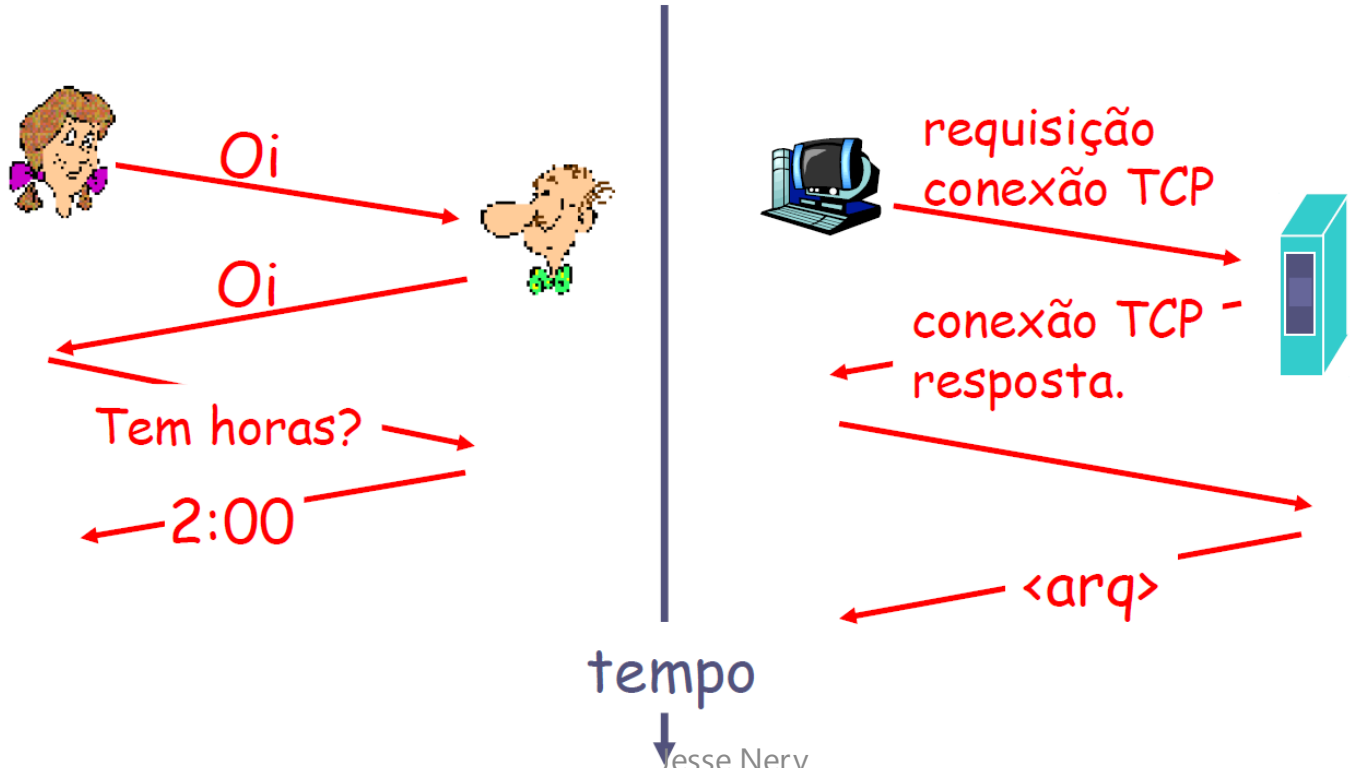
REDES DE COMPUTADORES

Modelo OSI / ISO

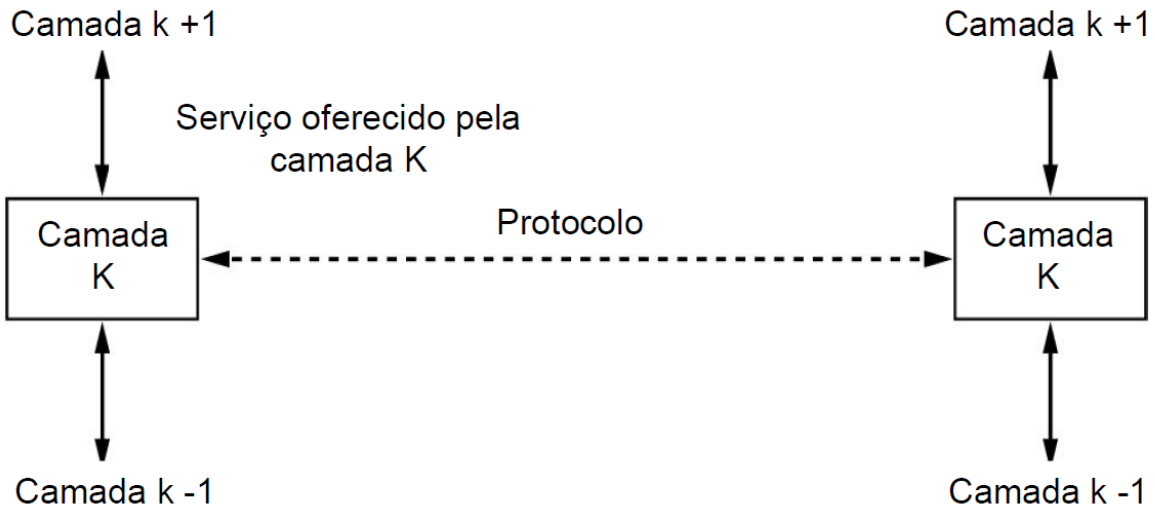
Protocolos

Protocolos definem formatos, ordens de mensagens enviadas e recebidas entre entidades e ações a serem tomadas.



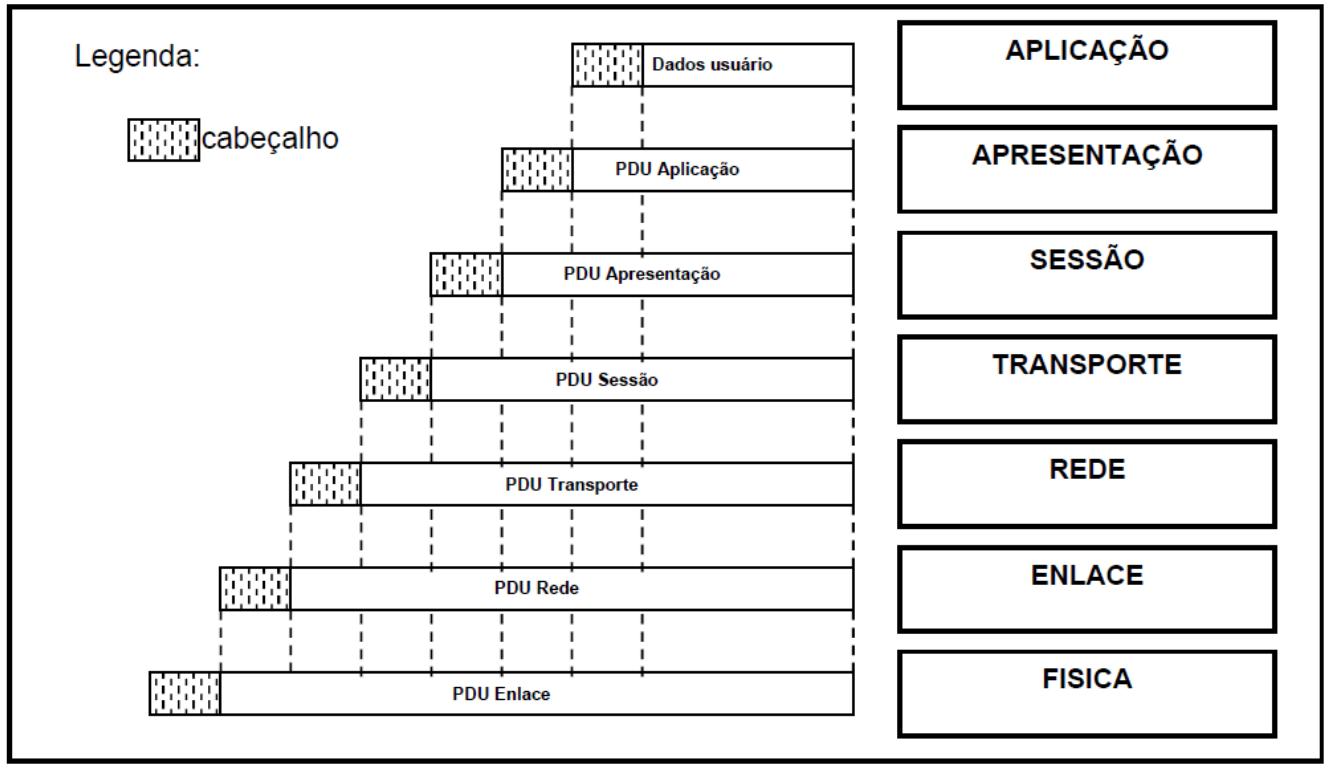
Serviço X Protocolo

- Serviço: “Conjunto de primitivas (operações) que uma camada oferece à camada situada acima dela.” [TAN, 03]
- Protocolo: “Conjunto de regras que controla o formato e o significado dos pacotes ou mensagens que são trocadas pelas entidades pares contidas em uma camada.” [TAN, 03]



Modelo de Referência OSI da ISO

- ISO (*International Standards Organization*)
- OSI (*Open Systems Interconnection*);
- Criado no início dos anos 80, foi revisto em 1995, proposta de **padronização internacional dos protocolos empregados nas diversas camadas**. Não é uma arquitetura de rede, pois não define os serviços e protocolos que devem ser usados em cada camada.



Modelo de Referência OSI da ISO

O modelo OSI utiliza uma abordagem estratificada com certos conjuntos de funções alocados nas diferentes camadas que o compõem. Uma entidade é um elemento ativo em uma camada. Duas entidades em uma mesma camada são denominadas entidades pares. As entidades de uma camada prestam serviços às entidades da camada imediatamente acima e, por sua vez, recebem serviços da camada situada imediatamente abaixo. Por exemplo, as entidades da camada de apresentação prestam serviços à camada de aplicação e recebem serviços da camada de sessão.

Camada Física e de Enlace

Física: Ativação e desativação das conexões físicas, mediante solicitação da camada de enlace de dados. Transmissão dos bits por uma conexão física em modo síncrono ou assíncrono. Tratamento das atividades de gerência da camada física, inclusive a ativação e o controle de erros.

Enlace de Dados: Estabelecimento e liberação de conexões de enlace de dados. Sincronização da recepção de dados que tiverem sido partidos por várias conexões físicas. Detecção e correção de erros de transmissão, com retransmissão de quadros, se necessário.

Camada de Rede e de Transporte

Rede: Determinação de um roteamento ótimo sobre as conexões de rede que podem existir entre dois endereços de rede. Provisão de uma conexão de rede entre duas entidades de transporte. Multiplexação de múltiplas conexões de rede em uma única conexão de enlace de dados. Tratamento das atividades da camada de rede, inclusive ativação e controle de erros.

Transporte: Colocação em sequencia das unidades de dados transferidas, para garantir que sejam entregues na mesma sequencia em que foram enviadas. Detecção de erros e recuperação após erros. Controle de fluxo de dados para evitar sobrecarga dos recursos da rede. Realização das atividades de supervisão da camada de transporte.

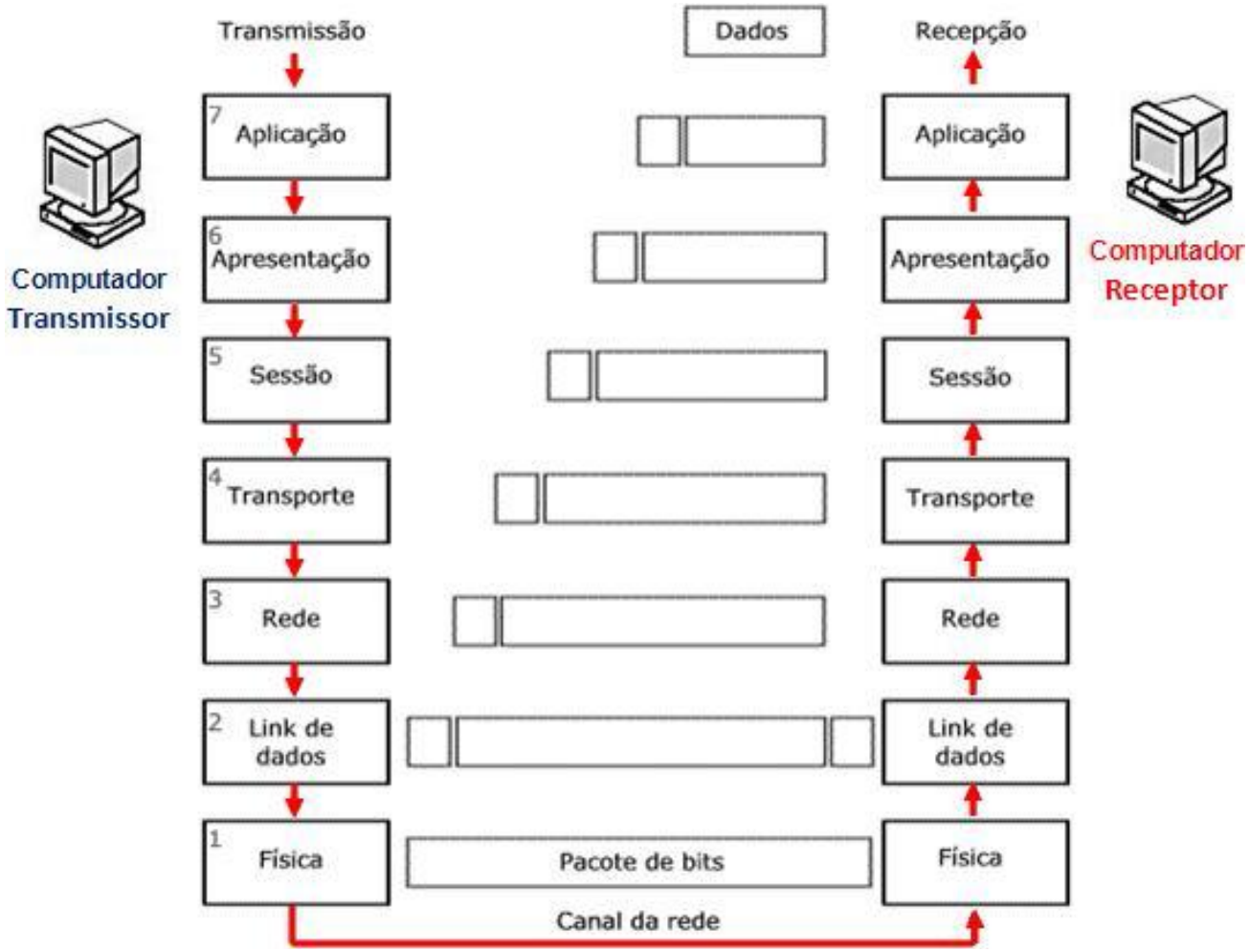
Camada de Sessão e de Apresentação

Sessão: Provimento de um mapeamento um-para-um entre uma conexão de sessão e uma conexão de apresentação, em qualquer momento. Evitar que uma entidade de apresentação seja sobrecarregada de dados, pelo uso do controle de fluxo de transporte. Restabelecimento de uma conexão de transporte para suportar uma conexão de sessão. Realização das atividades de gerência da camada de sessão.

Apresentação: Emissão de uma solicitação para que a camada de sessão estabeleça uma sessão. Iniciação da transferência de dados entre entidades de aplicação ou usuários. Execução de quaisquer transformações ou conversões de dados que forem requeridas. Emissão de uma solicitação para que a camada de sessão encerre a sessão.

Camada de Aplicação

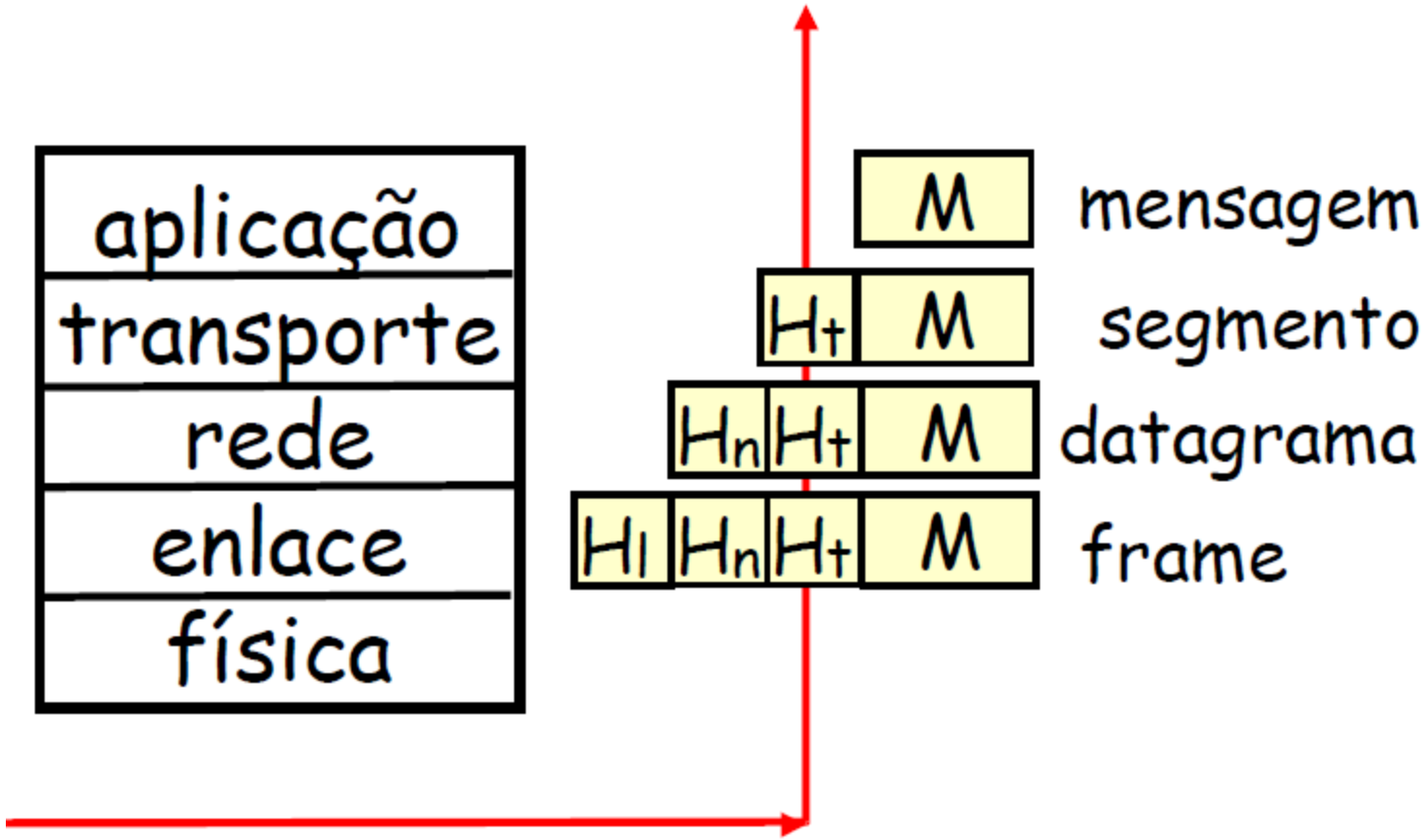
Aplicação: Execução das funções de aplicação comuns, que são funções que proporcionam capacidades úteis a muitas aplicações. Execução das funções de aplicação específicas, que são funções necessárias para atenderem aos requisitos de uma aplicação em particular.



Modelo TCP/IP

Definido pela primeira vez em 1974 e revisto em 1988, quando foi oferecida uma nova perspectiva;

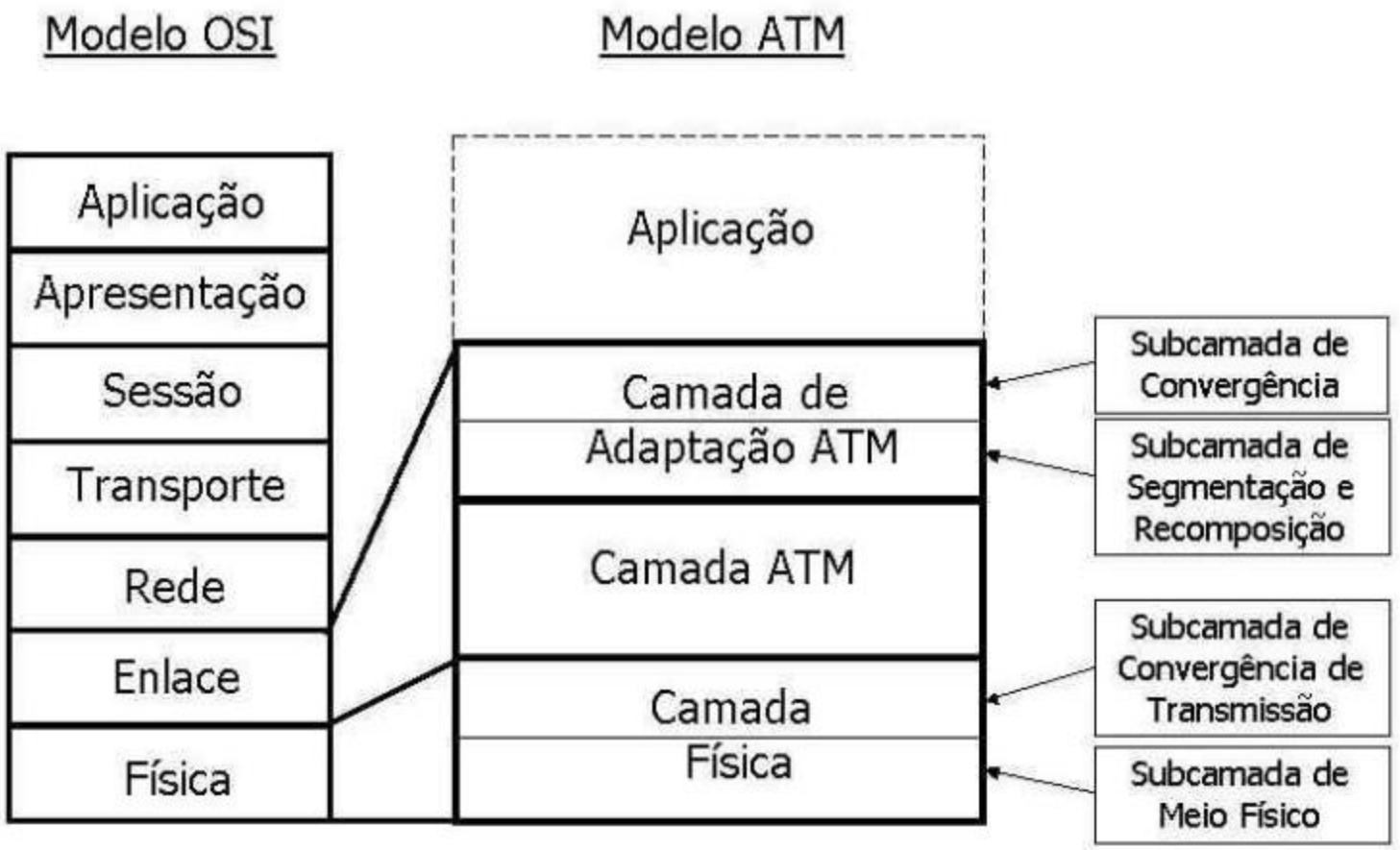
O Modelo TCP/IP é uma arquitetura de rede, pois especifica como os serviços devem ser oferecidos e quais os protocolos a serem utilizados.



ATM (*Asynchronous Transfer Mode*)

- Criado no início da década de 1990, para redes geograficamente distribuídas
- Foi lançado com estardalhaço, mas não vingou;
- Orientado a conexões;
- Permite a criação de circuitos virtuais permanentes;
- Envia pequenos pacotes chamados células;
- Possibilidade de uma entrada ser replicada em várias saídas;
- Velocidades de 155Mbps ou 622Mbps.

ATM (*Asynchronous Transfer Mode*)

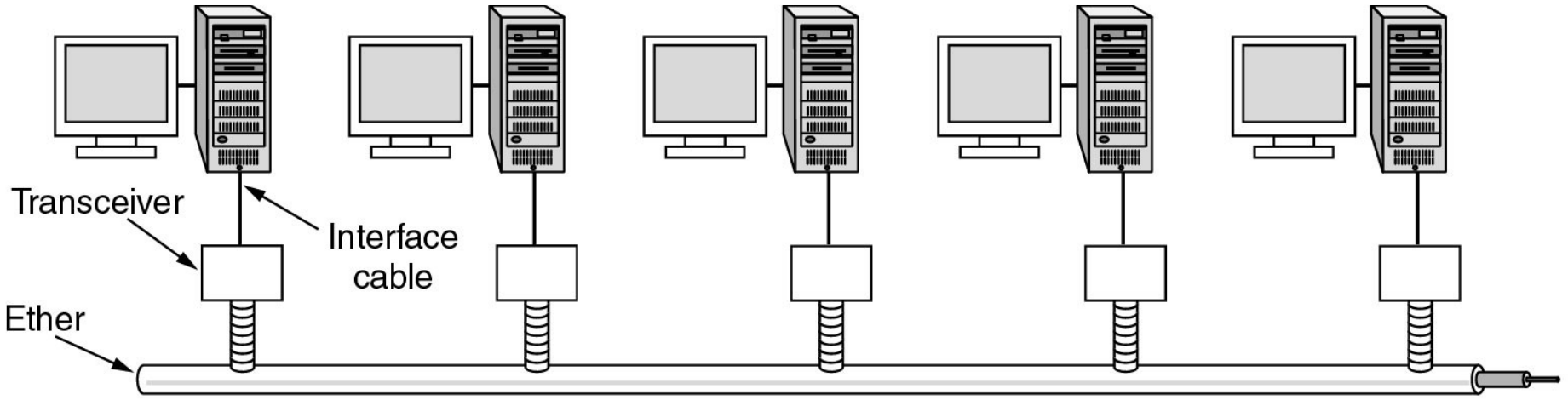


Redes Ethernet

- Criado para redes geograficamente distribuídas e LANS;
- Originou-se a partir da ALOHANET em1976 no PARC *(Palo Alto Research Center*) da Xerox (Metcalfe e Boggs)*;*
- Em 1978 criou-se o padrão Ethernet de 10Mbps, chamado de DIX (Dell, Intel e Xerox) que virou o padrão IEEE 802.3 em 1983;
- Metcalfe cria a 3COM para vender adaptadores ethernet para computadores;
- No mesmo período o comitê padronizava a rede *Token Ring*, o barramento de *tokens* IEEE 802.4 (GM) e a rede de *tokens* em anel IEEE 802.5 (IBM);

Redes Ethernet

- Meio compartilhado com velocidade inicial de 2,96Mbps;
- Utiliza um *token* para controle da transmissão.



Camada de Aplicação

Camada de Aplicação

A camada de aplicação é o nível que possui o maior número de protocolos existentes, devido ao fato de estar mais perto do usuário e os usuários possuírem **necessidades** diferentes.

Esta camada fornece ao usuário uma interface que permite acesso a diversos serviços de aplicação, convertendo as diferenças entre diferentes fabricantes para um denominador comum.

Por exemplo, em uma transferência de arquivos entre máquinas de diferentes fabricantes pode haver convenções de nomes diferentes (por exemplo, antigamente o sistema operacional DOS tinha uma limitação de somente 8 caracteres para o nome de arquivo, o UNIX nunca teve essa limitação), formas diferentes de representar as linhas, e assim por diante.

Aplicação X Protocolo de Aplicação

Aplicação: processos distribuídos em comunicação:

- rodam nos computadores usuários da rede como programas de usuário
- trocam mensagens para realização da aplicação
- e.x., email, ftp, Web

Protocolos de aplicação:

- fazem parte das aplicações
- definem mensagens trocadas e as ações tomadas
- usam serviços de comunicação das camadas inferiores

Arquiteturas de Aplicações de Redes

Arquitetura Cliente-Servidor

Servidor: sempre em funcionamento, atende as requisições dos clientes, endereço fixo;

Cliente: pode estar em funcionamento ou não, requisita algo ao servidor, geralmente endereço variável, geralmente inicia a comunicação;

Arquitetura P2P

Não há um servidor sempre funcionando;

Pares de hospedeiros (*peers*) comunicam-se diretamente entre si;

Possui alta escalabilidade;

Pelo fato de ser distribuído e descentralizado, não há garantias de um arquivo (por exemplo) ser encontrado num exato momento;

Ex.: rede Gnutella.

Comunicação de Aplicações

Quem se comunica não é a aplicação é um **processo**.

Processo: programa executado num host.

- dentro do mesmo host: comunicação interprocessos;
- processos sendo executados em diferentes hosts se comunicam através da troca de mensagens pela rede.

Numa aplicação de rede temos um par de processos, um encarregado de enviar mensagens requisitando algo (cliente) e outro para a recepção e resposta destas requisições (servidor).

“No contexto de uma sessão de comunicação entre um par de processos, o processo que inicia a comunicação (o primeiro a contatar o outro no início da sessão) é rotulado de **cliente**. O processo que espera ser contatado para iniciar a sessão é o **servidor**.”

SOCKETS

Uma mensagem enviada de um processo para outro, em hosts distintos, tem de passar pela rede. Um processo recebe e envia mensagens para rede através de seu socket.

Socket é a interface entre a camada de aplicação e a camada de transporte.

Socket também denominada **Interface de Programação da Aplicação (*Application Programming Interface* – API),** através da qual uma aplicação utiliza a rede para enviar e receber mensagens.

Serviços Necessários a uma Aplicação

Transferência Confiável de Dados

- Devemos saber se a aplicação é tolerante a perdas

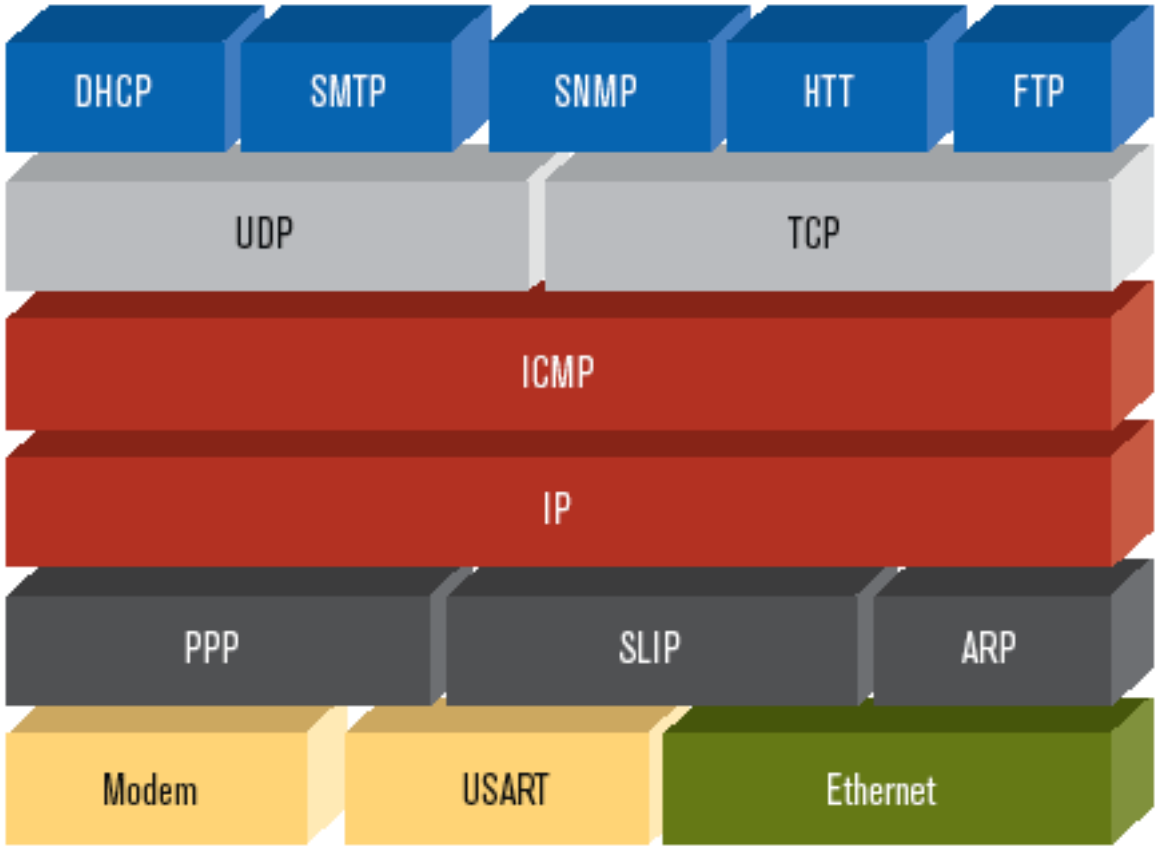
Largura de Banda

- Algumas aplicações necessitam de uma banda mínima
- Algumas aplicações melhoram consideravelmente com o aumento da banda

Temporização

- Algumas aplicações são sensíveis a atrasos na entrega das mensagens

Modelo TCP/IP e suas camadas



TCP x UDP

Serviços TCP

- ***orientado á conexão****:* conexão requerida entre cliente e servidor;
- ***transporte confiável****:* dados perdidos na transmissão são recuperados;
- ***controle de fluxo****:* compatibilização de velocidade entre o transmissor e o receptor;
- ***controle de congestionamento****:* protege a rede do excesso de tráfego;
- *não oferece:* garantias de temporização e de banda mínima.

Serviço UDP

- ***Não*** *orientado á conexão*
- transferência de dados **não** confiável entre os processos transmissor e receptor;
- **Não** oferece controle de fluxo;
- **Não** oferece controle de congestionamento;
- **Garantia de temporização e de banda mínima**.

DNS

DNS (*Domain Name System* – Sistema de Nomes de Domínio), usa UDP, porta 53;

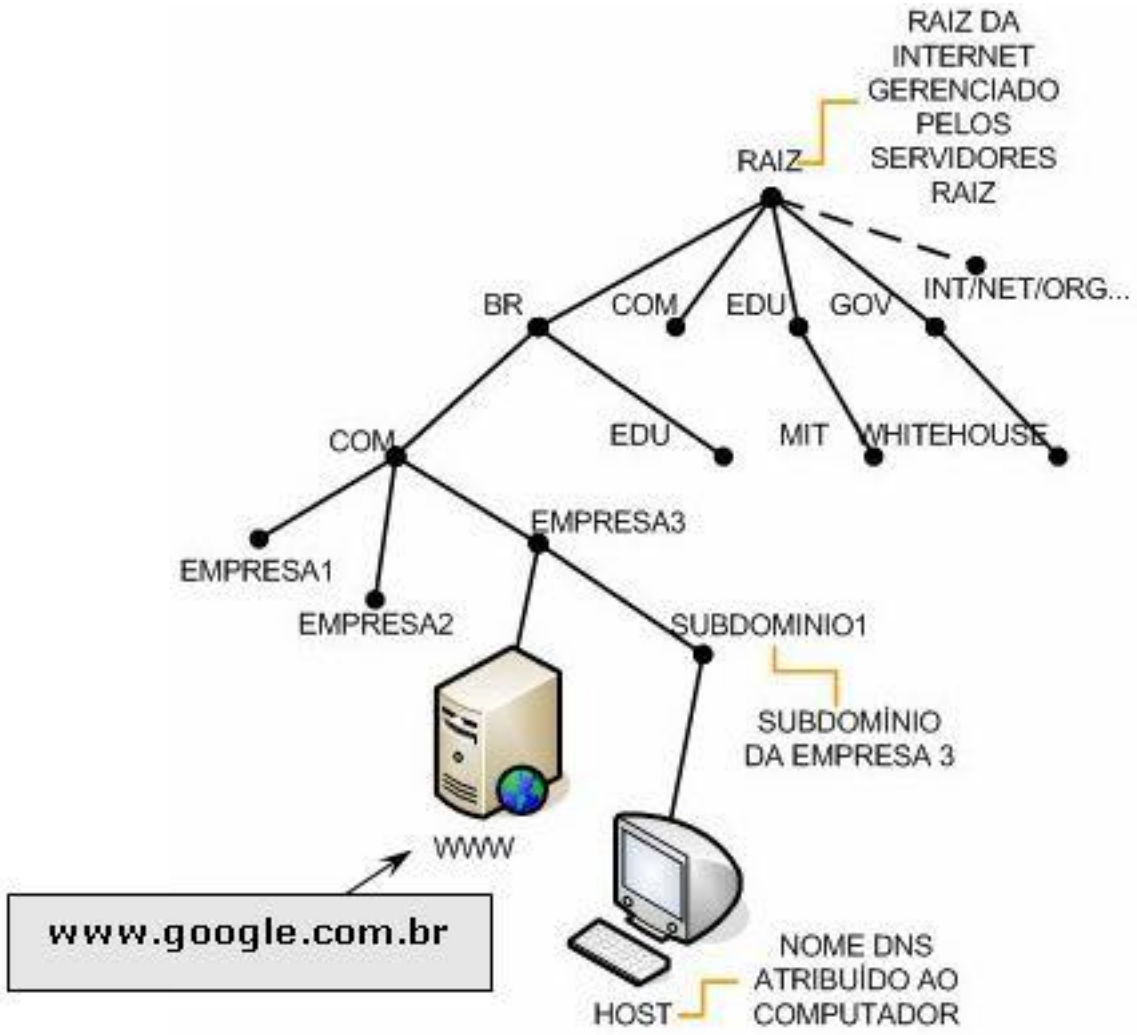
Definido nos RFC's 1034 e 1035;

É um banco de dados distribuído implementado em uma hierarquia de servidores de nome (DNS *Servers*).

É um protocolo de camada de aplicação que permite que hospedeiros consultem o BD distribuído; Serviço de diretórios que traduz nomes de hospedeiros para endereços IP.

Árvore DNS

Há uma **padronização** da nomenclatura onde cada nó da árvore é separado no nome por pontos. No nível mais alto podemos ter: COM (organizações comerciais), EDU (instituições educacionais), GOV (instituições governamentais), MIL (órgãos militares), ORG (outras organizações), NET (Netwotking), etc.



Tipos de Servidores DNS

Servidores de Nomes Raiz:

Existem 13 no mundo: Cada servidor, na verdade, é um conglomerado de servidores replicados;

Servidores de Nomes de Domínio de Alto Nível (TLD – *Top-Level Domain*).:

São servidores responsáveis por domínios de alto nível, como com, org, net, edu e gov. Responsáveis pelos domínios de países, como br, uk, fr;

Servidores de Nomes com Autoridade:

Toda organização com *hosts* que possam ser acessados publicamente pela internet, deve fornecer registros que mapeiem os nomes dos seus *hosts* para endereço IP. Geralmente grandes empresas e grandes universidades possuem servidores DNS primários e secundários (*backup*);

Servidor DNS Local

Não pertence propriamente a hierarquia de servidores DNS; Mantidos por ISP's; Funciona como um proxy DNS;

Exemplo de Uso do DNS

Um ***browser*** é executado e é digitado um endereço HTTP (www.ifbaiano.edu.br/reitoria), o *browser* retira o nome do hospedeiro (ex: www.ifbaiano.edu.br) e o passa para o cliente DNS (executado na própria máquina do usuário);

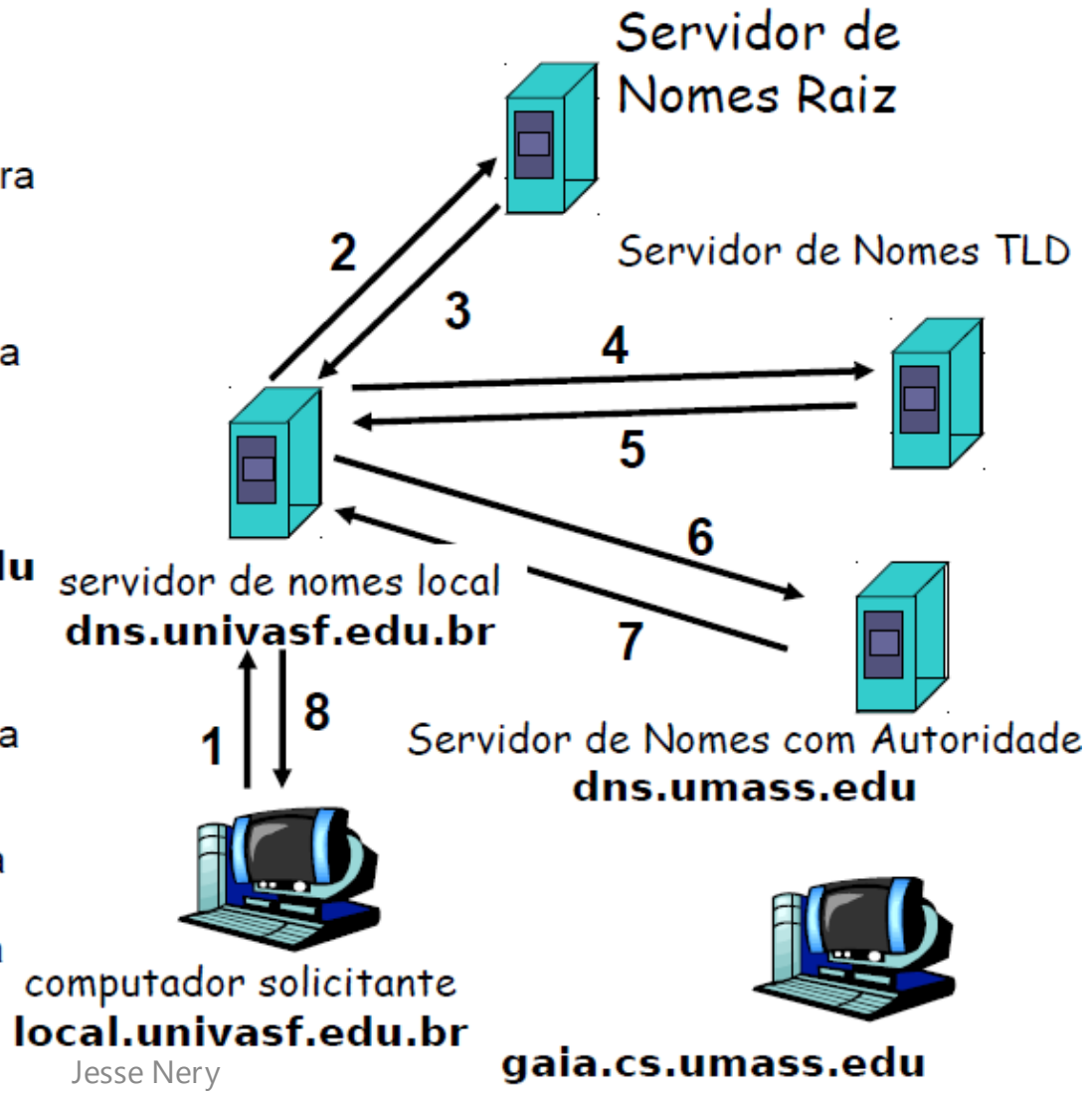
O **cliente** DNS envia uma consulta contendo o nome do hospedeiro para um **servidor** DNS;

O **cliente** recebe do servidor DNS uma resposta contendo o endereço IP do hospedeiro e o envia para o *browser*;

O ***browser*** faz a conexão com o hospedeiro através do **endereço IP** recebido.

Exemplo de Requisição DNS

1. O *host* requisita o endereço de **gaia.cs.umass.edu**, que é repassado para o DNS local **dns.univasf.edu.br**;
2. O servidor local repassa a consulta para um servidor DNS raiz;
3. O servidor raiz verifica o sufixo **edu** e retorna uma lista de endereços IP de servidores de nomes TLD responsáveis por **edu**;
4. O servidor de nomes local retransmite a mensagem para um dos servidores da lista;
5. O servidor TLD observa o sufixo **umass.edu** e retorna o endereço IP para o servidor DNS com autoridade para a *University of Massachussetts*;
6. O servidor DNS local envia a mensagem para o servidor de nomes com autoridade para o endereço **dns.umass.edu**;
7. O servidor responde com o endereço IP para **gaia.cs.umass.edu**;
8. O Servidor de nomes local repassa o IP para o computador solicitante.



Outros Serviços - DNS

Apelidos de Hospedeiros: um host pode ter um outro nome mais simples (apelido) ao invés de usar o seu **nome canônico**. ex: a24.g.akamai.net pode ter entre outros o apelido www.amd.com

Apelidos de Servidor de Correio: simplificar nomes de servidores de correio. ex: o servidor do yahoo real é rc.yahoo.akadns.net

Distribuição de Carga: alguns endereços possuem mais de um servidor à sua disposição para atender aos chamados, para auxiliar na utilização o servidor DNS pode relacionar um nome a vários endreços IP e a cada requisição envia todos mas altera a ordem (rodízio de endereços IP).

Protocolo HTTP

HTTP (***HiperText Transfer Protocol***) – Protocolo de Transferência de HiperTexto;

modelo cliente/servidor:

- *cliente:* ***browser que solicita, recebe e apresenta objetos da Web***
- *servidor:* ***envia objetos em resposta a pedidos***
- http1.0: RFC 1945
- http1.1: RFC 2616

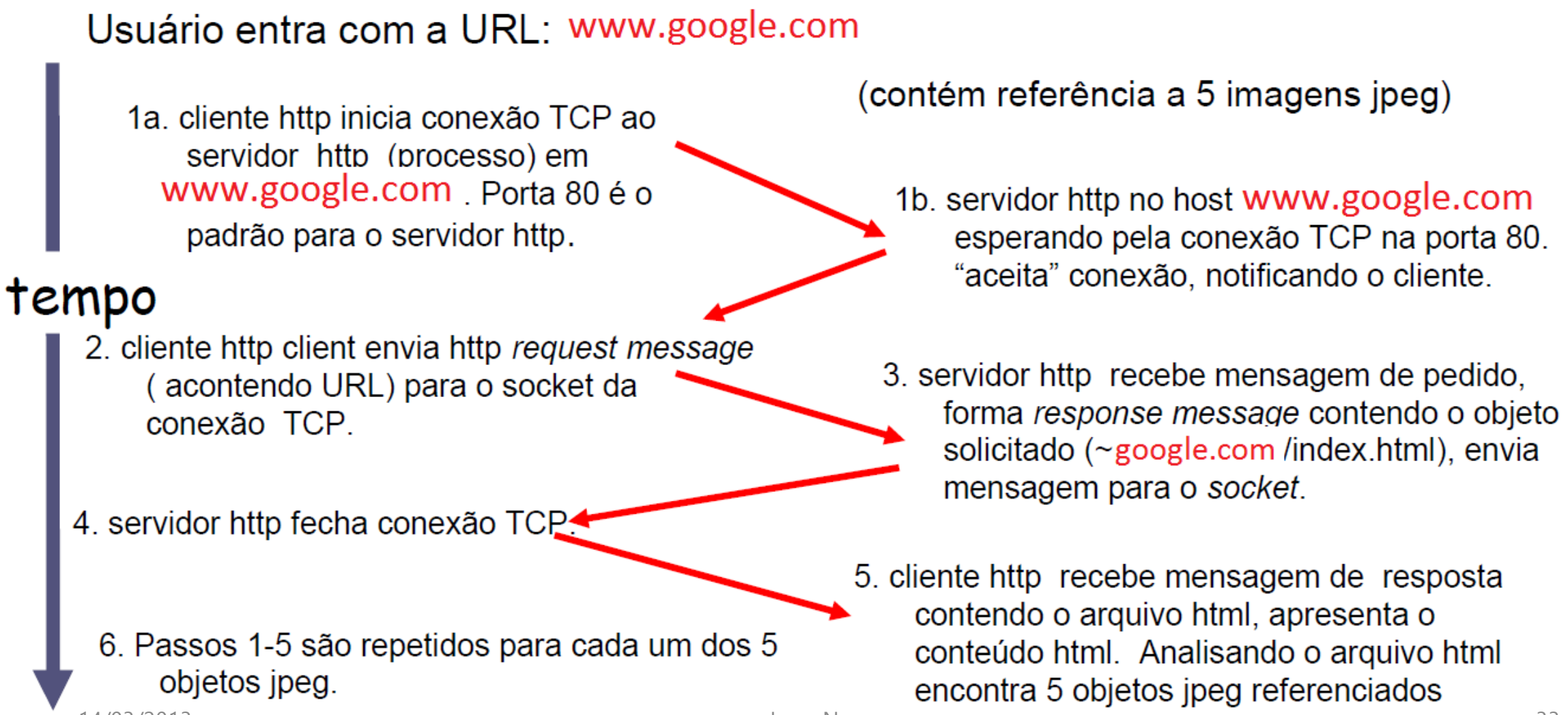
Protocolo HTTP

Uma página web é constituída de objetos; Os objetos são o código-fonte da página, as imagens etc;

Browser é um agente usuário para a Web, ele apresenta a página requisitada ao usuário e fornece numerosas características de configuração e navegação;

Servidor Web, abriga objetos Web, cada um endereçado por um URL (***Uniform Resource Locator***, o endereço de um recurso).

Funcionamento do Protocolo HTTP



Conexões Persistentes e Não-Persistentes

Conexões Não-Persistentes

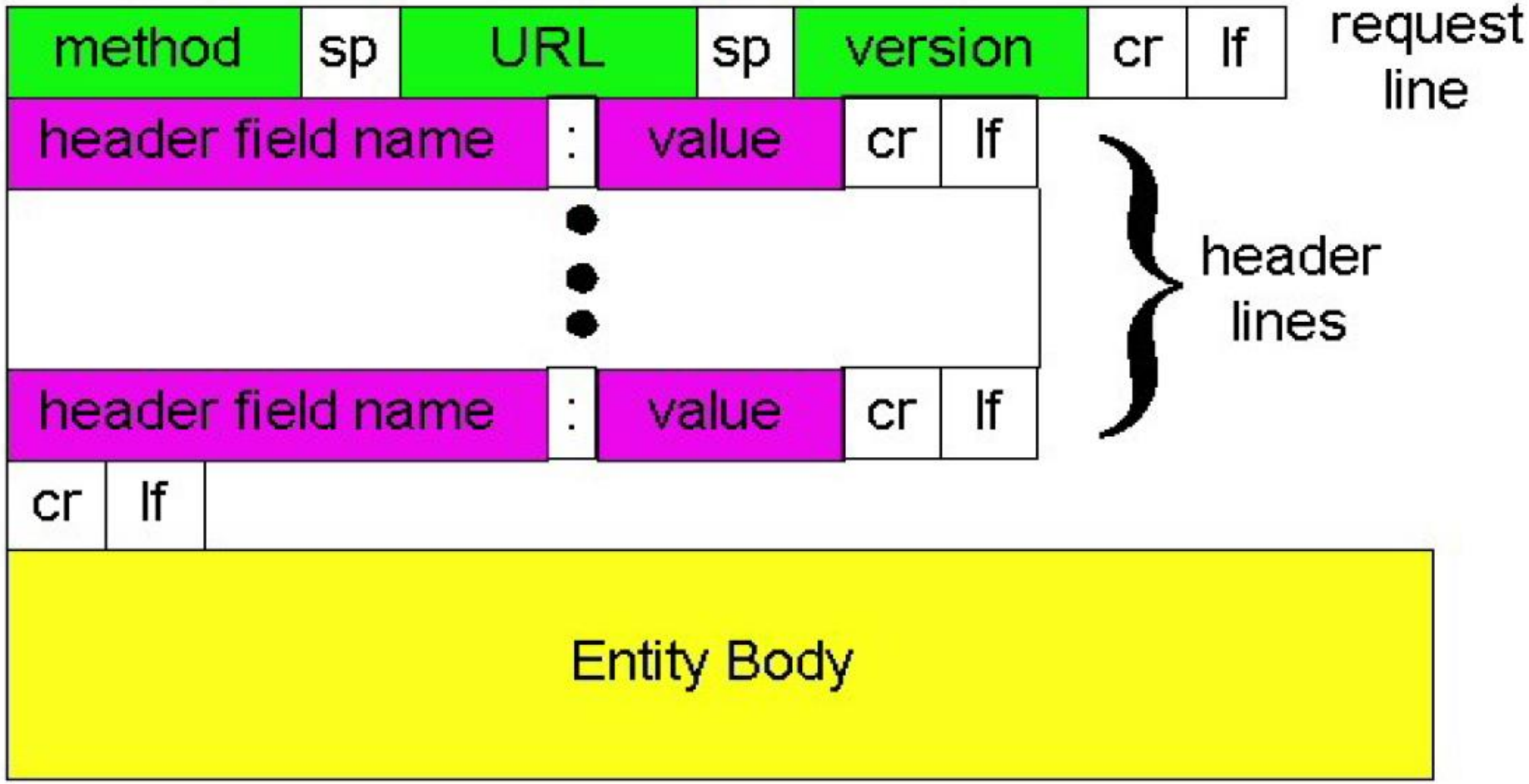
- HTTP/1.0: servidor analisa pedido, envia resposta e fecha a conexão TCP;
- 2 RTTs (*Round Trip Time* – tempo de viajem de ida e volta) para obter um objeto;
    - Conexão TCP
    - solicitação e transferência do objeto
- cada transferência sofre por causa do mecanismo de ***slow-start*** (partida lenta) do TCP;
- Os *browsers atualmente* abrem várias conexões paralelas.

Conexões Persistentes e Não-Persistentes

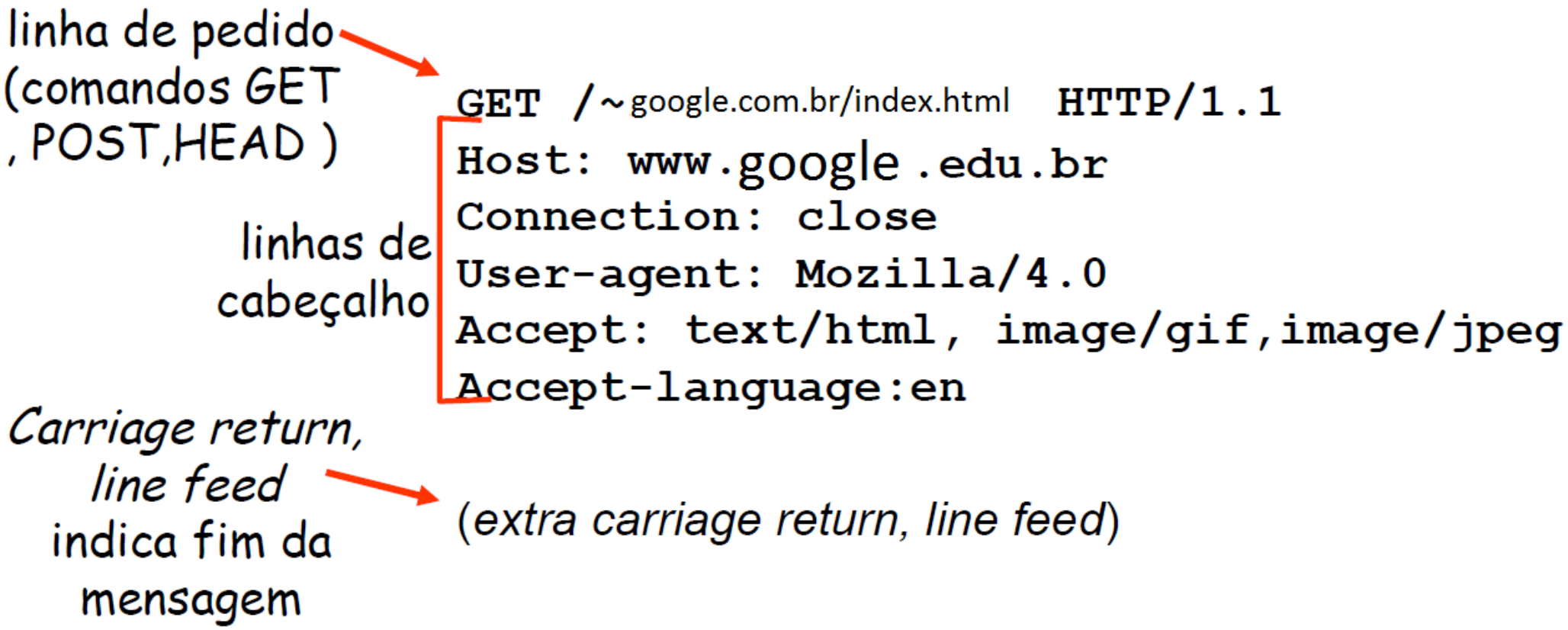
Conexões Persistentes

- modo *default* (padrão) para HTTP/1.1;
- na mesma conexão TCP são trazidos vários objetos;
- o cliente envia pedido para todos os objetos referenciados tão logo ele recebe a página HTML básica;
- poucos RTTs, menos *slow start*.

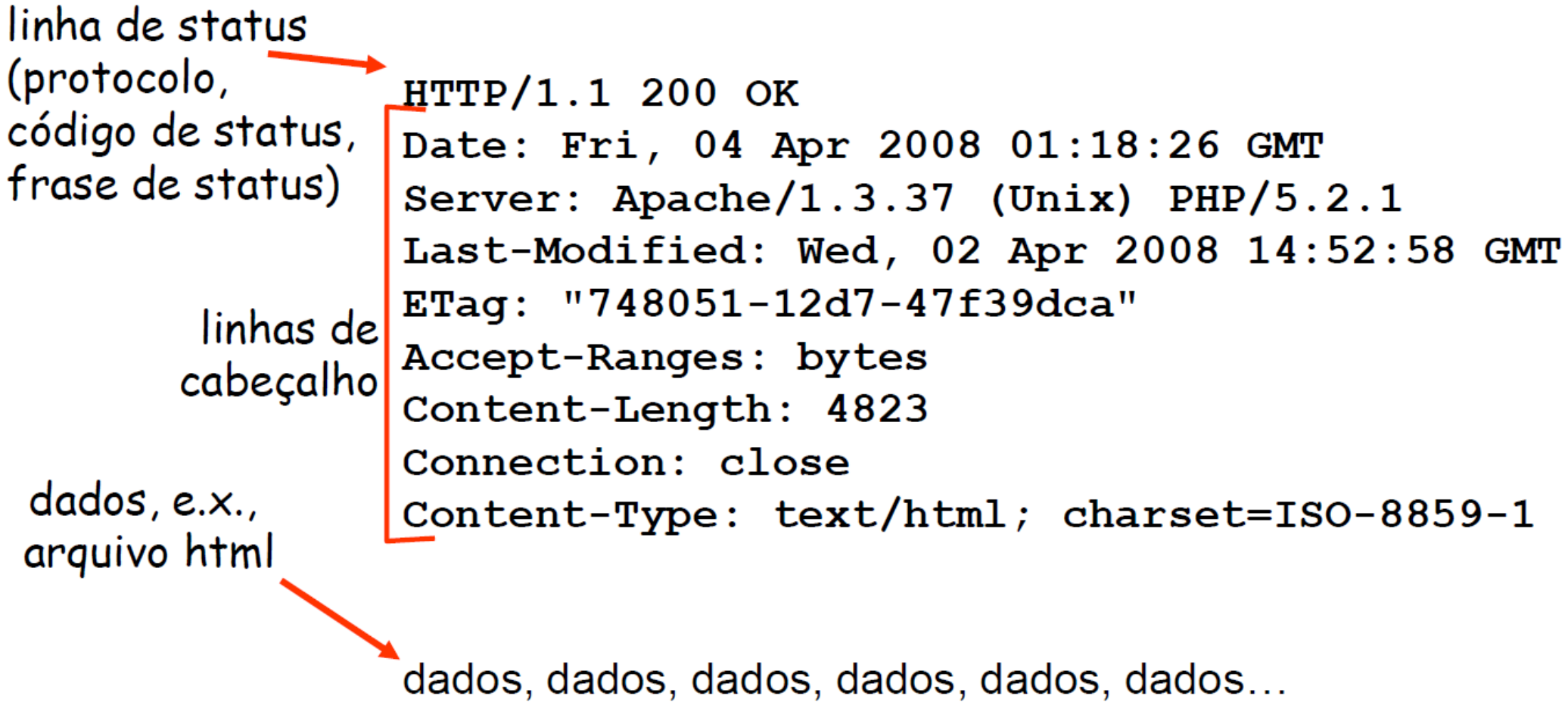
Formato de uma mensagem HTTP



Mensagem de requisição HTTP



Mensagem resposta HTTP



Código de Status das Mensagens HTTP

200 OK: requisição bem-sucedida e a informação é entregue com a resposta

301 Moved Permanently: objeto requisitado removido, nova localização informada adiante no cabeçalho Location:, neste ponto é informado a nova URL

400 Bad Request: requisição não entendida pelo servidor

404 Not Found: O objeto requisitado não encontrado no servidor

505 HTTP Version Not Supported: Versão do protocolo HTTP não suportada pelo servidor

Métodos HTTP

GET: Requisita um objeto do servidor;

POST: Enviar dados para um servidor (ex: formulário);

HEAD: Confirma a existência de um objeto no servidor;

PUT: Envia um objeto para o servidor;

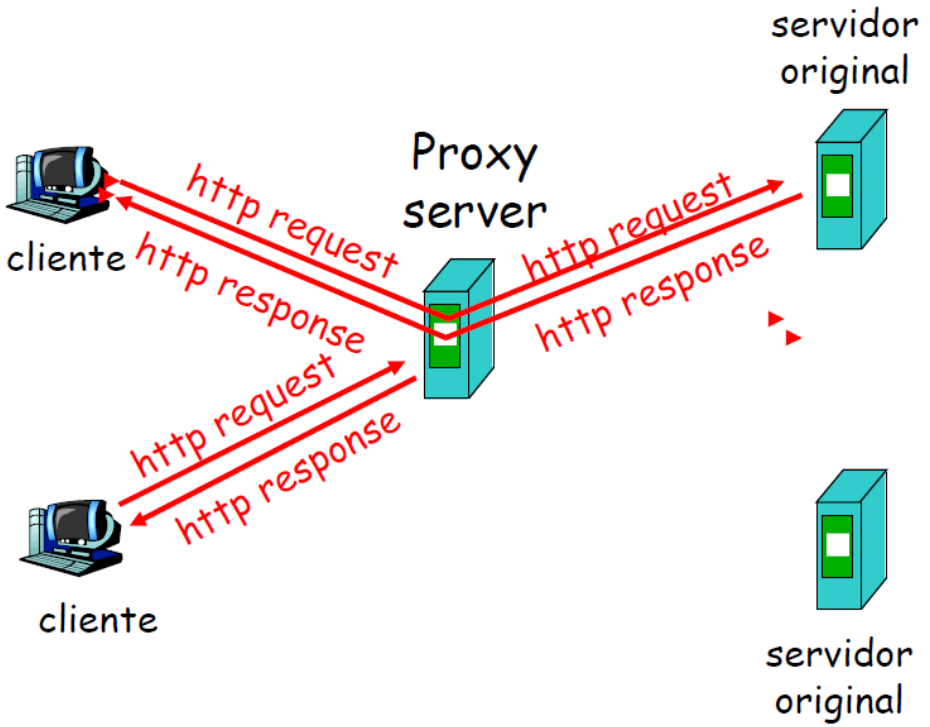
DELETE: Apaga um objeto no servidor.

Solução para um Protocolo *Stateless* - Cookie

- Utilizado para monitoramento e/ou acompanhamento do usuário numa conexão;
- Pode ser usado para criar uma camada de sessão de usuário;
- Um cookie possui quatro componentes:
    - Uma linha de cabeçalho de cookie na resposta HTTP (SET COOKIE: número);
    - Uma linha de cabeçalho de cookie na mensagem de requisição HTTP (COOKIE: número);
    - Um arquivo de cookies mantido pelo sistema e gerenciado pelo browser;
    - Um banco de dados no site da web.'

SERVIDOR PROXY

- Atende a requisições HTTP no lugar do servidor Web de origem;
- Funciona como cliente e servidor ao mesmo tempo;
- Diminui o tempo de resposta e o tráfego no enlace de acesso a internet;
- Utiliza um método chamado GET condicional



Telnet – Network Virtual Terminal Protocol

O Telnet é um protocolo **cliente-servidor** de comunicações usado para permitir a comunicação entre computadores ligados numa rede (exemplos: rede local / LAN, Internet), baseado em TCP. Basicamente este protocolo permite fazer a **emulação de terminal**, através do cliente pode acessar os recursos do servidor de Telnet. Antes de existirem os famosos chats em IRC (Internet Relay Chat) o Telnet já permitia este tipo de serviços.

Porém para acesso a console de servidores com **segurança** é melhor fazer uso de outro tipo de recurso como o SSh (Secure Shell) cujo conteúdo é encriptado antes de ser enviado. A senha que é enviada ao servidor pode ser capturada em muitos casos com o modo de rede promiscuo e com o uso de um analisador de rede.

Protocolo FTP

- Protocolo de **transferência** de arquivos de e para o computador remoto;
- Comunicação no modelo cliente/servidor: Cliente: inicia a conexão. Servidor: host remoto.
- RFC 959, porta 21 (controle) e porta 20 (dados)
- Trabalha com **duas** conexões:
    - Conexão de controle: permanente, utilizada para informações de controle; ex: id, senha, comandos...
    - Conexão de dados: temporária, utilizada para transferência de dados.

Protocolo FTP

- Envia as suas informações de controle fora da banda;
- Durante toda a sessão FTP o servidor mantêm informações de estado sobre o usuário;
- A conexão de controle é associada a um usuário;
- Por manter o estado da sessão, o servidor tem uma limitação alta de usuários conectados simultâneamente;

Comandos FTP

USER usermane Envia a identificação do usuário

PASS password Envia a senha do usuário

LIST Pede a listagem do diretório corrente, a lista é enviada por uma conexão de dados

RETR filename Pede que o servidor envie um arquivo <filename>

STOR filename Envia um arquivo <filename> para o servidor

331 Username OK, password required Nome do usuário OK, senha requisitada

125 Data connection already open; starting Transfer Conexão de dados aberta, iniciando a transferência

425 Can't open data connection Não é possível abrir a conexão de dados

452 Error writing file Erro ao escrever o arquivo

Protocolo SMTP

Definido no RFC 2821, usa a porta **25**;

As mensagens são em código ASCII de 7 bits;

Utiliza o protocolo **TCP** para transporte;

Usualmente não são utilizados servidores

intermediários para entregar a correspondência;

Protocolo SMTP

Etapas para envio de um e-mail:

1. O cliente SMTP faz uma conexão TCP com um servidor SMTP;
2. Feita a conexão é feito o *handshaking* (apresentação, identificação, endereço de entrega e origem);
3. É enviada a mensagem;
4. O processo se repete para cada e-mail.

SMTP versus HTTP

Quanto a comunicação: HTTP é um protocolo de **recuperação** de informações (*pull protocol*). A conexão é feita por quem quer receber o arquivo.

SMTP é um protocolo de **envio** de informações (*push protocol*). A conexão é feita por quem quer enviar o arquivo.

Quanto ao envio de dados: SMTP exige que a mensagem e os dados enviados estejam no formato ASCII de 7 bits, todos os objetos componentes da mensagem são enviados de uma só vez; HTTP **encapsula** cada objeto em sua própria mensagem.

Extensão MIME

MIME – *Multipurpose Internet Mail Extensions* (extensões multiuso do correio da internet), RFC

2045 e RFC 2046 (extensões do RFC 822);

Utilizado para **enviar** conteúdo que não seja no formato ASCII de 7bits;

São cabeçalhos extras **adicionados** ao já existentes do SMTP;

Permite o envio de arquivos diversos e informa a ação relacionada aos mesmos;

Comandos usuais:

Content-Type: permite a realização de uma ação específica pelo agente usuário destinatário com a mensagem;

Podem ser: text, video, application, audio, image.

Content-Transfer-Encoding: informa o tipo de codificação utilizada

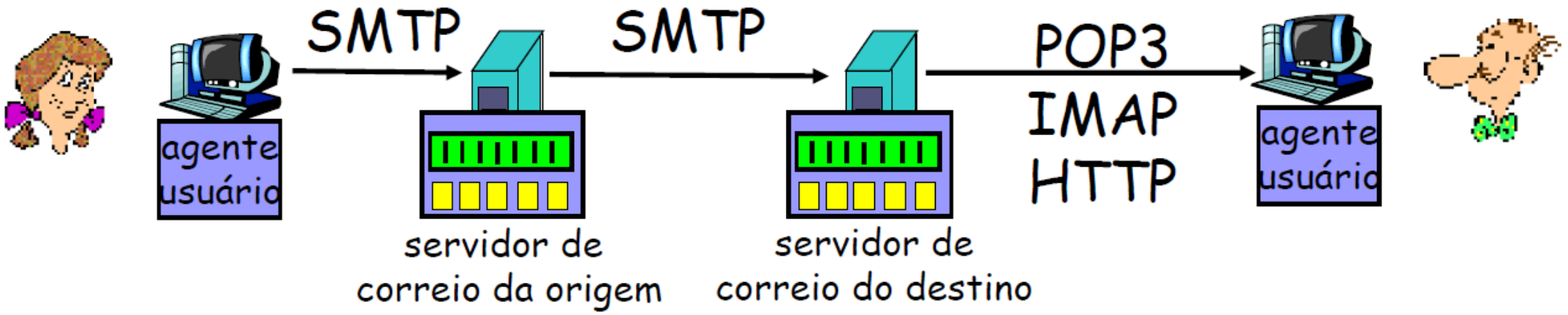
Exemplo

From: jessenery@Hotmail.com
To: alguem@msn.com
Subject: turma de redes
MIME-Version: 1.0
Content-Transfer-Encoding: base64
Content-Type: image/jpeg

base64 encoded data .....
.....
.....
..... base64 encoded data

Recuperando E-mails

Protocolos de Acesso



POP3

Definido no RFC 1939, porta 110;

Um protocolo **simples e com poucas** funcionalidades;

Uma vez que é feita a conexão TCP com o servidor o protocolo tem três fases:

- Autorização;
- Transação;
- Atualização

Comandos POP3

user <nome> inicia a identificação do usuário enviando o seu “nome”

pass <senha> envia a senha do usuário

List solicita uma listagem das mensagens existentes no servidor

retr <id da mensagem> solicita o envio da mensagem <id da mensagem>

dele <id da mensagem> solicita a remoção da mensagem <id da mensagem>

Quit termina a conexão

IMAP

- Definida no RFC 2060;
- Oferece mais recursos que o POP3 e muito mais complexo;
- Um servidor IMAP associa cada mensagem a uma pasta;
- Oferece **comandos** de criação, remoção e pesquisa de **pasta**, entre outros;
- Mensagens podem ser baixadas por pedaços.

DNS+(SMTP+IMAP+POP3)xHTTP

- Necessário conhecer o IP da pagina web para visualizar o recebimento/envio e-mails;
- A comunicação, seja POP3 ou IMAP, é realizada no servidor HTTP e as informações são enviadas para o usuário via HTTP;
- Ao enviar uma mensagem ocorre o sentido inverso.

Camada de Transporte

TCP

O TCP tem como finalidade básica fornecer o **transporte confiável**, através de um circuito lógico robusto de conexão entre um par de processos. Este protocolo se preocupa exclusivamente com a parte de transporte.

Aplicações que necessitam de transporte confiável se utilizam do protocolo de transporte TCP, porque este **verifica** se os dados são enviados de forma correta, na sequência apropriada, pela rede.

Características fundamentais do TCP

Orientado à conexão: A aplicação envia um pedido de conexão para o destino e usa a conexão para transferir dados.

Ponto a ponto: uma conexão TCP é estabelecida entre dois pontos.

Confiabilidade: O TCP garante a entrega dos dados sem perdas, duplicação ou outros erros.

Full-duplex: Pode haver troca de dados em simultâneo, em ambas as direções, pelos dois pontos da conexão.

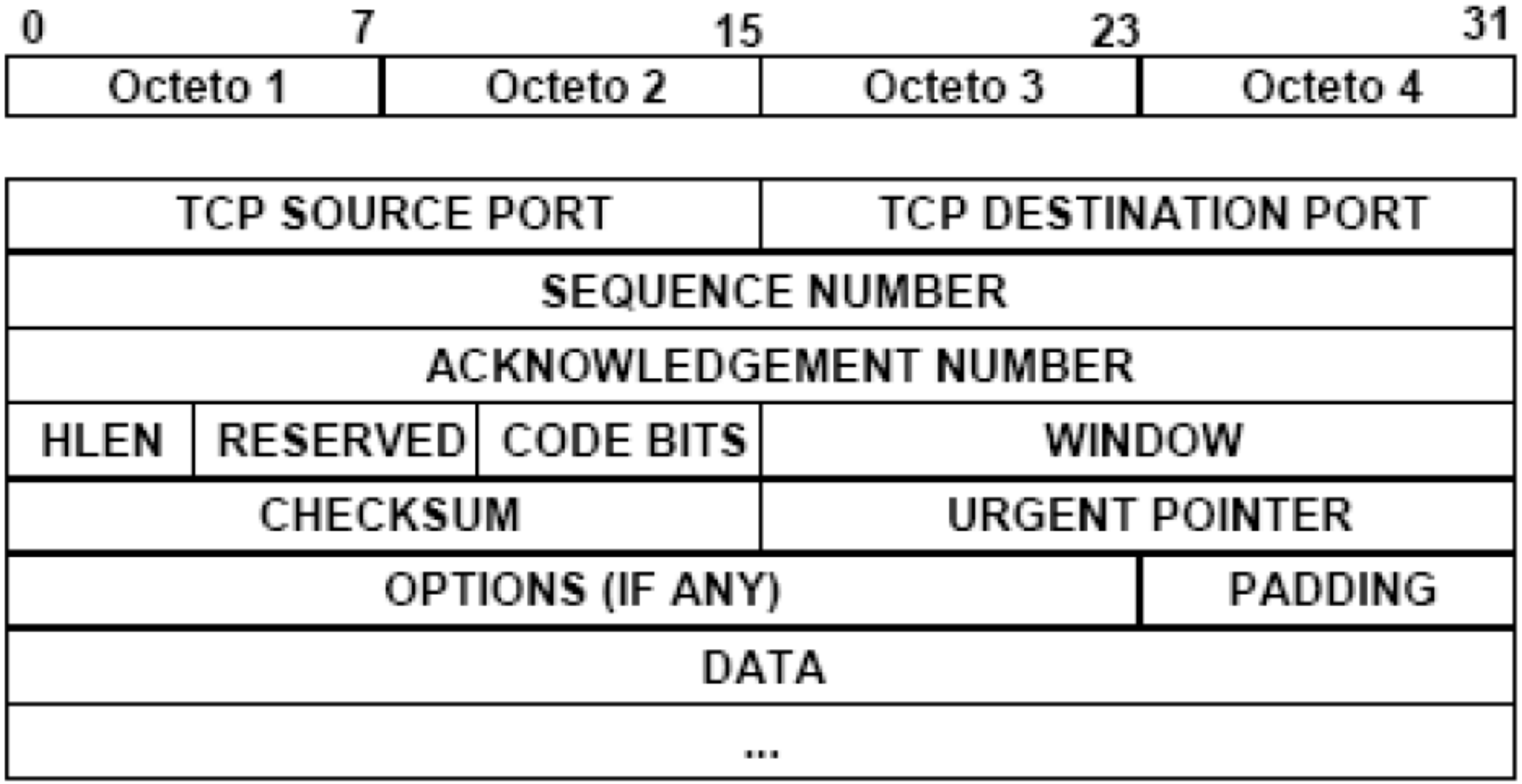
Características fundamentais do TCP

Interface Stream: Fluxo contínuo de dados; o TCP não garante que os dados sejam recebidos nos mesmos blocos em que foram transmitidos.

3-way Handshake: Mecanismo fiável de conexão em 3 vias, garantindo uma inicialização fiável e sincronizada entre os pontos.

Finalização da conexão controlada: O TCP garante a entrega de todos os dados depois de terminada a ligação.

Pacote TCP


0
7
15
23
31
Octeto 1
Octeto 2
Octeto 3
Octeto 4
TCP SOURCE PORT
TCP DESTINATION PORT
SEQUENCE NUMBER
ACKNOWLEDGEMENT NUMBER
HLEN
RESERVED
CODE BITS
WINDOW
CHECKSUM
URGENT POINTER
OPTIONS (IF ANY)
PADDING
DATA
...

Pacote TCP

TCP SOURCE PORT (bits 0-15): Porta origem da mensagem.

TCP DESTINATION PORT (bits 16-31): Porta destino da mensagem.

SEQUENCE NUMBER (bits 32-63): Este campo indica o número de **seqüência** dos dados sendo transmitidos. Se o bit SYN=1 então este número de seqüência SQN (Sequence Number) é o inicial ISN (Initial Sequence Number), ou seja, se SYN=1 então SQN=ISN que é atribuído durante o estabelecimento da conexão. Este número é utilizado nas subsequentes transmissões para determinar o próximo número a ser utilizado na seqüência (este número nunca deve ser 0 ou 1, a seqüência começa com um valor aleatório). Quando o bit ACK=1 então o ISN passa a ser o SQN comum. Vale a pena mencionar que ambos os números de seqüência dos fluxos de dados (de A para B e de B para A) são completamente diferentes, já que os dados transmitidos por um e outro lado são diferentes.

Pacote TCP

ACKNOWLEDGE NUMBER (bits 64-95): Esse campo possui um número que significa o **reconhecimento** dos dados recebidos até então no sentido inverso. São trocados ACK de um sentido a outro com se levando em consideração o número de SEQUENCE NUMBER inicial praticado pela outra máquina. O valor de ACK informa sempre o próximo byte ainda não recebido do conjunto contíguo de bytes recebidos do transmissor.

HLEN ou DATA OFFSET (bits 96-99): Esse campo informa o número de palavras de 32 bits contidas no cabeçalho do TCP.

RESERVED (bits 100-103): Campo reservado para o uso futuro.

CODE BITS (bits 104-111): São formados por oito bits: CWR, ECE, URG, ACK, PSH, RST, SYN e FIN, cuja utilização é mostrada abaixo:

Pacote TCP

CWR – bit de controle de **congestionamento** utilizado pelo ECN (Explicit Congestion Notification – veja a RFC 3168). O bit CWR (Congestion Window Reduced) é utilizado pelo transmissor para informar ao receptor que a janela de congestionamento foi reduzida. Quando a janela de congestionamento é reduzida, implica que menos dados são enviados por unidade de tempo, isto com o único propósito de satisfazer a carga (volume total de tráfego) da rede, ou seja, na presença de congestionamento na rede o bit CWR é ativado.

ECE – bit utilizado também pelo ECN (veja a RFC 3268). O bit ECE (ECN Echo) é utilizado pela pilha de protocolos TCP/IP do receptor para dizer ao transmissor que ele recebeu um pacote com indicação de **congestionamento** CE. Os bits CWR e ECE inicialmente faziam parte do campo RESERVED e de vido a isto alguns computadores, não estariam habilitados para entender o significado destes bits, sendo assim, eles simplesmente ignorarão ou rejeitarão os pacotes que tenham CWR = 1 e ECE = 1.

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 URGENT POINTER que indica o fim dos dados urgentes.

Pacote TCP

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.

RST – bit de RESET: Informa o destino que a conexão foi abortada neste sentido pela origem.

SYN – bit de **Sincronismo**: ë o bit que informa que este é um dos dois primeiros segmentos de estabelecimento da conexão.

FIN – bit de **Terminação**: indica que este pacote é um dos pacotes de finalização da conexão.

Pacote TCP

WINDOW (bits 112-127): Este campo informa o **tamanho** disponível em bytes na janela de recepção da origem deste pacote. Isso ajuda a efetuar o controle de fluxo adequado, evitando o estouro de buffer do receptor.

CHECKSUM (bits 128-143): O cálculo do Checksum de todo o cabeçalho TCP é alocado neste campo. Se o cabeçalho não finaliza em um comprimento de 16 bits, os bits que faltam (para dar 16 bits) são zerados. Durante o cálculo do checksum, este campo é zerado. Neste campo também é considerado o pseudo-cabeçalho de 96 bits que contem os campos DESTINATION, SOURCE ADDRESS, PROTOCOL, e TCP LENGTH. Isto dá uma segurança extra. Diferente do protocolo UDP, no TCP o campo CHECKSUM nunca é opcional.

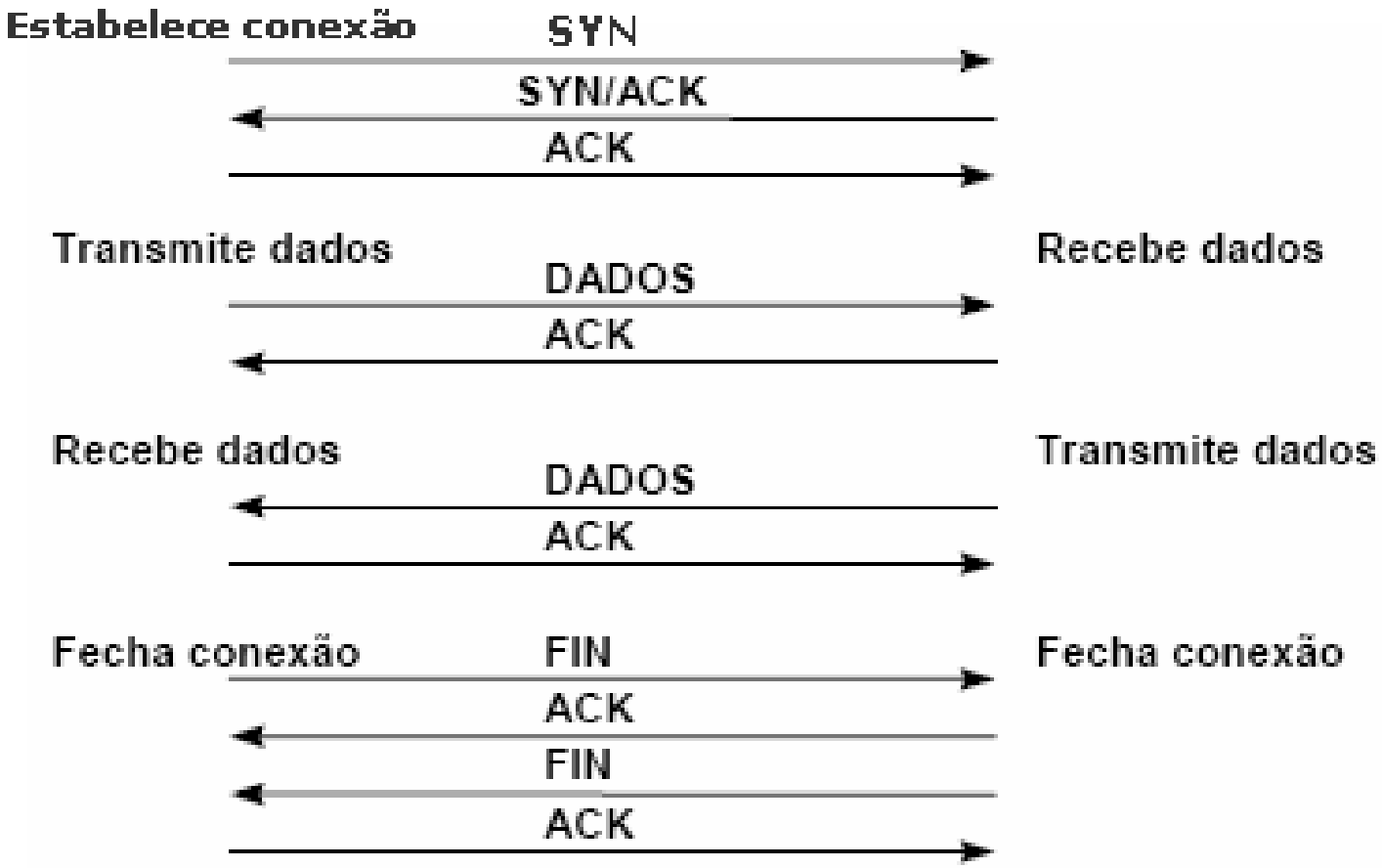
URGENT POINTER (bits 144-159): Este é um ponteiro que aponta para o fim dos dados os quais são considerados urgentes. Se a conexão tem dados importantes a serem processados pelo receptor, o transmissor pode ativar o bit URG e dizer que o campo URGENT POINTER aponte onde terminam os dados urgentes. Este campo indica um número positivo que corresponde ao valor de Offset do número de seqüência para este segmento em particular. Se o bit URG é 1 então este campo aponta para o número de seqüência do último Byte correspondente a uma seqüência de dados urgentes.

Pacote TCP

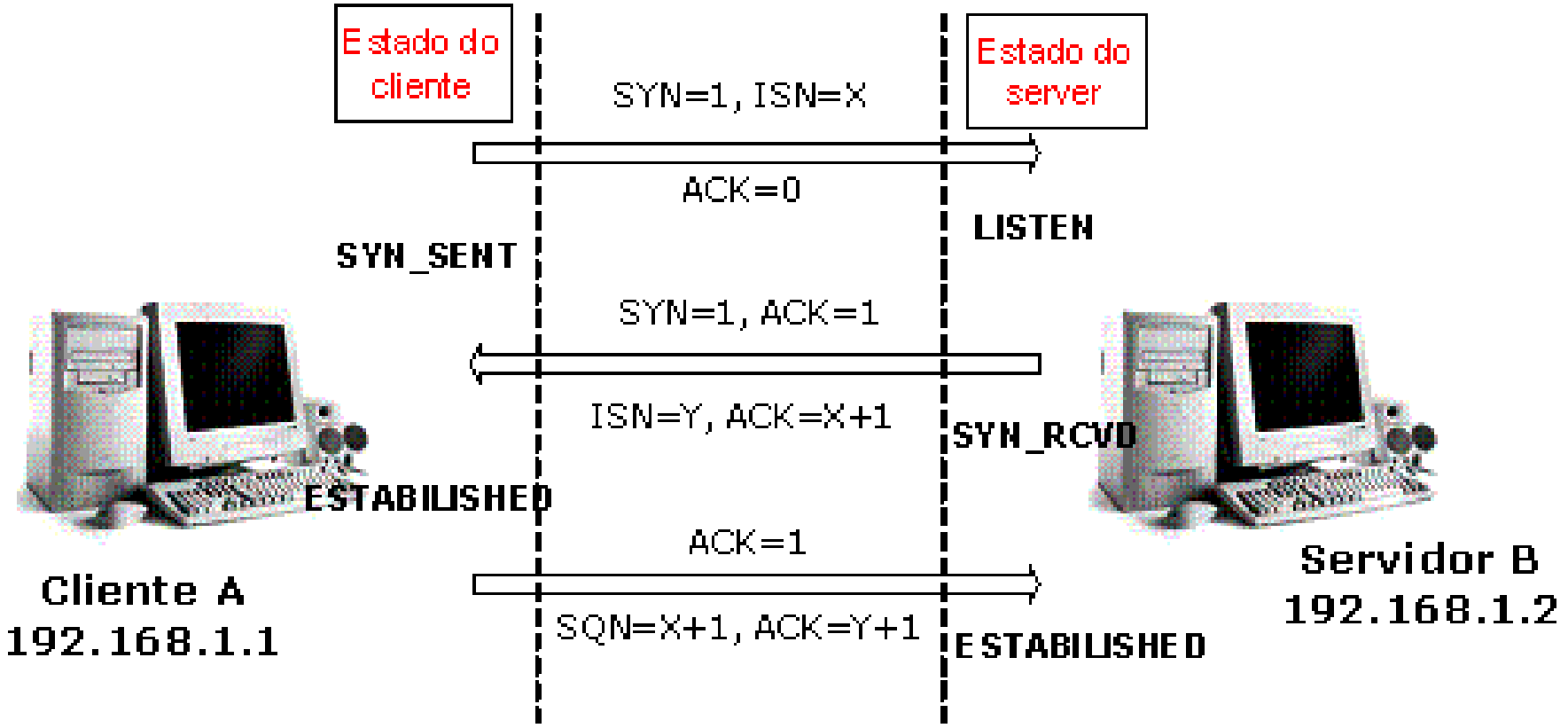
OPTIONS (bits 160-)**: Este campo é de comprimento variável e informa sobre as varias opções que o TCP pode transmitir. Basicamente, este campo possui 3 subcampos. Um subcampo inicial que diz o comprimento do campo OPTIONS, um Segundo subcampo que diz quais as opções que estão sendo utilizadas, e finalmente temos o subcampo das opções propriamente ditas. Para mais informações sobre as opções TCP veja o seguinte Link: http://www.iana.org/assignments/tcp-parameters.

PADDING (bits **): Este campo também é de comprimento variável e é utilizado para assegurar que o cabeçalho TCP termine e o campo de dados inicie com um comprimento de 32 bits, se isto não ocorrer, então bits 0 serão adicionados (padded) neste campo para dar o comprimento requisitado de 32 bits.

Etapas do TCP



Estabelecimento De Uma Conexão TCP



Troca De Dados Em Uma Conexão TCP

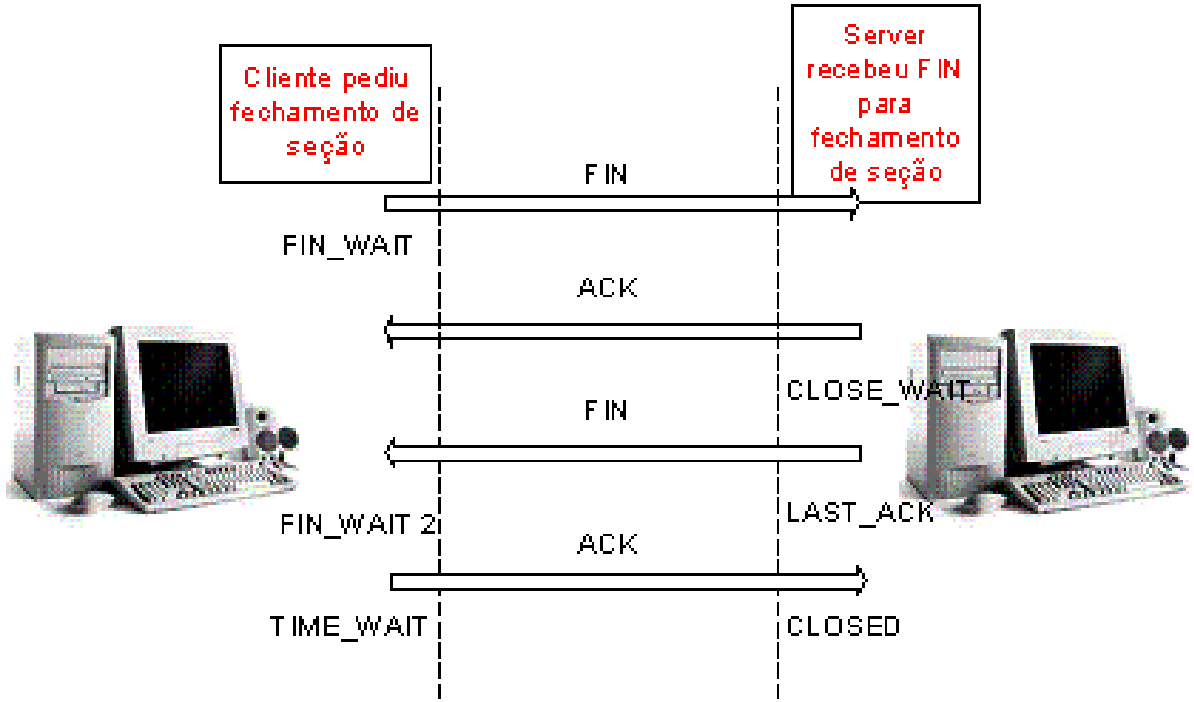
Durante a fase de transferência o TCP está equipado com vários mecanismos que asseguram a **confiabilidade e robustez**: números de seqüência que garantem a entrega ordenada, código detector de erros (checksum) para detecção de falhas em segmentos específicos, confirmação de recepção e temporizadores que permitem o ajuste e contorno de eventuais atrasos e perdas de segmentos.

Troca De Dados Em Uma Conexão TCP

Como se pode observar do cabeçalho TCP, existem permanentemente um par de **números de seqüência**, referidos como número de seqüência e número de confirmação positiva (positive ACKnowledgement) ou +ACK. O emissor determina o seu próprio número de seqüência e o receptor confirma o segmento usando como número ACK o número de seqüência do emissor. Para manter a **confiabilidade**, o receptor confirma os segmentos indicando que recebeu um determinado número de Bytes contíguos. Uma das melhorias introduzidas no TCP foi a possibilidade do receptor confirmar blocos fora da ordem esperada. Esta característica designa-se por ACK seletivo (selective ACK) ou apenas SACK.

Finalização Da Conexão TCP

Da mesma forma que a abertura de sessão, o protocolo TCP também realiza um fechamento formal de uma sessão exigindo uma troca de flags entre os computadores, de maneira a confirmar, explicitamente, que a sessão TCP será fechada.



Definição De Porta

Uma porta é um objeto **abstrato** que deve ser usado para identificar processos de aplicações. Para cada nível do modelo OSI existe um campo no protocolo da camada que indica para quem os dados encapsulados devem ser entregues.

Por exemplo, no nível de enlace, o campo TYPE (Tipo de protocolo) indica qual é o protocolo que está encapsulado no quadro Ethernet, um valor igual a 0x0800 neste campo indica que os dados devem ser passados para o IP.

Definição De Porta

No nível de rede, o campo PROTOCOL no cabeçalho do pacote IP identifica o protocolo para o qual o datagrama deve ser repassado, por exemplo, um valor de 7 neste campo indica que o pacote deve ser transferido para o protocolo de transporte UDP e se o valor deste campo for 6 então o pacote deve ser encaminhado para o TCP.

De maneira similar, para distinguir dentre as várias aplicações das camadas superiores, o nível de transporte associa um identificador a cada processo de aplicação. Esse identificador é chamado como "Número de Porta" (PORT NUMBER).

Portas Reservadas Do TCP

O TCP introduz o conceito de porta tipicamente associado a um serviço da camada de Aplicação para fazer uma tarefa (ligação) específica. Assim, cada um dos intervenientes na conexão dispõe de uma porta associada (com um valor de 16 bits) que dificilmente será o mesmo do interlocutor.

Alguns serviços (que fazem uso de protocolos específicos) são tipicamente acessíveis em portas fixas predefinidas denominadas como **Well-known ports** (portas bem conhecidas), que são aquelas portas numeradas do 1 a 1023.

Além destas, existem ainda duas gamas de portas, registradas e privadas ou dinâmicas. As portas bem conhecidas são atribuídas pela IANA (Internet Assigned Numbers Authority) e são tipicamente utilizadas por processos com direitos de sistema ou super-usuário.

5 RJE	43 NIXNAME	20 FTP-DATA	95 SUPDUP
7 ECHO	53 DOMAIN	21 FTP-CONTROL	101 HOSTNAME
9 DISCARD	67 BOOTPS	23 TELNET	102 ISO-TSAP
11 USERS	68 BOOTPC	25 SMTP	113 AUTJ
13 DAYTIME	69 TFTP	37 TIME	117 UUCP-PATH
15 NETSTAT	75 Any private dial-out	39 RLP	123 NTP
17 QUOTE	77 Any private RJE ser	42 NAMESERVER	
19 CHARGEN	79 FINGER	80 HTTP	

Processo De Retransmissão TCP

O TCP mantém um timer interno que é inicializado e decrementado no momento que um segmento é transmitido.

Se houver algum problema no processo de comunicação como link ruim, cabeamento com problema, erro causado por ruído, ou alta latência na rede e o segmento chegar com erros, atrasado ou mesmo destruído, o computador destino simplesmente descartará este segmento, não enviando um ACK de resposta.

O que pode causar **retransmissão** em uma rede?

- Aplicações lentas e mal construídas. Muito comum em aplicações antigas como Cobol e Clipper.
- Redes com problemas na infra-estrutura física como cabeamento com problemas de ruídos e erros de transmissão.
- Dispositivos ativos (Switches, Hubs) com problemas de Hardware.
- Placas de rede com problemas de Hardware.
- Drivers de placas de rede.
- Excesso de tráfego.
- Redes com problemas de projetos.

UDP (User Datagram Protocol)

Este protocolo de transporte pode ser considerado uma versão **econômica** do TCP, um protocolo que emagreceu demais e que dá às aplicações acesso direto ao serviço de entrega de datagramas.

O protocolo UDP faz o envio do datagrama, mas **não garante** que ele chegará efetivamente ao destino, portanto, é pouco confiável, isto devido a que é um protocolo não orientado para conexão.

O "pouco confiável" significa que não há técnicas no protocolo para confirmar que os dados chegaram ao destino corretamente ou se realmente chegaram.

Formato do cabeçalho UDP

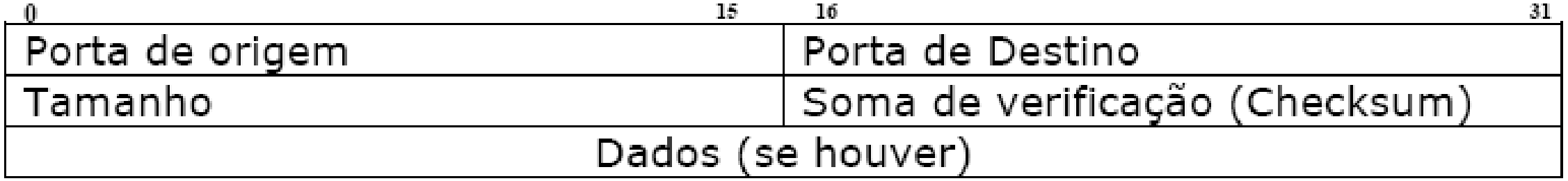
PORTA DE ORIGEM: Número da porta do computador transmissor.

PORTA DE DESTINO: Número da porta da aplicação solicitada no computador receptor.

TAMANHO DO SEGMENTO: Tamanho do cabeçalho UDP e dados UDP.

CRC: Checagem de redundância cíclica ou soma de verificação dos campos de cabeçalho e dados do UDP.

DADOS: Dados do nível superior, isto é, os dados do usuário.



Funcionamento Básico Do UDP

Assim como para o TCP, os pontos de acesso do UDP são geralmente designados por "Portas de protocolo" ou simplesmente "portas", em que cada unidade de transmissão de dados UDP identifica o endereço IP e o número de porta do destino e da fonte da mensagem, os números podendo ser diferentes em ambos os casos.

O UDP por sua vez é uma redução do TCP, feito para transmitir dados pouco sensíveis, como streaming de áudio e vídeo que não requerem de retransmissões.

No UDP não **existem checagens** e nem confirmação alguma. Os dados são transmitidos apenas uma vez, incluindo apenas um frágil sistema de CRC. Os pacotes que chegam corrompidos são simplesmente descartados, sem que o emissor sequer saiba do problema.

A ideia é justamente transmitir dados com o **maior desempenho possível**, eliminando dos pacotes quase tudo que não sejam dados em si. Apesar da pressa, o UDP tem seus méritos, afinal você não gostaria que quadros fantasmas ficassem sendo exibidos no meio de um vídeo, muito menos se isso ainda por cima causasse uma considerável perda de performance.

Neste sentido, aplicações como o SNMP podem tirar vantagem do UDP, o SNMP faz o monitoramento de rede. Nesse tipo de serviço existe o envio de mensagens intermitentes e um fluxo constante de atualizações de status e alertas, principalmente quando esta sendo utilizado em uma grande rede.

Se ele fosse utilizado numa conexão TCP no lugar de UDP, isso geraria uma sobrecarga muito grande na rede (gerada por ter que abrir e fechar uma conexão TCP para cada uma das mensagens enviada).

Portas UDP

53 Consultas de nomes DNS (Domain Name System, sistema de nomes de domínios)

69 Trivial File Transfer Protocol (TFTP)

137 Serviço de nomes de NetBIOS

138 Serviço de datagrama de NetBIOS

161 SNMP (Simple Network Management Protocol)

520 Routing Information Protocol (RIP, protocolo de informações de roteamento)

TCP X UDP

TCP	UDP
Seqüenciado	Não seqüenciado
Confiável	Não confiável
Orientado a conexão	Sem conexão
Circuito virtual	Pouca sobrecarga
Three-way handshake	Sem reconhecimento
Controle de fluxo por janelas	Sem janela ou controle de fluxo

Camada de Rede

Funções da camada de rede

Repasse Conduzir o pacote que chega pelo enlace de entrada até o enlace de saída apropriado através de uma tabela de repasse.

Roteamento Determinar a rota tomada pelos pacotes de um destinatário a um remetente através de um algoritmo de roteamento.

Estabelecimento de Conexão Troca de mensagens entre roteadores ao longo do caminho escolhido desde a fonte até o destino, com a finalidade de estabelecer o estado da conexão antes do início do envio de dados.

Tipos de Pacotes

Pacotes de Dados: São os responsáveis de transportar os dados dos usuários entre os diferentes tipos de redes (inter- rede). Por exemplo, protocolos sensíveis a roteamento tais como o IP e IPX.

Pacotes de Atualização de Rotas: São os responsáveis para atualizar a informação dos roteadores, através do anuncio e manutenção de tabelas de rotas. Entre os protocolos que fazem manutenção de tabelas de roteamento assim como da disponibilidade da mesma temos o RIP (v1 e v2), o EIGRP e o OSPF.

Estrutura Do Endereço IP

O IP tem um endereço de 32 bits, este endereço traz o ID (identificador) da Rede e o ID (identificador) do computador dentro dessa rede.

Para os endereços IP se utiliza o **sistema binário** para efetuar a representação dos números. Neste formato binário o endereço IP tem a aparência abaixo, onde os x podem ter o valor de 0 ou 1 (bits):

Byte 1	.	Byte 2	.	Byte 3	.	Byte 4
xxxxxxxx	.	xxxxxxxx	.	xxxxxxxx	.	xxxxxxxx

Sistema Binário

Neste formato não fica claro visualizar o endereço IP, mas efetuando a conversão para o formato decimal o endereço IP torna-se legível. Nesse exemplo a conversão do endereço IP resulta igual a 192.168.0.1, que resulta muito mais familiar.

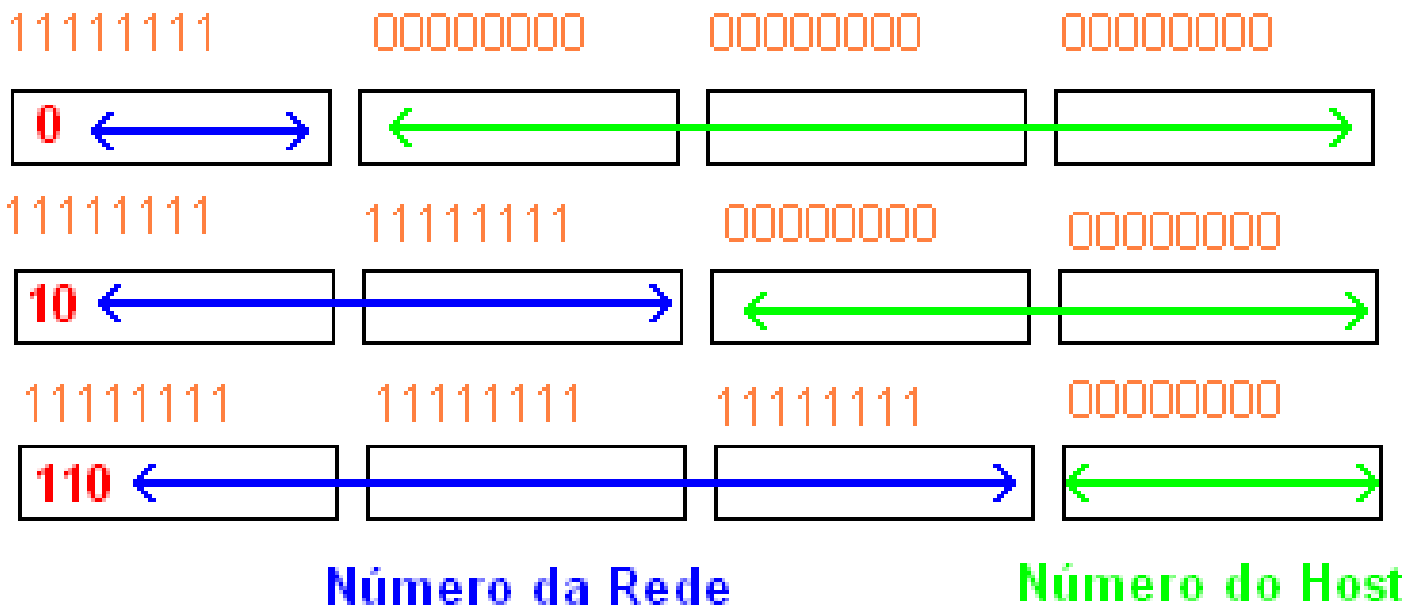
Binário	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	Resultado em Sistema Decimal
Decimal	128	64	32	16	8	4	2	1	255
00000001	0	0	0	0	0	0	0	1	1
00000011	0	0	0	0	0	0	1	1	3
01000100	0	1	0	0	0	1	0	0	68

Formato E Categorias IP Versão 4 (Ipv4)

A versão 4 do protocolo IP pode suportar 5 **classificações** para endereçar redes e computadores na Internet, a saber, essa classificação IP é dada por redes Classe A, B, C, D e E.

Normalmente na Internet são utilizados os endereços de classe A, B e C. As redes tipo Classe D e E são reservadas. O que diferencia entre um e outro tipo classe é o número de Bytes que serão utilizados para a identificação da rede e para a identificação do computador dentro dessa rede, denominado **mascara de rede**.

Mascara de Rede



Redes Classe A

Esta classe foi definida como tendo o primeiro bit do número IP como sendo igual a zero. Com isso o primeiro número IP somente poderá variar de 1 até 126.

O número 127 não é utilizado como rede Classe A, pois é um número especial, reservado para fazer referência ao próprio computador. O número 127.0.0.1 é um número especial, conhecido como *localhost* ou endereço de loopback. Ou seja, sempre que um programa fizer referência a *localhost* ou ao número 127.0.0.1, estará fazendo referência ao computador onde o programa está sendo executado.

A máscara de sub-rede padrão de uma rede Classe A, foi definida como sendo: **255.0.0.0**. Com esta máscara de sub-rede observe que temos 8 bits para o endereço da rede e 24 bits para o endereço da máquina dentro da rede. Com base no número de bits para a rede e para as máquinas, podemos determinar quantas redes Classe A podem existir e qual o número máximo de máquinas por rede.

Número De Redes Classe A

Sabe-se que o número de bits para esta classe é 7. Como o primeiro bit sempre é zero, este não varia. Por isso sobram 7 bits (8-1) para formar diferentes redes, fazendo uso da fórmula anterior temos o seguinte resultado:

2n – 2 → 2^7 – 2 = 128 – 2 = 126 redes Classe A

Redes Classe A

	0	**1**	**1**	**1**	**1**	**1**	**1**	**1**
Multiplica por:	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
equivale a:	128	64	32	16	8	4	2	1
Multiplicação:	0x128	1x64	1x32	1x16	1x8	1x4	1x2	1x1
Resulta em:	0	64	32	16	8	4	2	1
Somando tudo:	**0+64+32+16+8+4+2+1**							
Resulta em:	**127**							

Número De Máquinas Em Uma Rede Classe A

O número de bits para identificar o endereço da máquina dentro da rede é 24.

Isto é 2^24 – 2 = 16777216 – 2 = 16777214 máquinas em cada rede classe A

Como é possível observar, pelos cálculos anteriores, nas redes Classe A existe apenas um pequeno número de redes disponíveis, porém um grande número de máquinas em cada rede.

Redes Classe B

Esta classe foi definida com os dois primeiros bits do número IP sendo sempre iguais a 1 e 0.

Com isso o primeiro número do endereço IP somente poderá variar de 128 até 191.

A máscara de sub-rede padrão de uma rede Classe B, foi definida como sendo: **255.255.0.0**.

Com esta máscara de sub-rede temos 16 bits para o endereço da rede e 16 bits para o endereço da máquina dentro da rede.

Redes Classe B

	1	**0**	**1**	**1**	**1**	**1**	**1**	**1**
Multiplica por:	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
equivale a:	128	64	32	16	8	4	2	1
Multiplicação:	1x128	0x64	1x32	1x16	1x8	1x4	1x2	1x1
Resulta em:	128	0	32	16	8	4	2	1

Somando tudo: **128+0+32+16+8+4+2+1**

Resulta em: **191**

Número De Redes Classe B

O número de bits para esta classe de rede é 14. Como o primeiro e o segundo bit são sempre 10, fixos, não variam, sobram 14 bits (16-2) para formar diferentes redes:

214 – 2 = 16384 – 2 = 16382 redes Classe B

Número De Máquinas Em Uma Rede Classe B

O número de bits para identificar o endereço da máquina dentro da rede é 16.

216 – 2 = 65536 – 2 = 65534 máquinas em cada rede classe B

Redes Classe C

Esta classe foi definida com os três primeiros bits do número IP sempre iguais a 110. Com isso o primeiro número do endereço IP somente poderá variar de 192 até 223. Como o terceiro bit é sempre 0, o valor do terceiro bit que é 32 nunca é somado para o primeiro número IP, com isso o valor máximo fica em: 255 – 32 = 223

Redes Classe C

	1	**1**	**0**	**1**	**1**	**1**	**1**	**1**
Multiplica por:	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
equivale a:	128	64	32	16	8	4	2	1
Multiplicação:	1x128	1x64	0x32	1x16	1x8	1x4	1x2	1x1
Resulta em:	128	64	0	16	8	4	2	1

Somando tudo: **128+64+0+16+8+4+2+1**

Resulta em: **223**

Número De Redes Classe C

O número de bits para a este tipo de rede é 21. Como o primeiro, o segundo e o terceiro bit são sempre 110 e fixos, não variam, sobram 21 bits (24-3) para formar diferentes redes: 221 – 2 = 2.097.152 – 2 = 2.097.150 redes Classe C.

Número De Máquinas Em Uma Rede Classe C

O número de bits para identificar a máquina: 8, portanto, 28 – 2 = 256 – 2 = 254 máquinas em cada rede classe C

Observa-se que em redes Classe C temos um **grande número de redes disponíveis**, com, no máximo, 254 máquinas em cada rede. É o ideal para empresas de pequeno porte. Mesmo com a Classe C, existe um grande desperdício de endereços. Imagine uma pequena empresa com apenas 20 máquinas em rede. Usando um endereço Classe C, estariam sendo desperdiçados 234 endereços. Conforme já descrito anteriormente, esta questão do desperdício de endereços IP pode ser resolvida através da utilização de sub-redes.

Redes Classe D

Esta classe de redes foi definida com os quatro primeiros bits do número IP iguais a 1110. A classe D é uma classe especial, reservada para os chamados endereços de Multicast. Com esse valor de bits iniciais temos para esta classe os endereços desde 224.0.0.0 a 239.255.255.255.

Redes Classe E

Esta classe foi definida com os quatro primeiros bits do número IP sempre iguais a 1111.

A classe E é uma classe especial e está reservada para uso futuro e/ou funções especiais.

Por que Criar Sub-Redes IP?

Para efetuar a expansão da rede: Pode ser que seja necessário extrapolar as limitações físicas da rede e que seja necessário adicionar mais computadores e efetuar a criação de uma sub-rede com dispositivos como um roteador inclusive.

Para reduzir o congestionamento: O trafego entre computadores (nós) em uma rede utiliza parte da banda da rede. Nesse sentido, quantos mais computadores ou dispositivos a rede possuir, mais banda será consumida. A divisão de uma rede em varias redes (menores e separadas) reduz o número de computadores numa mesma rede.

Por que Criar Sub-Redes IP?

Para reduzir o uso da CPU: Mesmo que um pacote de rede não seja endereçado a um computador em especifico a placa de rede faz a analise do mesmo antes de efetuar o descarte do pacote. Então quantos mais computadores existirem em uma rede maior será o Broadcasts entre os equipamentos; Esta característica de broadcasting pode ser utilizada para, por exemplo, divulgar serviços ou para efetuar descobertas de computadores.

Facilita a resolução e isola os problemas de rede: Depois de subdividida a rede é possível identificar com maior facilidade problemas com relação a um computador em especifico.

Melhoria da Segurança de Rede: Com um analisador de protocolo de rede (ou Sniffer) e a configuração adequada.

Criando Sub-Redes

1. Determinar o número de bits de máquina a serem usados para sub-redes.

2. Listar as novas identificações de sub-redes.

3. Listar os endereços IP para cada nova identificação de sub-rede.

Considerações

A principal mudança que se nota ao trabalhar com sub-redes é uma mudança na mascara do endereço IP que ira variar conforme a quantidade de bits usada para o endereçamento de sub-rede. Inclusive é bom pensar muito bem antes de escolher o número de bits de máquina, pois se deve levar em conta o crescimento da rede (número de computadores) e a quantidade de sub-redes a serem criadas (para evitar transtornos ou o trabalho de trocar os dados IP).

Calculo de Sub-rede

Para poder criar sub-redes em uma rede, a única forma é alterar a máscara de rede padrão, isto é, devemos **emprestar** um ou mais bits 0 que correspondem ao endereçamento dos computadores dentro da rede, esses bits emprestados farão parte dos **bits de rede** (bits 1).

Por exemplo, temos o endereço de rede dado por **192.168.1.0**, este endereço corresponde a uma rede Classe C. Agora, para se ter duas sub-redes em esta rede Classe C basta emprestar um único bit do grupo de bits de máquina (bits 0) e invertê-lo para que faça parte do grupo de bits de rede (bits 1).

Calculo de Sub-rede

Lembremos que para este tipo de redes Classe C a máscara padrão é de 24 bits “1”, esses 24 bits correspondem ao endereço de rede e os 8 bits 0 correspondem para endereçar os computadores dentro dessa rede, ou seja,

11111111.11111111.11111111.00000000 = **255.255.255**.0

Agora, empresta-se dois bits dos “bits de máquina” (bits “0”) para, assim, termos 2 “bits de rede”, ou seja, 26 bits “1”, o resultado seria uma nova máscara de rede dada por:

11111111.11111111.11111111.11000000 = 255.255.255.**192**

Exemplo

Dado o IP 129.45.32.0 crie duas sub-redes utilizando dois bits do grupo de bits de máquina (bits 0), mostre os endereços de broadcasting para essas sub-redes e calcule quantos computadores cada uma delas suportará.

Reservados:

1000 0001 . 0010 1101 . 0010 0000 . 0000 0000 Subrede 00 = Endereço de Rede

1000 0001 . 0010 1101 . 0010 0000 . 1100 0000 Subrede 11 = BroadCast

Validos: 6^2 – 2 maquinas em cada rede

1º Subrede 1000 0001 . 0010 1101 . 0010 0000 . 0100 0000 = 129.45.32.64 **BroadCast =** 129.45.32.127

2º Subrede 1000 0001 . 0010 1101 . 0010 0000 . 1000 0000 = 129.45.32.128 **BroadCast =** 129.45.32.191

Exemplo 2

Deseja-se dividir a seguinte rede classe C: 129.45.32.0/255.255.255.0. As especificações são as seguintes: ter como mínimo 10 sub-redes.

Nestas condições, pede-se determinar o seguinte:

- Quantos bits serão necessários para fazer a divisão e obter pelo menos 10 sub-redes?
- Quantos números IP (para as máquinas) estarão disponíveis em cada sub-rede?
- Qual a nova máscara de sub-rede?
- Listar a faixa de endereços de cada sub-rede.

Resposta do Exemplo 2

Sub-rede 01: 129.45.32.0 →129.45.32.15

Sub-rede 02: 129.45.32.16 → 129.45.32.31

Sub-rede 03: 129.45.32.32 → 129.45.32.47

Sub-rede 04: 129.45.32.48 → 129.45.32.63

Sub-rede 05: 129.45.32.64 → 129.45.32.79

Sub-rede 06: 129.45.32.80 → 129.45.32.95

Sub-rede 07: 129.45.32.96 → 129.45.32.111

Sub-rede 08: 129.45.32.112 → 129.45.32.127

Sub-rede 09: 129.45.32.128 → 129.45.32.143

Sub-rede 10: 129.45.32.144 → 129.45.32.159

Sub-rede 11: 129.45.32.160 → 129.45.32.175

Sub-rede 12: 129.45.32.176 → 129.45.32.191

Sub-rede 13: 129.45.32.192 → 129.45.32.207

Sub-rede 14: 129.45.32.208 → 129.45.32.223

Sub-rede 15: 129.45.32.224 → 129.45.32.239

Sub-rede 16: 129.45.32.240 → 129.45.32.255

IPv6 – Sua importancia

O IPv4, é um número de 32 bits (4 Bytes) que equivale a nada menos do que 232 = 4.294.967.296 combinações. Destes, pouco mais de 3.7 bilhões de endereços são aproveitáveis, já que os endereços iniciados com 0, 10, 127 e de 224 em diante são reservados.

Além disso, a maior parte das faixas de endereços de classe A, que englobam as faixas iniciadas com de 1 a 126 são propriedade de grandes empresas, que acabam utilizam apenas uma pequena faixa deles. Por exemplo, apenas a HP, sozinha, tem direito a duas faixas inteiras, uma ganha durante a distribuição inicial das faixas de endereços IP classe A e a segunda herdada com a compra da DEC.

No início de 2007, já restavam apenas 1.3 bilhões de endereços disponíveis. Se a procura se mantiver nos níveis atuais, teremos o esgotamento dos endereços disponíveis em 2014. Caso ela cresça, impulsionada pela popularização das conexões 3G, uso do ADSL em países desenvolvidos, aumento do número de servidores Web, popularização do ADSL nos países mais pobres e assim por diante, podemos chegar a uma situação caótica ainda em 2013!

Uma solução provisória

Um dos fatores que vem reduzindo a pressão sobre os escassos endereços disponíveis é o uso do NAT.

Graças a ele, você pode compartilhar uma única conexão (e, conseqüentemente, um único endereço), entre vários micros. É possível até mesmo adicionar um segundo, terceiro, quarto, ou mesmo quinto nível de compartilhamento, recompartilhando uma conexão já compartilhada.

É muito comum, por exemplo, que um provedor de acesso via rádio use um único IP para um prédio inteiro, dando endereços de rede interna para os assinantes. Muitos destes criam redes domésticas e compartilham novamente a conexão, adicionando uma segunda camada de NAT, e assim vai.

Apesar disso, o NAT não é a solução para tudo. Você não pode usar NAT em um Datacenter, por exemplo, precisa de um endereço "real e válido" para cada servidor disponível para o mundo exterior.

Endereçamento IPv6

O IPv6 é a nova versão do Protocolo de Internet, a qual deverá substituir progressivamente o protocolo atual da Internet, o IPv4, estendendo o espaço de endereçamento corrente para (nada mais nem nada menos do que) 128 bits. O numero 340.282.366.920 seguido por mais 27 casas decimais!

IPv6 é igual ao IPv4 ? "232.234.12.43.45.65.132.54.45.43.232.121.45.154.34.78", algo impraticável.

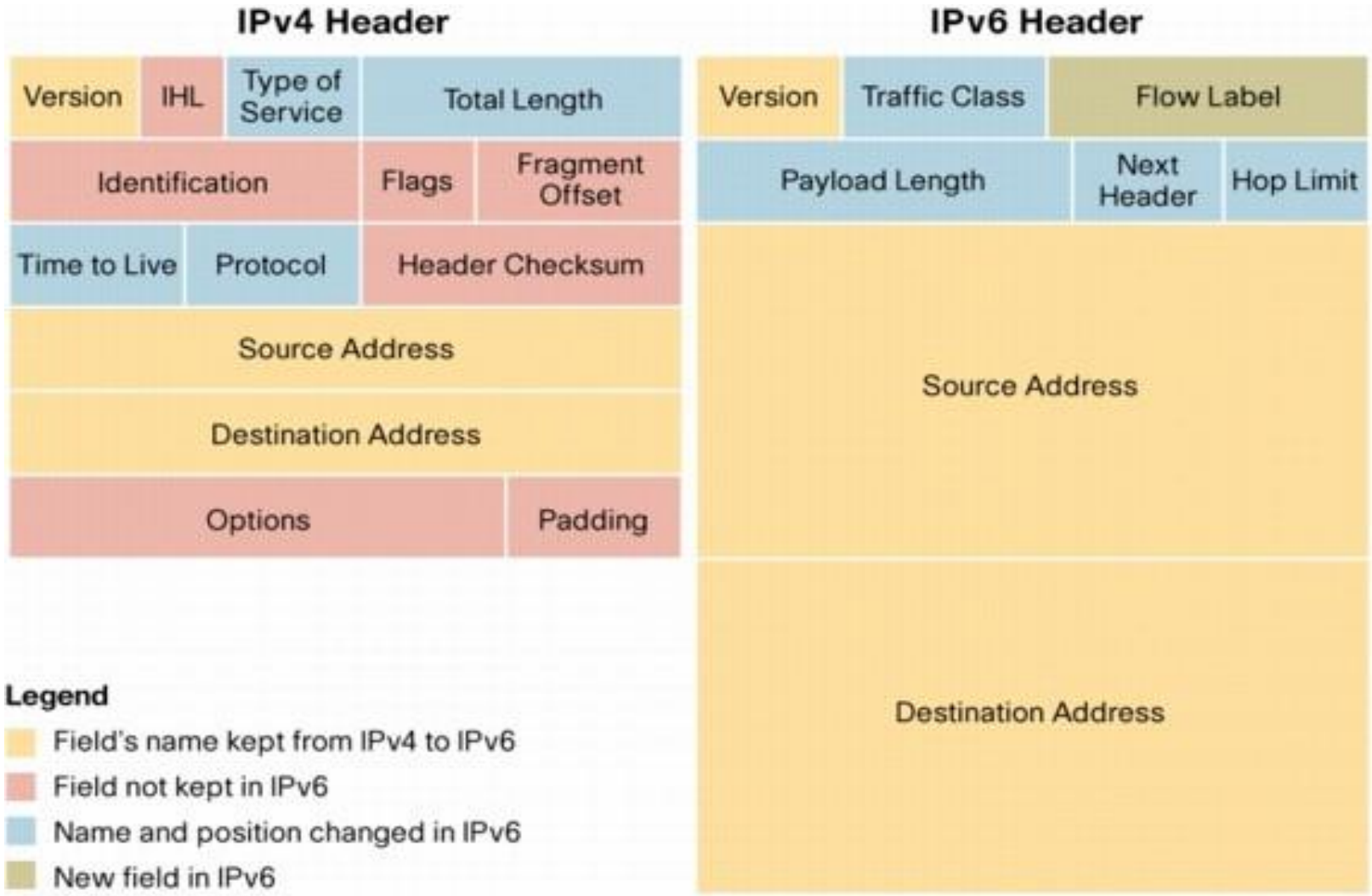
Os endereços IPv6 utilizam 8 quartetos de caracteres em Hexadecimal separados por "dois pontos" (:).

Exemplo: 2001:BCE4:5641:3412:341:45AE:FE32:65.

Forma compacta "::1" ou "FEC::1" e também "0341" == "341"

"2001:BCE4:0:0:0:0:0:1" == "2001:BCE4::1"

Pacote IPv4 x IPv6

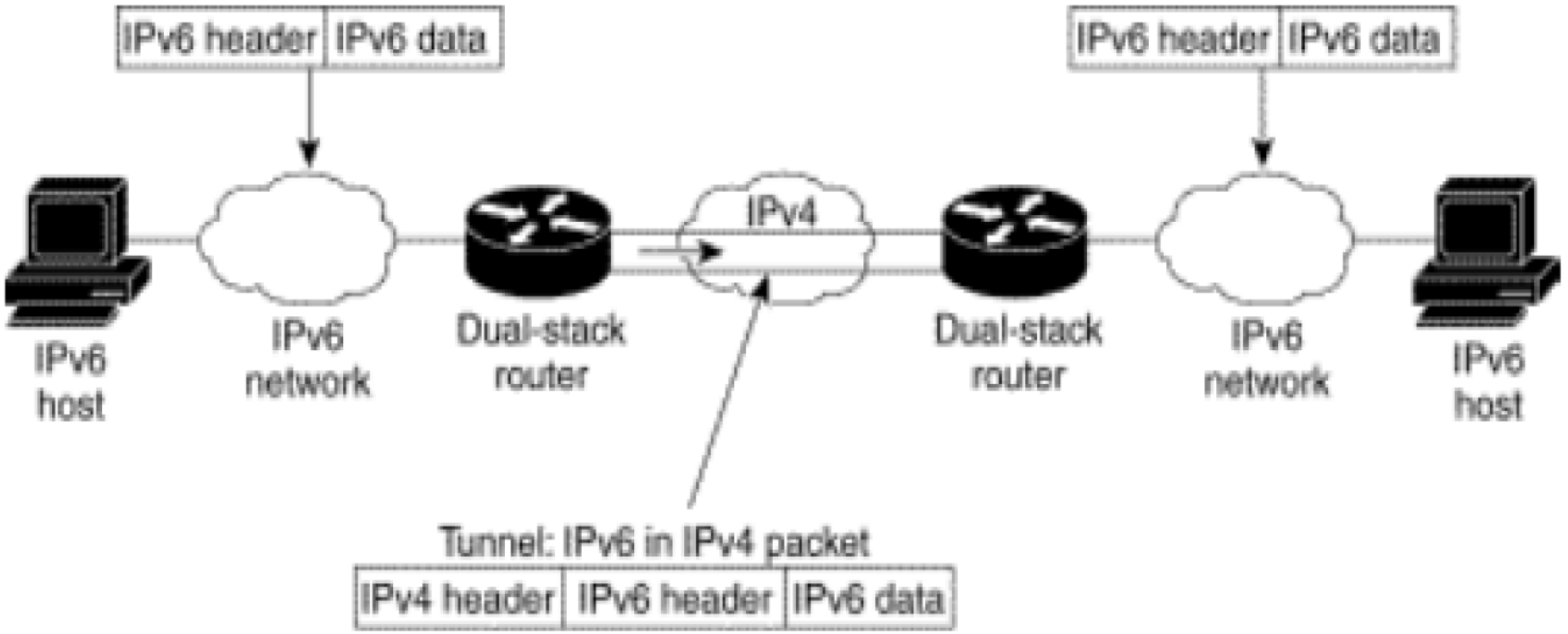


Migração

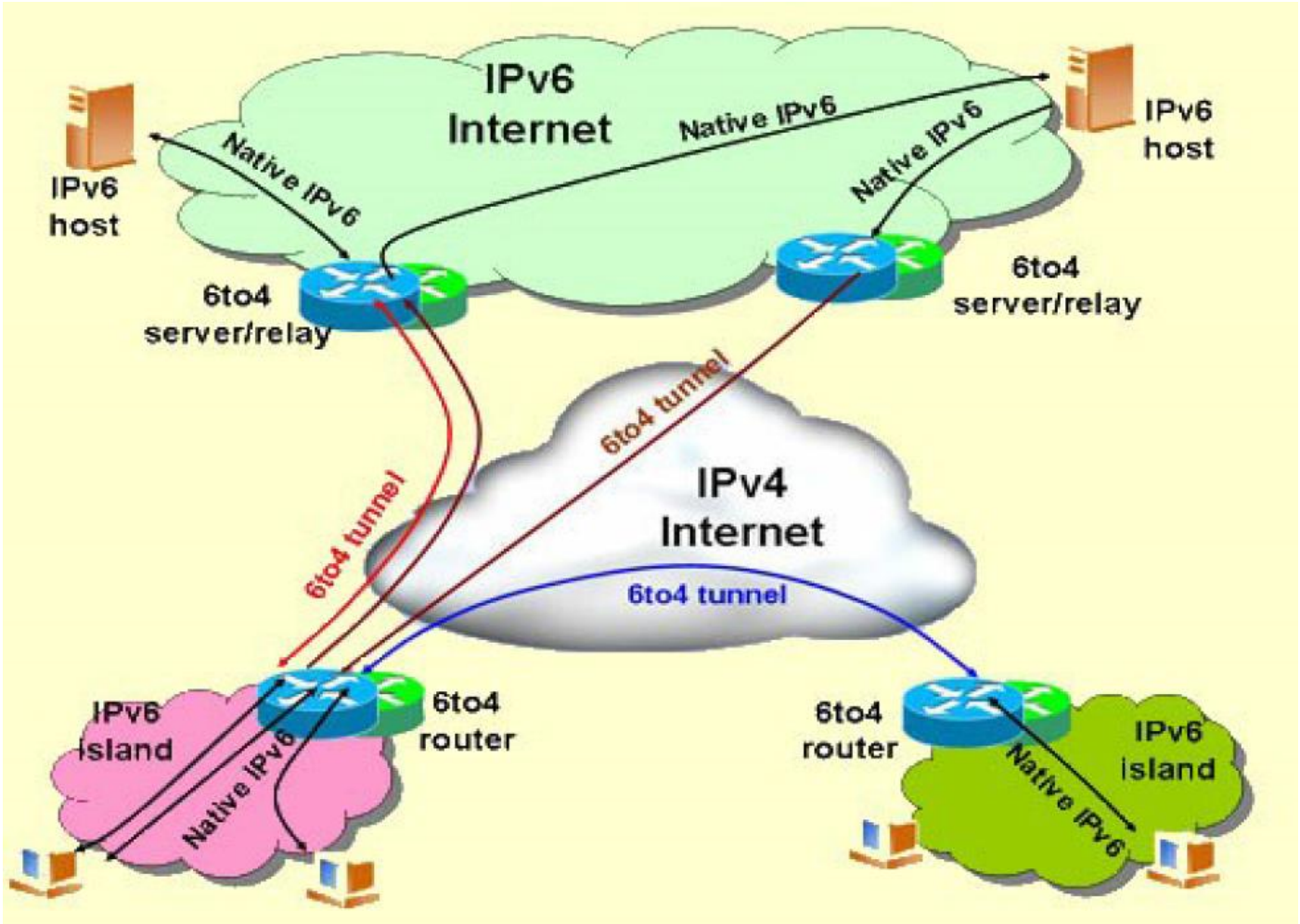
A terceira opção é utilizada na representação de endereçamento compatível IPv6-IPv4, sendo útil no período de migração e de coexistência de ambos os protocolos. Assim utilizamos a representação X:X:X:X:X:X:Z.Z.Z.Z, onde os "X" indicam números hexadecimais (16 bits) e os "Z" são valores que representam os 8 bits referentes ao endereço IPv6 :

0:0:0:0:0:0:192.168.1.1 (**IPv6**) = :192.168.1.1 (**IPv4**)

Túnel


IPv6 header
IPv6 data
IPv6 header
IPv6 data
IPv4
IPv6 host
IPv6 network
Dual-stack router
Dual-stack router
IPv6 network
IPv6 host
Tunnel: IPv6 in IPv4 packet
IPv4 header
IPv6 header
IPv6 data

Situação IPv4 + IPv6



DHCP- *Dynamic Host Configuration Protocol*

O DHCP fornece aos protocolos TCP/IP, as informações iniciais de configuração da máquina tais como endereço IP, máscara de sub-rede, roteadores default, rotas, servidores de Boot, servidores de nome e diversas outras informações. Este protocolo pode ser utilizado pra efetuar a administração centralizada de máquinas TCP/IP.

O BOOTP (Bootstrap Protocol) é o protocolo mais antigo e o DHCP (Dynamic Host Control Protocol) está aos poucos o substituindo.

O DHCP é mais complexo e mais versátil e vem sendo usado para simplificar a administração de endereços e outros parâmetros de configuração de grandes instalações de máquinas TCP/IP. O DHCP consegue efetuar a configuração automática de estações, sem necessidade de criação de uma tabela de configuração para cada máquina (que é necessária no caso do BOOTP).

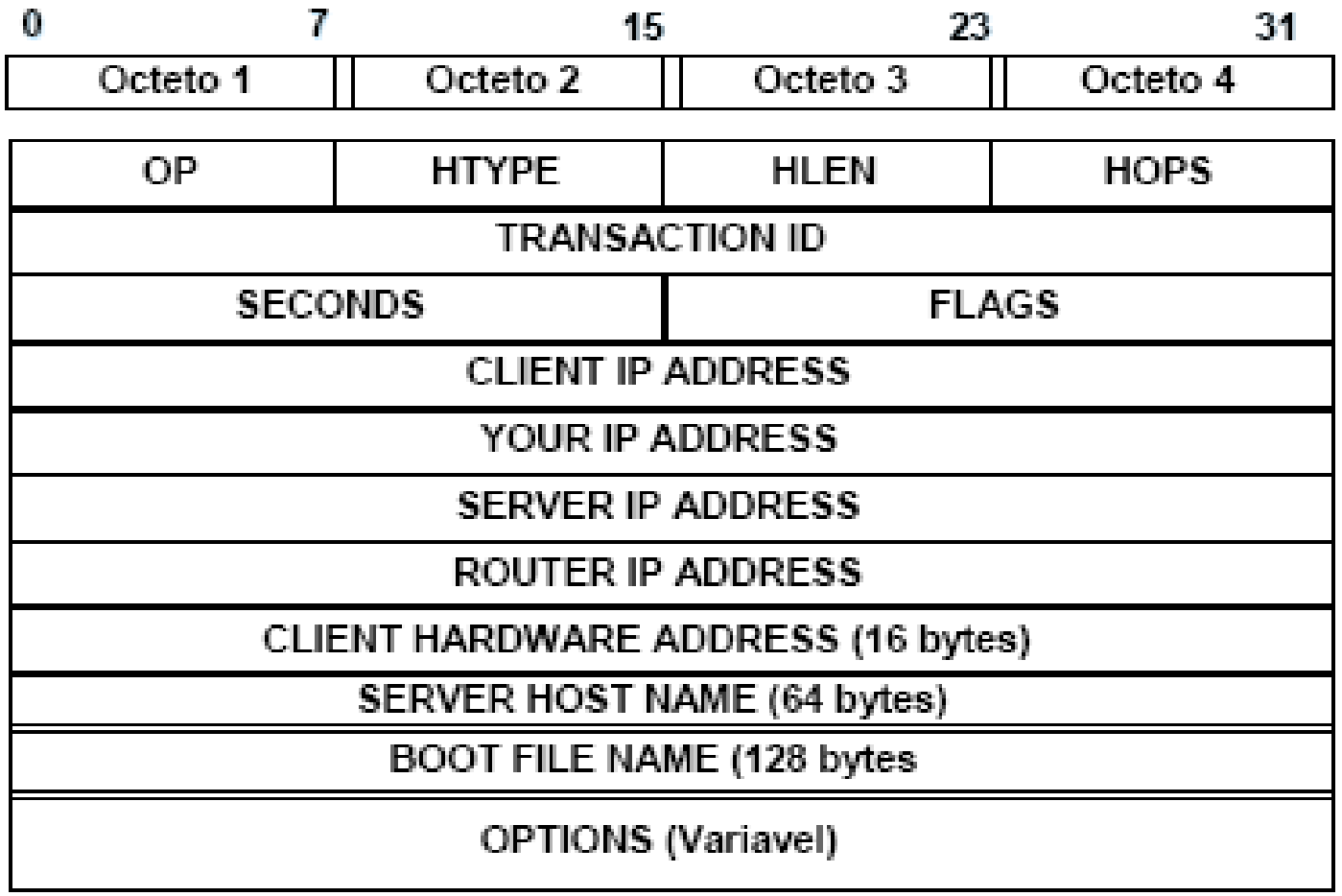
DHCP - métodos de fornecimento

Empréstimo (leasing) de endereço aleatório por tempo limitado: Se escolhe um endereço IP de um range e se fornece ao cliente por um tempo pré-determinado.

Empréstimo de endereço aleatório por tempo infinito: O servidor atribui um endereço do range ao cliente na primeira vez que este cliente contatar o servidor. Nas demais vezes se consultam o MAC do cliente e se fornece o mesmo endereço a este cliente, mesmo que as duas máquinas sejam desligadas e ligadas. Este método simplifica a atribuição de endereços para uma quantidade grande de máquinas.

Empréstimo de endereço fixo: Nesse tipo de fornecimento existe a associação explícita entre o endereço IP e o endereço MAC da máquina origem, estipulado em uma tabela de configuração.

Pacote DHCP


0
7
15
23
31
Octeto 1
Octeto 2
Octeto 3
Octeto 4
OP
HTYPE
HLEN
HOPS
TRANSACTION ID
SECONDS
FLAGS
CLIENT IP ADDRESS
YOUR IP ADDRESS
SERVER IP ADDRESS
ROUTER IP ADDRESS
CLIENT HARDWARE ADDRESS (16 bytes)
SERVER HOST NAME (64 bytes)
BOOT FILE NAME (128 bytes
OPTIONS (Variavel)

Comandos DHCP

DHCP DISCOVER – Uma solicitação de resposta enviada a um servidor pelo cliente

DHCP OFFER – Uma oferta de IP do servidor para o cliente. O cliente pode receber várias ofertas de diferentes servidores DHCP.

DHCP REQUEST – É uma requisição de um endereço específico do servidor. Um broadcast é gerado apesar de ser endereçado a um único servidor para que os demais saibam da escolha.

DHCP DECLINE – Comunica que a oferta contém parâmetros incorretos (Erro).

DHCP ACK – Mensagem de OK do servidor sobre a atribuição do endereço para a requisição do cliente.

DHCP NAK - Servidor rejeita o fornecimento do endereço previamente oferecido, isso ocorre por erro ou por demora do cliente a requisitar o endereço solicitado.

DHCP RELEASE - Cliente libera o endereço IP utilizado. Difícil de se ver na prática, pois geralmente o cliente é desligado sem liberar o endereço. Esse endereço volta ao conjunto de endereços disponíveis no servidor devido ao estouro do tempo de leasing.

DHCP INFORM - Cliente que já possui endereço IP pode requisitar outras informações de configuração respectivas àquele endereço.

Funcionamento DHCP

O cliente utiliza o protocolo User Datagram Protocol pacote (**UDP**), com o destino de difusão de 255.255.255.255 ou o endereço de broadcast de sub-rede específica. Um cliente DHCP também pode solicitar o seu último endereço IP conhecido (exemplo, 192.168.1.100).

Se o cliente permanece conectado a uma rede IP para o qual este é válido, o servidor pode satisfazer o pedido. Caso contrário, ele depende se o servidor está configurado como autoridade ou não. Um servidor com autoridade irá negar o pedido, fazendo com que o cliente pedir um novo endereço IP imediatamente. Um servidor não-autorizada simplesmente ignora o pedido, levando a um limite de tempo dependente da implementação para o cliente a desistir do pedido e pedir um novo endereço IP

NAT - *Network Address Translator*

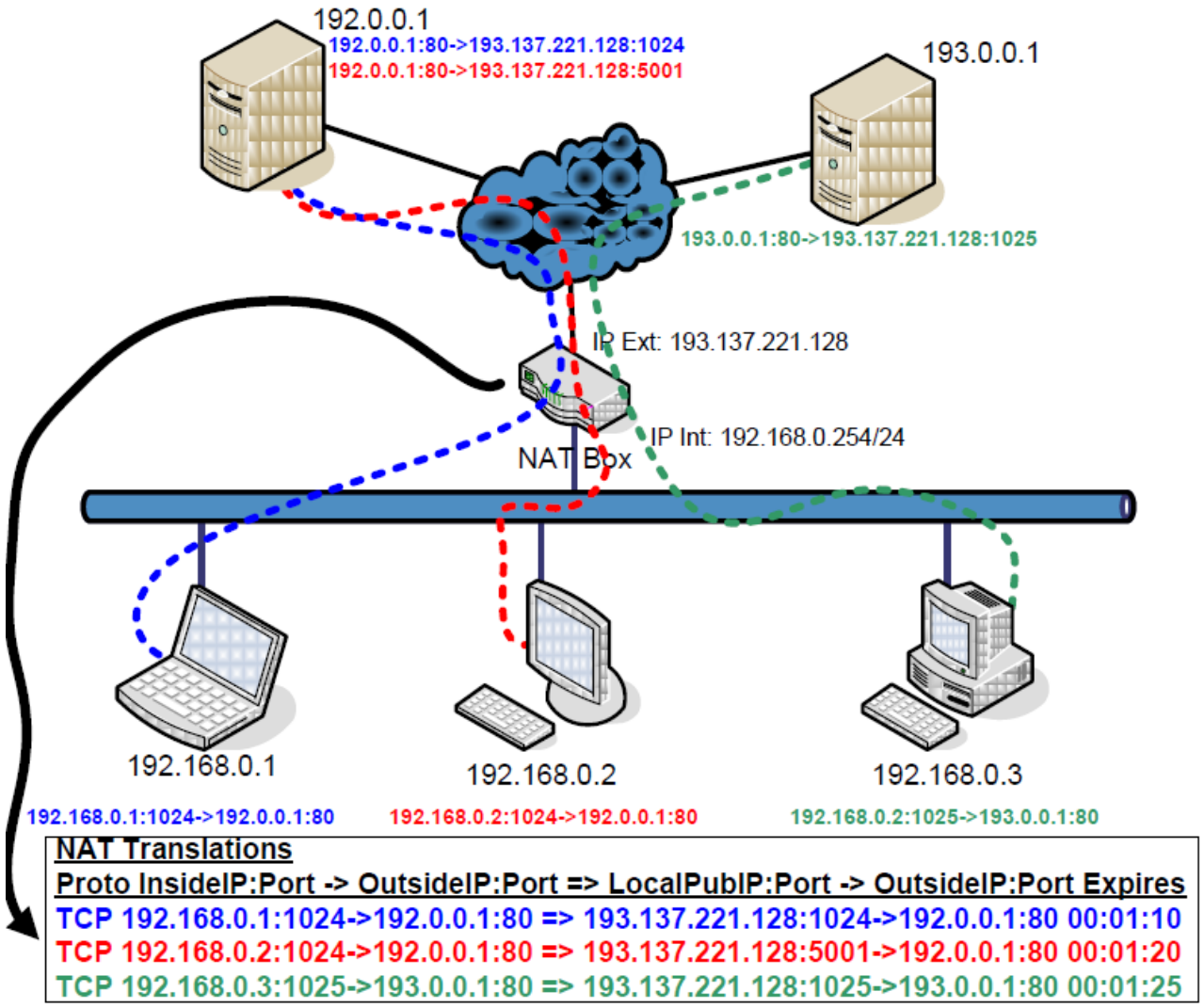
Tradução de Endereços na Rede (*Network Address Translator* – NAT), RFC 2663 e RFC 3022;

Utilizada para possibilitar uma rede local com endereços privados acessar a internet através de um servidor NAT com um **único** endereço válido, porém cada uma com o seu endereço individual interno.

Utiliza a tabela de tradução NAT para direcionar respostas a requisições locais;

Motivação principal: Escassez de endereços devido ao desperdício de endereços na atribuição clássica por classes a cada entidade e do consumo desproporcional praticado por muitas entidades.

Funcionamento do NAT



Porquê usar endereços privados atrás de NAT?

Se se usar uns quaisquer, pode acontecer ter-se necessidade de comunicar com a rede que os usa legalmente e não se consegue, porque todas as máquinas da nossa rede assumem que conseguem falar diretamente com as máquinas da "sua" rede.

Usar os blocos de endereços *Private Networks*(RFC1918) para endereçar as redes internas

- 10.0.0.0/8 –1 Classe A
- 172.16.0.0/12 –16 Classes B
- 192.168.0.0/16 –256 Classes C

ICMP- *Internet Control Message Protocol*

Este protocolo faz parte da pilha de protocolos **TCP/IP**, enquadrando-se na camada de rede (nível 3), a mesma camada do protocolo IP. O seu uso mais comum é feito pelos utilitários ping e traceroute.

O **ping** evia pacotes ICMP para verificar se um determinado computador está disponível na rede.

O **traceroute** faz uso do envio de diversos pacotes UDP ou ICMP e, através de um pequeno truque, determina a rota seguida para alcançar um determinado computador.

EMAIL:

jesse.filho@bonfim.ifbaiano.edu.br

MATERIAL

http://softwarelivre.org/jessenery/jesse-nery-filho