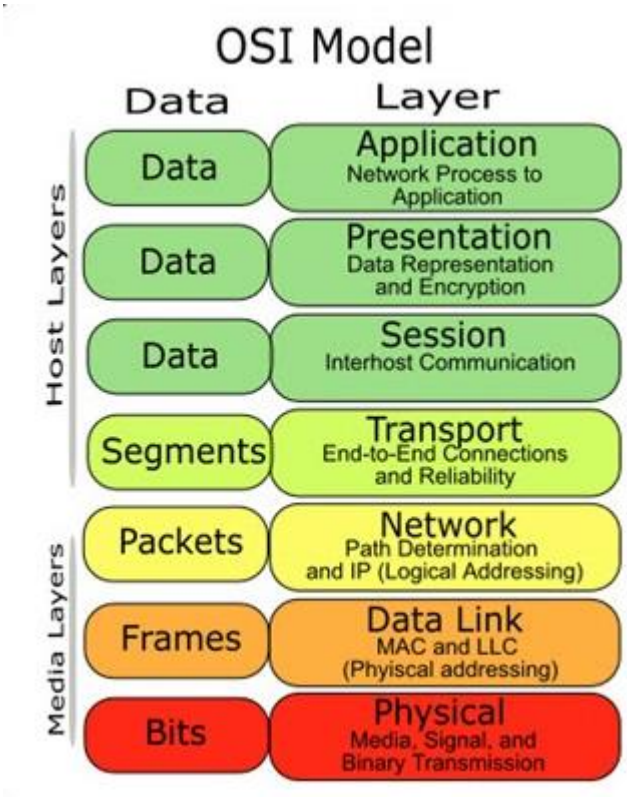
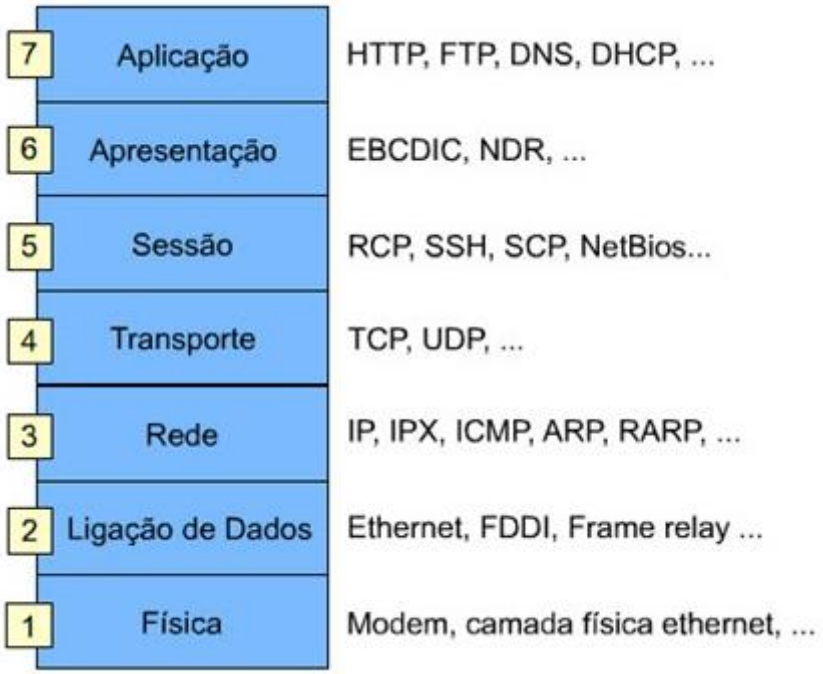
Resumo IRC – 2019/2020

Modelo OSI



- **Aplicação (Application):**
  - Fornece serviços às aplicações do utilizador.
- **Apresentação (Presentation):**
  - Encriptação e compressão de dados.
  - Assegura a compatibilidade entre camadas de aplicação de sistemas diferentes
- **Sessão (Session):**
  - Controla (estabelece, faz a gestão e termina), as sessões entre aplicações.
- **Transporte (Transport):**
  - Controlo de fluxo de informação, segmentação e controlo de erros
- **Rede (Network):**
  - Encaminhamento (routing) de pacotes
  - Esquema de endereçamento lógico
- **Dados (Data Link):**
  - Controla o acesso ao meio físico de transmissão.
  - Controlo de erros da camada física
- **Física (Physical):**
  - Define as características do meio físico de transmissão da rede, conectores, interfaces, codificação ou modulação de sinais.

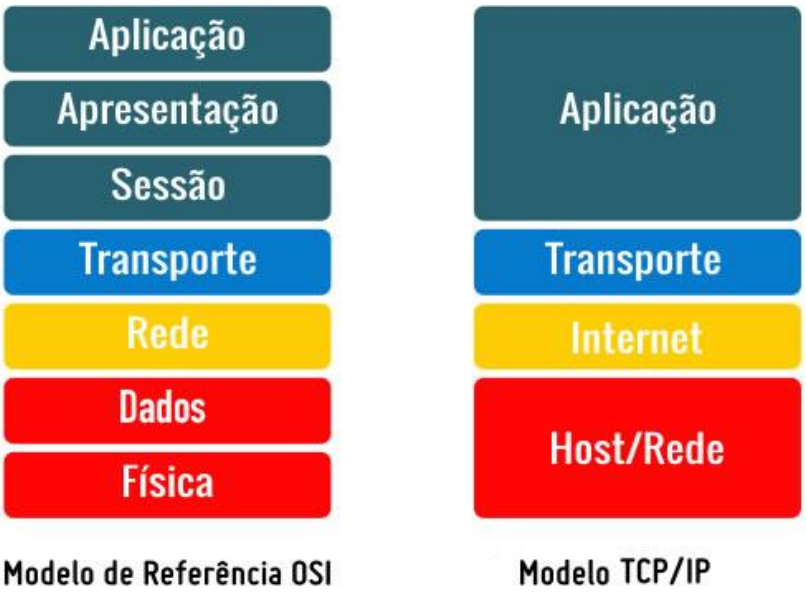
Modelo OSI – Protocolos



- Na **camada de aplicação**, o browser (aplicação) serve de interface para apresentação da informação ao utilizador. Para este pedido (cliente-> servidor), foi usado o protocolo HTTP
- O formato dos dados é tratado na **camada de apresentação**. Os formatos tradicionais da Web incluem HTML, XML, PHP, GIF, JPG, etc. Adicionalmente são usados mecanismos de encriptação e compressão para a apresentação da informação.
- Na **camada de sessão** é estabelecida a sessão entre o computador cliente (onde estamos a fazer pedido via browser) e o servidor web (que aloja a página requisitada).
- O protocolo TCP fornece garantia na entrega de todos os pacotes entre um PC emissor e um PC recetor (neste caso, a entrega de toda a informação da página web do servidor para o cliente). Isso é uma funcionalidade da **camada de transporte**.
- Tanto o PC cliente como servidor possuem um endereço lógico (endereço IP neste caso). Isso é uma funcionalidade **da camada de rede**. Adicionalmente os routers determinam qual o melhor caminho para que os pacotes possam fluir (encaminhamento) entre cliente e servidor web.
- O endereço IP (endereço lógico) é então "traduzido" para o endereço físico (endereço MAC da placa de rede. Isto é funcionalidade da **camada de dados.**

- Cabos de cobre, fibra ótica, placas de rede, hubs e outros dispositivos, ajudam na ligação física entre o cliente e o servidor que acontece na **camada física**.
    - **HUB** funciona a nível da camada 1 (camada física);
    - **Switch** na camada 2 (camada de dados).
        - Há **switchs** capazes de funcionar também na camada 3;
    - **Router** na camada 3 (camada de rede).

TCP/IP



Meios de Transmissão

10Base-2	10 Mbps (half-duplex)	Ethernet + Cabo Coaxial Fino	BUS
10Base-5	10 Mbps (half-duplex)	Ethernet + Cabo Coaxial Grosso	BUS
10Base-T	10 Mbps (half-duplex ou full-duplex)	Ethernet + Cabo Pares Entraçados (UTP)	Ponto-a-ponto
100Base-FX	100 Mbps (full-duplex)	Ethernet + Fibra Ótica	Ponto-a-ponto (anel e estrela)

- **Bus (nível 1):**
  - Em cada instante apenas pode transmitir um único dispositivo.
- **Hub (nível 1):**
  - Qualquer sinal recebido numa porta é reenviado pelas restantes;
  - *Multiport-repeater*;
  - Opera apenas no meio físico, não interpretando os quadros.
- **Bridge (nível 2):**
  - Apenas reencaminha quadros sem erros;
  - Com base nos endereços de origem, a ponte vai aprendendo, de um modo dinâmico, de que lado se encontram as máquinas;
  - Quando o destino faz parte da tabela, os quadros apenas são reencaminhados se necessário;
  - Expansão de domínios de difusão;
  - Segmentação de domínios de colisão.
    - Qualquer sinal que resulte de uma colisão não é reencaminhado pela ponte.
- **Switch (nível 2):**
  - Multiport bridge;
  - O aumento do débito efetivo, deve-se:
    - Com full-duplex;
    - Ausenciia de colisões;
  - Modos de operação:
    - *Store and Forward*
    - *Cut-Through*
    - *Fragment-Free*
    - *Algoritmo Spanning-Tree*
- **Router (nível 3)**

Domínios de Colisão

- Hubs e repetidores expandem os domínios de colisão;
- Segmentação:
    - Bridges;
    - Switches;
    - Routers.

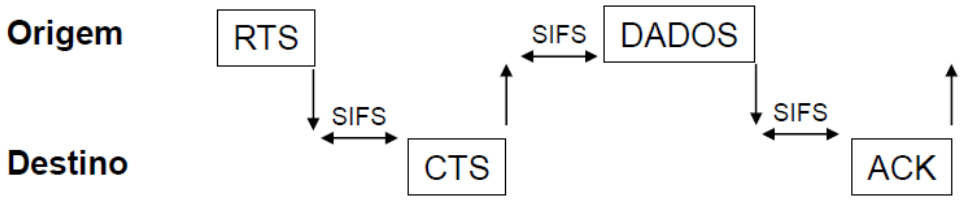
Domínios de Difusão

- Segmentação:
    - Router (nível 3);
- Expansão:
    - Hubs;
    - Bridges;
    - Switches.

WLAN

- **CSMA/CA (*CSMA with Collision Avoidance*):**
    - **Algoritmo:**
        - Gera um período de espera aleatório;
        - Inicia a contagem decrescente enquanto escuta o meio;
        - Suspende a contagem decrescente sempre que o meio estiver ocupado ou reservado;
        - Quando o contador atinge zero, inicia a transmissão do quadro.
    - A receção de quadros NÃO é garantida;
    - Como uma interface de rede ou se encontra em modo de transmissão ou em modo de receção, a aplicação do CSMA/CD não é viável.
    - **RTS/CTS (*Request to Send/Clear to Send*):**
        - Usado opcionalmente para garantir a receção dos quadros de dados;
        - Ajuda a minorar o problema do nodo escondido, i.e, duas estações ouvem um mesmo AP, mas não uma à outra;

- É usado o CSMA/CA ou outro mecanismo de controlo de acesso ao meio para enviar os quadros RTS;
- Os restantes quadros são transmitidos diretamente;
- Possibilita a reserva do meio de transmissão pelo período necessário à conclusão da transmissão;
- Os quadros RTS, CTS e ACK incluem um campo que indica a duração estimada até à conclusão do processo de transmissão (microssegundos).



SIFS (*Short Inter-Frame Space*): prevê o tempo necessário à comutação entre os modos de recepção e de transmissão.

LAN

- **Token:**
  - Aplicável a topologias em anel, bus e estrela;
  - Permite acessos ordenados;
  - Permite a realização de mecanismos de gestão de prioridades e de suporte a requisitos de qualidade de serviço (QoS);
  - Alguma complexidade devido à necessidade de funções de monitorização e de gestão.
  - **Modo de operação elementar:**
    - Existe um quadro de controlo, o testemunho, que vai passando pelos dispositivos num anel físico (topologia em anel) ou num anel lógico (topologia em bus ou em estrela);

        - O dispositivo que recebe o testemunho pode passá-lo ao seguinte no anel ou iniciar a transmissão de um quadro de dados;
        - O quadro de dados transmitido vai passando pelos dispositivos do anel, podendo estes ficar com uma cópia do respetivo conteúdo e alterar eventuais bits de resposta ou de reserva de prioridade, até regressar ao emissor;
        - O emissor do quadro de dados passa o testemunho ao dispositivo seguinte no anel (físico ou lógico).
- **CSMA/CD:**
    - *Carrier Sense Multiple Access with Collision Detection;*
    - Aplicável em topologias em bus e em estrela;
    - Acessos assíncronos com possibilidade de ocorrência de colisões;
    - Maior simplicidade devido à ausência de funções de gestão e ao seu carácter distribuído;
    - Torna difícil a realização de mecanismos para gestão de prioridades e suporte de requisitos de QoS;
    - Usado em redes do tipo Ethernet, a tecnologia de redes locais mais comum na atualidade.

PROTOCOLO IP

- Responsável pelo endereçamento e encaminhamento;
- Não orientado a ligações;
- Datagramas;
- Recorre a vários protocolos auxiliares.

CABEÇALHO IP

- **Version** – 4 bits – IPV4
- **IHL** = Internet Header Length – 4 bits
    - Tamanho do cabeçalho em words
        - (1word = 4 bytes) : Min 5 – max 15
- **Type of Service** – 8 bits
- **Total Length** – 16 bits
    - Tamanho total do cabeçalho + dados em bytes

        - Tam. Máx. 64Kb (64535 bytes)
- **Identification** – 16 bits
    - Usado para identificação de datagramas fragmentados
- **Flags** – 3 bits
    - Para controlar ou identificar fragmentos
        - Bit0 : reservado, tem que ser = 0;
        - Bit1 : *Don't Fragment* (DF)
        - Bit2 : Mane Fragments (MF)
- **Fragment offset** – 13 bits
- **Time to live (TTL)** – 8 bits – hop count
    - quando o *datagram* chega a um router, este diminui o TTL em 1
    - quando o TTL chega a zero, o router desconecta do pacote e envia um ICMP *Time Exceeded Message* ao remetente
- **Protocol** – 8 bits
    - o protocolo é usado em data fild (TCP, UDP, ICMP, etc..)
- **Header Checksum** – 16 bits
    - Usado para error-checking do header
- **Source IP address** – 32 bits
- **Destination IP address** - 32 bits
- **Options** (if IHL > 5)

FRAGMENTAÇÃO

- Se o tamanho do pacote for maior que o MTU (*maximum transmission unit*) e o bit DF for definido como zero, então o router pode fragmentar o pacote;
- Tamanho máximo de cada fragmento = MTU – IP Header Size;
- O comprimento total do ficheiro é o tamanho do fragmento;
- O MF flag é definido para todos os fragmentos, exceto o último que é definido como zero;
- O campo de deslocamento do fragmento é definido com base no deslocamento do fragmento na carga de dados original; é medido em unidades de blocos de 8 bytes;
- O campo checkum do cabeçalho é recalculado.

ENDEREÇAMENTO IPV4

- Classe A → 0…netID + hostID (24 bits) : 1 – 126
- Classe B → 10… netID + hostID (16bits) : 126 – 191
- Classe C → 110… netID + hostID (8bits) : 192 – 223
- Classe D → 1110… netID + hostID (identificador de grupo multicast) : 224 – 239
- Classe E → 11110… reservado : 240 – 255

- Identificação / endereço de uma rede : todos os bits do hostID = 0
- Endereço de difusão : todos os bits do hostID = 1
- Nº de endereços úteis : 2^(nº de bits do hostID) – 2

- **ENDEREÇOS PRIVADOS**
  - Classe A : 10.0.0.0 – 10.255.255.255 ---------- 10.0.0.0/8
  - Classe B : 172.16.0.0 – 172.31.255.255 ------- 172.16.0.0/12
  - Classe C : 192.168.0.0 – 192.168.255.255 ---- 192.168.0.0/16

- **IP MULTICAST**
  - Classe D:
    - Correspondência direta entre endereços IP da classe D e endereços físicos – endereços Ethernet reservados para multicast:
      - 01:00:5e:00:00:00 – 01:00:5e:7f:ff:ff

- **MÁSCARAS**
  - Por omissão:
    - Classe A : 255.0.0.0
    - Classe B : 255.255.0.0
    - Classe C : 255.255.255.0
  - O resultado da aplicação de uma máscara é o mesmo para todos os endereços de uma mesma (sub)rede :
    - O endereço da (sub)rede → mantén os bits da netID e coloca a zero os bits de hostID.

ARP – Address Resolution Protocol

- Permite, dado um endereço IP unicast local, obter o endereço físico correspondente;
- Recorre aos mecanismos de difusão da camada de ligação;
- Cache / ARP-table → tabela onde são armazenados pares <endereço IP; endereços físicos > relativos aos troços de rede locais
- Send data to a device (destino ou router)
    - *Is the MAC address in my ARP cache?*
        - ***YES***
            - Send data
        - ***NO***
            - Send na ARP request (por difusão)
            - Get na ARP reply
            - Send data
- **Mensagem ARP:**
    - Hardware type;
    - Protocol tpe (IPV4);
    - HLen (hardware address length);
    - Plen (Protocal address length);
    - Operation (request, reply);
    - Sender HA (hardware address);
    - Sender PA (Protocol address);
    - Target HA;
    - Target PA;
    - RARP header struture;

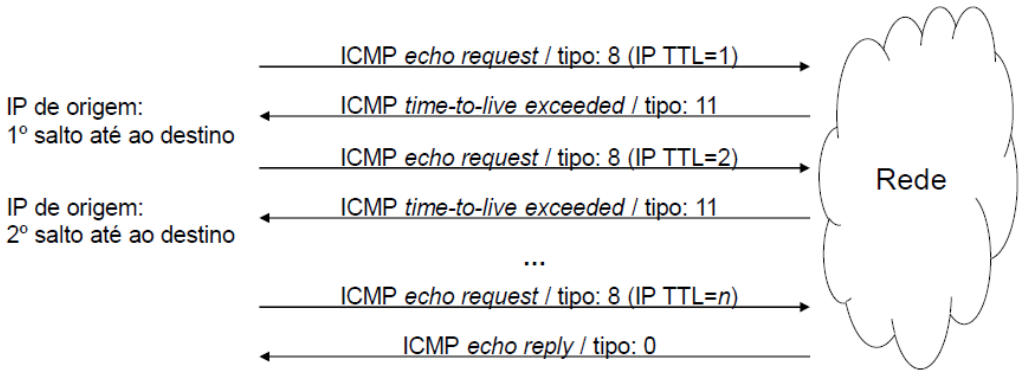
ICMP – Internet Control Message Protocol

- Dados transportador por datagramas IP

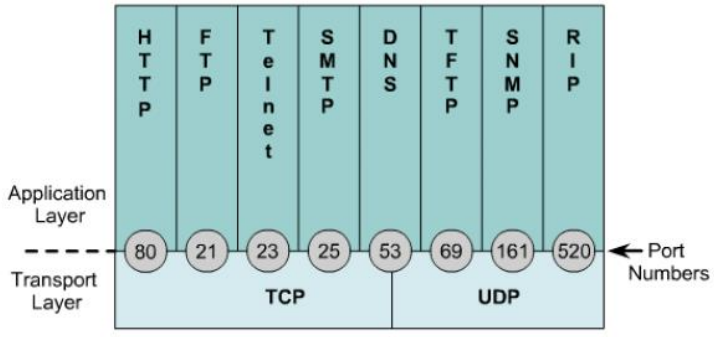
Cabeçalho IP	Tipo	Código	Checksum	Conteúdo (depende do tipo e código)	
20	1	1	2		Bytes

Tipo	Código	Descrição
0	0	echo reply
8	0	echo request
11	0	TTL expired in transit (time exceeded)
11	1	Fragment reassembly (time exceeded)

- **ping:**
  - envia messagens ICMP (tipo 8) e espera por uma resposta (tipo 0);
- **trace router:**



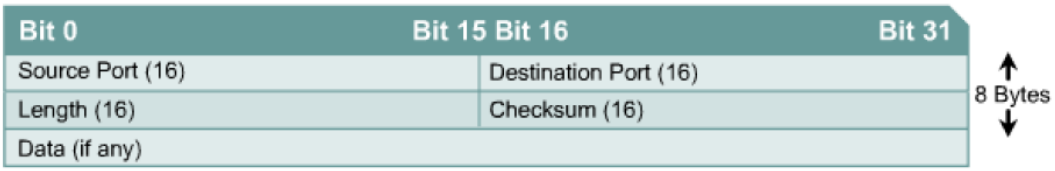
- **ports:**
  - a localização de um processo na Internet é dada por 3 coordenadas:
    - Endereço IP;
    - Protocolo de transporte;
    - Porto;



PROTOCOLO TCP/UDP

UDP – User Datagram Protocol

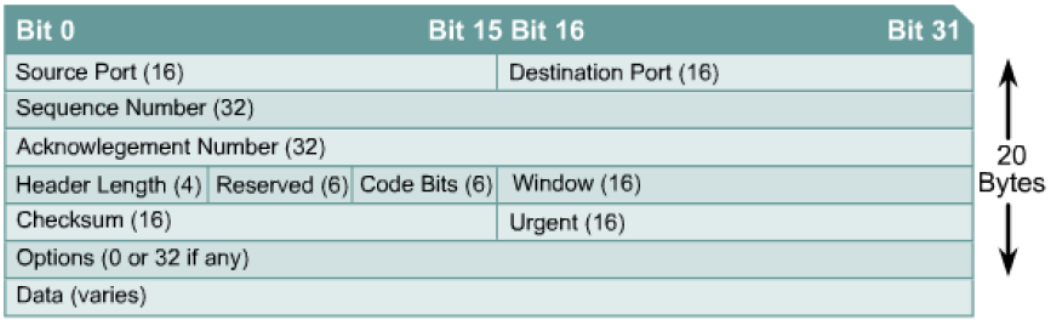
- do tipo não orientada a ligação;
- não fiável (serviço do tipo *best effort*);
- preferível ao TCP quando não é exigida uma fiabilidade de 100%, devido à sua reduzida sobrecarga protocolar;
- **Exemplo:**
  - TFTP e DNS recorrem ao UDP e realizam os seus próprios controlos de erros;
  - Transmissão de voz ou imagens digitalizadas em que erros ocasionais são aceitáveis;
- Cabeçalho de um datagrama UDP:



TCP – Transmisson Control Protocol

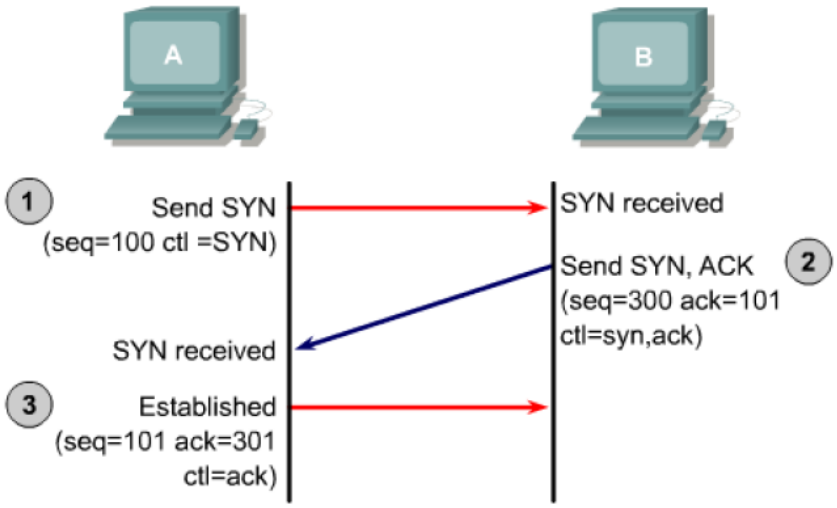
- Orientado a ligação;
- Estabelecimento da ligação antes da transferência de dados;
- Entrega fiável dos dados: sem erros ordenados, não duplicados;
- Fluxo de bytes bidirecional entre dois processos;
- Broadcasting & multicasting NÃO são suportados;
- Controlo de fluxo extremo–a–extremo;
- Fiabilidade do serviço de transporte baseado em:
  - Confirmação positiva;
  - *Timeout* de confirmação;
  - Retransmissão;
  - Reordenação;
  - Eliminação de segmentos duplicados;
  - Controlo de fluxo por janela deslizante.

- **<u>Fluxo de bytes:</u>**
  - Nos <u>recetores</u>, apenas existe um fluxo de bytes, sem qualquer separação baseada nos pedidos de envio efectuados ou segmentos gerados;
  - A interpretação dos conteúdos é da responsabilidade das aplicações;
  - Comparável ao tratamento de ficheiros.
- Cabeçalho de um segmento TCP:

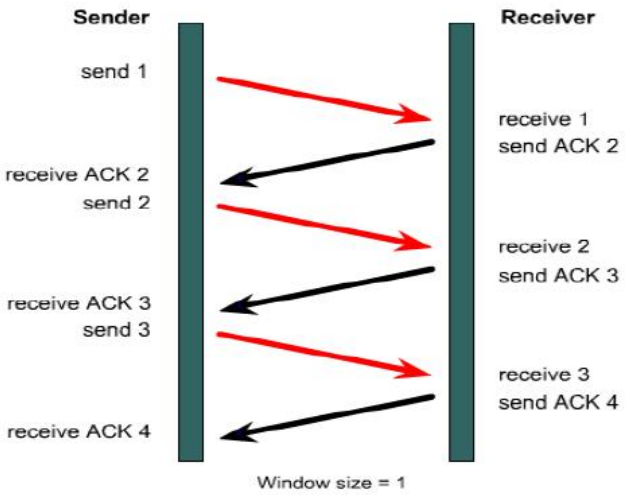


  - *<u>Campos</u>*
    - ***Code bits*** *(flag)*
      - SYN
        - Ativo durante o estabelecimento da ligação;
      - ACK
        - Indica se o campo **Número de Confirmação** deve ser considerado
      - URG
        - Indica de o campo **Ponteiro Urgente** deve ser considerado
      - PSH
        - O campo de dados deve ser enviado o mais rapidamente possível
      - RST
        - Reset de ligação
      - FIM
        - O emissor vai deixar de enviar dados
    - ***Window***
      - Controlo de fluxo

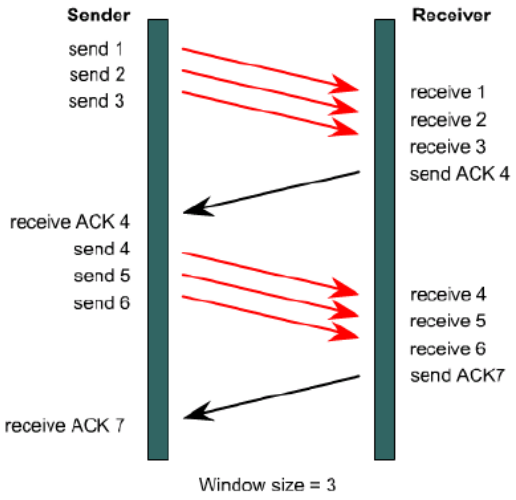
        - Número de *bytes* que o emissor do segmento pode aceitar
    - ***Checksum***
        - Abrange o cabeçalho e os dados
    - **Opções**
        - Tamanho máximo de segmento (MSS – *Maximum Segment Size)*
- Estabelecimento de uma ligação (*three-way handshake*)*:*



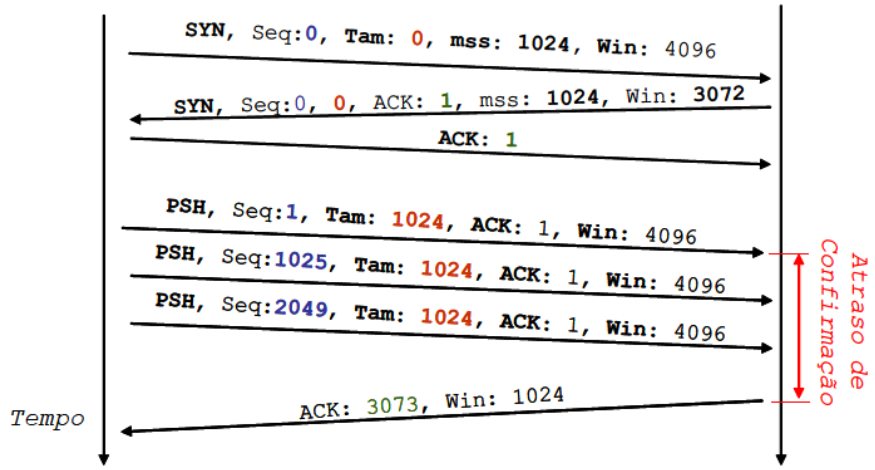
- Janela deslizante (tamanho = 1)



- Janela deslizante (tamanho = 3)

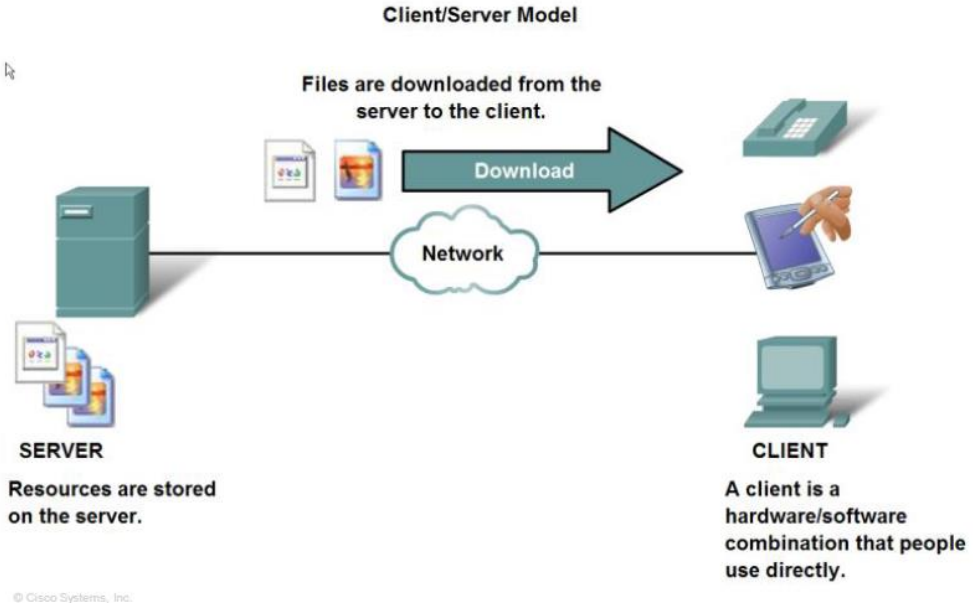


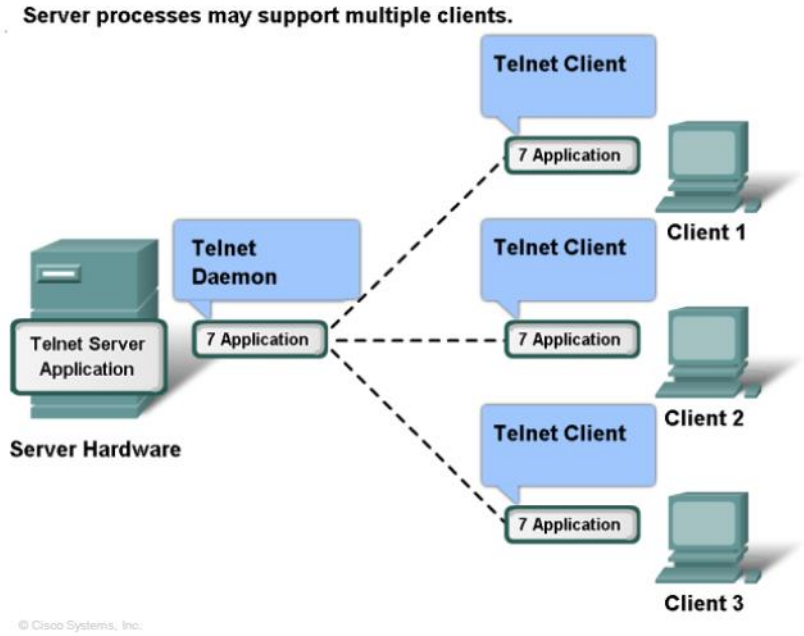
- No TCP, o tamanho da janela é definido em termos de quantidades de bytes e não de mensagens/segmentos



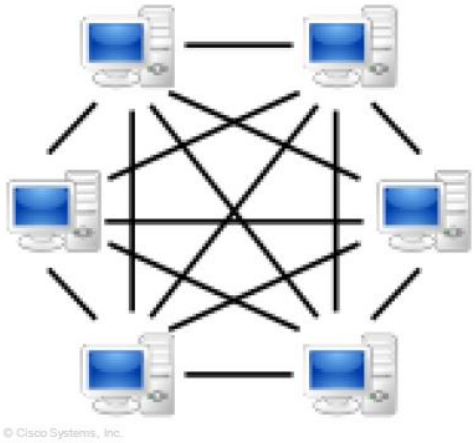
CAMADA DE APLICAÇÃO

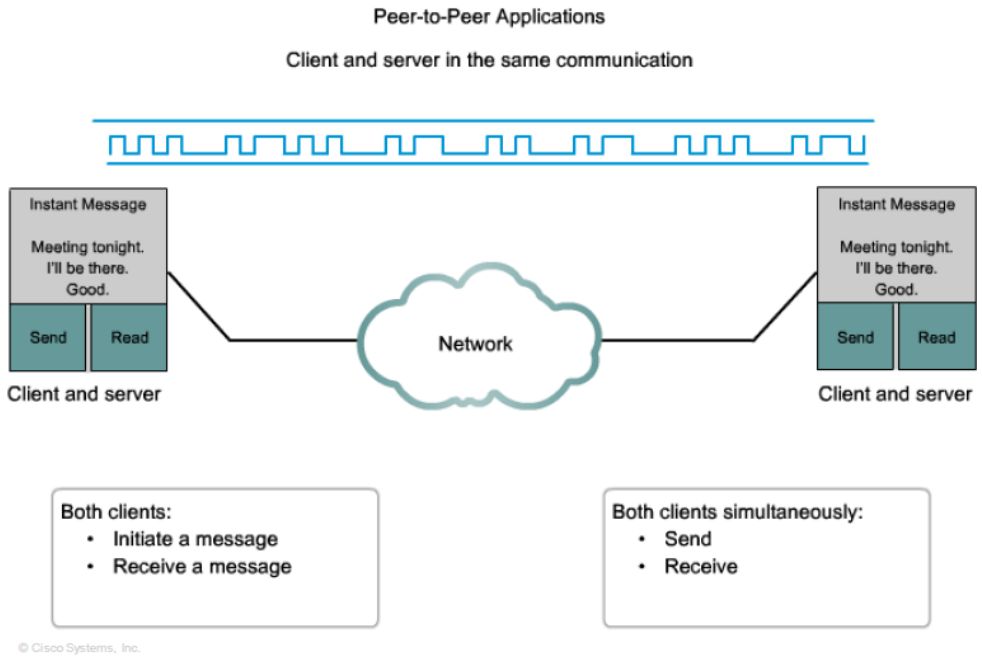
- **Modelo Cliente-Servidor**
  - Uma aplicação servidor oferece um serviço específico;
  - Associada a um porto well-known (pré-estabelecido);
  - Pode suportar vários clientes em simultâneo (servidor concorrente).





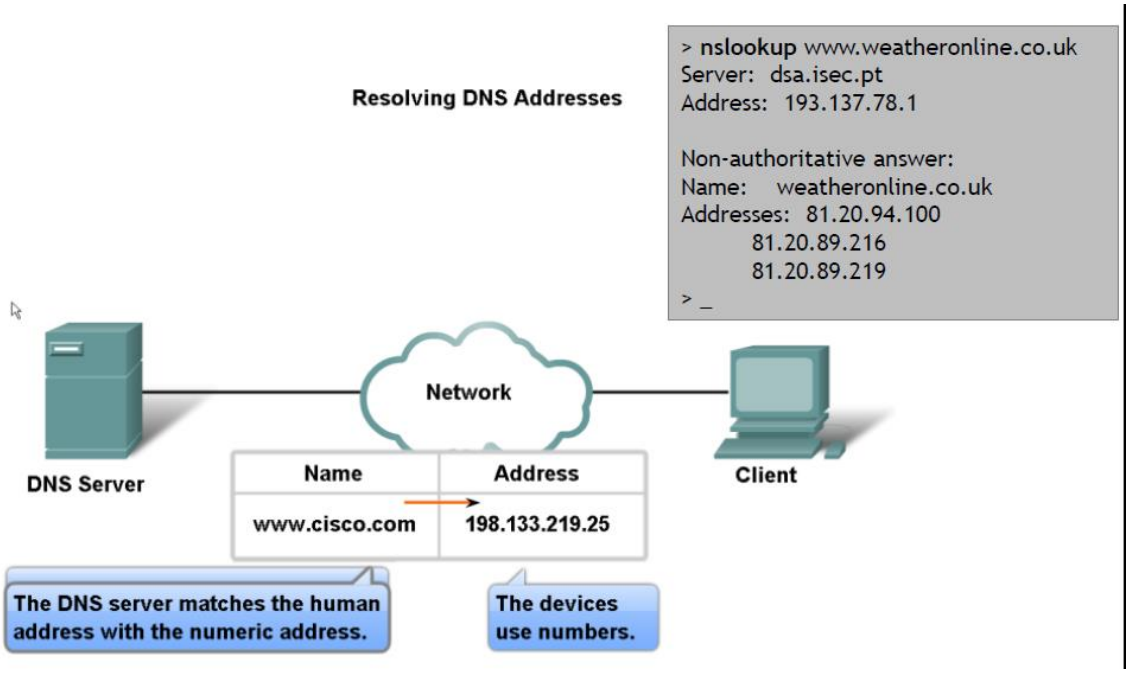
- **<u>Modelo Peer-to-Peer</u>**
  - Aplicações que operam tanto como servidores como clientes;
  - **<u>p.e.</u>** : instant messaging.





DNS – Domain Name System

- base de dados hierárquicos, distribuída e redundante que permite fazer o mapeamento entre nomes e endereços IP;
- serviço distribuído por vários servidores (DNS Servers/Name Servers);
- protocolo de transporte: UDP e TCP / porto por omissão: 53;



- principais serviços:
  - resolução de nomes de máquinas (nome → endereço IP);
  - resoluçao inversa (endereço IP → nome);
  - manter a base de dados atualizada e acessível na Internet.
- Recurso a *caching* para aumentar a eficiência do serviço;
- Udado sempre que, numa aplicação, o destino é identificado através de um nome em vez de um endereço IP.
- Espaço de nomeação organizado em domínios hierárquicos;
- A concessão de domínios é da responsabilidade da ICANN (*Internet Corporation for Assigned Names and Numbers*);
- Domínio de topo (TLD → *Top-Level domains*);

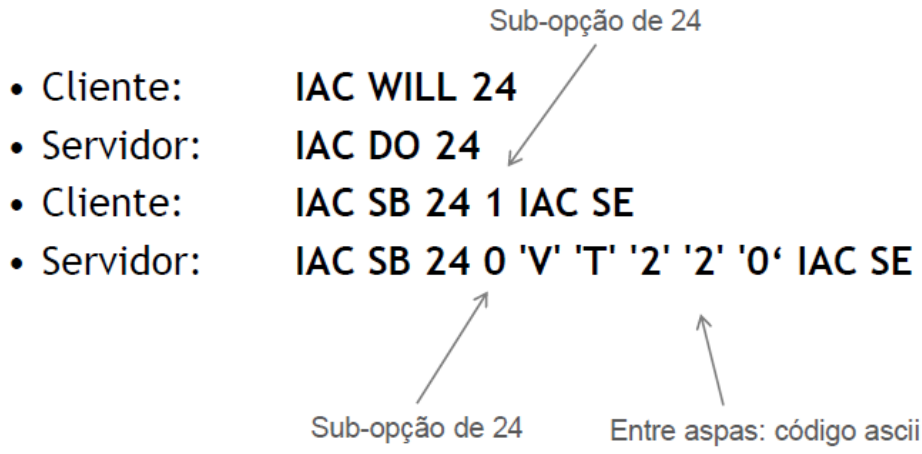
TELNET – Telecommunication Network Protocol

- Uma das aplicações mais antigas da Internet (1969);
- Protocolo de transporte: TCP;
- Protocolo de omissão: 23
- Funcionalidade:
  - Serviço de "terminal virtual" (VTY) que permite o acesso e a execusão remota de comandos noutro computador;
  - Um cliente Telnet permite estabelecer sessões com muitos serviços assentes em TCP e baseados em mensagens ASCII.
- Recorre ao *urgent mode* do protocolo TCP;

- Suporta autenticação mas ão encriptação;
- O SSH (*Secure Shell*) possui as mesmas funcionalidades que o Telmet mas encripta o diálogo entre o cliente e o servidor, permitindo um maior nível de segurança:
- As mensagens trocadas são comandos e texto;
- Os comandos são precedidios de carácter IAC (*Interpret As Command*) cujo código ASCII é 0xFF;

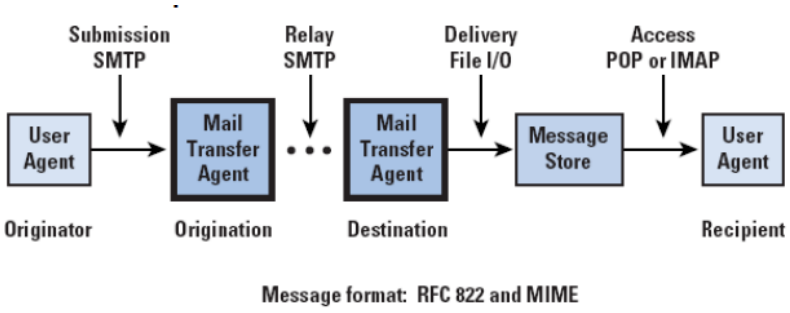
Code	Name	Description
240	SE	End of subnegotiation parameters.
241	NOP	No operation.
242	Data Mark	The data stream portion of a Synch. This should always be accompanied by a TCP Urgent notification.
243	Break	NVT character BRK.
244	Interrupt Process	The function IP.
245	Abort output	The function AO.
246	Are You There	The function AYT.
247	Erase character	The function EC.
248	Erase Line	The function EL.
249	Go ahead	The GA signal.
250	SB	Indicates that what follows is subnegotiation of the indicated option.
251	WILL (option code)	Indicates the desire to begin performing, or confirmation that you are now performing, the indicated option.
252	WON'T (option code)	Indicates the refusal to perform, or continue performing, the indicated option.
253	DO (option code)	Indicates the request that the other party perform, or confirmation that you are expecting the other party to perform, the indicated option.
254	DON'T (option code)	Indicates the demand that the other party stop performing, or confirmation that you are no longer expecting the other party to perform, the indicated option.
255	IAC	Data Byte 255.

- Exmeplo:
  - Se o cliente pretender saber qual é o tipo de terminal (número de opção 24) do servidor, é efectuada a seguinte troca de comandos:



E-MAIL: SMTP e POP

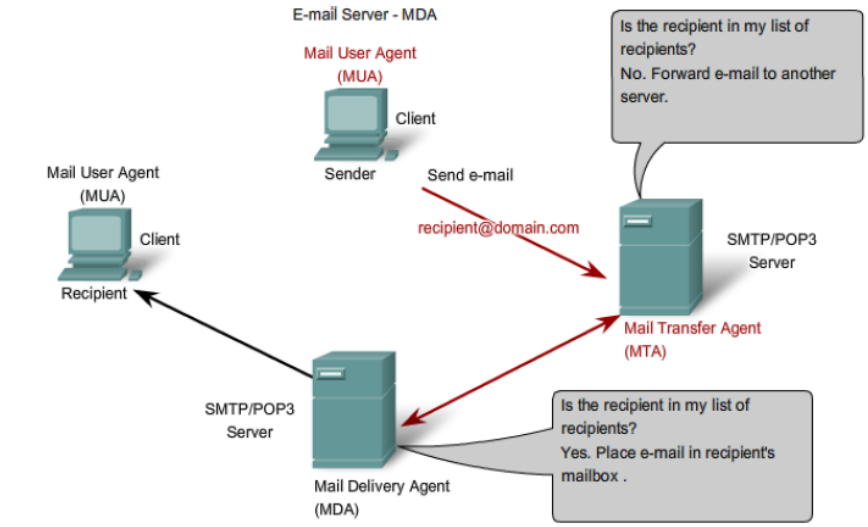
- Protocolos de apoio ao sistema de correio eletrónico;



- **SMTP – *Simple Mail Transport Protocol***
  - Protocolo: TCP;
  - Porto *well-known*: 25;
  - Mensagens de e-mail transportadas em formato ASCII;
- **POP – *Post Office Protocol* | IMAP – *Internet Message Access Protocol***
  - Permite a consulta/obtenção de mensagens de e-mail;
  - Versões seguras e mensagens encriptadas (POP3S / IMAPS);
  - Protocolo: TCP;
  - Porto *well-known*: 110;

- Noção de mensagem (de e-mail):

- **Cliente de mail (*Mail User Agent* - MUA)**
  - Acede às mensagens existentes no servidor de mail;
  - Programas: Eudora, Outlook, Netscape, etc..
  - Protocolos: POP3 e IMAP4;
- **Servidor de mail (*Message Transfer Agetn* - MTA)**
  - Encaminha as mensagens para os detinatários não locais;
  - Recebe as mensagens destinadas aos utilizadores locais;
  - Programas: Sendmail, Exchange, Outlook, etc..



The Mail Delivery Agent process governs delivery of e-mail between servers and clients.

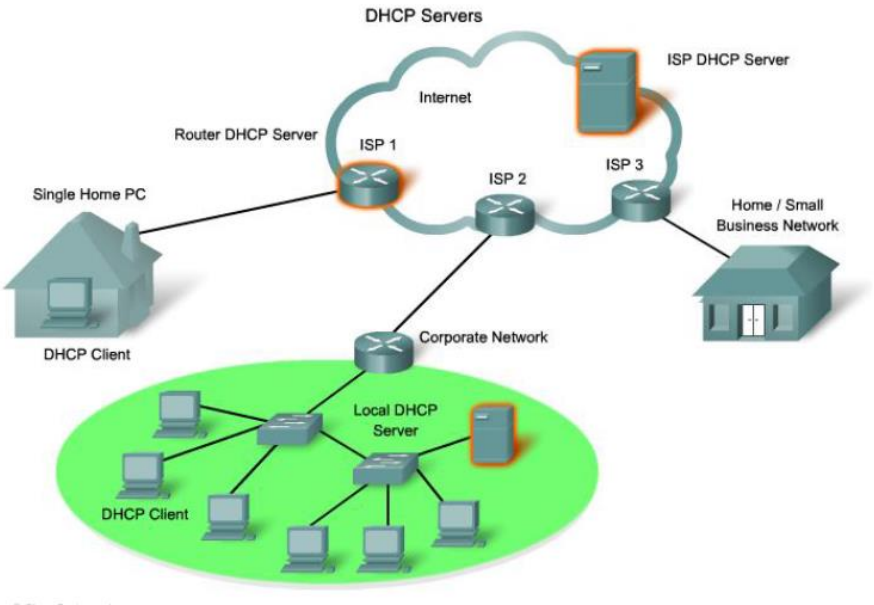
- Exemplo de diálogo **SMTP** entre um cliente e um servidor de e-mail:

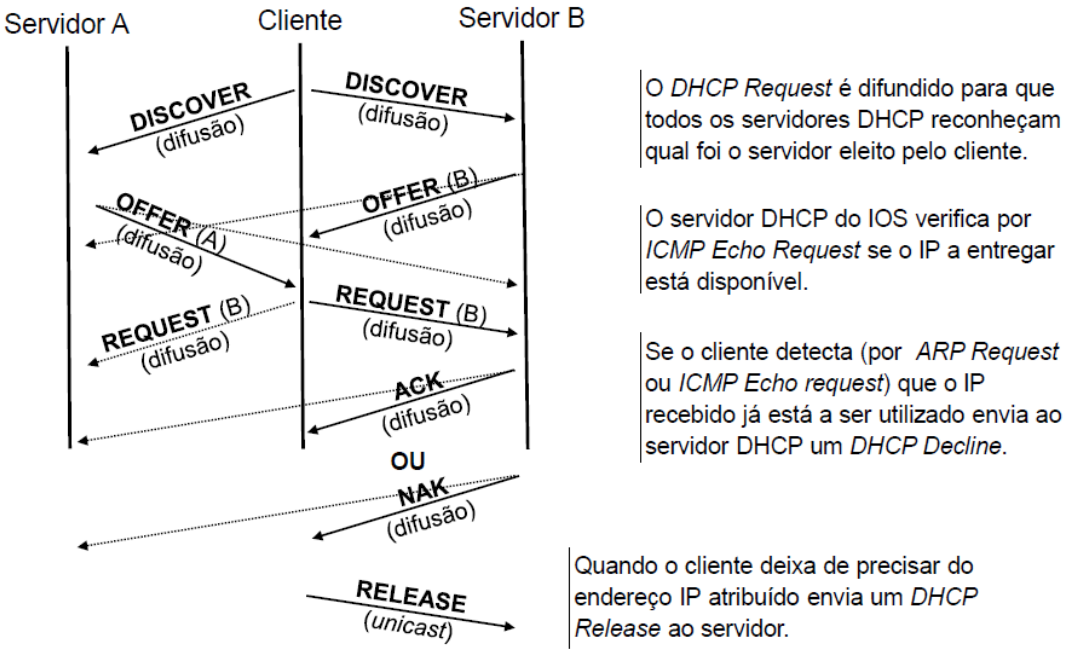
```
S: 220 www.example.com ESMTP Postfix
C: HELO mydomain.com
S: 250 Hello mydomain.com
C: MAIL FROM: sender@mydomain.com
S: 250 Ok
C: RCPT TO: friend@example.com
S: 250 Ok
C: DATA
S: 354 End data with <CR><LF>.<CR><LF>
C: Subject: test message
C: From: sender@mydomain.com
C: To: friend@example.com
C:
C: Hello,
C: This is a test.
C: Goodbye.
C: .
S: 250 Ok: queued as 12345
C: quit
S: 221 Bye
```

- Exemplo de diálogo **POP3** entre um cliente e um servidor de e-mail:

```
S:  +OK Microsoft Exchange Server 2003 POP3 server
    version 6.5.7638.1 (frontmail.isec.pt) ready.
C:  user Mike
S:  +OK
C:  pass 1234
S:  +OK User successfully logged on.
C:  stat
S:  +OK 1 3524
C:  quit
S:  +OK Microsoft Exchange Server 2003 POP3 server
    version 6.5.7638.1 signing off.
```

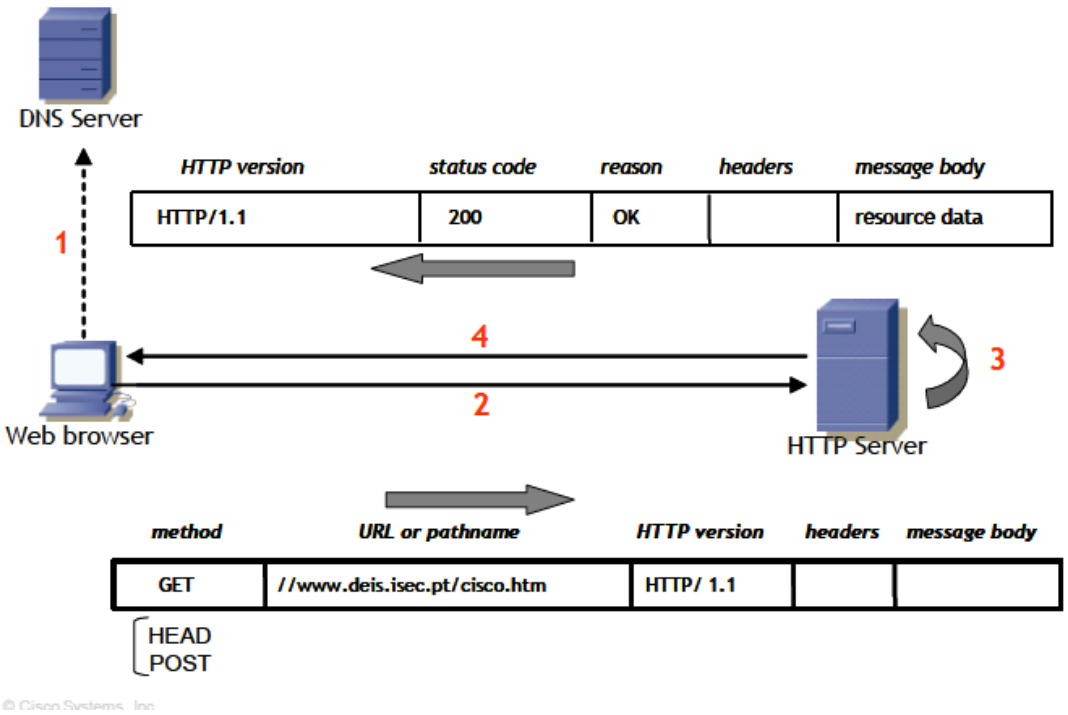
- **DHCP – *Dynamic Host Configuration Rpotocol***
  - Proceso alternativo à configuração estática do endereço IP e de outros parâmetros (router por omissão, DNS, etc..);
  - Tem por origem o protocolo BOOTP (BOOT *Protocol*);
  - Os endereços são "emprestados" por um servidor de DHCP por um determinado período e mediante o pedidio dos clientes;
  - Um servidor de DHCP possui uma ou várias gamas de endereços IP que vai enprestando e gerindo;
  - Protoocolo: UDP;
  - Portos *well-known*: 67 (servidor) e 68 (cliente);





- **HTTP – *HyperText Transfer Protocol***
  - Permite aceder ao "mundo WWW";
  - Os clientes (*browsers* obtêm ficheiros disponíveis nos servidores HTTP (*web*);
  - *Uniform Resource Locator* (URL):
    - Forma de especificar a localização de um recurso;
    - Formato:
      - **Protocolo://[endereço_host[:porto]][/caminho_recurso]**
      - Campo rotocolo: file, ftp, http, mailto;
  - Exemplos:
    - http://www.isec.pt
    - http://www.isec.pt/secretariavirtual
    - **etc…**
  - os *browsers* seleccionam, com base no campo protocolo, em que tipo de cliente devem "transformar-se";
  - mensagens de texto;
  - HEAD: pedido igual ao GET, sendo dispensado o conteúdo na resposta;
  - GET: solicitação de um recurso;
  - POST: *Upload* de parâmetros a serem processados;

- PUT: *Upload* de recursos;
- DELETE: apaga determinado recurso;
- OPTIONS: lista de métodos suportados pelo servidor;
- **Pedido/Resposta:**



- Exemplo de pedido/resposta HTTP:

PEDIDO

```
GET http://www.isec.pt HTTP/1.1
```

RESPOSTA

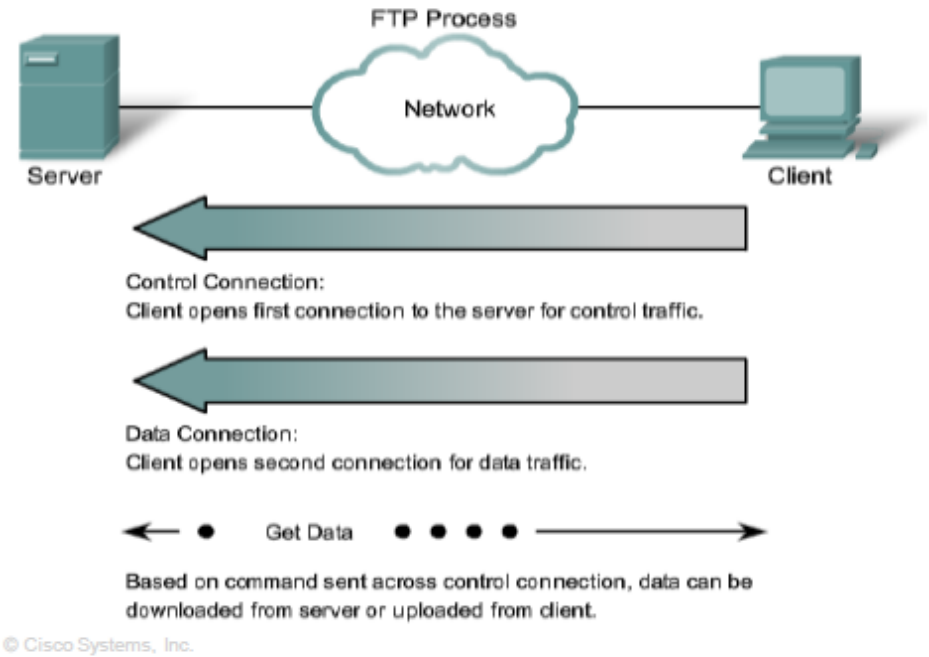
```
HTTP/1.1 200 OK
Date: Thu, 13 Nov 2014 16:20:31 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 64621

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="pt-
pt" xml:lang="pt-pt">
                              ...
</body>
</html>
```

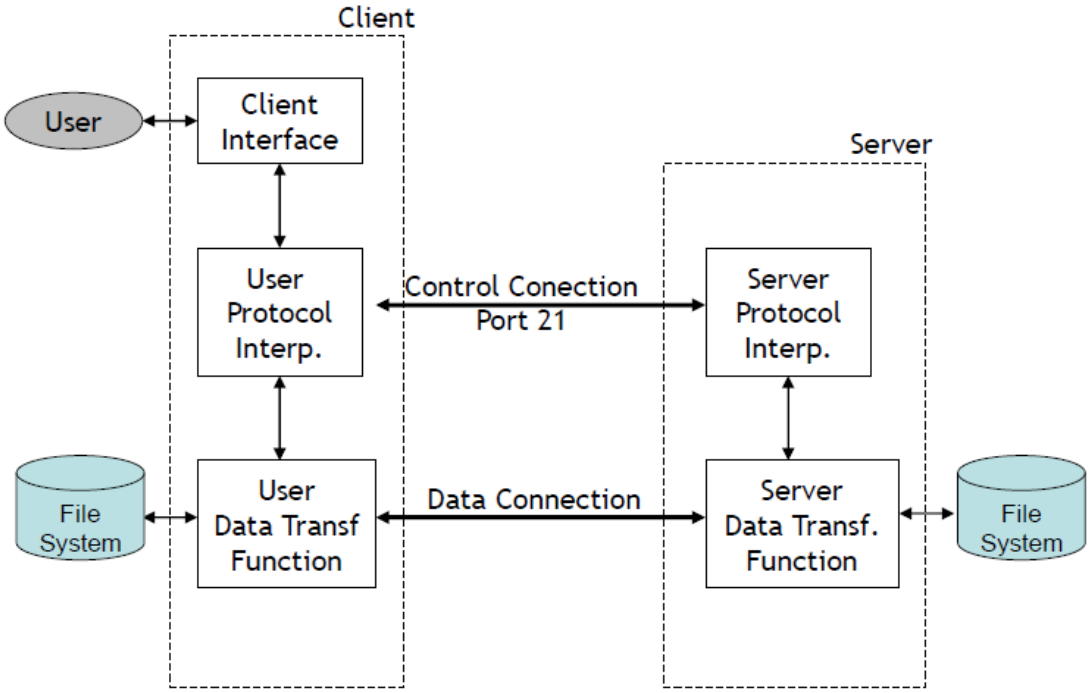
Código de resposta

Cabeçalho

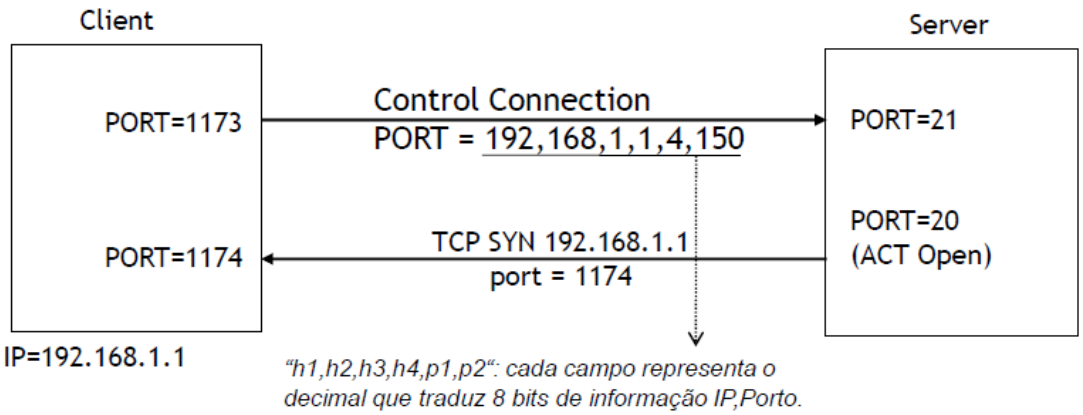
Conteúdo da resposta / recurso solicitado (opcional)

- **FTP – *File Transfer Protocol***
  - Transferência de ficheiros entre sistemas sistintos;
  - Protocolo: TCP;
  - Porto *well-known* (controlo): 21;
  - Modos de operação: Ativo, Passivo estendido;
  - Autenticação do cliente (username + password ou anónimo):
    - Operação básica insegura (senha enviada em *clear text*)
    - Existem estenções;
  - Suporta um número limitado de tipos de dados (ASCII, EBCDIC e binário);
  - Por regra, os *browsers web* possuem clientes FTP:
    - **ftp(s)://<login>:<password>@<ftpserveraddress>:<port>**
  - controlo do tipo *out-of-band*
    - ligação TCP para troca de dados de controlo (comandos);
    - ligações TCP (temporárias) para troca de dados;





- **Modo ativo:**
  - O cliente abre um *socket* num porto automático e estabelece uma sessão TCP (destinada a mensagens de controlo) com o porto 21 servidor;
  - Sempre que existe necessidade de transferência de dados, o cliente abre um novo *socket* num porto automático e anuncia-o ao servidor, estabelecendo este numa sessão TCP para o novo porto do cliente com origem no seu porto 20;
  - Uma sessão TCP temporária é encerrada após a conclusão da transferência dos dados para a qual foi aberta;
  - Cria problemas às *firewalls*, pelo que o modo passivo é preferido-



- **Modo passivo:**
  - O cliente abre um *socket* num porto automático e estabelece uma sessão TCP (destinada a mensagens de controlo) com o porto 21 do servidor;
  - O servidor abre um novo *socket* num porto automático e anuncia-o ao cliente, o qual estabelece uma sessão TCP temporária para o novo porto do servidor (destinada à transferência de dados) com origem num porto automático;
  - Uma sessão TCP temporária é encerrada após a conclusão da transferência dos dados para a qual foi aberta.