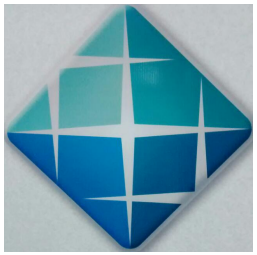
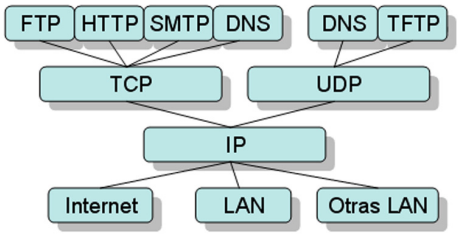
Protocolos de Redes

Aula 01

Evandro J.R. Silva[1]

[1] Bacharelado em Ciência da Computação
Estácio Teresina





Sumário

Introdução

Introdução

- Alguém já assistiu *The Crown* da Netflix?
- A visita oficial de qualquer Chefe de Estado ou Governo à Rainha Elizabete II (agora ao Rei Charles III) precisa obedecer a uma série de **protocolos**
  - Vestimenta adequada;
  - Gestos apropriados, no momento correto;
  - O que falar, e quando falar.
- Neste caso (e outros semelhantes), não seguir o protocolo pode causar transtornos diplomáticos!

Introdução

Introdução

- A comunicação entre os dispositivos também precisa utilizar protocolos para estabelecer a comunicação (o que e como enviar)
  - São dispositivos diferentes, que *entendem* os dados de forma diferente.
  - Os protocolos servem para que todos se entendam.

Introdução

- A comunicação entre os dispositivos também precisa utilizar protocolos para estabelecer a comunicação (o que e como enviar)
  - São dispositivos diferentes, que *entendem* os dados de forma diferente.
  - Os protocolos servem para que todos se entendam.
- Algumas regras definidas pelos protocolos de rede
  - Como a mensagem é formada e estruturada;
  - Como dispositivos de rede compartilham informações sobre rotas;
  - Como e quando mensagens de erro são passadas entre dispositivos;
  - A configuração e término das sessões de transferência de dados.

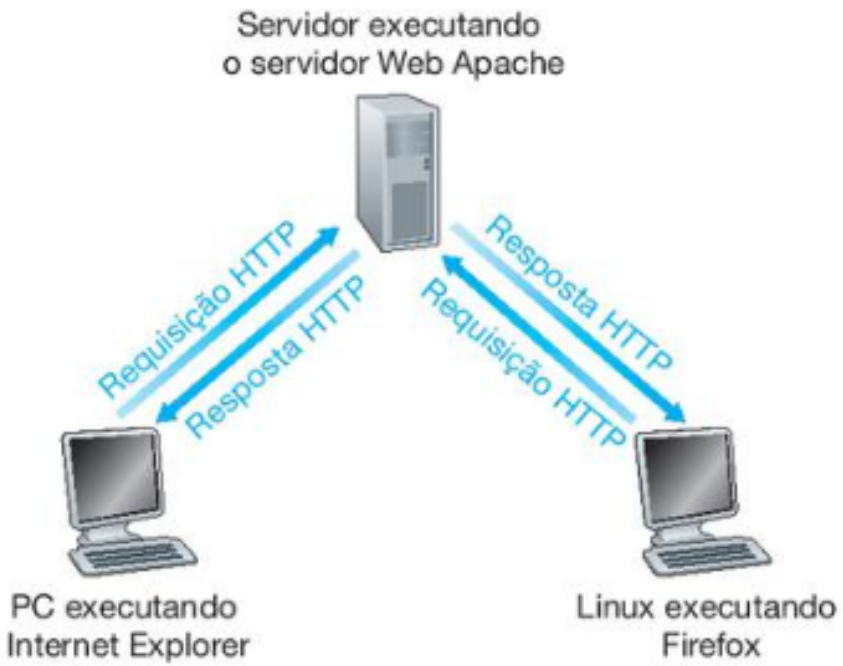
Introdução

ARQUITETURA TCP/IP	PROTOCOLOS TCP/IP
APLICAÇÃO	Telnet SMTP HTTP FTP DNS SMNP DHCP
TRANSPORTE	TCP UDP
REDE	IPv4, IPv6 ICMPv4, ICMPv6 ARP, RARP
ENLACE DE DADOS /FÍSICA	Ethernet PPP Frame Relay ATM WLAN

HTTP

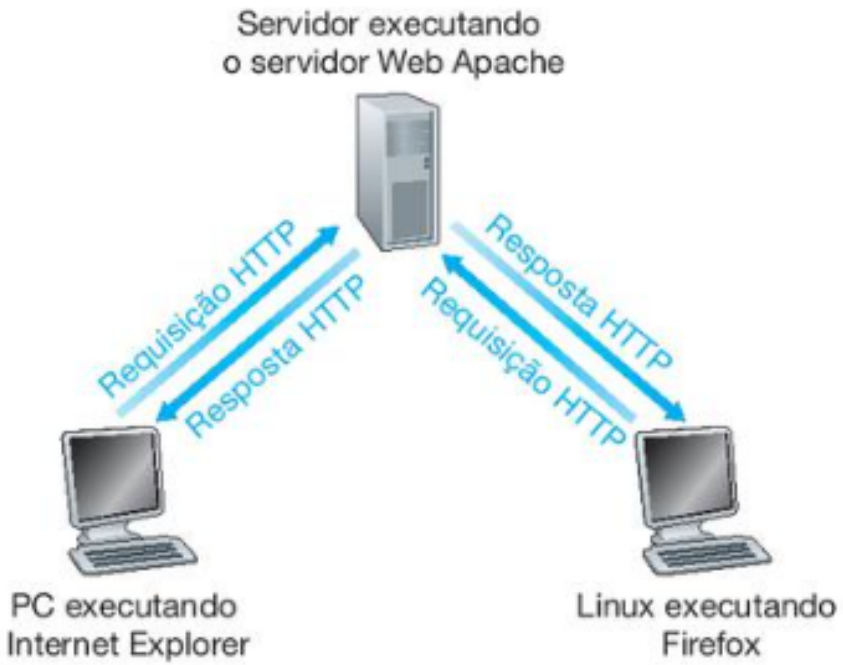
HTTP

- HTTP (*Hypertext Transfer Protocol*) definido no RFC 1945 e no RFC 2616.
- Duas últimas atualizações: RFC 9110 e RFC 9112
- O protocolo é executado em dois programas: um cliente e um servidor.
- A troca de mensagens entre os dois programas é feita através de mensagens HTTP (*Request* e *Response*).
- Utiliza o TCP para transporte de suas mensagens



HTTP

- HTTP (*Hypertext Transfer Protocol*) definido no RFC 1945 e no RFC 2616.
- Duas últimas atualizações: RFC 9110 e RFC 9112
- O protocolo é executado em dois programas: um cliente e um servidor.
- A troca de mensagens entre os dois programas é feita através de mensagens HTTP (*Request* e *Response*).
- Utiliza o TCP para transporte de suas mensagens
  - Veremos esse protocolo em outra aula.



HTTP

- Mensagem de requisição (*request*)

```
GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
Connection: close
User-agent: Mozilla/5.0
Accept-language: fr
```

HTTP

- Mensagem de requisição (*request*)

```
GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
Connection: close
User-agent: Mozilla/5.0
Accept-language: fr
```

  - Três elementos principais: **linha de requisição**, **linhas de cabeçalho** e **corpo da entidade/mensagem**.

HTTP

- Mensagem de requisição (*request*)

```
GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
Connection: close
User-agent: Mozilla/5.0
Accept-language: fr
```

  - Três elementos principais: **linha de requisição**, **linhas de cabeçalho** e **corpo da entidade/mensagem**.
  - **Linha de requisição**: `Método` sp `URI` sp `Versão HTTP` crlf

HTTP

- Mensagem de requisição (*request*)

```
GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
Connection: close
User-agent: Mozilla/5.0
Accept-language: fr
```

  - Três elementos principais: **linha de requisição**, **linhas de cabeçalho** e **corpo da entidade/mensagem**.
  - **Linha de requisição**: `Método` sp `URI` sp `Versão HTTP` crlf
    - Métodos: **OPTIONS**, **GET**, **HEAD**, **POST**, **PUT**, **DELETE**, **TRACE**, **CONNECT**.

HTTP

- Mensagem de requisição (*request*)

```
GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
Connection: close
User-agent: Mozilla/5.0
Accept-language: fr
```

  - Três elementos principais: **linha de requisição**, **linhas de cabeçalho** e **corpo da entidade/mensagem**.
  - **Linha de requisição**: `Método` sp `URI` sp `Versão HTTP` crlf
    - Métodos: **OPTIONS**, **GET**, **HEAD**, **POST**, **PUT**, **DELETE**, **TRACE**, **CONNECT**.
    - URI: *Uniform Resource Identifier*, usado para idenficiar um recurso em um servidor.

HTTP

- Mensagem de requisição (*request*)

```
GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
Connection: close
User-agent: Mozilla/5.0
Accept-language: fr
```

  - Três elementos principais: **linha de requisição**, **linhas de cabeçalho** e **corpo da entidade/mensagem**.
  - **Linha de requisição**: `Método` sp `URI` sp `Versão HTTP` crlf
  - **Linhas de cabeçalho**: `Campo` : `valor`

HTTP

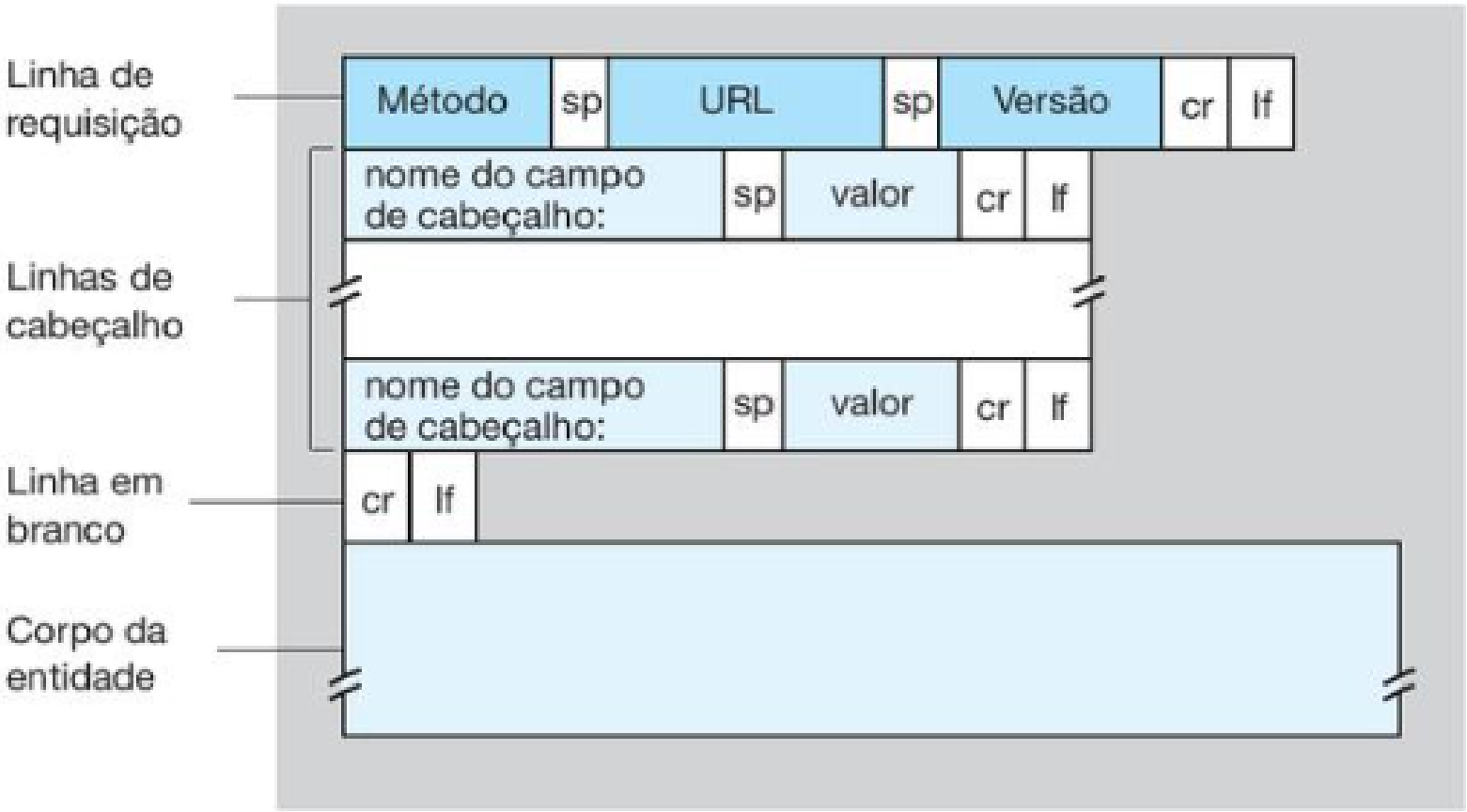
- Mensagem de requisição (*request*)

```
GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
Connection: close
User-agent: Mozilla/5.0
Accept-language: fr
```

  - Três elementos principais: **linha de requisição**, **linhas de cabeçalho** e **corpo da entidade/mensagem**.
  - **Linha de requisição**: `Método` sp `URI` sp `Versão HTTP` crlf
  - **Linhas de cabeçalho**: `Campo` : `valor`
    - Campos: **Accept**, **Accept-Charset**, **Accept-Encoding**, **Accept-Language**, **Authorization**, **Expect**, **From**, **Host**, **If-Match**, **If-Modified-Since**, **If-None-Match**, **If-Range**, **If-Unmodified-Since**, **Max-Forwards**, **Proxy-Authorization**, **Range**, **Referer**, **TE**, **User-Agent**.

HTTP

FORMATO GERAL DE UMA MENSAGEM DE REQUISIÇÃO HTTP



HTTP

- Mensagem de resposta (*response*)

```
HTTP/1.1 200 OK
Connection: close
Date: Tue, 09 Aug 2011 15:44:04 GMT
Server: Apache/2.2.3 (CentOS)
Last-Modified: Tue, 09 Aug 2011 15:11:03 GMT
Content-Length: 6821
Content-Type: text/html

(dados dados dados dados dados ...)
```

  - Três elementos principais: **linha de estado**, **linhas de cabeçalho** e **corpo de entidade/mensagem**.

HTTP

- Mensagem de resposta (*response*)

```
HTTP/1.1 200 OK
Connection: close
Date: Tue, 09 Aug 2011 15:44:04 GMT
Server: Apache/2.2.3 (CentOS)
Last-Modified: Tue, 09 Aug 2011 15:11:03 GMT
Content-Length: 6821
Content-Type: text/html

(dados dados dados dados dados ...)
```

  - Três elementos principais: **linha de estado**, **linhas de cabeçalho** e **corpo de entidade/mensagem**.
  - **Linha de estado**: `Versão HTTP` sp `Código de status` sp `Frase do Código` crlf

HTTP

- Mensagem de resposta (*response*)

```
HTTP/1.1 200 OK
Connection: close
Date: Tue, 09 Aug 2011 15:44:04 GMT
Server: Apache/2.2.3 (CentOS)
Last-Modified: Tue, 09 Aug 2011 15:11:03 GMT
Content-Length: 6821
Content-Type: text/html

(dados dados dados dados dados ...)
```

  - Três elementos principais: **linha de estado**, **linhas de cabeçalho** e **corpo de entidade/mensagem**.
  - **Linha de estado**: `Versão HTTP` sp `Código de status` sp `Frase do Código` crlf
    - 1xx : Informacional - requisição recebida, processo em continuação;
    - 2xx : Sucesso - a ação foi recebida com sucesso, entendida e aceita;
    - 3xx : Redirecionamento - mais ações precisam ser tomadas para completar a requisição;
    - 4xx : Erro do Cliente - a requisição contém erro de sintaxe ou não pode ser atendida;
    - 5xx : Erro de Servidor - o servidor falhou em atender ou requisição supostamente válida.

HTTP

- Código de status e sua respectiva frase:
  - 100 : Continue
  - 101 : Switching Protocols
  - 200 : OK
  - 201 : Created
  - 202 : Accepted
  - 203 : Non-Authoritative Information
  - 204 : No Content
  - 205 : Reset Content
  - 206 : Partial Content
  - 300 : Multiple Choices
  - 301 : Moved Permanently
  - 302 : Found
  - 303 : See Other
  - 304 : Not Modified
  - 305 : Use Proxy
  - 307 : Temporary Redirect
  - 400 : Bad Request
  - 401 : Unauthorized
  - 402 : Payment Required
  - 403 : Forbidden
  - 404 : Not Found
  - 405 : Method Not Allowed
  - 406 : Not Acceptable
  - 407 : Proxy Authentication Required
  - 408 : Request Time-out
  - 409 : Conflict
  - 410 : Gone
  - 411 : Length Required
  - 412 : Precondition Failed
  - 413 : Request Entity Too Large
  - 414 : Request-URI Too Large
  - 415 : Unsupported Media Type
  - 416 : Requested range not satisfiable
  - 417 : Expectation Failed
  - 500 : Internal Server Error
  - 501 : Not Implemented
  - 502 : Bad Gateway
  - 503 : Service Unavailable
  - 504 : Gateway Time-out
  - 505 : HTTP Version not supported

HTTP

- Mensagem de resposta (*response*)

```
HTTP/1.1 200 OK
Connection: close
Date: Tue, 09 Aug 2011 15:44:04 GMT
Server: Apache/2.2.3 (CentOS)
Last-Modified: Tue, 09 Aug 2011 15:11:03 GMT
Content-Length: 6821
Content-Type: text/html

(dados dados dados dados dados ...)
```

  - Três elementos principais: **linha de estado**, **linhas de cabeçalho** e **corpo de entidade/mensagem**.
  - **Linha de estado**: `Versão HTTP` sp `Código de status` sp `Frase do Código` crlf
  - **Linhas de cabeçalho**: `Campo` : `valor`

HTTP

- Mensagem de resposta (*response*)

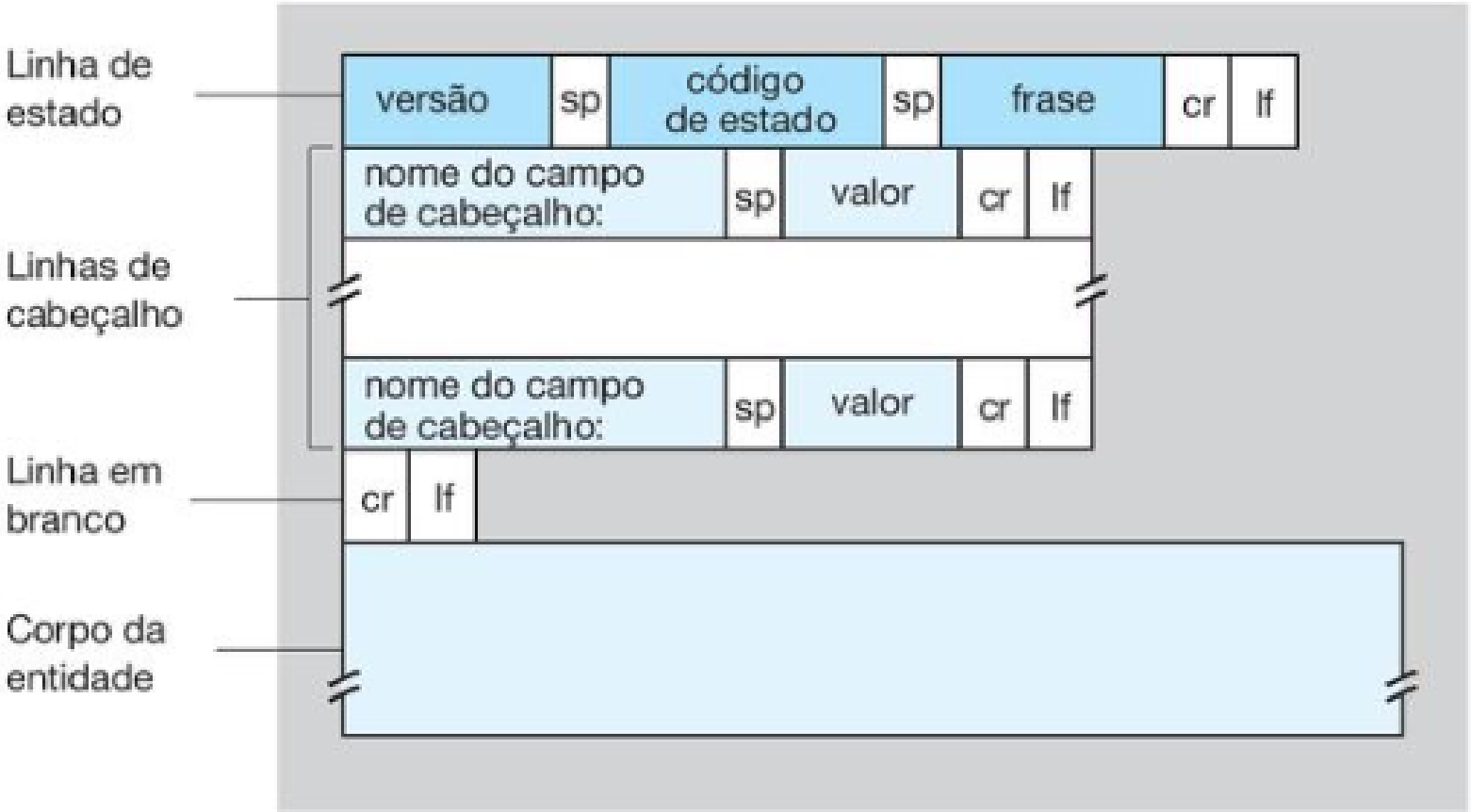
```
HTTP/1.1 200 OK
Connection: close
Date: Tue, 09 Aug 2011 15:44:04 GMT
Server: Apache/2.2.3 (CentOS)
Last-Modified: Tue, 09 Aug 2011 15:11:03 GMT
Content-Length: 6821
Content-Type: text/html

(dados dados dados dados dados ...)
```

  - Três elementos principais: **linha de estado**, **linhas de cabeçalho** e **corpo de entidade/mensagem**.
  - **Linha de estado**: `Versão HTTP` sp `Código de status` sp `Frase do Código` crlf
  - **Linhas de cabeçalho**: `Campo : valor`
    - *Response Header*: **Accept-Ranges**, **Age**, **ETag**, **Location**, **Proxy-Authenticate**, **Retry-After**, **Server**, **Vary**, **WWW-Authenticate**.
    - *Entity Header*: **Allow**, **Content-Encoding**, **Content-Language**, **Content-Length**, **Content-Location**, **Content-MD5**, **Content-Range**, **Content-Type**, **Expires**, **Last-Modified**.

HTTP

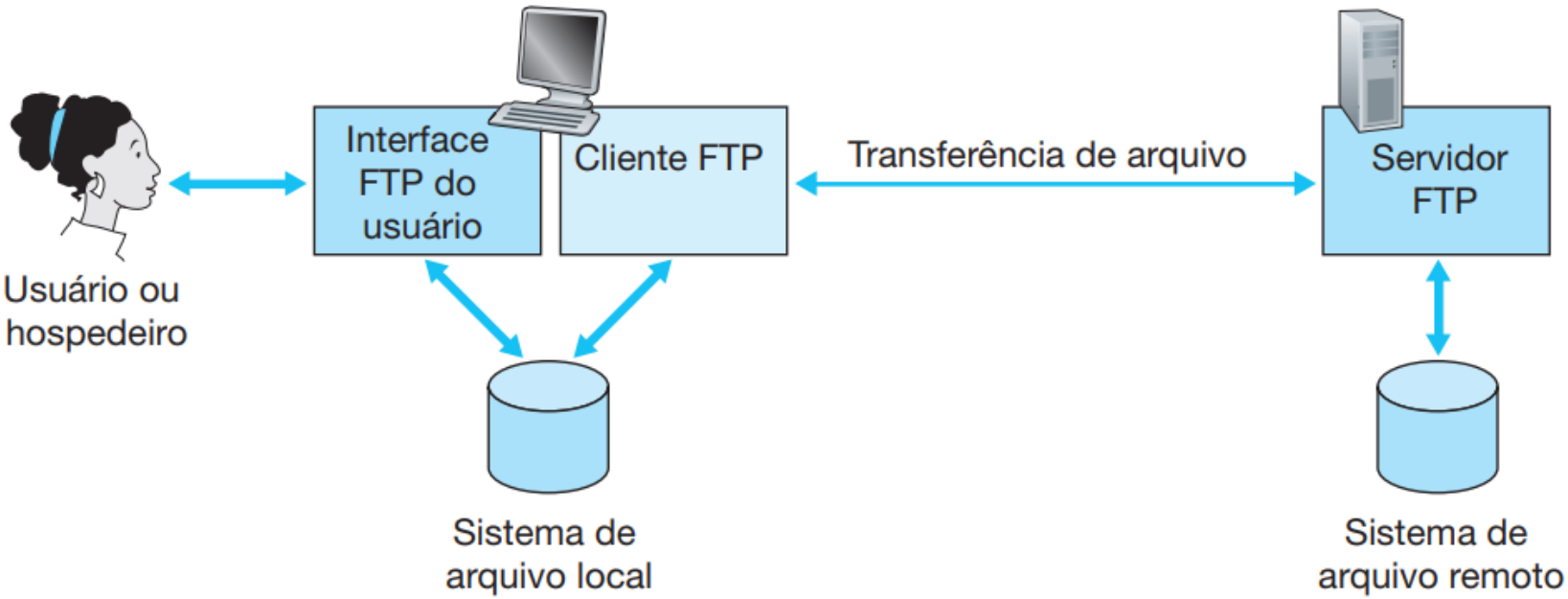
FORMATO GERAL DE UMA MENSAGEM DE RESPOSTA HTTP



FTP

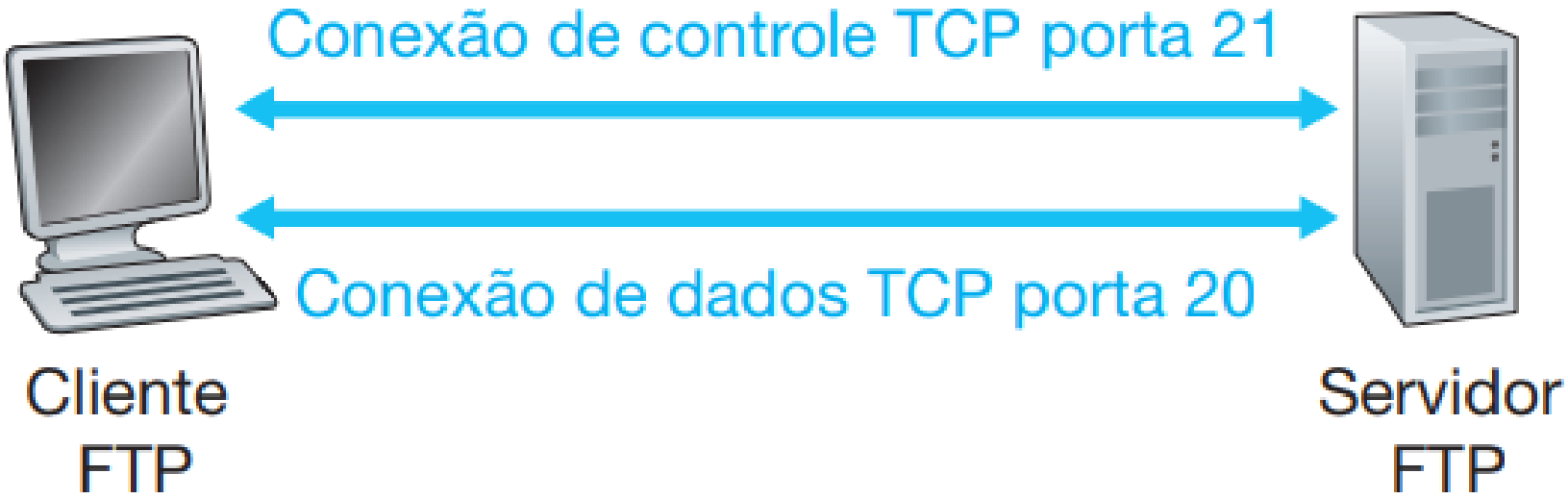
FTP

- FTP (*File Transfer Protocol*) foi primeiro definido no RFC 114.
- Atualmente é definido por: RFC 959, RFC 2228, RFC 2640, RFC 2773, RFC 3659, RFC 5797 e RFC 7151.
- A IETF, entretanto, decidiu aposentá-lo.
- É um protocolo utilizado para transferência de arquivos de/para um hospedeiro remoto
  - Para acessar a conta remota, o usuário deve fornecer uma identificação e uma senha.
  - Após fornecer essas informações de autorização, pode transferir arquivos do sistema local de arquivos para o sistema remoto e vice-versa.



FTP

- O FTP usa duas conexões TCP paralelas para transferir um arquivo: uma **conexão de controle** e uma **conexão de dados**
  - A primeira é usada para enviar informações de controle entre os dois hospedeiros — como identificação de usuário, senha, comandos para trocar diretório remoto e comandos de "enviar" (*put*) e "receber" (*get*) arquivos.
  - A segunda, ou seja, a conexão de dados é usada para enviar de fato um arquivo.



FTP

- Como acontece uma transferência de arquivo:

FTP

- Como acontece uma transferência de arquivo:
  1. Quando um usuário inicia uma sessão FTP com um hospedeiro remoto, o lado cliente do FTP (usuário) inicia primeiro uma conexão TCP de controle com o lado servidor (hospedeiro remoto) na porta número 21 do servidor e envia por essa conexão de controle a identificação e a senha do usuário, além de comandos para mudar o diretório remoto.

FTP

- Como acontece uma transferência de arquivo:

2. Quando o lado servidor recebe, pela conexão de controle, um comando para uma transferência de arquivo (de ou para o hospedeiro remoto), abre uma conexão TCP de dados para o lado cliente.

FTP

- Como acontece uma transferência de arquivo:

3. O FTP envia exatamente um arquivo pela conexão de dados e em seguida fecha-a.

FTP

- Como acontece uma transferência de arquivo:

- Se, durante a mesma sessão, o usuário quiser transferir outro arquivo, o FTP abrirá outra conexão de dados (conexão **não persistente**).

FTP

- Os comandos, do cliente para o servidor, e as respostas, do servidor para o cliente, são enviados por meio da conexão de controle no formato ASCII de 7 bits.
- Alguns dos comandos mais comuns:

FTP

- Os comandos, do cliente para o servidor, e as respostas, do servidor para o cliente, são enviados por meio da conexão de controle no formato ASCII de 7 bits.
- Alguns dos comandos mais comuns:
  - `USER nome`: comando para enviar o nome de usuário para o servidor;
  - `PASS senha`: comando para enviar a senha do usuário para o servidor;
  - `LIST`: comando para que o servidor envie uma lista com todos os arquivos existentes no atual diretório remoto. A lista é enviada pela conexão de dados.
  - `RETR arquivo`: comando para obter um arquivo do diretório remoto atual.
  - `STOR arquivo`: comando para armazenar um arquivo no diretório remoto atual.

FTP

- Os comandos, do cliente para o servidor, e as respostas, do servidor para o cliente, são enviados por meio da conexão de controle no formato ASCII de 7 bits.
- Alguns dos comandos mais comuns:

- Cada comando é seguido de uma resposta do servidor. Alguns exemplos:

FTP

- Os comandos, do cliente para o servidor, e as respostas, do servidor para o cliente, são enviados por meio da conexão de controle no formato ASCII de 7 bits.
- Alguns dos comandos mais comuns:

- Cada comando é seguido de uma resposta do servidor. Alguns exemplos:
  - `331`: nome de usuário OK, senha requisitada;
  - `125`: conexão de dados já aberta; iniciando transferência;
  - `425`: não é possível abrir conexão de dados;
  - `452`: erro ao escrever o arquivo.

Telnet

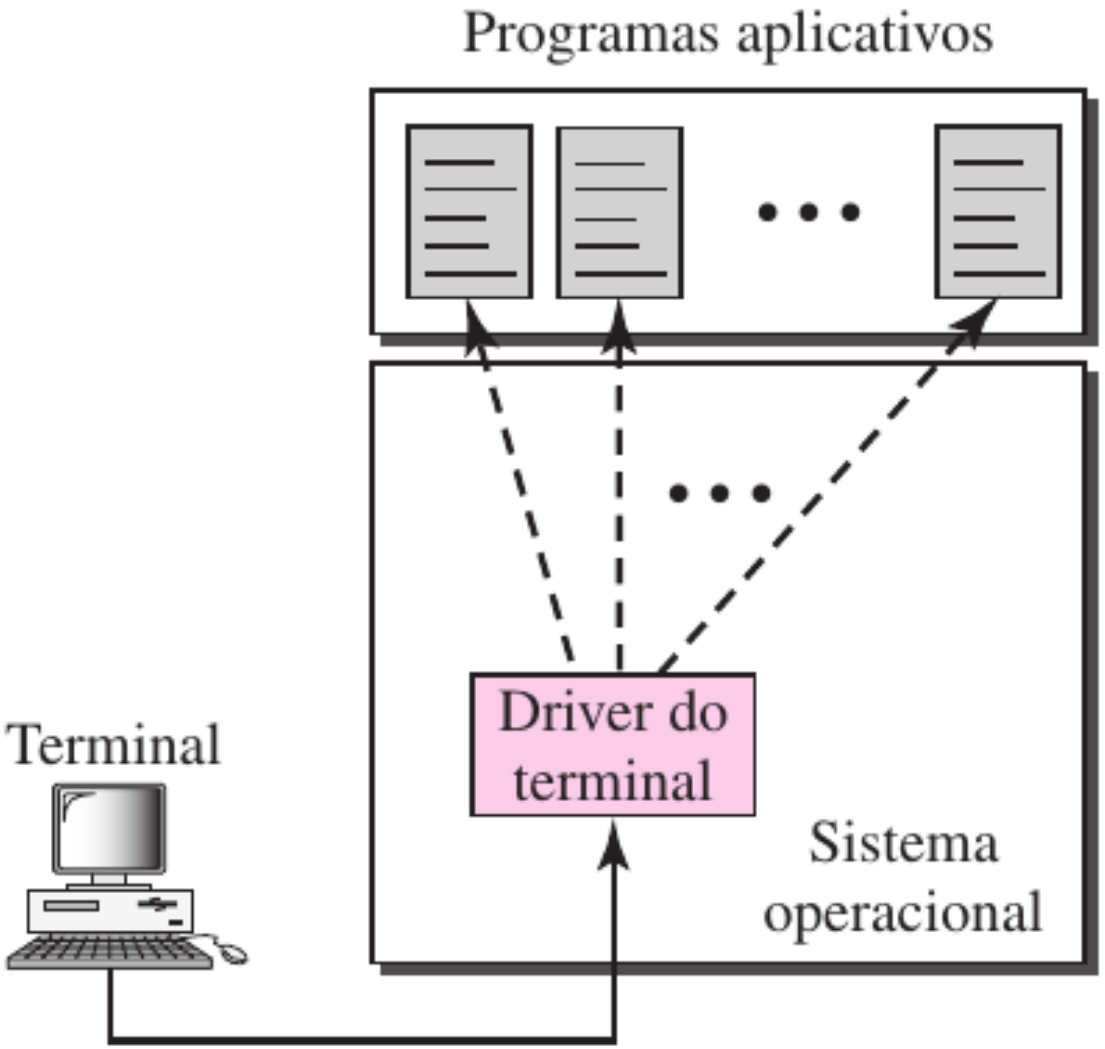
Telnet

- Telnet (*TELetype NETwork*) é a aplicação padrão da Internet para serviços de terminal virtual.
- É um protocolo de aplicação cliente/servidor.
- Permite que um terminal local estabeleça uma conexão virtual a um sistema remoto de tal maneira que o terminal local se comporte exatamente como se fosse um terminal do sistema remoto.
- Definido como STD 8, o qual é composto pelos RFCs 854 e 855.
- Outros RFCs do Telnet que possuem o status de Padrão da Internet:
  - RFC 856, conhecido como STD 27: *Telnet Binary Transmission*.
  - RFC 857, conhecido como STD 28: *Telnet Echo Option*.
  - RFC 858, conhecido como STD 29: *Telnet Suppress Go Ahead Option*.
  - RFC 859, conhecido como STD 30: *Telnet Status Option*.
  - RFC 860, conhecido como STD 31: *Telnet Timing Mark Option*.
  - RFC 861, conhecido como STD 32: *Telnet Extended Options: List Option*.

Telnet

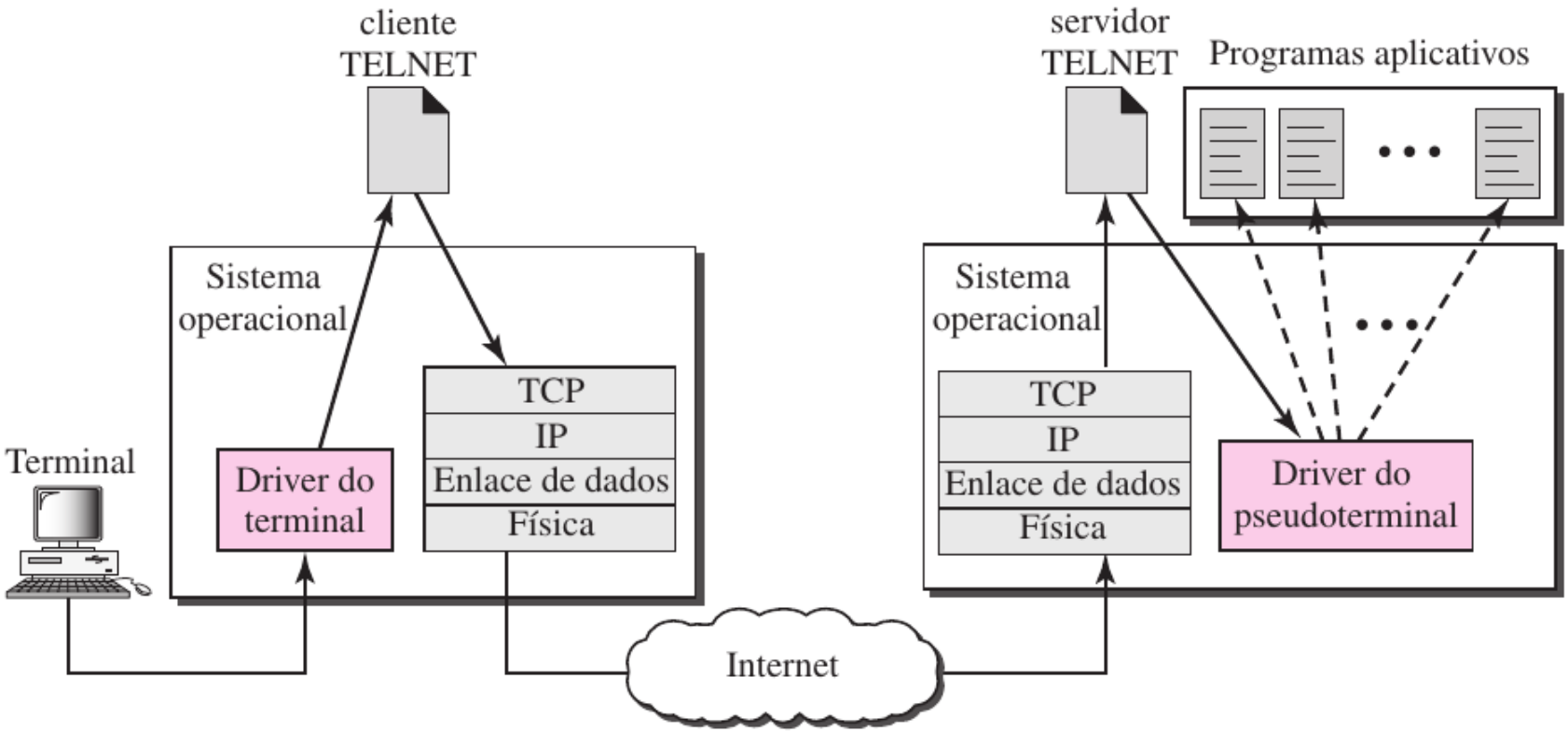
- O Telnet foi desenvolvido em uma época em que a maioria dos sistemas operacionais funcionava em um ambiente de tempo compartilhado.
- Ou seja, havia um *mainframe* que suportava vários usuários ao mesmo tempo, cada um com seu monitor, teclado e mouse. Cada usuário estava em um terminal.

Telnet


Programas aplicativos
Driver do terminal
Terminal
Sistema operacional

Telnet

- Quando um usuário se conecta ao sistema local de tempo compartilhado, ele realiza um login local.
- Quando um usuário quer acessar um programa aplicativo ou utilitário, localizado em uma máquina remota, ele realiza um login remoto.
  - É aqui que os programas Telnet no cliente e no servidor entram em ação.



Telnet

- O Telnet utiliza apenas 1 conexão TCP, e o servidor estará *ouvindo* na porta 23.
- Só tem como utilizar via linha de comando, ou seja, não existe interface gráfica.
- Não possui criptografia, por isso não é nada recomendável acessar um servidor usando telnet e logando com seu usuário e senha, e ainda enviando ou pegando arquivos sensíveis.

Telnet

- Um pouco mais detalhes, com boa explicação, você pode encontrar no capítulo 26 do livro do Forouzan:
  FOROUZAN, Behrouz A. **Comunicação de dados e redes de computadores**. AMGH, 4 ed., Porto Alegre, 2010.

FIM