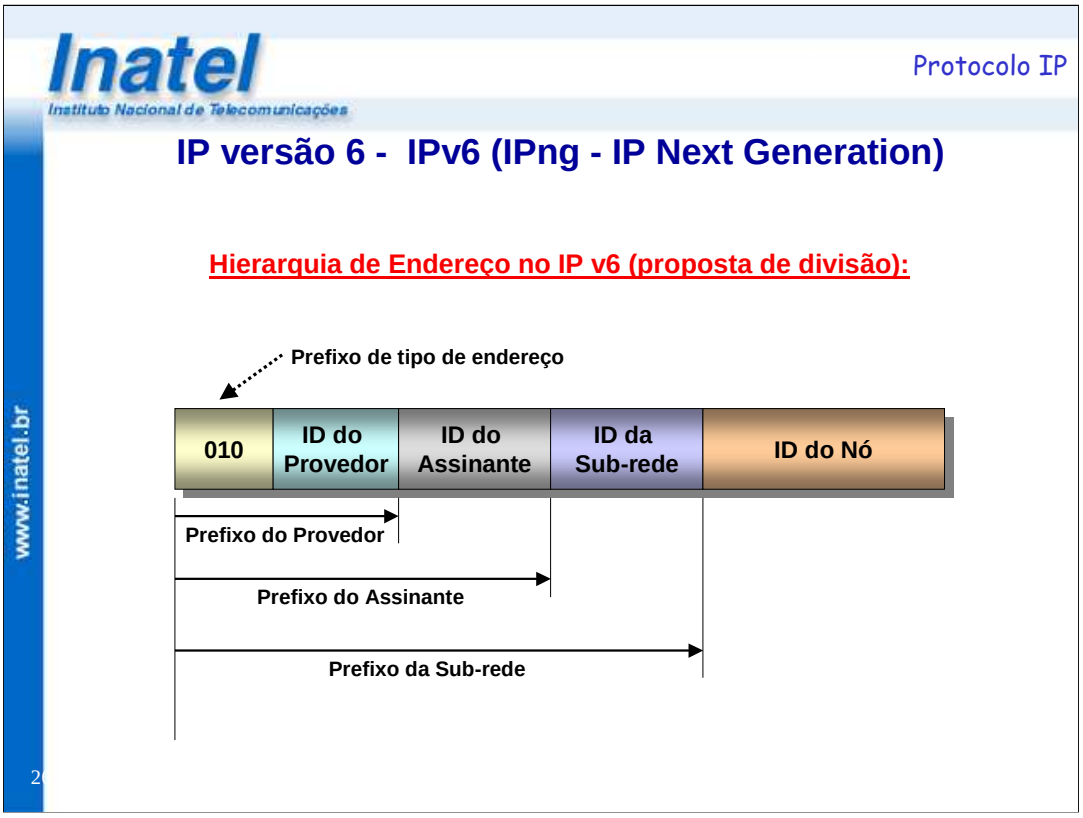


Prefixo Binário	Tipo de Endereço	Espaço do Endereçamento
0000 0000	Compatibilidade com IP v4	0.39 %
0000 0001	Reservado	0.39 %
0000 001	Endereços NSAP	0.78 %
0000 010	Endereços IPX	0.78 %
0000 011	Reservado	0.78 %
0000 100	Reservado	0.78 %
0000 101	Reservado	0.78 %
0000 110	Reservado	0.78 %
0000 111	Reservado	0.78 %
0001	Reservado	6.25 %
001	Reservado	12.5 %
010	Provedores de acesso	12.5 %
011	Reservado	12.5 %
100	Geográfico	12.5 %
101	Reservado	12.5 %
110	Reservado	12.5 %
1110	Reservado	6.25 %
1111 0	Reservado	3.12 %
1111 10	Reservado	1.56 %
1111 110	Reservado	0.78 %
1111 1110	Disponível para uso local	0.39 %
1111 1111	Usado para Multicast	0.39 %

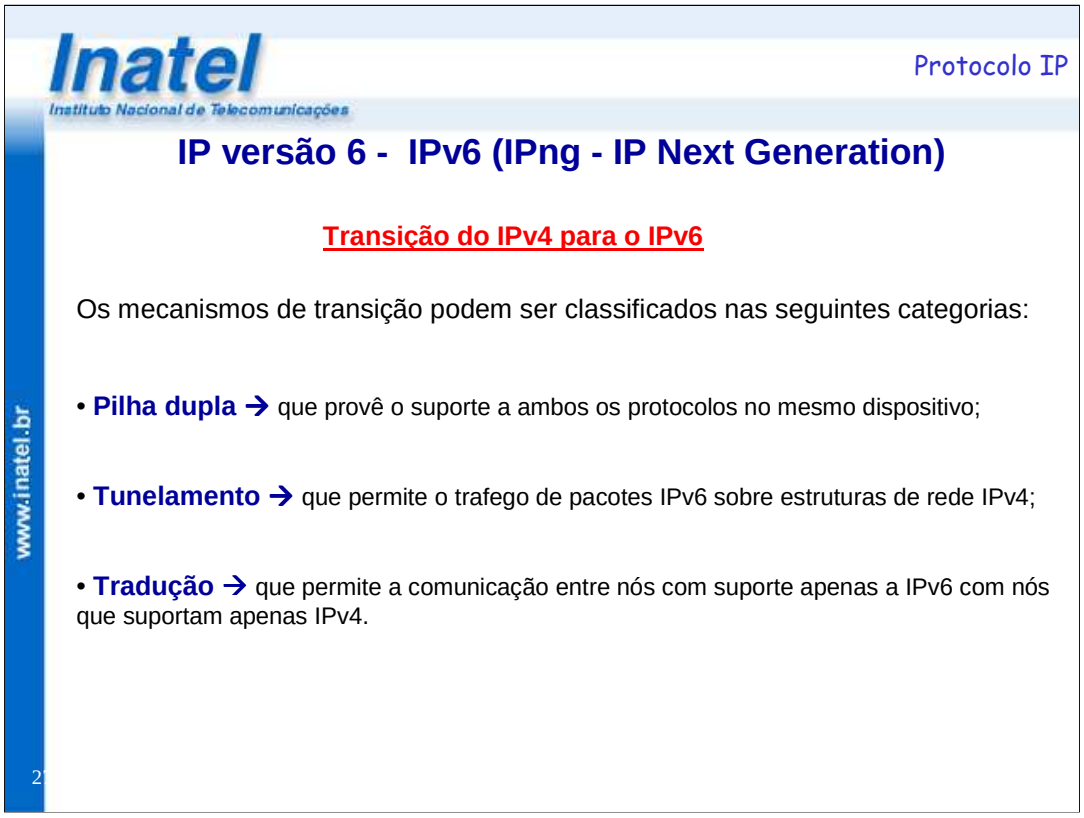
Como ilustrado, os projetistas do IP v6 propõem a atribuição de classes de endereço de modo semelhante ao esquema usado para IP v4. Embora os oito primeiros bits de um endereço sejam suficientes para identificar seu tipo, o espaço de endereço não é partilhado em seções de igual tamanho.

O espaço de endereços no IP v6 fica então organizado usando-se prefixos de formato, semelhante aos códigos de telefone de países e áreas, que logicamente divide-o na forma de uma árvore a fim de que a rota de uma rede para outra possa ser facilmente encontrada.

A forma de codificar um endereço de IP v4 em um endereço de IP v6 não soluciona o problema de tornar as duas versões interoperacionais. Além da codificação do endereço, a conversão de datagramas é necessária quando um computador IP v6 gerar um datagrama para o endereço de destino de um computador ainda com IP v4. Será necessário que o computador IP v6 envie o datagrama para um conversor que usa o IP v4 para se comunicar com o destino. Quando o conversor recebe uma resposta do destino, converte o datagrama do IP v4 para IP v6 e o devolve à origem.

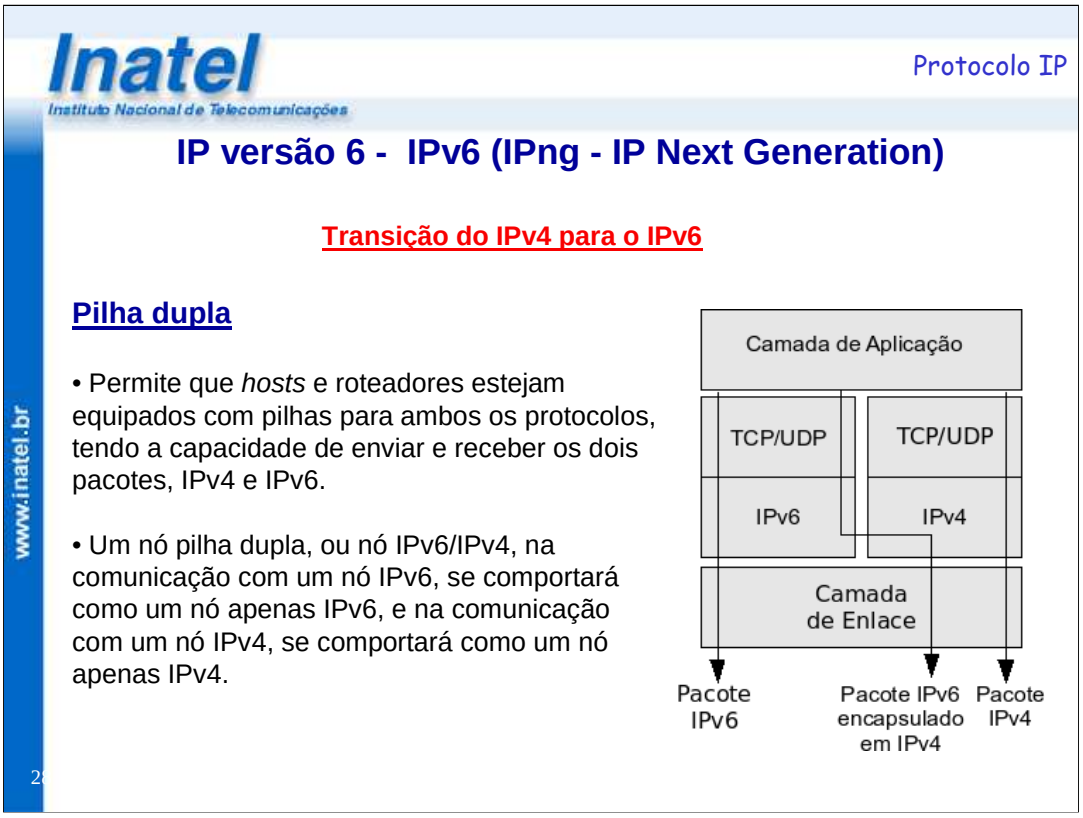


Um exemplo, irá ajudar a esclarecer como os projetistas imaginam o uso de endereços IP v6. A hierarquia de endereço do IP v6 para um endereço atribuído por um provedor de acesso à rede. A autoridade da Internet (IANA) atribui a cada provedor uma única ID, o provedor atribui a cada assinante uma única ID e o assinante atribui uma ID a cada sub-rede e cada nó (host).



Com o intuito de facilitar o processo de transição entre as duas versões do Protocolo Internet (IP), algumas técnicas foram desenvolvidas para que toda a base das redes instaladas sobre IPv4 mantenha-se compatível com o protocolo IPv6, sendo que nesse primeiro momento de co-existência entre os dois protocolos, essa compatibilidade torna-se essencial para o sucesso da transição para o IPv6.

Como o período de co-existência entre os dois protocolos pode durar indefinidamente, a implementação de métodos que possibilitem a interoperabilidade entre o IPv4 e o IPv6, poderá garantir uma migração segura para o novo protocolo, através da realização de testes que permitam conhecer as opções que estes mecanismos oferecem, além de evitar, no futuro, o surgimento de “ilhas” isoladas de comunicação.

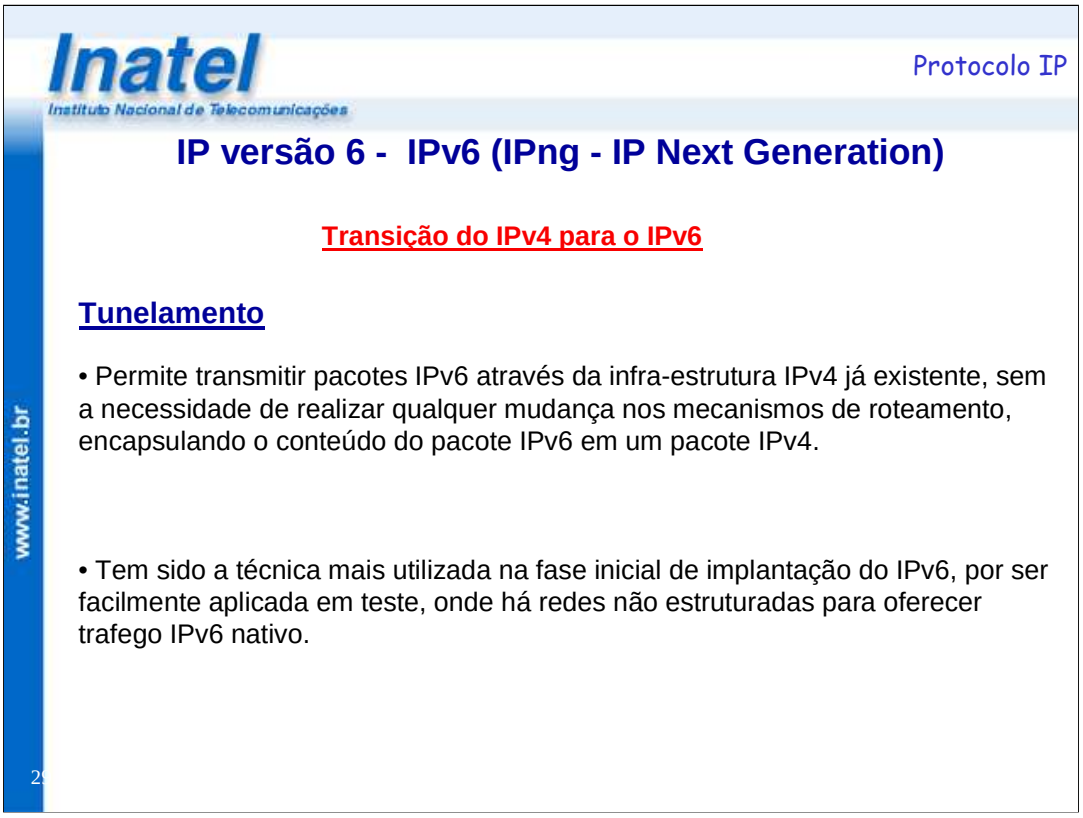


Pilha dupla

Nesta fase inicial de implementação do IPv6, ainda não é aconselhável ter nós com suporte apenas a esta versão do protocolo IP, visto que muitos serviços e dispositivos de rede ainda trabalham somente sobre IPv4. Deste modo, uma possibilidade é a de se introduzir o método conhecido como **pilha dupla**. A utilização deste método permite que *hosts* e roteadores estejam equipados com pilhas para ambos os protocolos, tendo a capacidade de enviar e receber os dois pacotes, IPv4 e IPv6. Com isso, um nó pilha dupla, ou nó IPv6/IPv4, na comunicação com um nó IPv6, se comportará como um nó apenas IPv6, e na comunicação com um nó IPv4, se comportará como um nó apenas IPv4.

Cada nó IPv6/IPv4 é configurado com ambos endereços, utilizando mecanismos IPv4 (ex. DHCP) para adquirir seu endereço IPv4, e mecanismos do protocolo IPv6 (ex. auto-configuração e/ou DHCPv6) para adquirir seu endereço IPv6.

Este método de transição pode facilitar o gerenciamento da implantação do IPv6, por permitir que este seja feito de forma gradual, configurando pequenas seções do ambiente de rede de cada vez. Além disso, caso no futuro o IPv4 não seja mais usado, basta simplesmente desabilitar a pilha IPv4 de cada nó.



A técnica de criação de túneis, ou tunelamento, permite transmitir pacotes IPv6 através da infra-estrutura IPv4 já existente, sem a necessidade de realizar qualquer mudança nos mecanismos de roteamento, encapsulando o conteúdo do pacote IPv6 em um pacote IPv4.

Essa técnica, tratada na RFC 4213, têm sido a mais utilizada na fase inicial de implantação do IPv6, por ser facilmente aplicada em teste, onde há redes não estruturadas para oferecer trafego IPv6 nativo.

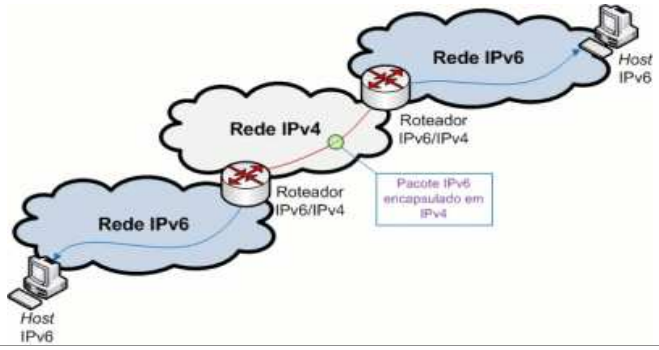
IP versão 6 - IPv6 (IPng - IP Next Generation)

Transição do IPv4 para o IPv6

Tunelamento (continuação)

Os túneis podem ser configurados nos seguintes modos:

• **Roteador-a-Roteador** – roteadores IPv6/IPv4, conectados via rede IPv4, podem trocar pacotes IPv6 entre si, ligando um segmento no caminho entre dois *hosts*;



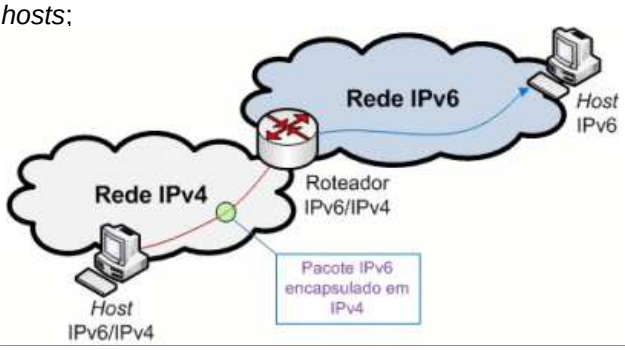
IP versão 6 - IPv6 (IPng - IP Next Generation)

Transição do IPv4 para o IPv6

Tunelamento (continuação)

Os túneis podem ser configurados nos seguintes modos:

• **Host-a-Roteador** - *hosts* IPv6/IPv4 enviam pacotes IPv6 a um roteador IPv6/IPv4 intermediário via rede IPv4, ligando o primeiro segmento no caminho entre dois *hosts*;



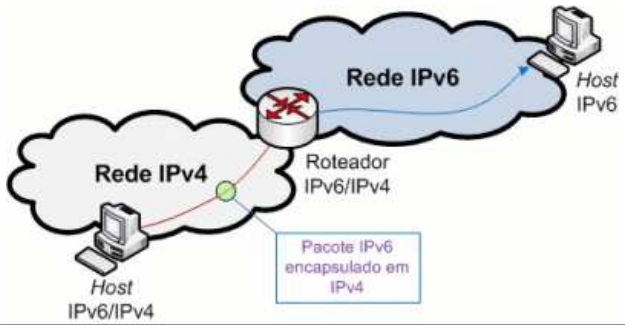
IP versão 6 - IPv6 (IPng - IP Next Generation)

Transição do IPv4 para o IPv6

Tunelamento (continuação)

Os túneis podem ser configurados nos seguintes modos:

• **Roteador-a-Host** - roteadores IPv6/IPv4 enviam pacotes IPv6 ao destino final IPv6/IPv4, ligando o último segmento do caminho entre dois *hosts*;



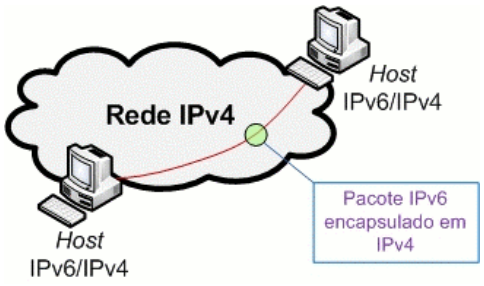
IP versão 6 - IPv6 (IPng - IP Next Generation)

Transição do IPv4 para o IPv6

Tunelamento (continuação)

Os túneis podem ser configurados nos seguintes modos:

• **Host-a-Host** - *hosts* IPv6/IPv4, conectados via rede IPv4, trocam pacotes IPv6 entre si, ligando todo o caminho entre os dois *hosts*.



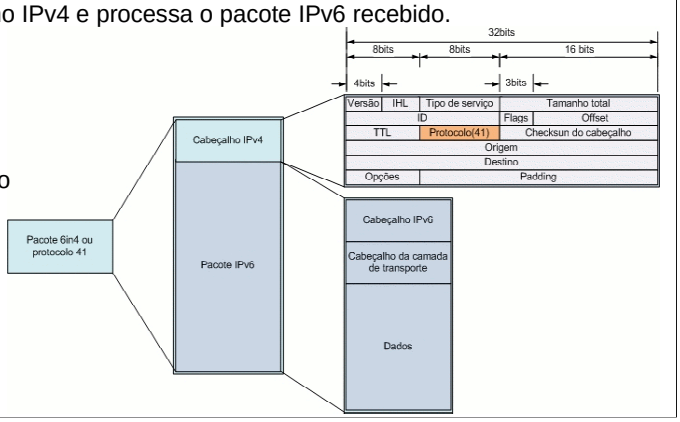
IP versão 6 - IPv6 (IPng - IP Next Generation)

Transição do IPv4 para o IPv6

Tunelamento (continuação)

Encapsulamento dos Pacotes → O nó de entrada do túnel, cria um cabeçalho IPv4 com o pacote IPv6 encapsulado e o transmite através da rede IPv4. O nó de saída recebe o pacote encapsulado, retira o cabeçalho IPv4 e processa o pacote IPv6 recebido.

Este processo de encapsulamento, conhecido como **6in4**, é identificado como protocolo do tipo 41 e sua utilização é comum em algumas técnicas de tunelamento, como 6to4, ISATAP e Tunnel Broker.



IP versão 6 - IPv6 (IPng - IP Next Generation)

Transição do IPv4 para o IPv6

Tradução

• Possibilitam um roteamento transparente na comunicação entre nós que apresentem suporte apenas a uma versão do protocolo IP, ou utilizem pilha dupla.

• Podem atuar de diversas formas e em camadas distintas, traduzindo cabeçalhos IPv4 em cabeçalhos IPv6 e vice-versa, realizando conversões de endereços, de APIs de programação, ou atuando na troca de trafego TCP ou UDP.

Os principais mecanismos de tradução utilizados são:

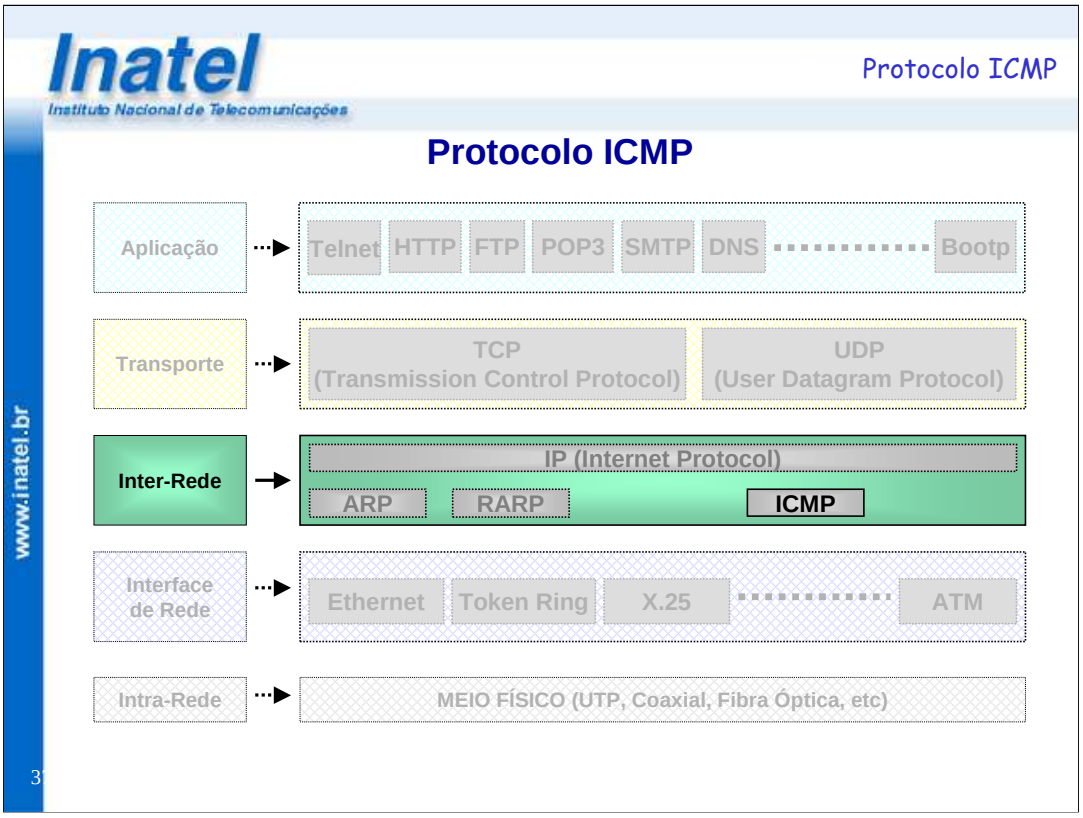
SIIT (*Stateless IP/ICMP Translation Algorithm*)
NAT-PT (*Network Address Translation with Protocol Translation*)
NAPT-PT (*Network Address Port Translation and Packet Translation*)
BIS (*Bump in the Stack*)
BIA (*Bump in the API*)
TRT (*Transport Relay Translator*)
SOCKS64 (*Socks-Based IPv6/IPv4 Gateway*)
ALG (*Application Layer Gateway*)

Inatel
Instituto Nacional de Telecomunicações

www.inatel.br

Capítulo 4 – Protocolo ICMP

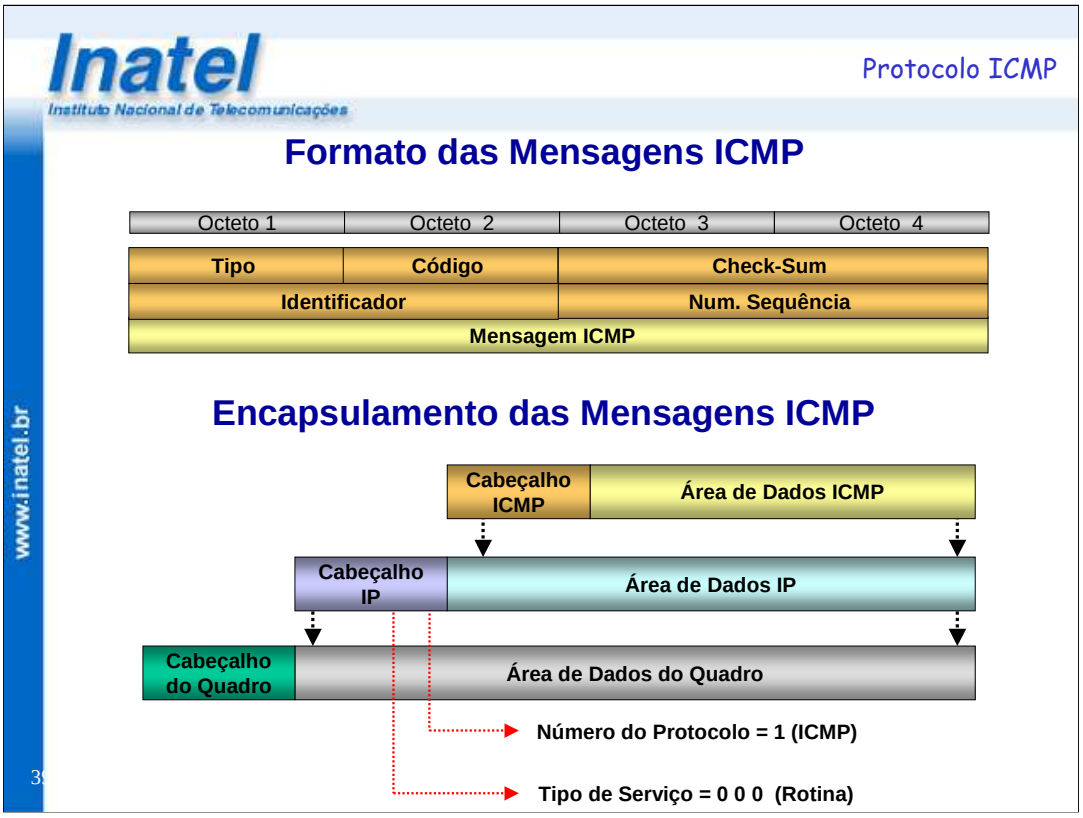
- Formato das Mensagens ICMP
- Tipos de Mensagens ICMP
  - Solicitação de Eco / Resposta de Eco
  - Destino Inatingível
  - Tempo Esgotado (time-out)
  - Source Quench
  - Redirecionamento



O ICMP (Internet Control Message Protocol), é um protocolo que faz parte da camada Inter-Rede na pilha de protocolos TCP / IP. É usado para permitir que os roteadores de uma interligação inter-redes informem os erros ou forneçam informações sobre ocorrências inesperadas. É considerado uma parte necessária do IP e deve ser incluído em cada implementação de IP.

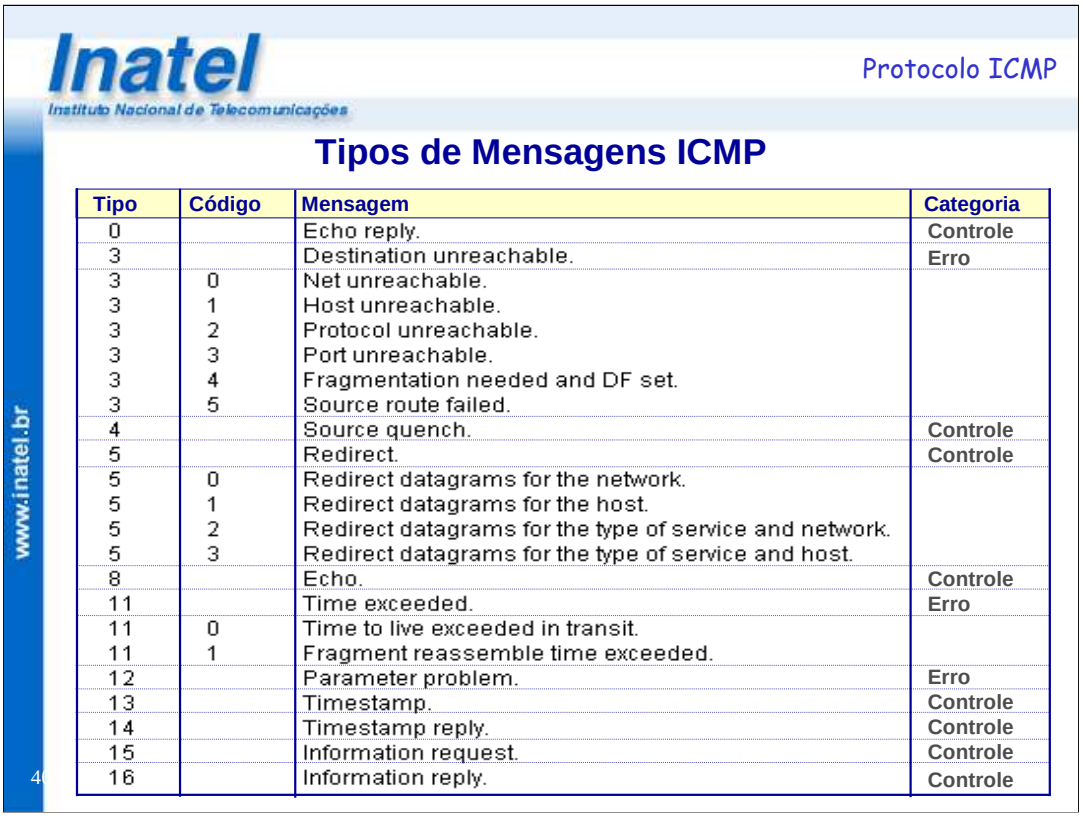
Assim como ocorre com o restante do tráfego, as mensagens ICMP trafegam pela interligação em redes na parte de dados dos datagramas IP. Entretanto, o destino final de uma mensagem ICMP não é um programa aplicativo ou do usuário na máquina de destino, mas sim o software IP daquela máquina. Ou seja, quando chega uma mensagem de erro ICMP, o módulo do software ICMP trata dela. Naturalmente, se o ICMP determinar que determinado protocolo de alto nível ou um programa de aplicativo causou o problema, informará ao módulo apropriado.

Quando um roteador ou host de destino precisa informar o host de origem sobre os erros no processamento de um datagrama, ele usa o ICMP.



As mensagens ICMP precisam de dois níveis de encapsulamento. Cada mensagem ICMP trafega pela interligação em redes na parte dos dados de um datagrama IP que, por sua vez, trafega por cada rede física na parte dos dados de um quadro. Datagramas que transportam mensagens ICMP são roteados exatamente como os que levam informações aos usuários; não há nenhuma confiabilidade ou prioridade adicional. Portanto, as próprias mensagens de erro podem ficar perdidas ou descartadas. Além disso, numa rede já congestionada a mensagem de erro pode causar mais congestionamento.

É importante lembrar que, embora mensagens ICMP sejam encapsuladas e enviadas usando IP, o ICMP não é considerado um protocolo de alta prioridade. O motivo de usar IP para conduzir mensagens ICMP é que elas precisam trafegar por várias redes físicas para alcançar seu destino final. Portanto, não podem ser entregues somente através do transporte físico.



Tipo	Código	Mensagem	Categoria
0		Echo reply.	Controle
3		Destination unreachable.	Erro
3	0	Net unreachable.	
3	1	Host unreachable.	
3	2	Protocol unreachable.	
3	3	Port unreachable.	
3	4	Fragmentation needed and DF set.	
3	5	Source route failed.	
4		Source quench.	Controle
5		Redirect.	Controle
5	0	Redirect datagrams for the network.	
5	1	Redirect datagrams for the host.	
5	2	Redirect datagrams for the type of service and network.	
5	3	Redirect datagrams for the type of service and host.	
8		Echo.	Controle
11		Time exceeded.	Erro
11	0	Time to live exceeded in transit.	
11	1	Fragment reassemble time exceeded.	
12		Parameter problem.	Erro
13		Timestamp.	Controle
14		Timestamp reply.	Controle
15		Information request.	Controle
16		Information reply.	Controle

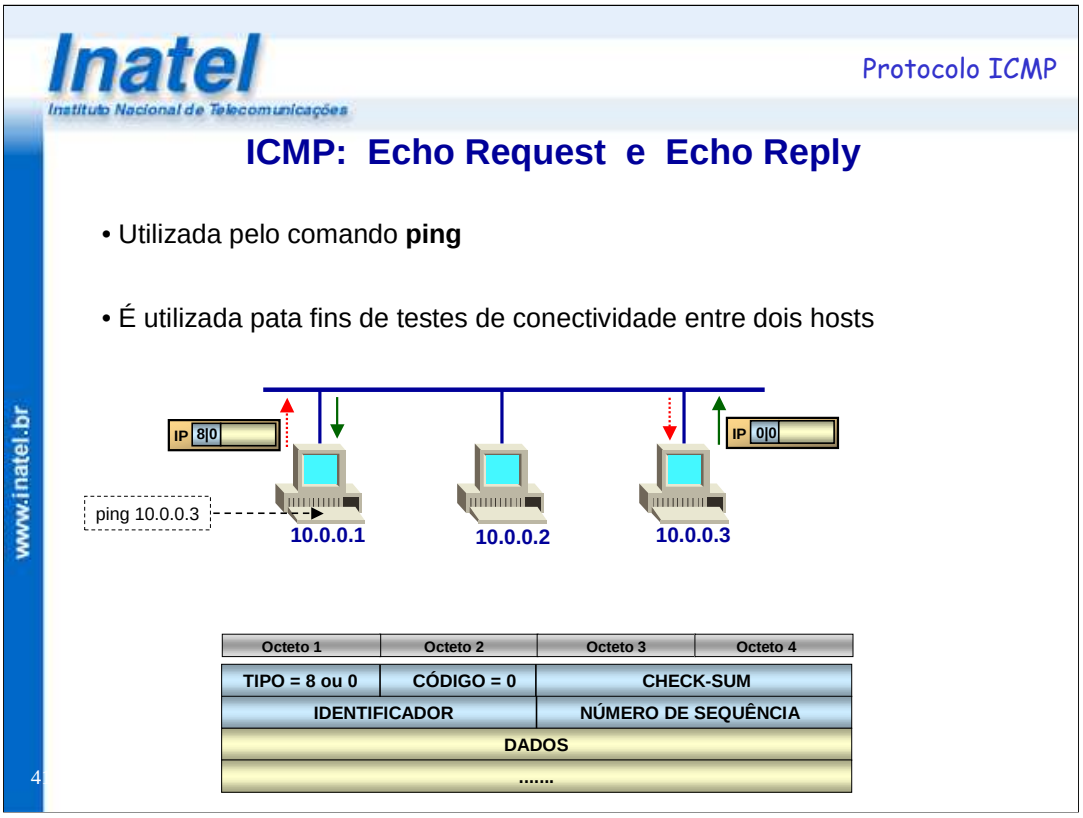
Embora cada mensagem ICMP tenha seu próprio formato, todas começam com os mesmos três campos:

TIPO: É o tipo de mensagem, com oito bits, que identifica a mensagem ICMP.

CÓDIGO: Com oito bits, que fornece informações adicionais sobre o tipo de mensagem.

CHECK-SUM: É a soma de verificação, com 16 bits, que faz a somatória em todos os bytes de mensagem ICMP.

Além disso, as mensagens ICMP que tratam de erros sempre incluem o cabeçalho e os primeiros 64 bits (8 bytes) de dados do datagrama que causou o problema. A razão para retornar mais do que apenas o cabeçalho do datagrama é permitir que o receptor determine com maior precisão qual(is) protocolo(s) e qual programa de aplicação foram responsáveis pelo datagrama. Como veremos mais adiante, os protocolos de alto nível na seqüência TCP / IP são projetados para que essa informação crucial esteja codificada nos primeiros 64 bits (8 bytes).

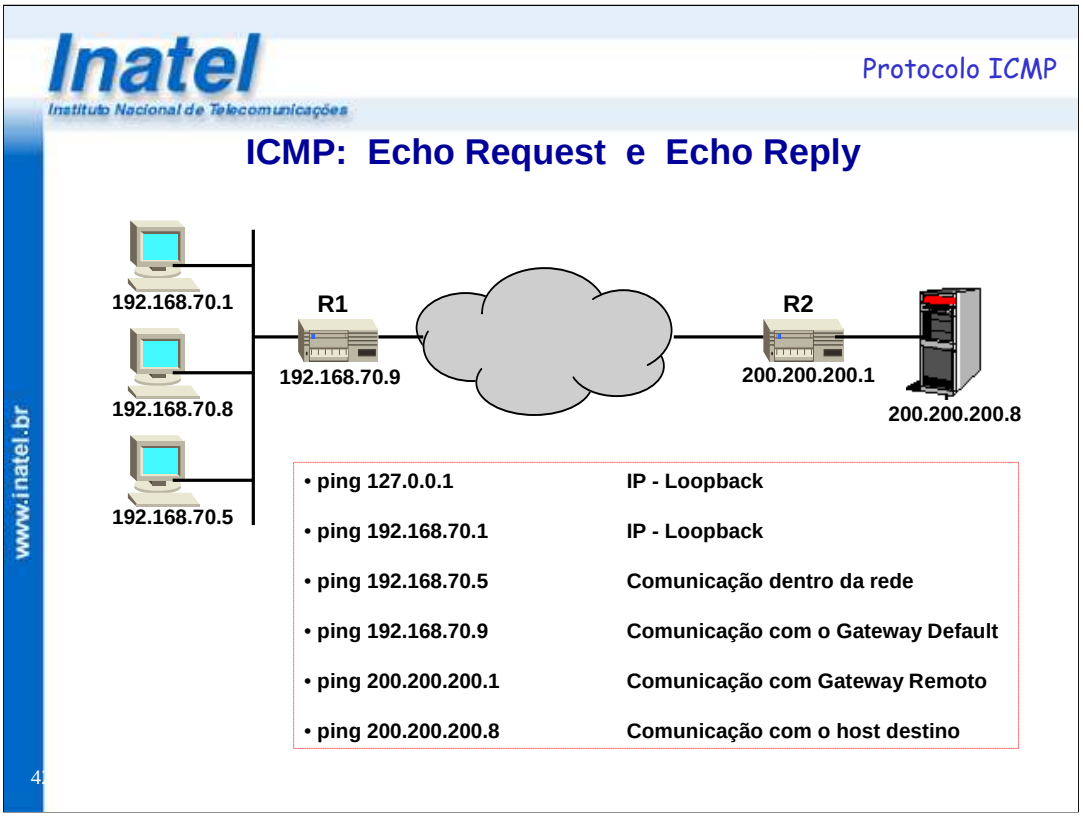


Echo Request e Echo Reply

Utilizada pelo comando ping, a mensagem Echo Request enviada para um host causa o retorno de uma mensagem Echo Reply. É utilizada principalmente para fins de testes de conectividade entre as duas máquinas.

O ping usa as mensagens de echo request e eco reply para determinar se um host está ao alcance. Ele envia um ou mais datagramas IP para um host de destino específico solicitando uma resposta e mede o tempo da viagem de ida e volta.

Tradicionalmente, se você pudesse “pingar” um host, outros aplicativos tais como o Telnet ou o FTP também poderiam alcançar aquele host. Com o advento das medidas de segurança na Internet, especialmente os firewalls, que controlam o acesso às redes pelo protocolo de aplicação e/ou número da porta, isto não é mais necessariamente verdade. Entretanto, o primeiro teste de alcance para um host ainda é o ping.

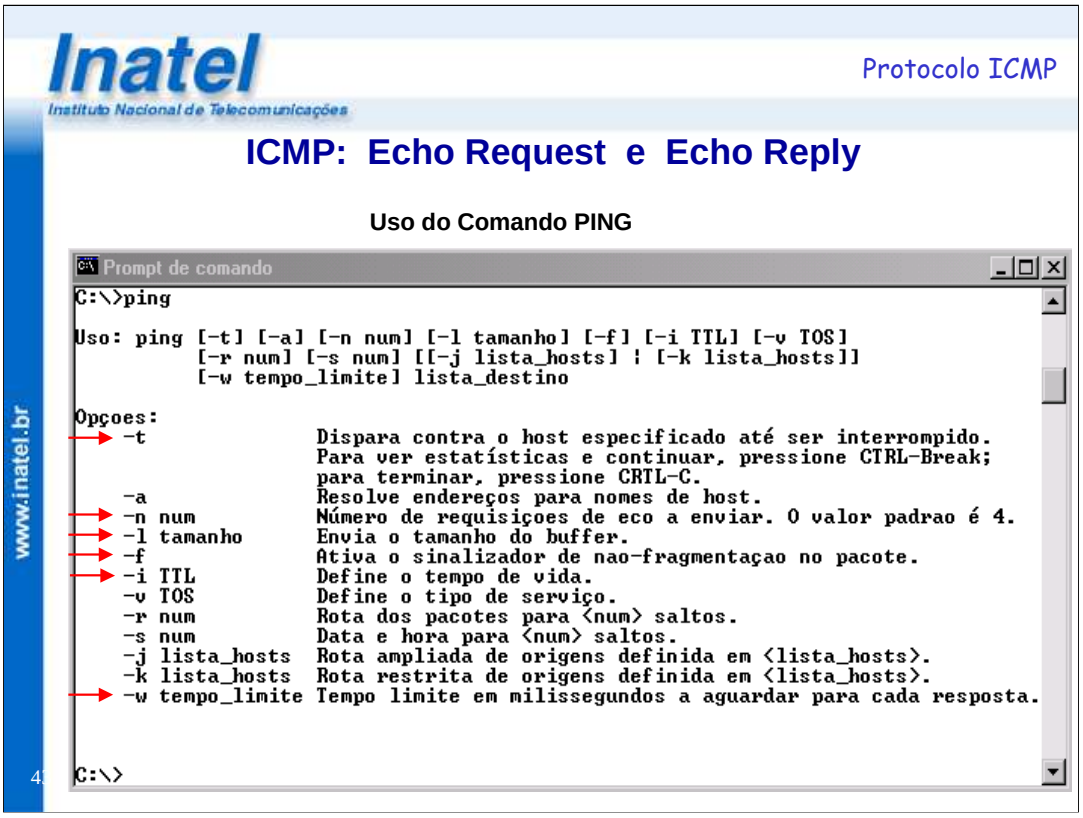


Caso você venha a ter problemas de comunicação, todas as pilhas TCP/IP, independente de qual sistema operacional, trazem o utilitário ping para testar a conectividade entre dois hosts TCP/IP. Siga o seguinte procedimento:

• .**ping 127.0.0.1**. Este endereço IP é um loopback, ou seja, não vai para a rede, fica no computador que originou a mensagem. Se o ping acusar o recebimento da resposta, significa que a pilha TCP/IP está instalada e ativa no computador onde foi realizado o teste.

• **ping no seu endereço IP**. Tem o mesmo efeito do anterior (ping 127.0.0.1). Ver tabela de rotas da máquina.

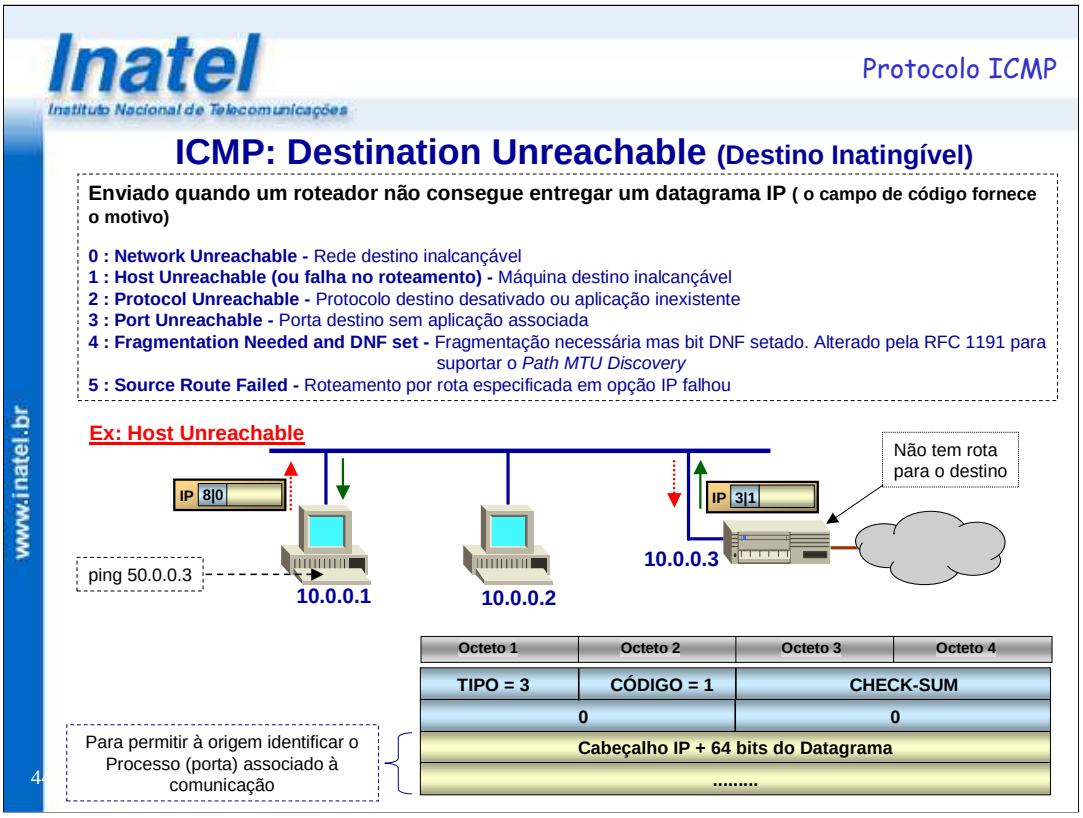
• **ping endereço IP de outro host na rede local**. Agora vamos testar a comunicação dentro da rede local onde o computador de origem está localizado. Garanta que o computador dono do ip_na_minha_rede está com o TCP/IP e a sua placa de rede ativos, segundo os dois testes acima. Se não funcionar, você tem um problema de cabos ou em uma placa de rede, ou simplesmente as suas máscaras de rede e endereços IP estão incorretos.



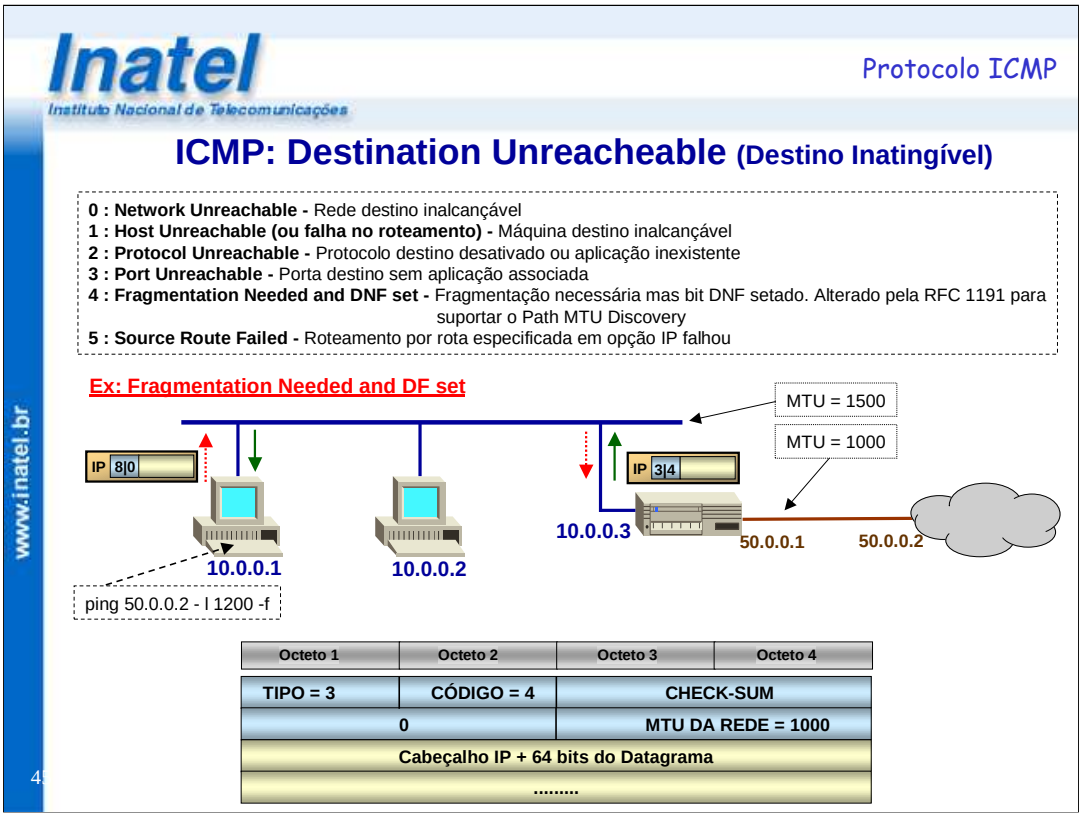
•.**ping no endereço do Gateway default**. Se a comunicação dentro da minha rede local está OK, temos que verificar se o default gateway da minha rede está no ar, pois todos os pacotes que saem da minha rede local passam por ele.

• .**ping no endereço IP do Gateway remoto**. Digamos que o meu default gateway esteja diretamente conectado na rede destino. Eu tenho que testar se a interface de rede que liga o default gateway a esta rede está no ar. Então eu dou um ping no endereço IP desta placa. Se o default gateway não estiver diretamente conectado na rede destino, eu repito os passos (4) e (5) para cada equipamento que esteja no caminho entre origem e destino.

• .**ping no endereço IP do host destino**. Sabendo que a outra rede pode ser alcançada via TCP/IP, resta saber se eu consigo me comunicar com o computador desejado.

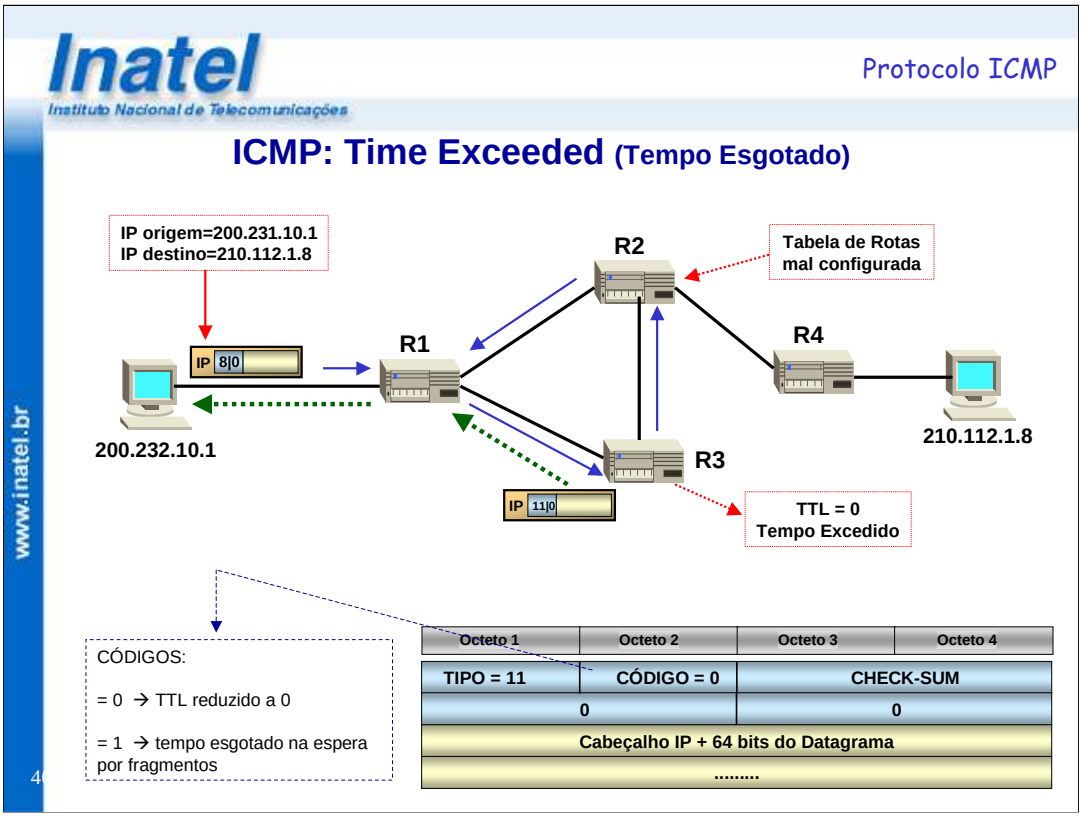


No exemplo mostrado, vemos o envio de uma mensagem ping (ICMP Echo Request) de um host da rede 10.0.0.0 para um host em outra rede (50.0.0.0) . A mensagem é enviada ao roteador (roteamento indireto) e ele verifica que a rota de destino para a mensagem não existe em sua tabela de rotas, então o roteador envia uma mensagem de erro ICMP para o host cliente que enviou a mensagem com o tipo de mensagem igual a 3 (Destino Inatingível) e o código igual a 1 (Host Inatingível). O host cliente ao receber esta mensagem ICMP, na camada inter-rede pelo software ICMP deve tratar esta mensagem de erro e o aplicativo do usuário informar que o destino não está disponível.

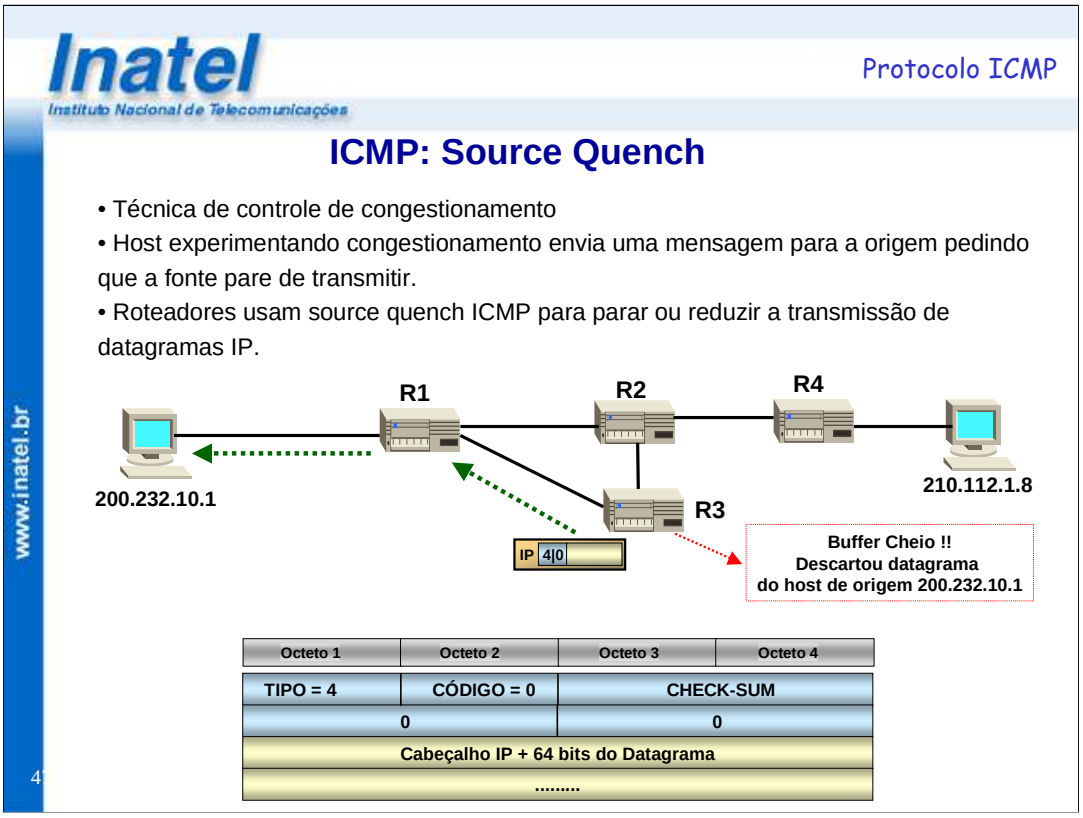


O sub-tipo Fragmentation Needed and DNF set é utilizado como forma de um host descobrir o menor MTU nas redes que serão percorridas entre a origem e o destino. Por meio desta mensagem, é possível enviar pacotes que não precisarão ser fragmentados, aumentando a eficiência da rede. Esta técnica, que forma um protocolo é denominado de ICMP MTU Discovery Protocol, definido na RFC 1191.

A operação é simples. Todo pacote IP enviado é marcado com o bit DNF (Do Not Fragment), que impede sua fragmentação nos roteadores. Desta forma, se uma pacote IP, ao passar por um roteador para chegar a outra rede com MTU menor, deva ser fragmentado, o protocolo IP não irá permitir e enviará uma mensagem ICMP Destination Unreacheable para o destino. Para suportar esta técnica, a mensagem ICMP foi alterada para informar o MTU da rede que causou o ICMP. Desta forma, a máquina origem saberá qual o valor de MTU que causou a necessidade de fragmentação, podendo reduzir o MTU de acordo, nos próximos pacotes.

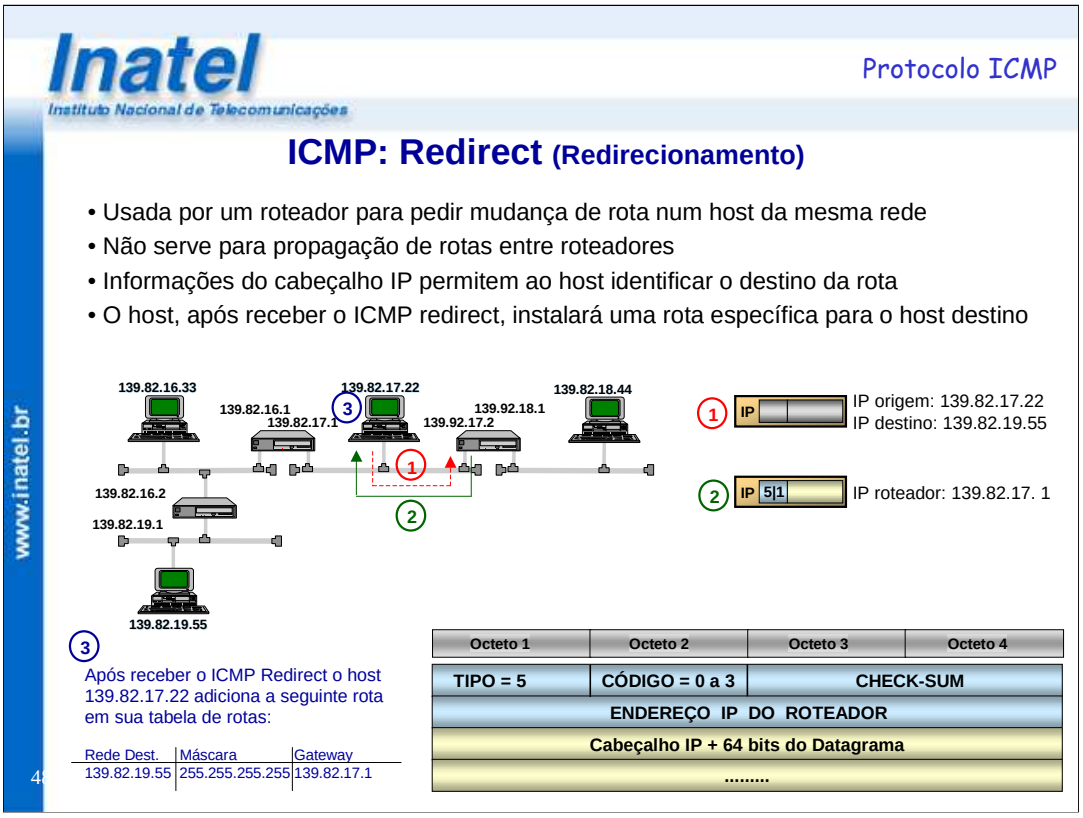


Neste exemplo, vemos uma situação onde uma mensagem foi enviada de um host origem para um host destino e por problemas na tabela de roteamento de um dos roteadores da interligação em redes, este provocou um loop na mensagem e, naturalmente o TTL foi sendo decrementado a cada roteador até chegar em zero (R3). Este roteador então envia uma mensagem de erro ICMP com tipo=11 ao host origem . Isto significa que a mensagem passou pelo limite de roteadores possíveis na rede, e por isso foi descartada.



Esta mensagem é utilizada por um roteador para informar à origem, que foi obrigado a descartar o pacote devido a incapacidade de roteá-lo devido ao tráfego.

Este tipo de mensagem é usada como técnica de controle de congestionamento através da qual uma máquina experimentando congestionamento envia uma mensagem de volta para a origem dos pacotes pedindo que a fonte pare de transmitir. Em uma rede TCP/IP, os roteadores usam source quench ICMP para parar ou reduzir a transmissão de datagramas IP.



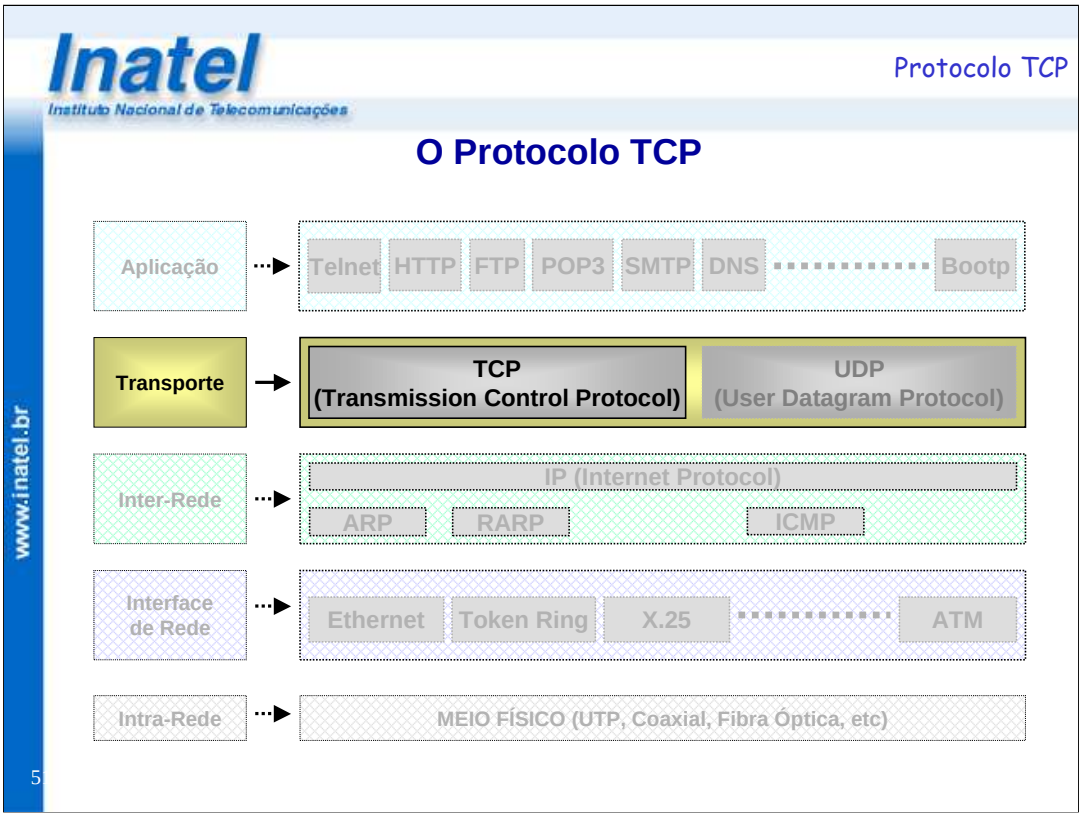
Esta mensagem, é utilizada por um roteador para informar ao host origem que existe uma rota direta mais adequada através de outro roteador. O host, após receber a mensagem ICMP, instalará uma rota específica para aquele host destino.

A operação do ICMP Redirect ocorre conforme a figura apresentada. Note que a rota instalada é uma rota específica para host, com máscara 255.255.255.255, não servindo para outras máquinas na mesma rede. Se uma máquina se comunica com 10 máquinas em outra rede e se basear em ICMP Redirect para aprender as rotas, ele instalará pelo menos 10 entradas na tabela de rede, uma para cada máquina

Algumas observações devem ser levadas em consideração com relação as mensagens de ICMP de erro. Nas descrições acima estão algumas condições onde uma mensagem ICMP de erro não é gerada como resposta, para evitar um broadcast storm na rede (explosão de mensagens).

Os capítulos anteriores exploram o serviço de transmissão de pacotes sem conexão, não-confiável, que forma a base para toda a comunicação de interligação em redes e o protocolo IP que a define. Este capítulo apresenta o segundo serviço conhecido e mais importante, em nível de rede, a transmissão de fluxo (stream) de dados confiável e o TCP (Transmission Control Protocol) que o define.

Veremos que o TCP acrescenta uma funcionalidade substancial aos protocolos já abordados, mas que sua implementação também é substancialmente mais complexa.



O TCP (Transmission Control Protocol) é um protocolo da camada de Transporte na pilha de protocolo TCP / IP.

O TCP fornece muito mais possibilidades para os aplicativos do que o UDP, principalmente recuperação de erros, controle de fluxo e confiabilidade. O TCP é um protocolo orientado para a conexão, ao contrário do UDP que é sem conexão. A maioria dos protocolos de aplicativo de usuário, como o Telnet e o FTP utilizam o TCP como transporte de dados.

O propósito inicial do TCP é fornecer um circuito lógico ou serviço de conexão confiável entre pares de processos. Como o TCP não conta com a confiabilidade dos protocolos de níveis inferiores (como o IP), ele deve garantir isto por si mesmo.

O TCP pode ser caracterizado pelas seguintes possibilidades que oferece aos aplicativos que fazem uso dele:

• **Transferência em fluxo (stream) de dados:** do ponto de vista do aplicativo, o TCP transfere pela rede um fluxo contínuo de bytes.

• **Confiabilidade:** O TCP designa um número seqüencial para cada pacote transmitido e espera um reconhecimento positivo (ACK) do TCP receptor. Se o ACK não for recebido dentro de um prazo estabelecido, os dados são retransmitidos.

• **Controle de fluxo:** Ao devolver um ACK para o remetente, o TCP receptor também indica ao remetente o número de bytes adicionais que ele pode receber além do último segmento TCP recebido, sem causar sobrecarga e estourar os buffers internos.

• **Multiplexação:** É alcançado através do uso de portas, assim como acontece com o UDP.

O conceito de portas e sockets oferece uma maneira uniforme e única de identificar as conexões e os programas e hosts que estão nela conectados, independente das identificações de processos específicas de cada aplicação.

Estes conceitos são necessários para determinar exatamente que processo local em um determinado host se comunica de fato com cada processo em cada host remoto e usando qual protocolo.

Portas: cada processo que queira comunicar-se com outro processo identifica-se para o conjunto de protocolos TCP/IP utilizando uma ou mais portas. Uma porta é um número de 16 bits, usado pelo protocolo host a host para identificar para qual protocolo de nível superior ou programas de aplicativos deve entregar as mensagens que chegarem. Há dois tipos de portas:

• Bem Conhecidas: pertencem a servidores padrão. São controladas e designadas pela IANA. O motivo da existência das portas bem conhecidas é permitir que clientes possam encontrar os servidores sem informação de configuração. O número de portas bem conhecidas tem valores entre 1 e 1023.

A interface socket é uma de várias APIs (Application Programming Interfaces - Interface de Programação de Aplicativos) para os protocolos de comunicação, e foi planejada para ser uma interface de programação genérica.

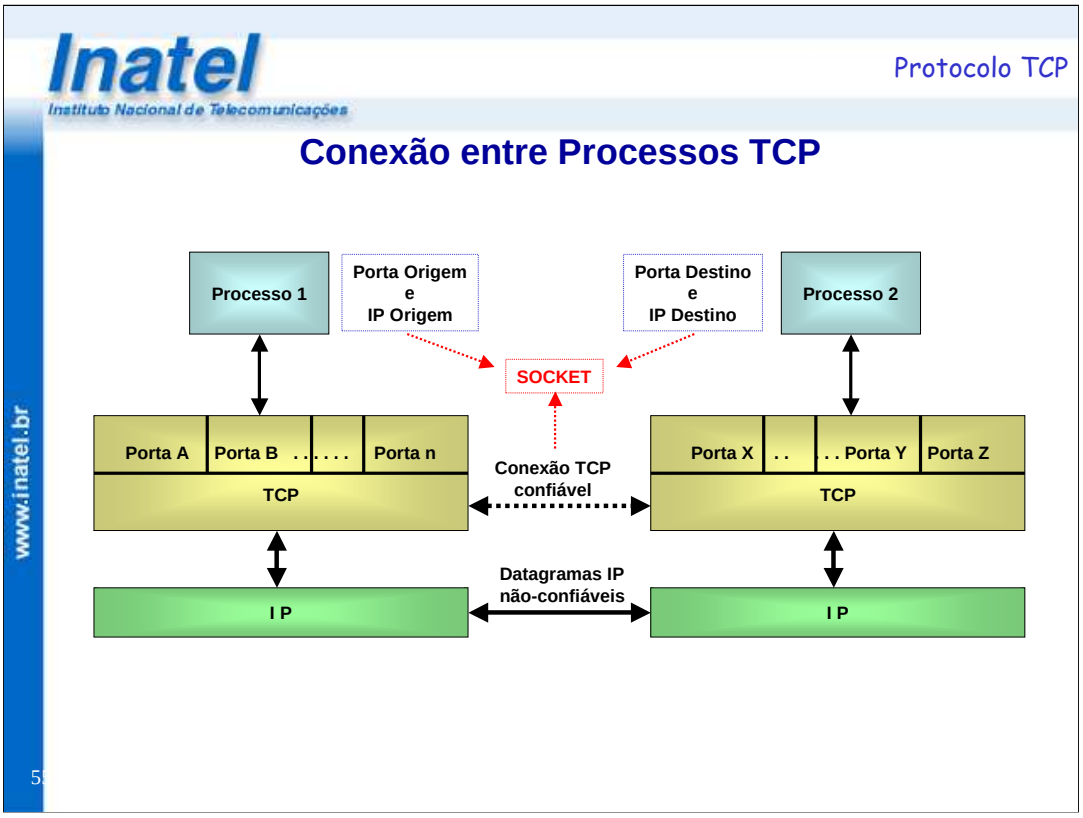
Um socket pode também ser definido como um tipo especial de handle de arquivo, que é usado por um processo para solicitar serviços de rede do sistema operacional.

Um endereço de socket é composto pelo trio:

{ protocolo, endereço-local, porta-local }

E uma associação é o quinteto que identifica os dois processos que estão na conexão:

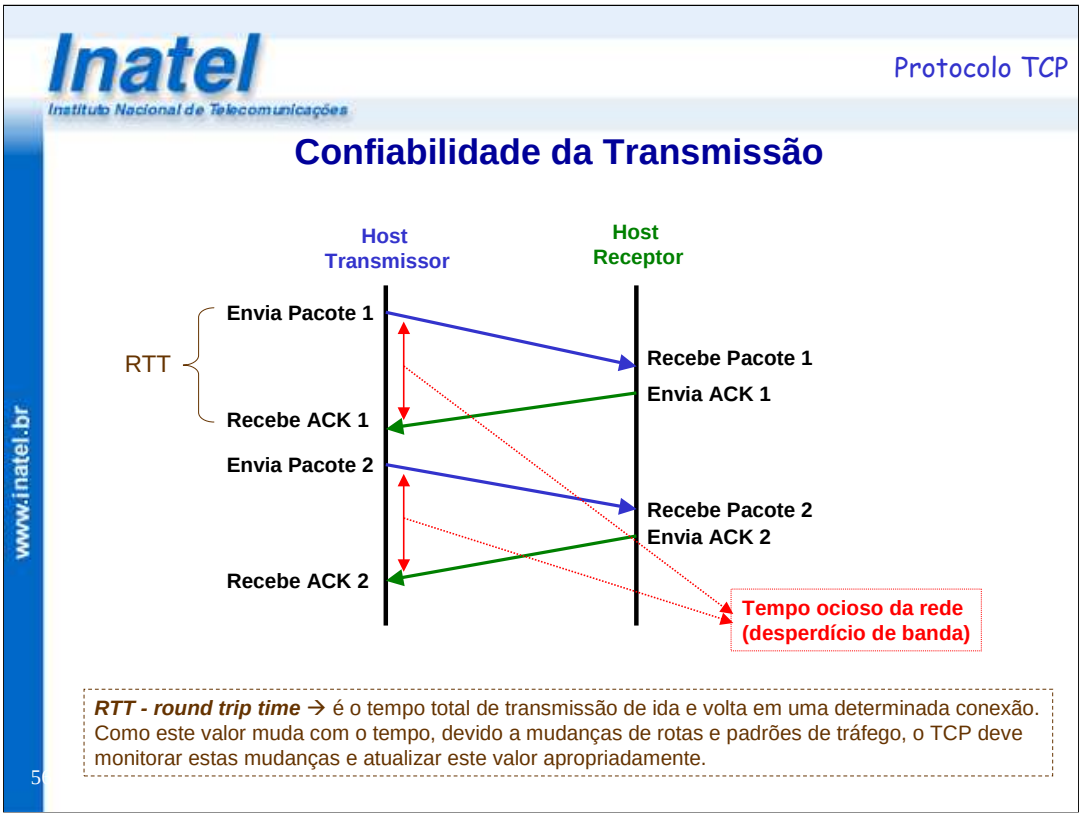
{ protocolo, endereço-local, porta-local, endereço-remoto, porta-remoto }



A comunicação entre dois processos de aplicativos no TCP é feito através de uma conexão chamada IPC (Interprocess Comunication – Comunicação entre processos), que utiliza o mesmo conceito de portas utilizado pelo UDP e define um novo conceito o de “sockets” (soquetes).

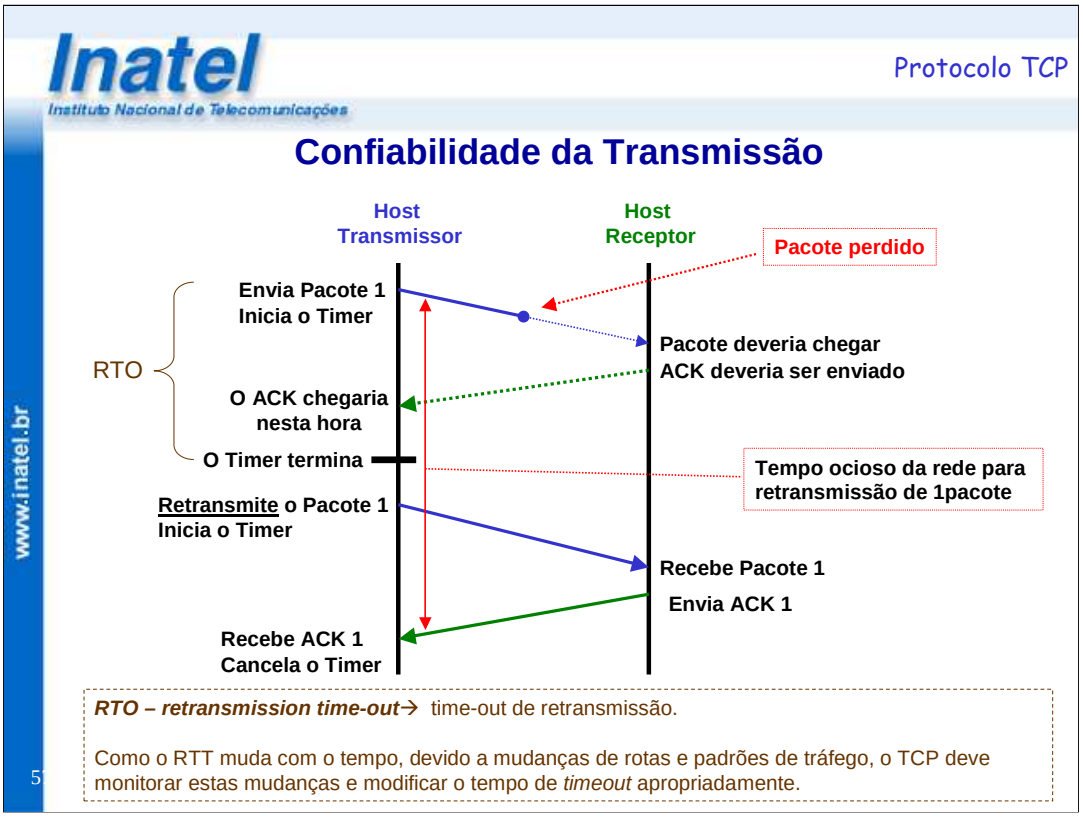
A interface soquete (**socket**) é uma das várias **APIs** (Interfaces de Programação de Aplicativos) para os protocolos de comunicação. Dois processos se comunicam via soquetes TCP. O modelo soquete fornece a um processo uma conexão full-duplex de fluxo de bytes (ou octetos) com outro processo. O aplicativo não precisa se preocupar com o gerenciamento deste fluxo; ele é fornecido pelo TCP.

O TCP usa portas efêmeras e bem conhecidas. Cada lado de uma conexão TCP tem um soquete que pode ser identificado pelo trio <***TCP, endereço IP, número da porta***>. Se dois processos estão se comunicando pelo TCP, eles têm uma conexão lógica que é identificada de maneira única pelos dois soquetes envolvidos, ou seja pela combinação <***TCP, endereço IP local, porta local, endereço IP remoto, porta remota***>. Os processos de servidores são capazes de gerenciar várias conversações através de uma única porta.



Um serviço de transmissão de stream (fluxo) de dados confiável deve garantir a entrega de um stream enviado de uma máquina a outra, sem duplicação ou perdas.

Podemos ficar com a seguinte dúvida: “como um protocolo pode oferecer uma transferência confiável se o sistemas básico de comunicação proporciona apenas transmissão de pacotes não-confiável ?”. A resposta é explicada pela técnica conhecida como “confirmação positiva com retransmissão”. A técnica exige que o receptor comunique-se com a origem, retornando uma mensagem de confirmação (ACK), à medida que recebe os dados. O transmissor mantém um registro de cada pacote que envia e espera uma confirmação antes de enviar o próximo pacote. O transmissor também inicia um temporizador quando envia o pacote, e retransmite o pacote se esse temporizador se completar antes que chegue uma confirmação.



A figura mostra o mesmo diagrama de formato anterior, para mostrar o que acontece quando um pacote é perdido ou destruído. O transmissor inicia um temporizador após a transmissão de um pacote. Quando o temporizador termina, o transmissor considera o pacote perdido e o retransmite. Este fato é conhecido como Time-out de transmissão.

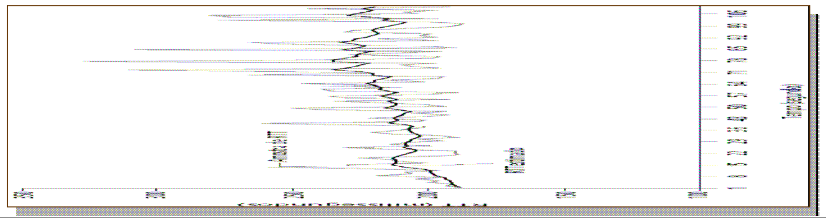
Confiabilidade da Transmissão

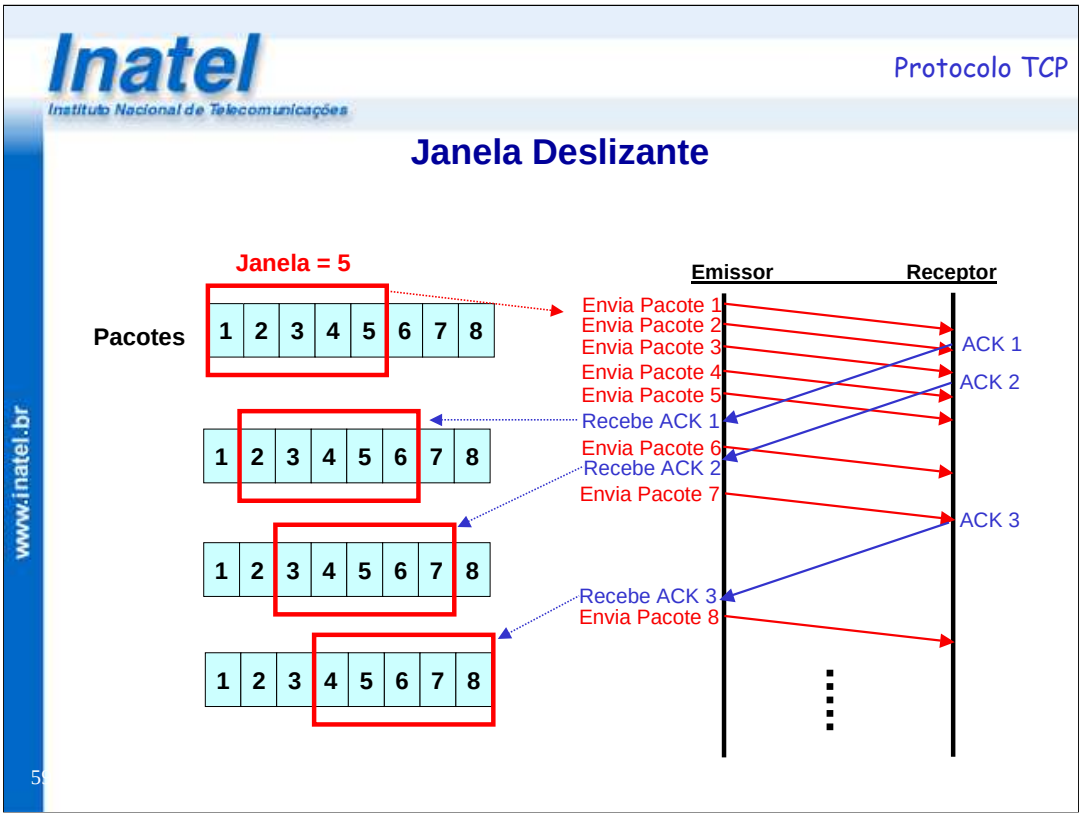
Problemas com a abordagem do RTO para Retransmissão:

• O cálculo do RTO não consegue se adaptar a flutuações muito altas no *RTT*, causando retransmissões desnecessárias.

- Timeout pequeno → gera retransmissões desnecessárias aumentando ainda mais a carga na rede, quando ela já está sobrecarregada
- Timeout grande → faz com que haja tempo longo de espera para retransmissão, subtilizando a rede.

• Solução: cálculo mais preciso do RTO baseado no desvio padrão do RTT (proposto por Van Jacobson)

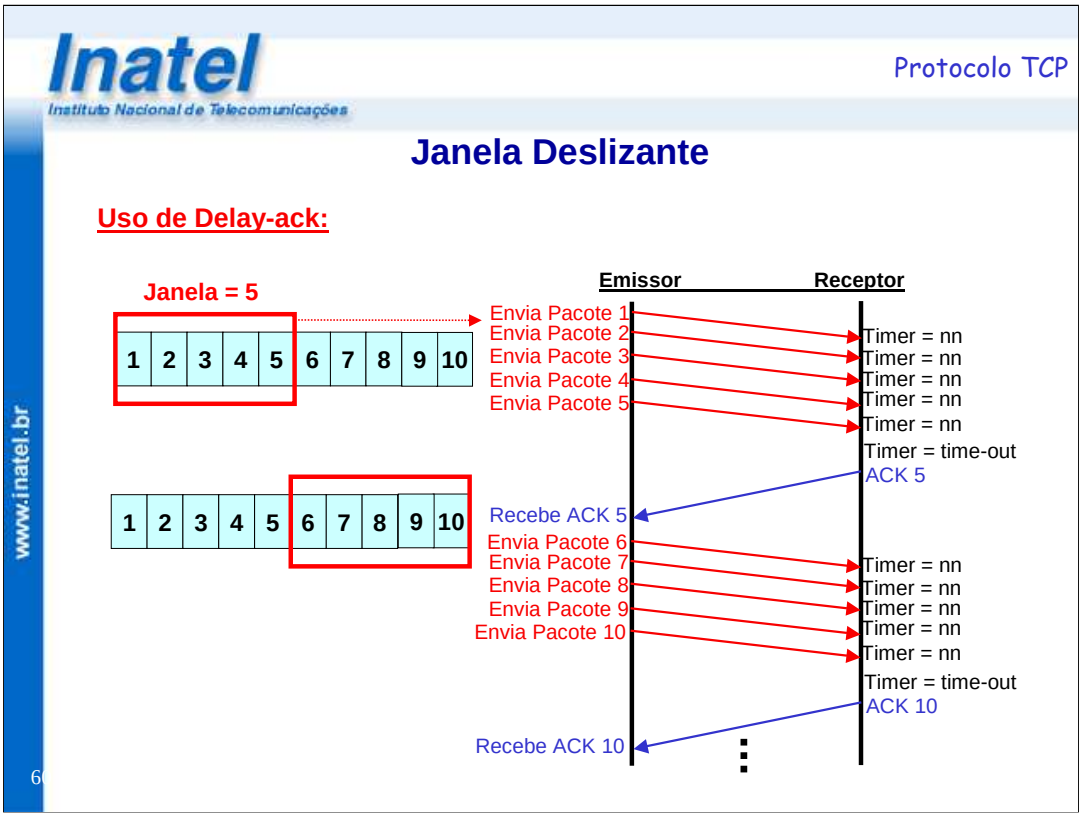




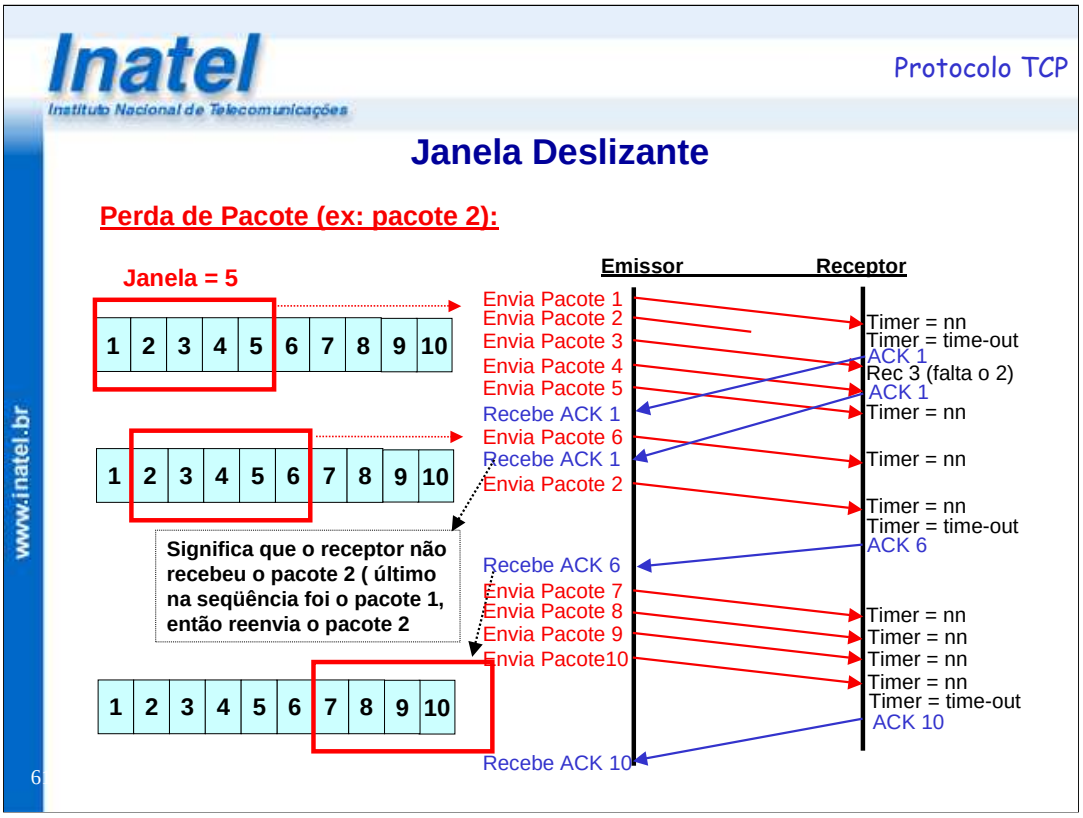
Um protocolo simples de transporte pode usar o princípio visto anteriormente, onde, enviar um pacote e a seguir esperar um reconhecimento do receptor antes de enviar o próximo. Se o ACK não for recebido dentro de um determinado tempo, o pacote é retransmitido.

Ao mesmo tempo que este mecanismo assegura a confiabilidade, ele usa apenas parte da largura de banda disponível na rede e a rede pode ficar completamente inativa durante o tempo em que a máquina retarda as respostas. Ou seja, um protocolo simples, de confirmação positiva, gasta uma substancial quantidade de largura de banda, porque precisa retardar a transmissão de um novo pacote, até que receba uma confirmação do pacote anterior.

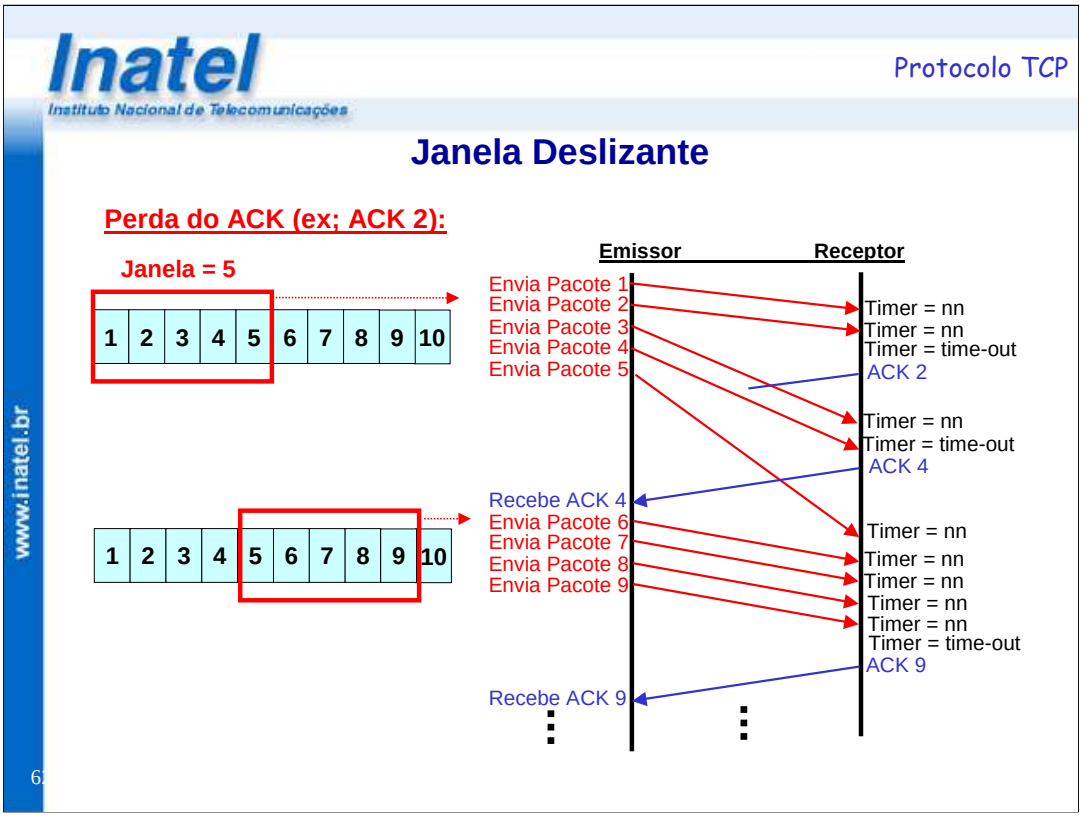
A técnica da janela deslizante é uma forma mais complexa de confirmação positiva e de retransmissão do que o método simples abordado anteriormente. Os protocolos de janela deslizantes utilizam melhor a largura de banda da rede, porque permitem que o transmissor transmita pacotes múltiplos antes de esperar uma confirmação.



O desempenho dos protocolos de janela deslizante depende do tamanho da janela e da velocidade com que a rede aceita novos pacotes. Com um tamanho de janela igual a um, um protocolo de janela deslizante é exatamente o mesmo que o protocolo simples de confirmação positiva. Aumentando-se o tamanho da janela, é possível eliminar completamente o tempo de inatividade da rede. Portanto, quando um protocolo de janela deslizante apropriado mantém a rede completamente saturada de pacotes, ele obtém um troughput substancialmente maior que um protocolo de confirmação positiva simples.



Neste exemplo (mostrado na figura) temos uma situação onde o pacote 2 é transmitido mas ele não chega ao destino. Portanto o emissor não receberá o ACK 2, assim a sua janela permanecerá na posição 1. A verdade, como o receptor não recebeu o pacote 2, ele reconhecerá os pacotes 3,4 e 5 e enviará um ACK 1 para confirmação, indicando que o pacote 1 foi o último pacote recebido na seqüência. No lado do emissor, se esgotará eventualmente o tempo definido para a chegada do ACK 2 que será então retransmitido. Veja que a recepção deste pacote pelo receptor gerará o ACK 5, já que agora recebeu com sucesso todos os pacotes de 1 a 5, e a janela do emissor deslizará quatro posições ao receber o ACK 5, e irá continuar a transmitir os pacotes.



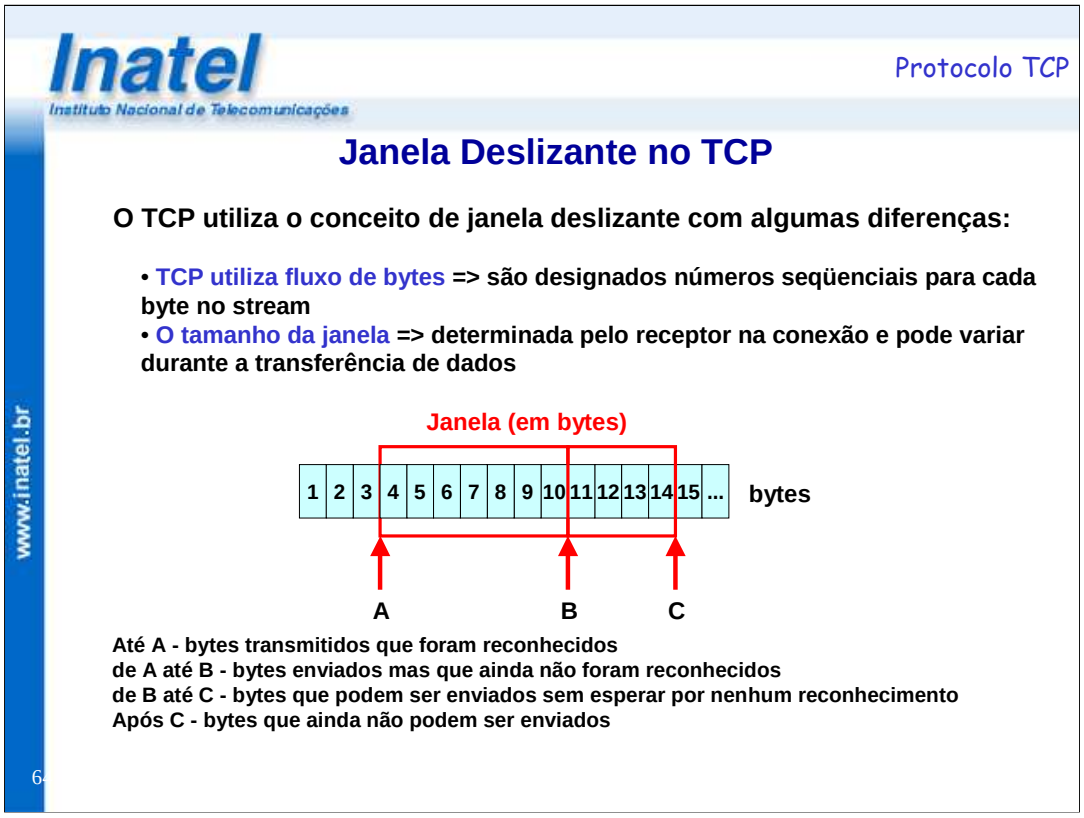
Neste outro exemplo iremos analisar a situação onde o pacote 2 chega ao destino, porém a confirmação deste pacote (ACK 2) não chega. Portanto, o emissor não recebe o ACK 2, mas receberá o ACK 4. O ACK 4 é um aviso de recebimento de todos os pacotes até o 4 (incluindo o pacote 2) e o remetente pode agora deslizar a sua janela para o pacote 4.

Vimos então que este mecanismo de janela deslizante assegura:

• Transmissão confiável dos dados

• Melhor utilização da largura de banda da rede (melhor troughput)

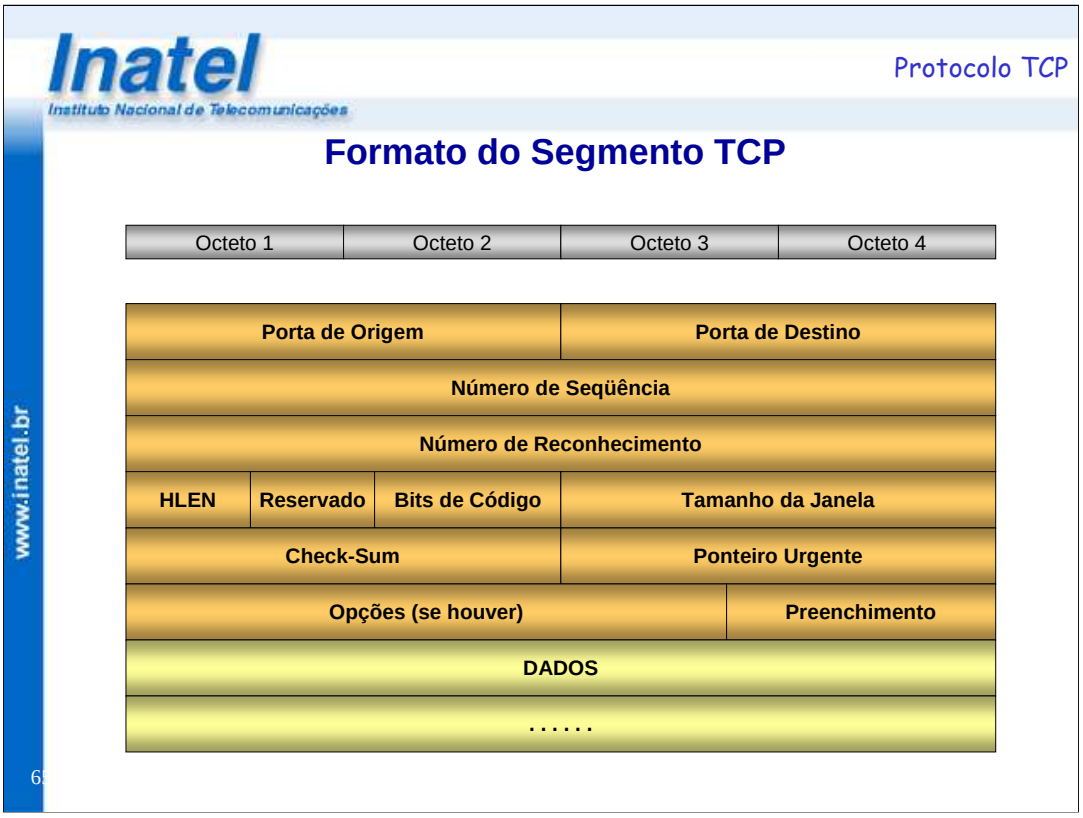
• Controle de fluxo, já que o receptor pode demorar para responder a um pacote com um reconhecimento, e tem conhecimento sobre seus buffers livres disponíveis e sobre o tamanho da janela de comunicação.



O mecanismo de janela deslizante no TCP opera em nível de bytes (octetos), e não de pacotes ou segmentos. Os bytes do stream (fluxo) de dados são numerados seqüencialmente, e o transmissor mantém três ponteiros associados a cada conexão. Os ponteiros definem uma janela deslizante, conforme a figura ilustra.

Todos os bytes à esquerda do ponteiro A já foram enviados e confirmados. Do ponteiro A até o B os bytes foram enviados mas ainda não foram confirmados. Do ponteiro B até o C os bytes que podem ser enviados antes que mais confirmações sejam recebidas. E após o ponteiro C os bytes que não podem ser enviados até que a janela se mova.

O TCP utiliza um tamanho de janela variável controlado pelo receptor . A vantagem de utilizar uma janela de tamanho variável é que ela fornece o controle de fluxo e uma transferência confiável. Se os buffers do receptor começam a ficar cheios, a janela não poderá suportar mais pacotes e, então, enviará um notificador de janela menor. Em casos extremos, o receptor indica um tamanho zero de janela para interromper todas as transmissões. Mais tarde, quando o espaço dos buffers tornar-se disponível, o receptor indicará um tamanho de janela diferente de zero para iniciar o fluxo de dados novamente.



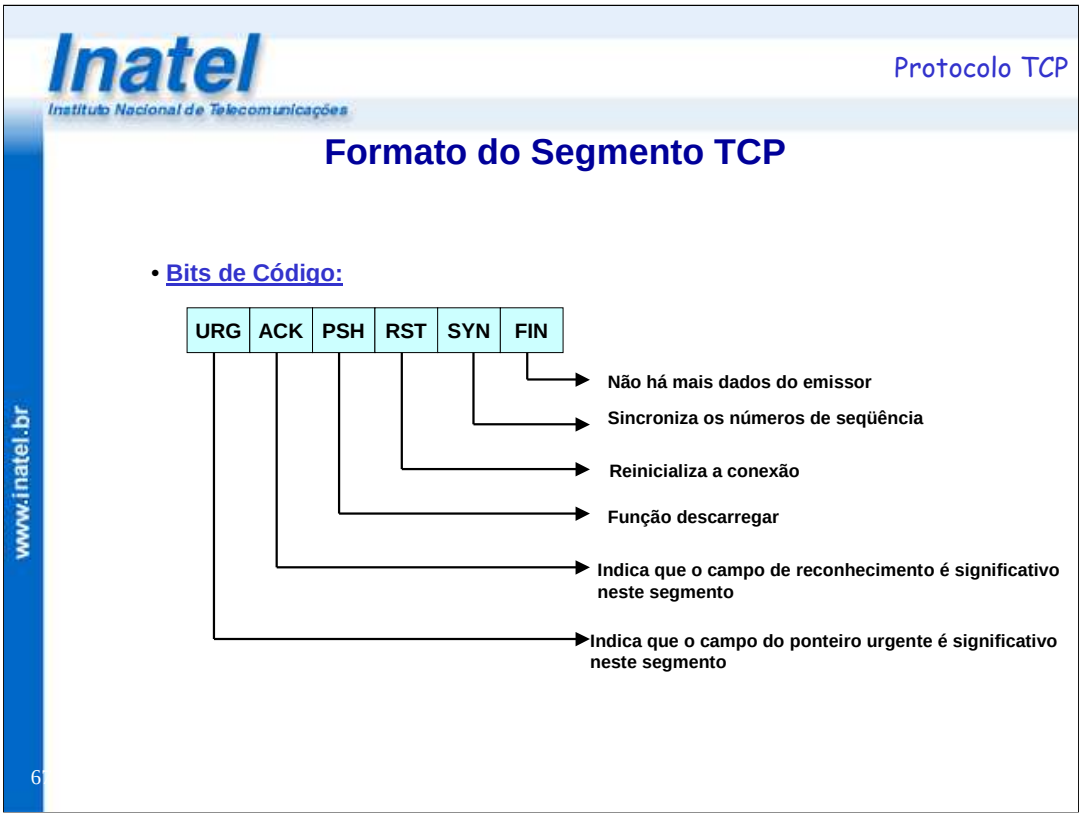
A unidade de transferência entre o software TCP de duas máquinas é denominado ***segmento***. Os segmentos são trocados para estabelecer conexões, transferir dados, enviar confirmações, informar tamanhos de janelas e encerrar conexões. O TCP utiliza o chamado ***piggyback***, que é quando uma confirmação que trafega da máquina A para a máquina B pode trafegar no mesmo segmento que os dados que trafegam da máquina A para a máquina B, embora a confirmação refira-se aos dados enviados de B para A.

Cada segmento TCP é dividido em duas partes: um cabeçalho seguido de dados. O cabeçalho, conhecido como cabeçalho TCP, transporta a identificação esperada e as informações de controle. Alguns segmentos transportam apenas uma confirmação, enquanto alguns transportam dados, outros transportam solicitações para estabelecer ou encerrar uma conexão.

PORTA DE ORIGEM: Porta origem da mensagem

PORTA DE DESTINO: Porta destino da mensagem

NÚMERO DE SEQÜENCIA: número de seqüência dos dados sendo transmitidos face ao conjunto total de dados já transmitidos. Este número indica a posição do primeiro byte de dados sendo transmitido em relação ao total de bytes já transmitidos nesta conexão. O primeiro número de seqüência utilizado não é zero ou um, mas começa de um valor aleatório. Logo se um pacote está transmitindo do 1234o. byte até o 2000o. byte de uma conexão e o NÚMERO DE SEQÜÊNCIA inicial utilizado nesta conexão foi 10000, o campo NÚMERO DE SEQÜÊNCIA conterá o valor 11234. O número de seqüência em um sentido da conexão (máquina A para B) é diferente do número de seqüência do sentido inverso, já que os dados transmitidos por um e outro lado são completamente distintos.

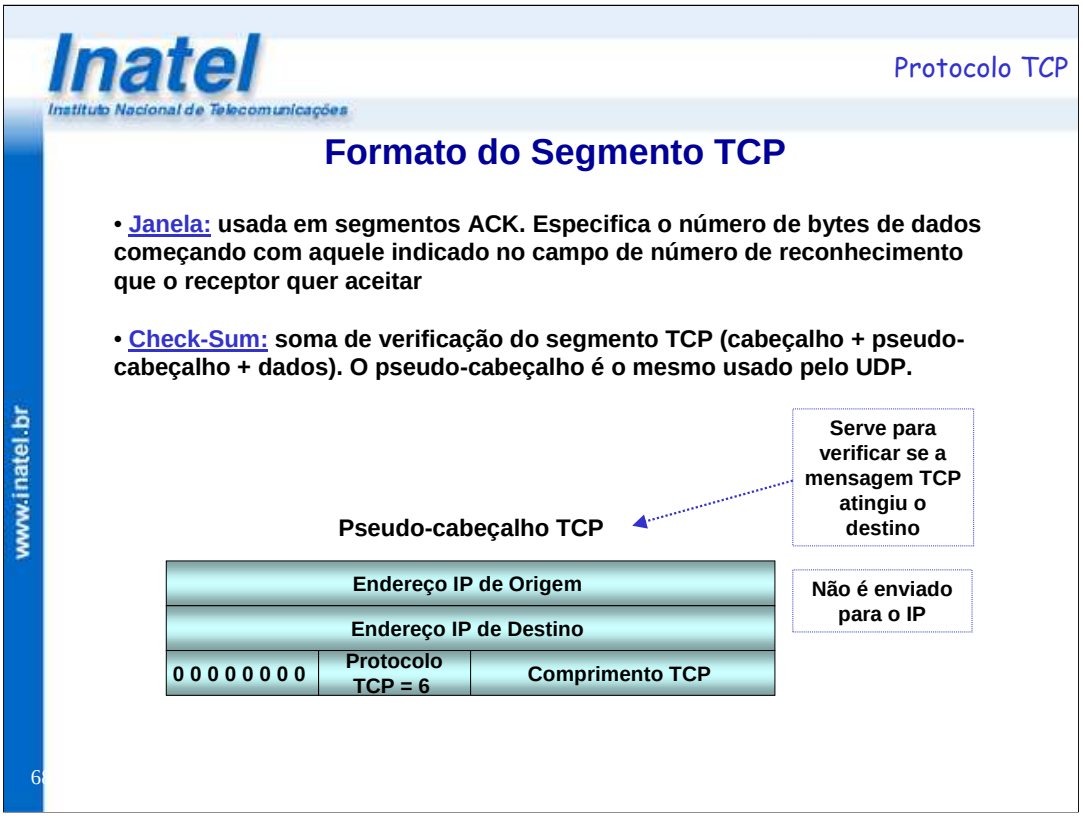


BITS DE CÓDIGO: São formados por seis bits, URG, ACK, PSH, RST, SYN e FIN, cuja utilização é mostrada abaixo:

URG: bit de Urgência: significa que o segmento sendo carregado contém dados urgentes que devem ser lidos com prioridade pela aplicação. A aplicação origem é responsável por acionar este bit e fornecer o valor do PONTEIRO URGENTE que indica o fim dos dados urgentes. Um exemplo da utilização desta facilidade é o aborto de uma conexão (por exemplo por Control-C), que faz com que a aplicação destino examine logo o pacote até o fim da área de urgência, descubra que houve um Control-C e termine a conexão.

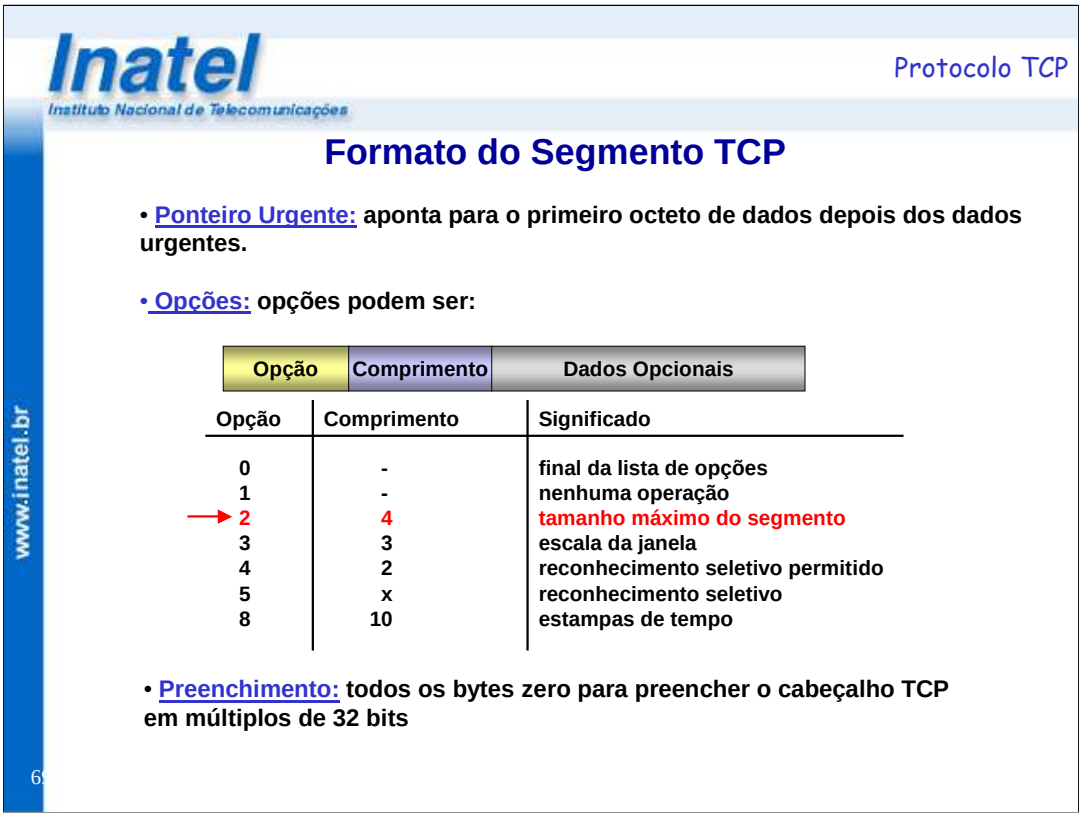
ACK: bit de Reconhecimento: indica que o valor do campo de reconhecimento está carregando um reconhecimento válido.

PSH: bit de PUSH: Este mecanismo que pode ser acionado pela aplicação informa ao TCP origem e destino que a aplicação solicita a transmissão rápida dos dados enviados, mesmo que ela contenha um número baixo de bytes, não preenchendo o tamanho mínimo do buffer de transmissão.



JANELA: Este campo informa o tamanho disponível em bytes na janela de recepção da origem deste pacote. Por meio deste valor, o TCP pode realizar um controle adequando de fluxo para evitar a sobrecarga do receptor. Quando este valor é igual a zero, o transmissor não envia dados, esperando receber um pacote com JANELA maior que zero. O transmissor sempre vai tentar transmitir a quantidade de dados disponíveis na janela de recepção sem aguardar um ACK. Enquanto não for recebido um reconhecimento dos dados transmitidos e o correspondente valor de JANELA > 0, o transmissor não enviará dados.

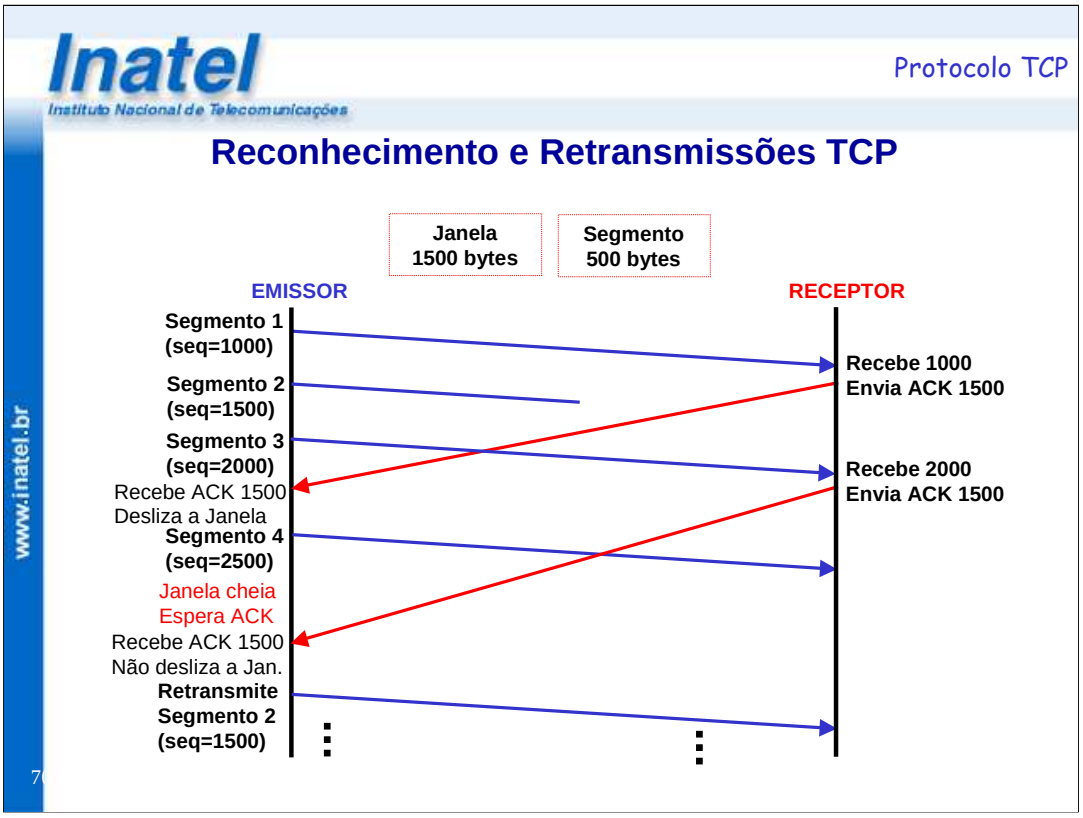
CHECK-SUM: Soma de todas as palavras de 16 bits em um pseudo-cabeçalho, mais o cabeçalho TCP e os dados TCP. Enquanto estiver calculando o CHECK-SUM, o próprio campo de CHECK-SUM é considerado zero. O pseudo-cabeçalho é o mesmo utilizado pelo UDP para calcular o CHECK-SUM. É usado apenas para o cálculo do CHECK-SUM.



PONTEIRO URGENTE: Aponta para o primeiro octeto de dados depois dos dados urgentes. Só faz sentido quando o bit de controle URG estiver em definido.

OPÇÕES: O campo de opções só possui uma única opção válida que é a negociação do MSS (Maximum Segment Size) que o TCP pode transmitir. O MSS é calculado através do MTU ou através do protocolo ICMP Path MTU Discovery. Nem todos os segmentos enviados através de uma conexão terão o mesmo tamanho. No entanto, as duas extremidades precisam concordar sobre o segmento máximo que irão transferir. Se os dois pontos terminais residirem na mesma rede física, o TCP normalmente calcula um tamanho máximo de segmento tal que os datagramas IP resultantes serão de acordo com o MTU da rede. Se os pontos terminais não residirem na mesma rede física, eles podem tentar descobrir o MTU mínimo ao longo do caminho entre eles, ou optar por um tamanho máximo de segmento de 536 octetos (o tamanho padrão de um datagrama IP, 576 menos o tamanho padrão de dados IP e cabeçalho TCP).

PREENCHIMENTO: Todos os bytes zero usados para preencher o cabeçalho TCP para que o comprimento total seja um múltiplo de 32 bits.

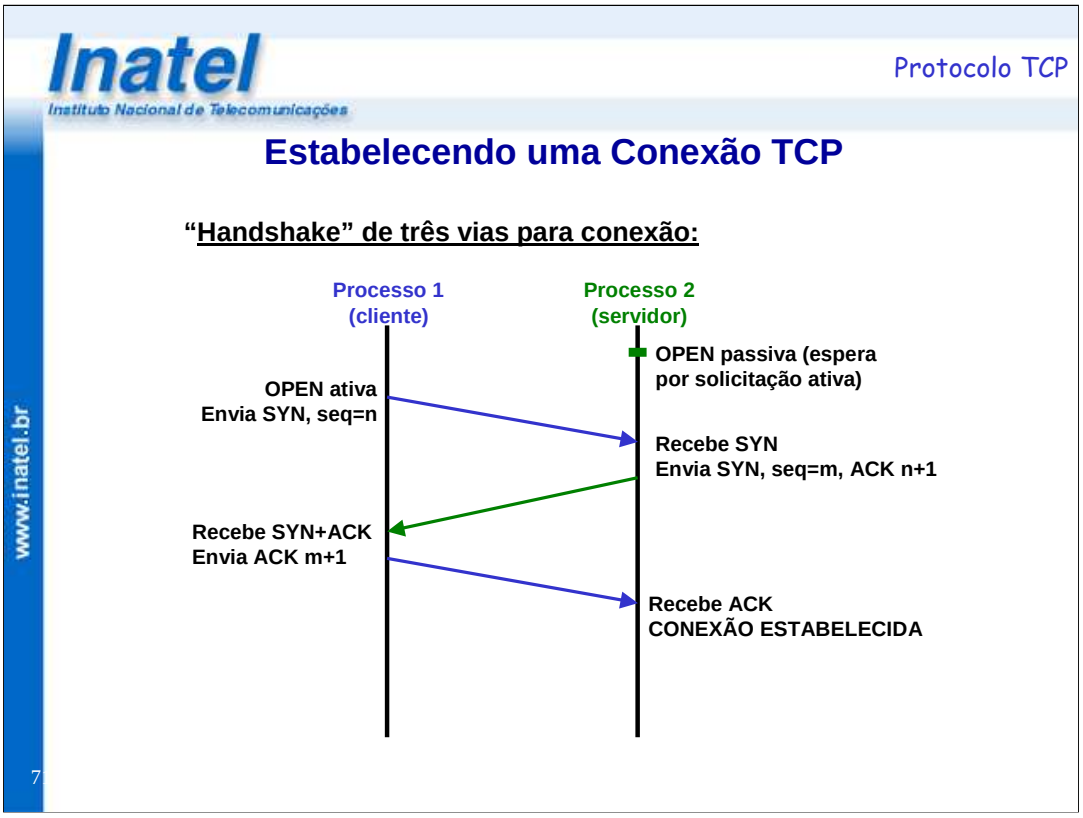


O TCP envia dados em segmentos de comprimento variável. Os números seqüenciais são baseados na contagem de bytes. Os reconhecimentos especificam o número seqüencial do próximo byte que o receptor espera receber.

Considere que um segmento se perde ou seja danificado. Neste caso, o receptor continuará a reconhecer todos os próximos segmentos bem recebidos indicando o primeiro byte do pacote perdido. O emissor irá parar de transmitir quando tiver enviado todos os bytes da janela. Eventualmente, um tempo esgotado irá ocorrer e o segmento perdido será retransmitido. (princípio da janela deslizante).

A figura ilustra um exemplo onde o tamanho de uma janela de 1.500 bytes e segmentos de 500 bytes são utilizados.

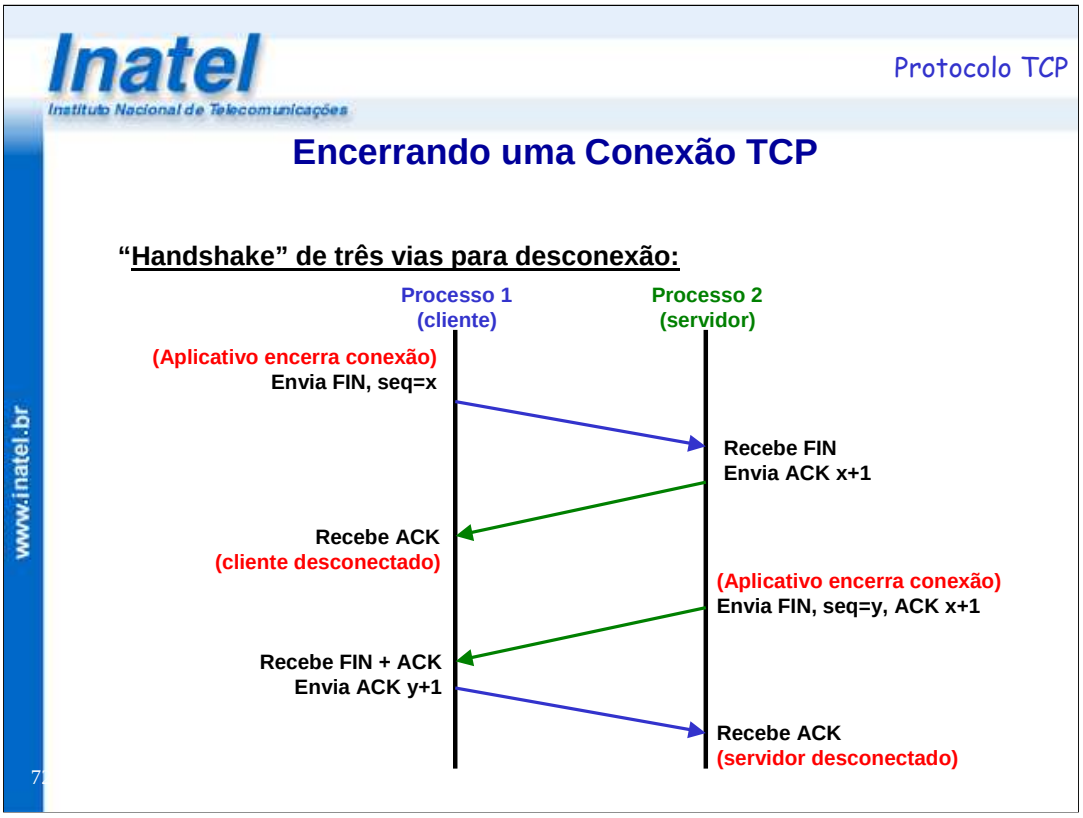
Podemos observar que irá surgir um problema, já que o emissor sabe que o segmento 2 está perdido ou danificado, mas não sabe nada sobre os segmentos 3 e 4. O emissor deveria pelo menos retransmitir o segmento 2, mas poderia também retransmitir os segmentos 3 e 4, já que estão na mesma janela. É possível que:



Antes que qualquer dado possa ser transferido, uma conexão tem que ser estabelecida entre os dois processos. Um dos processos (geralmente o servidor) emite uma chamada OPEN PASSIVA e permanece dormente até que outro processo tente se conectar através de uma chamada OPEN ATIVA. Após isto o TCP está pronto para estabelecer uma conexão utilizando um "handshake" de três vias. O primeiro segmento de um handshake pode ser identificado porque ele possui um conjunto de bits SYN no campo de código. A Segunda mensagem possui ambos os conjunto de bits SYN e ACK, indicando que confirma o primeiro segmento SYN e continua o handshake. A mensagem final de handshake é apenas uma confirmação e é utilizada simplesmente para informar ao destino que ambos os lados concordam em que uma conexão foi estabelecida.

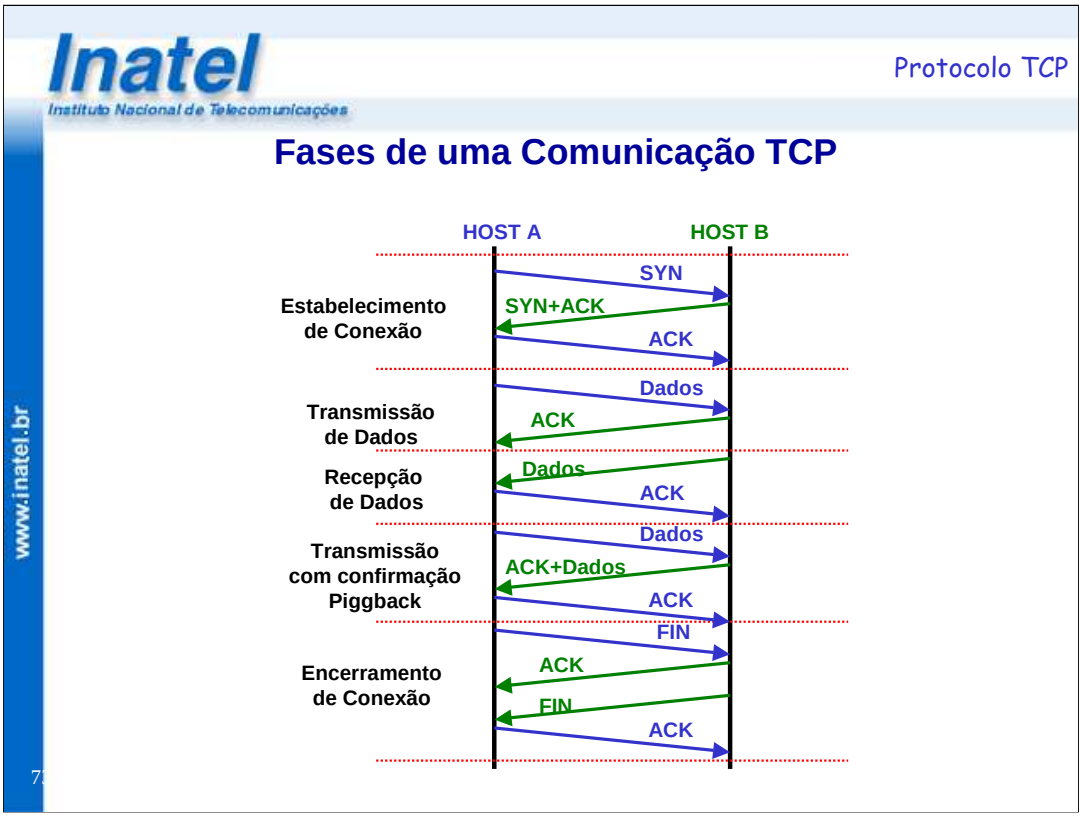
Quando a conexão tiver sido estabelecida, os dados podem fluir igualmente bem nas duas direções. Não há mestre ou escravo.

O handshake de três vias é necessário e suficiente para a sincronização correta entre as duas extremidades da conexão. Lembre-se que o TCP cria uma transmissão com base no serviço de transmissão não-confiável de pacotes, podendo as mensagens ser perdidas, retardadas, duplicadas, ou entregues fora de ordem.



O encerramento da conexão TCP é feito implicitamente pelo envio de um segmento TCP com o conjunto de bits FIN (fim de dados). Como a conexão é full-duplex, o segmento FIN fecha apenas a transferência dos dados em uma direção. O outro processo enviará os dados remanescentes que ainda tem para transmitir e também finalizará com um segmento TCP com bit FIN configurado. A conexão é encerrada já que o fluxo de dados está fechado em ambas as direções.

A diferença entre os handshakes de três vias usados para estabelecer e encerrar conexões ocorre depois que uma máquina recebe um segmento FIN inicial. Em vez de gerar um segundo segmento FIN imediatamente, o TCP envia uma confirmação e, a seguir, informa ao aplicativo sobre a solicitação a ser encerrada. Finalmente, quando o programa aplicativo instrui o TCP para encerrar a conexão, o TCP envia o segundo segmento FIN e o transmissor responde com uma terceira mensagem, um ACK.

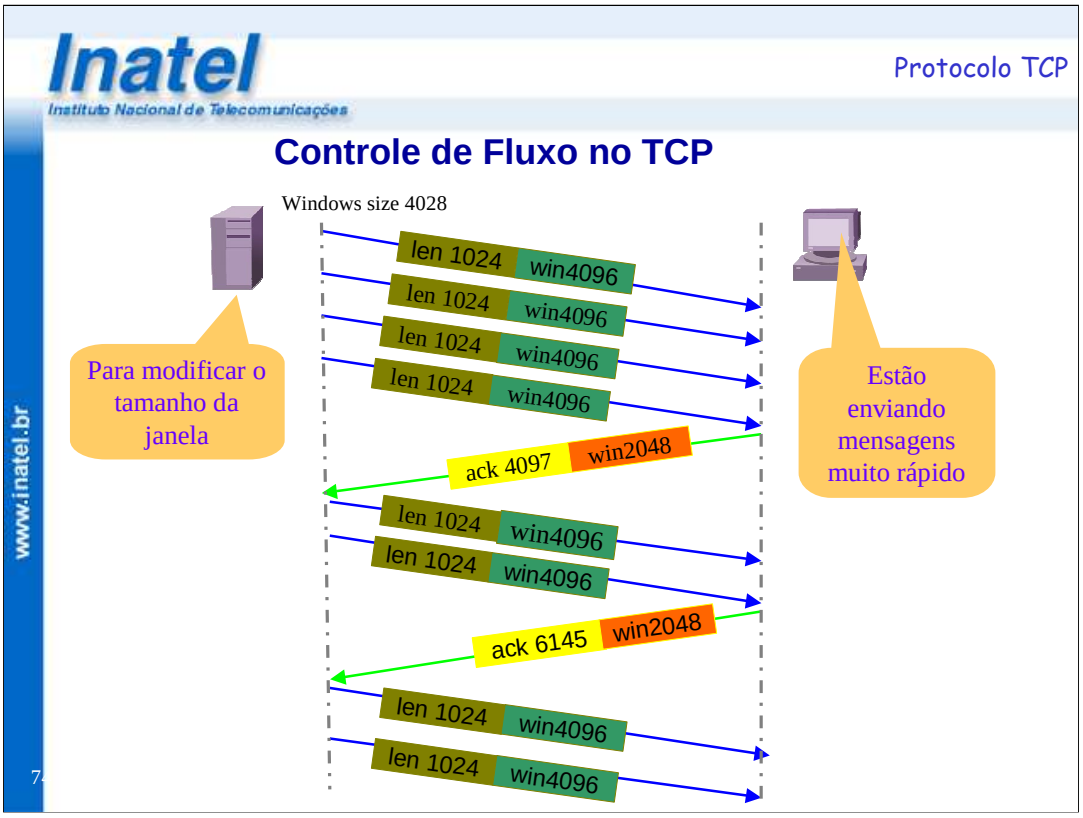


Podemos notar que uma transmissão TCP é formada por três fases:

• Estabelecimento de Conexão

• Transmissão / Recepção de Dados (inclusive Piggyback – envio de confirmação e dados no mesmo segmento)

• Encerramento da conexão



No protocolo TCP o tamanho da Janela pode ser usado como controle de fluxo de dados.

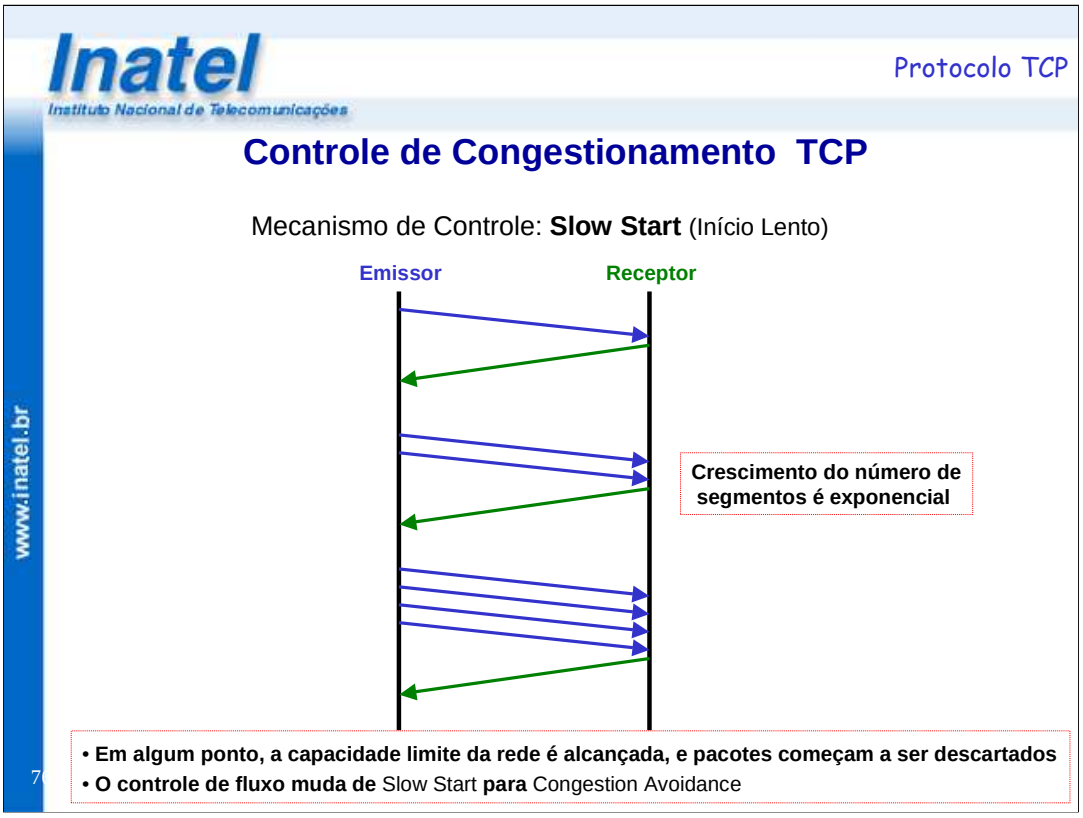
Uma grande diferença entre os protocolos TCP e UDP é o algoritmo de controle de congestionamento. O algoritmo de congestionamento TCP evita que um emissor ultrapasse a capacidade da rede. O TCP pode adaptar o ritmo do remetente à capacidade da rede, podendo também tentar evitar as situações de congestionamento potencial. As implementações modernas do TCP contêm quatro algoritmos como padrões básicos da internet:

Slow Start

Congestion Avoidance

Fast Recovery

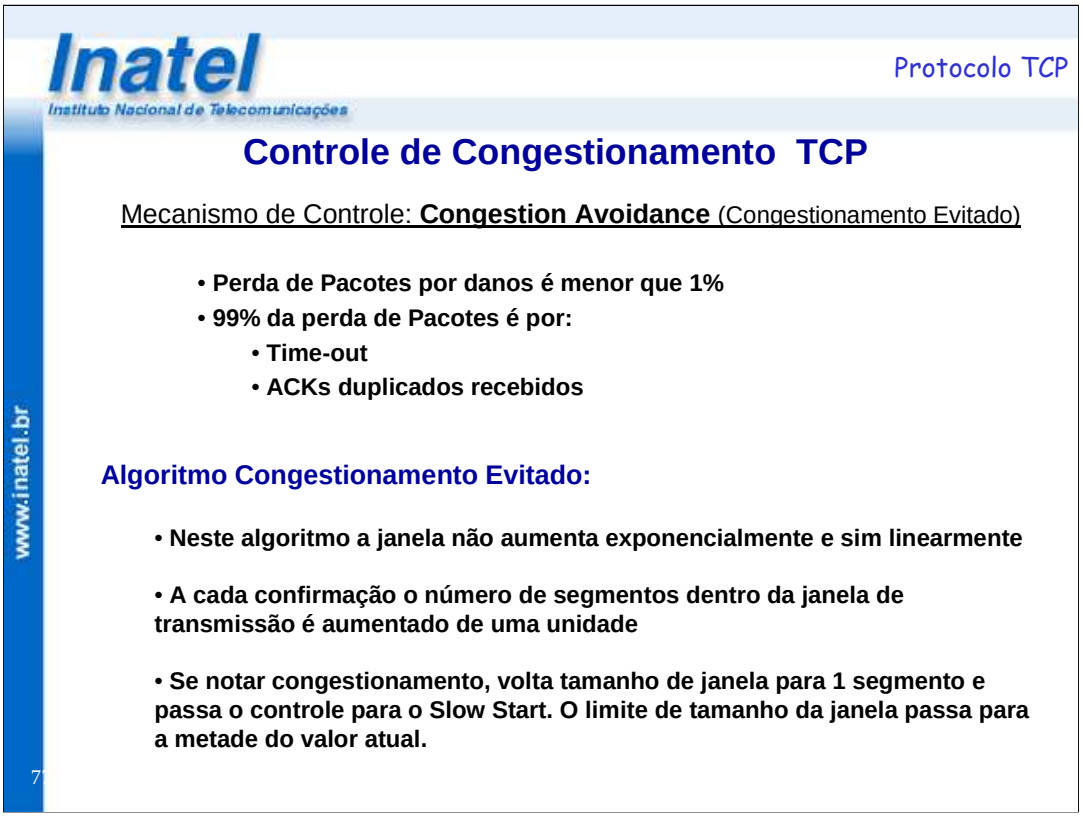
Fast Retransmit



Início Lento:

Implementações antigas do TCP iniciavam uma conexão com o emissor injetando vários segmentos na rede, até o tamanho da janela anunciado pelo receptor. Isto funciona bem quando os dois hosts estão na mesma rede local mas, se houver roteadores e enlaces mais lentos entre o emissor e o receptor, podem surgir problemas. Alguns roteadores intermediários podem não dar conta disto, resultando em pacotes perdidos, necessidade de retransmissão e desempenho degradado.

O algoritmo para evitar isto é chamado de Início Lento. Ele opera com base ma observação do ritmo em que os reconhecimentos são devolvidos pela outra ponta.



Congestionamento Evitado:

A suposição deste algoritmo é que a perda de pacotes causada por danos é muito pequena (muito menor que 1%), portanto uma perda de pacotes indica congestionamento em algum lugar da rede entre o origem e o destino. Há duas indicações para a perda de pacotes:

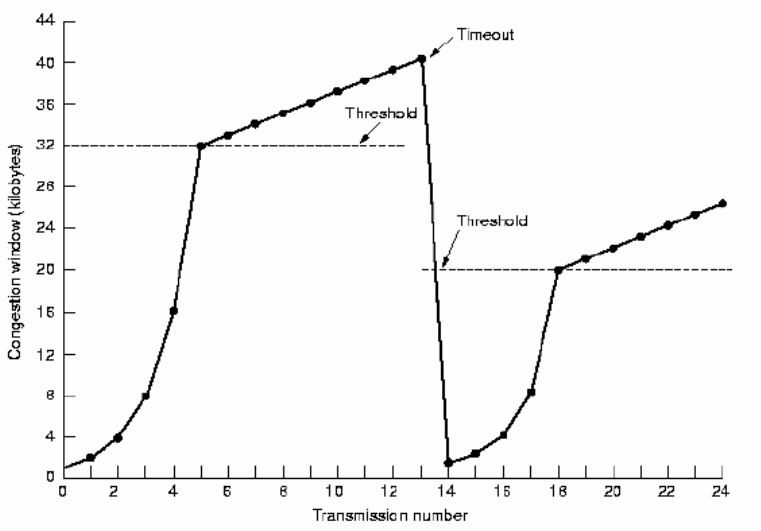
- Ocorrência de um tempo esgotado (time-out)
- ACKs duplicados foram recebidos

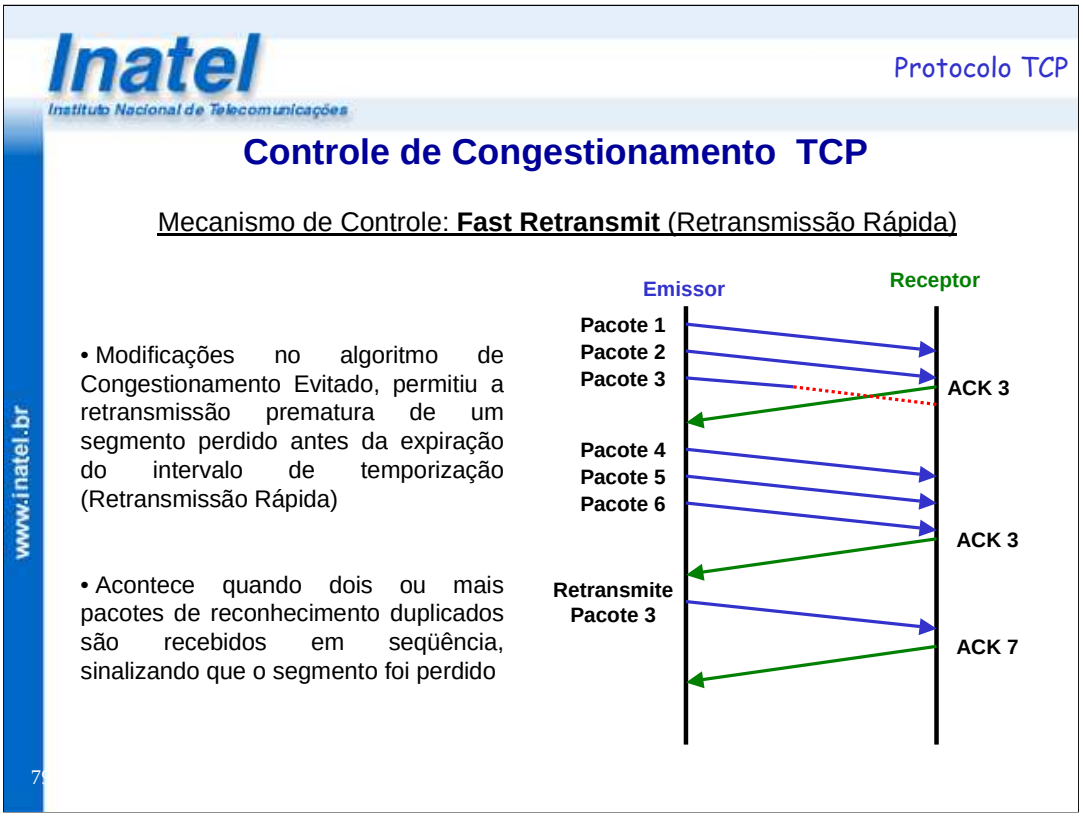
O congestionamento evitado e o início lento são algoritmos independentes com objetivos diferentes. Mas quando o congestionamento ocorre, o TCP deve diminuir sua taxa de transmissão de pacotes na rede, e então solicitar o início lento para dar continuidade ao processo.

O início lento continua até que o TCP esteja a meio caminho de onde estava quando ocorreu o congestionamento, daí em diante o congestionamento evitado assume o controle.

Controle de Congestionamento TCP

Mecanismo de Controle: **Congestion Avoidance** (Congestionamento Evitado)

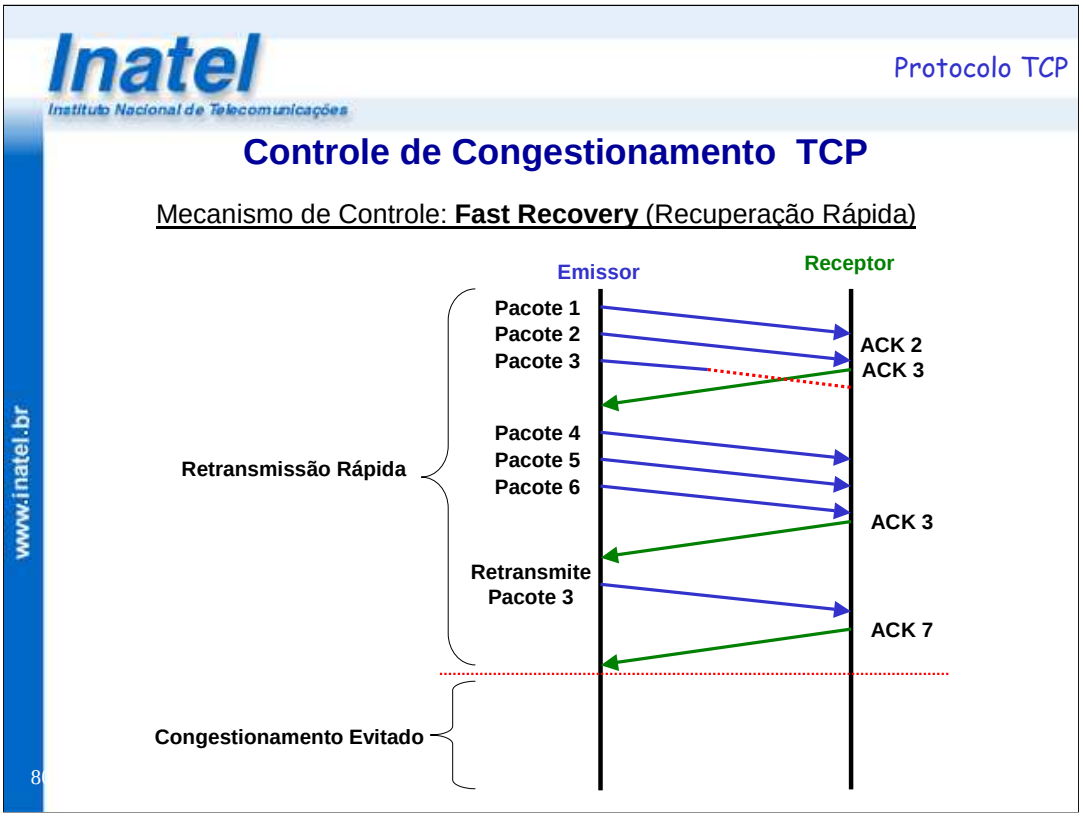




A retransmissão rápida evita que o TCP tenha que esperar por um tempo esgotado para enviar novamente os segmentos perdidos.

O TCP pode gerar um reconhecimento imediato (um ACK duplicado) quando um segmento fora de ordem for recebido. Este ACK duplo não deveria demorar para ser enviado. O propósito deste ACK duplo é avisar à outra ponta que um segmento foi recebido fora de ordem e dizer-lhe qual é o número seqüencial esperado.

Como o TCP não sabe se um ACK duplo foi provocado por um segmento perdido ou apenas por um novo pedido de segmentos, ele espera pelo recebimento de um pequeno número de ACKs duplos. Assume-se que se houver apenas uma nova solicitação de segmentos, haverá apenas um ou dois ACKs duplos antes que o segmento solicitado novamente seja processado, o que não produzirá um ACK. E três ou mais ACKs duplicados forem recebidos em seqüência, há um forte indício que um segmento foi perdido. O TCP realiza então uma retransmissão do que parece ser o segmento perdido, sem esperar que o cronômetro de retransmissão expire.



Depois que a retransmissão rápida envia o que parece ser o segmento perdido, o congestionamento evitado entra em funcionamento, não o início lento. Este é o algoritmo de Recuperação Rápida. É um aperfeiçoamento que permite alto fluxo sob congestionamento moderado, especialmente em janelas grandes.

Uma redução brusca do fluxo utilizando o Início Lento, não seria aconselhável, pois dados ainda podem estar trafegando entre dois nós, o que pode agravar a situação de congestionamento.

Observações sobre o Controle de Congestionamento no TCP

• Poucos sistemas operacionais (mesmo comerciais) implementam corretamente as RFC´s do TCP **(RFC 793 e RFC 2001)**;

• Os que implementam com mais fidelidade os algoritmos mencionados anteriormente são o **FreeBSD** e o **Solaris**;

• O **Windows** sequer implementa as RFC´s originais do IP e do TCP corretamente. O mesmo acontecendo em relação aos algoritmos da RFC 2001;

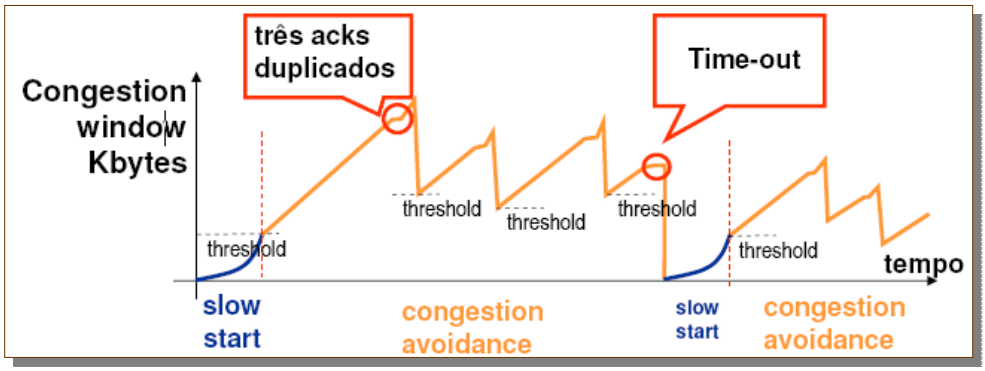
• O **AIX** implementa os algoritmos acima com algumas variações proprietárias;

• O **Linux** implementa perfeitamente alguns algoritmos mais avançados mas, assim como outros, deixa a desejar no simples algoritmo de *slow start*;

Dados da revista RNP – News Generation - vol.2, número 4 - 1998

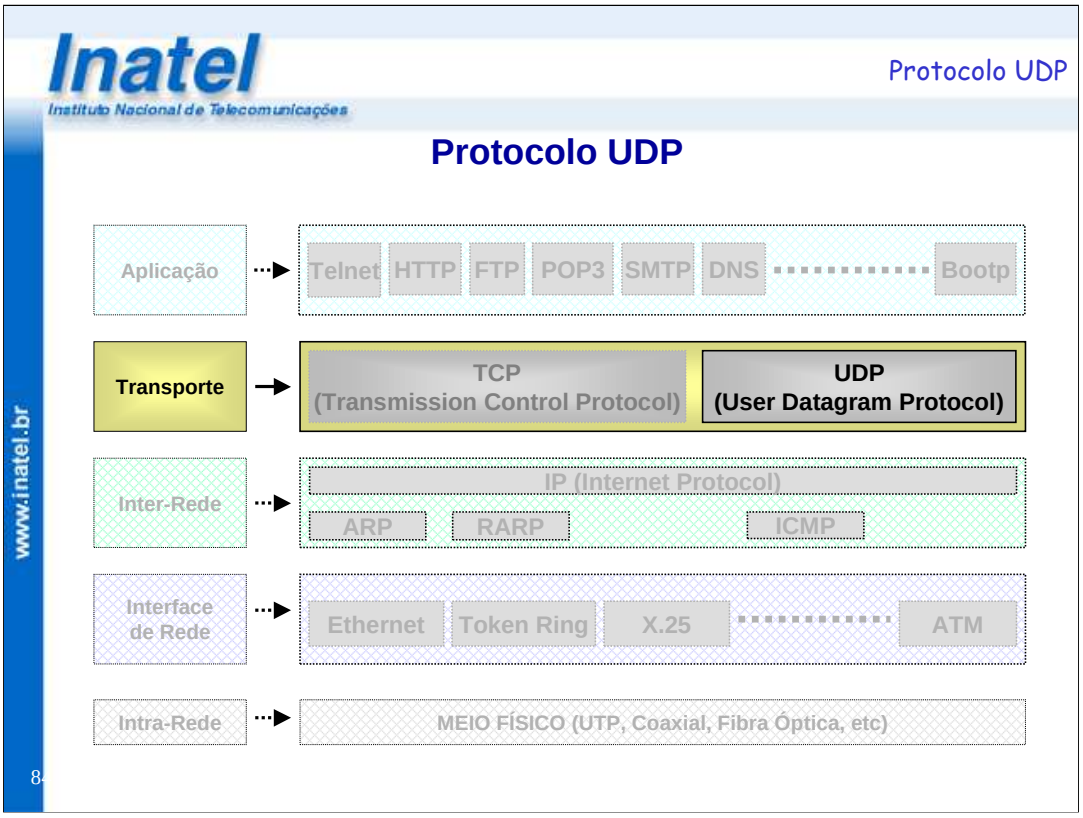
Comportamento do TCP na Internet

Vazão (**throughput)** ocorre em Dente de Serra



Neste capítulo, será mostrado o Protocolo UDP, que é um dos protocolo da camada de Transporte da pilha de protocolo TCP / IP. O UDP realiza a transferência de pacotes do usuário de maneira não-confiável e sem conexão, mas permite que várias aplicações utilizem o UDP para acessar um host. Esta técnica é realizada pelas portas de protocolo.

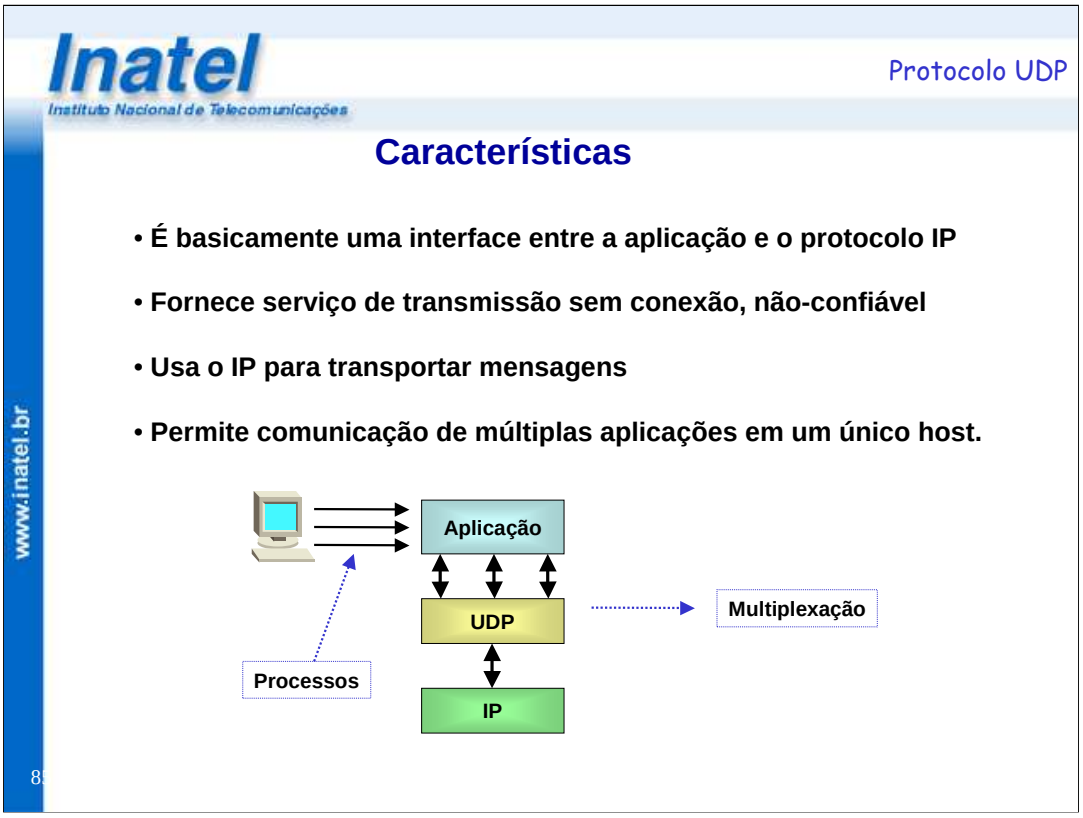
Será mostrado também como é o formato dos datagramas UDP e como eles são encapsulados pela camada IP.



Na pilha de protocolos TCP / IP, o UDP (User Datagram Protocol), fornece o mecanismo principal utilizado pelos programas aplicativos para enviar datagramas a outros programas iguais.

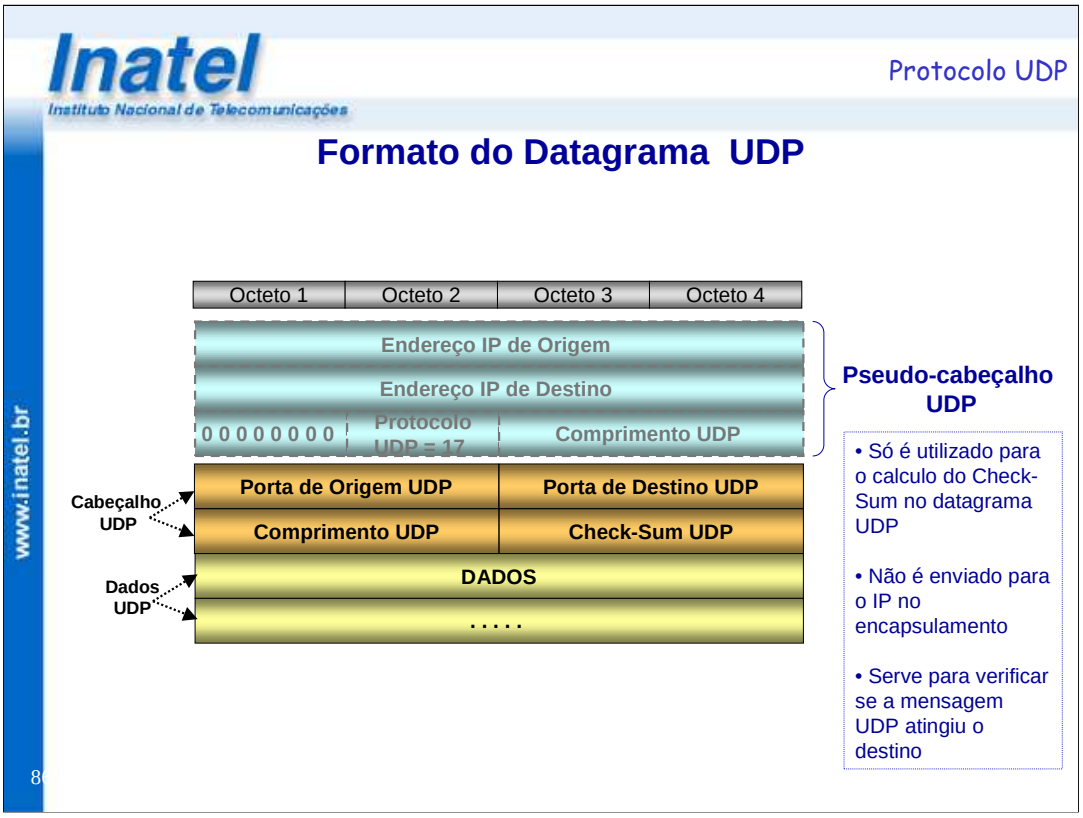
O UDP é basicamente uma interface de aplicação para o IP. O UDP fornece um mecanismo para que um aplicativo envie um datagrama para outro aplicativo. A camada UDP pode ser considerada como sendo extremamente fina e por isso opera com pouco “overhead”, mas precisa do aplicativo para cuidar da recuperação de erros, etc.

Os aplicativos que enviam datagramas para um host precisam identificar um alvo que seja mais específico que o endereço IP. Os datagramas são geralmente dirigidos a certos processos e não para o sistema como um todo. O UDP faz isto usando “portas”, que serão vistas mais adiante.



O protocolo **UDP** fornece uma forma simples de acesso ao sistema de comunicação, provendo um serviço sem conexão, não-confiável e sem correção de erros.

A principal função do nível de transporte implementada em UDP é a capacidade de multiplexação de acesso ao sistema de comunicação. Esta função permite que vários processos ou programas executando em um computador possam acessar o sistema de comunicação e o tráfego de dados respectivo a cada um deles seja corretamente identificado, separado e utilize buffers individuais. Um processo é o programa que implementa uma aplicação do sistema operacional, e que pode ser uma aplicação do nível de aplicação TCP/IP.

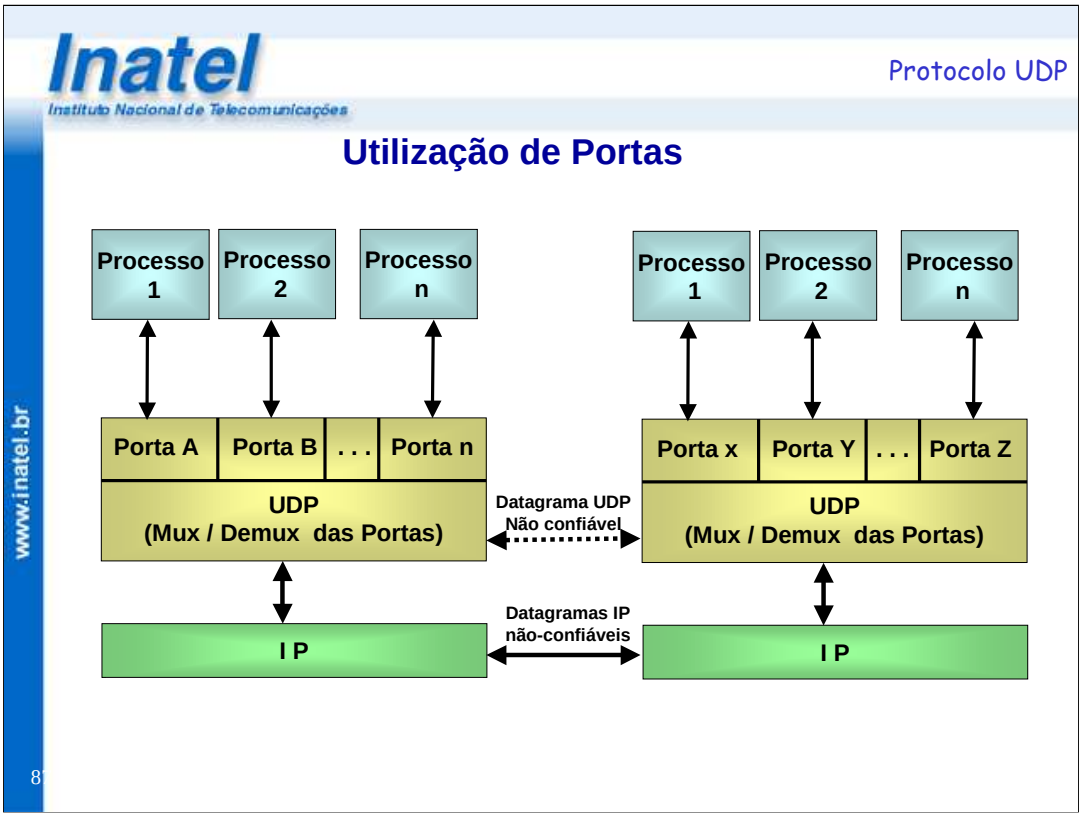


Cada mensagem UDP é conhecido como um datagrama do usuário e consiste de duas partes: um cabeçalho UDP e uma área de dados UDP. O cabeçalho está dividido em quatro campos de 16 bits que especificam a porta da qual a mensagem foi enviada, a porta à qual a mensagem é destinada, o comprimento da mensagem e a soma de verificação UDP.

Cada datagrama UDP é enviado dentro de um único datagrama IP. Apesar do datagrama IP poder ser fragmentado durante a transmissão, a implementação IP do receptor irá reagrupá-lo antes de apresentá-lo à camada UDP no destino. Exige-se que todas as implementações IP aceitem datagramas de 576 octetos, o que significa que, permitindo o tamanho máximo de cabeçalho IP de 60 octetos, um datagrama UDP de 516 bytes é aceitável em todas as implementações TCP / IP.

O datagrama UDP é descrito em detalhes a seguir:

PORTA DE ORIGEM: indica a porta do processo que está enviando o datagrama UDP. E é a porta para o qual as respostas devem ser endereçadas.



A forma de identificação de um ponto de acesso de serviço (SAP) do modelo OSI é a **porta** de protocolo em TCP/IP. A porta é a unidade que permite identificar o tráfego de dados destinado a diversas aplicações. A identificação única de um processo acessando os serviços TCP/IP é, então, o endereço IP da máquina e a porta (ou portas) usadas pela aplicação. Cada processo pode utilizar mais de uma porta simultaneamente, mas uma porta só pode ser utilizada por uma aplicação em um dado momento. Uma aplicação que deseje utilizar os serviços de comunicação deverá requisitar uma ou mais portas para realizar a comunicação. A mesma porta usada por uma aplicação pode ser usada por outra, desde que a primeira tenha terminado de utilizá-la.

A forma de utilização de portas mostra uma distinção entre a parte cliente e a parte servidora de uma aplicação TCP/IP. O programa cliente pode utilizar um número de porta qualquer, já que nenhum programa na rede terá necessidade de enviar uma mensagem para ele. Já uma aplicação servidora deve utilizar um número de porta bem conhecido (Well-know ports) de modo que um cliente qualquer, querendo utilizar os serviços do servidor, tenha que saber apenas o endereço IP da máquina onde este está executando. Se não houvesse a utilização de um número de porta bem conhecido, a arquitetura TCP/IP deveria possuir um mecanismo de diretório para que um cliente pudesse descobrir o número da porta associado ao servidor.

No modelo de camadas de protocolo o UDP situa-se na camada que fica acima da camada do IP. Em tese, os programas aplicativos acessam o UDP, que usa o IP para enviar e receber datagramas. Posicionar o UDP em uma camada acima do IP significa que uma mensagem UDP completa, incluindo o cabeçalho UDP e dados UDP, está encapsulada em um datagrama IP, que por sua vez, como já foi visto anteriormente está encapsulado em um quadro físico de rede, e este transporta os dados através de uma interligação em redes.

O encapsulamento do UDP, significa que ele anexa inicialmente um cabeçalho aos dados que o usuário envia e passa-o ao IP. A camada IP inicialmente anexa um cabeçalho ao que ele recebe de UDP. Finalmente, a camada de interface de rede embute o datagrama IP em um quadro antes de enviá-lo de uma máquina para outra. O formato do quadro depende da tecnologia básica da rede utilizada. Normalmente os quadros de rede incluem um cabeçalho adicional.

Podemos portanto concluir que: A camada IP é responsável apenas pela transferência de dados entre um par de hosts em uma interligação em redes, enquanto que a camada UDP é responsável apenas pela diferenciação entre múltiplas origens ou destinos em um host (através das portas).

Utilização do UDP

- Utilização
  - tráfego multimídia
  - DNS
  - SNMP
- Transmissão confiável sobre UDP: adicionar funções para garantir confiabilidade na camada de aplicação
  - recuperação de erro específica para aplicação!

Desse modo, apenas o cabeçalho IP identifica os hosts de origem e destino e a camada UDP apenas identifica as portas de origem e destino em um host.

Alguns aplicativos que usam o UDP são: TFTP (Trivial File Transfer Protocolo): protocolo de Transferência de Arquivos Trivial, Servidor de Nomes DNS, RPC (Remote Procedure Call): Chamada de Procedimentos Remotos, SNMP (Simple Network Management Protocol): Protocolo de Gerenciamento de Rede Simples, LDAP (Lightweight Directory Access Protocol): Protocolo de Acesso a Diretórios