

# Redes de computadores

## Camada de aplicação

Prof. Luís Eduardo Tenório Silva  
[luiz.silva@garanhuns.ifpe.edu.br](mailto:luiz.silva@garanhuns.ifpe.edu.br)

- Aplicação de rede
- Arquitetura de aplicação
- Serviços de transporte
- Protocolos de camada de aplicação
- HTTP
- Transferência de uma página HTML
- Realizar conexão TCP
- Requisitar arquivo
- Formato da mensagem HTTP
- Código de estado HTTP
- Respostas de sucesso (200-299)
- Redirecionamento (300-399)
- Resposta de erro do cliente (400-499)
- Resposta de erro do servidor (500-599)
- Referências

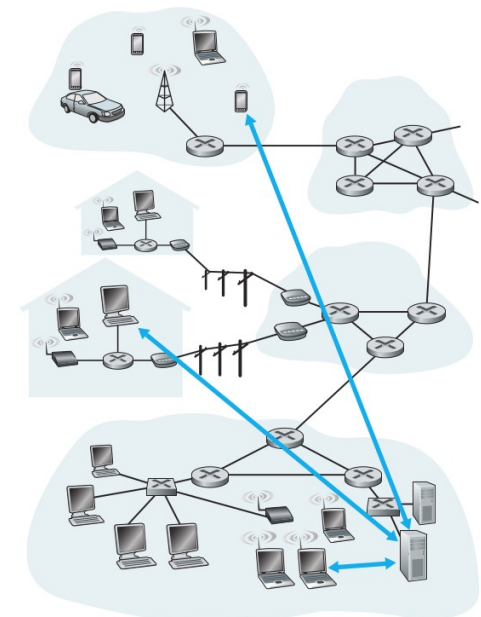
- Aplicações são usados por sistemas finais diferentes que se comunicam entre si;
- Aplicação Web
  - » Navegador: Cliente web
  - » Servidor Web: Responde as requisições do cliente
- Aplicação torrent
  - » Mesmo aplicativo para cliente/servidor

- **Arquitetura cliente/servidor**

- » Existe um hospedeiro sempre em funcionamento (servidor)
- » Requisita dados a ao hospedeiro (cliente)
- » Características
  - Clientes não se comunicam uns com os outros
  - Servidor possui um endereço fixo (endereço IP)

- Ex:

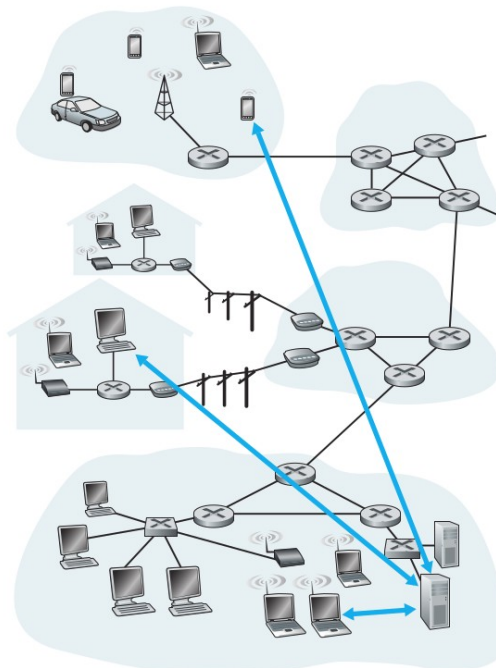
- » Servidor Web
- » FTP
- » Telnet/SSH
- » e-mail



a. Arquitetura cliente-servidor

- **Problemas arquitetura cliente/servidor**

- » Um único hospedeiro pode ser incapaz de atender todas as requisições
  - Necessário um **datacenter** para atender todas as requisições.



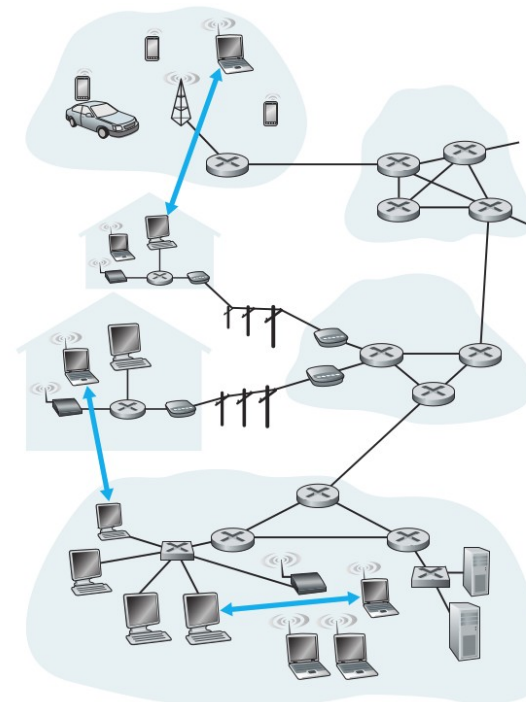
a. Arquitetura cliente-servidor

- **Arquitetura P2P**

- » Comunicação direta entre duplas de hospedeiros conectados (*peers*)
- » Os próprios clientes são ao mesmo tempo cliente/servidor;
- » Autoescalabilidade
  - *Peers* podem entrar e requisitar/disponibilizar arquivos a qualquer momento

- Ex:

- » Bittorrent
- » Skype
- » IPTV



b. Arquitetura P2P

- Dois protocolos podem ser utilizados pelas aplicações para transportar os dados:
  - » TCP
  - » UDP
- Cada um dos protocolos oferecem um **conjunto diferente de serviços** para as aplicações.

Aplicação	Perda de dados	Vazão	Sensibilidade ao tempo
Transferência / download de arquivo	Sem perda	Elástica	Não
E-mail	Sem perda	Elástica	Não
Documentos Web	Sem perda	Elástica (alguns kbits/s)	Não
Telefonia via Internet/ videoconferência	Tolerante à perda	Áudio: alguns kbits/s – 1Mbit/s Vídeo: 10 kbits/s – 5 Mbits/s	Sim: décimos de segundo
Áudio/vídeo armazenado	Tolerante à perda	Igual acima	Sim: alguns segundos
Jogos interativos	Tolerante à perda	Poucos kbits/s – 10 kbits/s	Sim: décimos de segundo
Mensagem instantânea	Sem perda	Elástico	Sim e não

- **Orientado a conexão**

- » Cliente e servidor trocam informação de controle antes de encaminhar os dados (*three-way handshake*);
- » Após a apresentação (informação de controle) uma **conexão** TCP é estabelecida;
- » A conexão é *full-duplex*;
- » Ao finalizar a troca de dados, a conexão é finalizada.

- **Entrega confiável**

- » Os dados são encaminhados sem erro e na ordem correta
- » Sem falta de bytes ou bytes duplicados

- **Controle de congestionamento**

- » Limita a capacidade de transmissão de dados para não sobrecarregar a largura de banda



- **Simplificado**
- **Leve**
- **Não orientado a conexão**
  - » O protocolo não oferece garantias que a mensagem chegará ao destino
- **Sem controle de congestionamento**
  - » Pode bombear dados sem controle de taxa

# Serviços de transporte

10

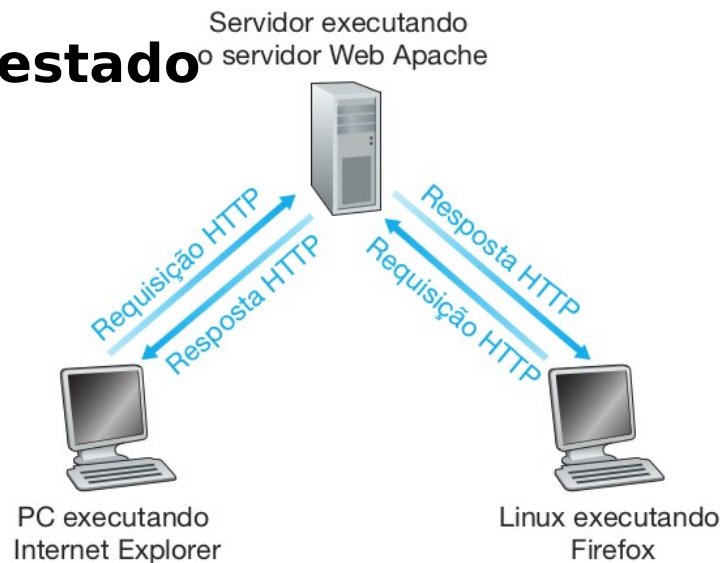
Aplicação	Protocolo de camada de aplicação	Protocolo de transporte subjacente
Correio eletrônico	SMTP [RFC 5321]	TCP
Acesso a terminal remoto	Telnet [RFC 854]	TCP
Web	HTTP [RFC 2616]	TCP
Transferência de arquivos	FTP [RFC 959]	TCP
Multimídia em fluxo contínuo	HTTP (por exemplo, YouTube)	TCP
Telefonia por Internet	SIP [RFC 3261], RTP [RFC 3550] ou proprietária (por exemplo, Skype)	UDP ou TCP

- Define como processos de uma aplicação **trocam mensagens** entre si
- Definem:
  - » Tipos de mensagens trocadas;
  - » Sintaxe das mensagens;
  - » Semântica dos campos;
  - » Regras de quando e como certas mensagens serão enviadas/respondidas.
- Principais protocolos:
  - » HTTP (RFC 2616)
  - » FTP (RFC 959)
  - » SNMP (RFC 5321)
  - » DNS
  - » Bittorrent

- Protocolo de Transferência de Hipertexto (RFC 2616)
- Executado em dois programas: **cliente** e **servidor**;
  - » Trocam mensagens HTTP
- Servidores Web fornecem **páginas HTML** e **objetos Web**
  - » Pode possuir objetos, imagens
- O arquivo HTML referencia outros objetos através de **URL**
  - » <http://www.school.com/departament1/picture.gif>
    - www.school.com: Nome do servidor
    - departament1/picture.gif : Caminho do objeto
- Servidores web
  - » Apache, IIS, nginx

- Protocolo HTTP define como os clientes **requisitam página** aos servidores;
- Usa **TCP** como protocolo de transporte;
- Porta padrão de um servidor Web: **80**;
- HTTP não mantém informações sobre clientes

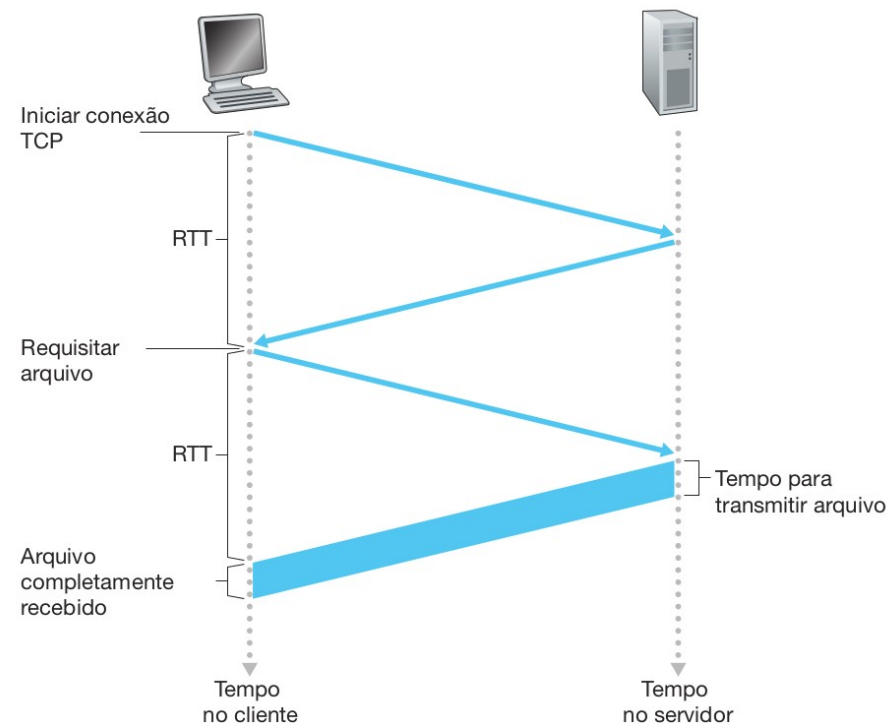
» **Protocolo sem estado**



# HTTP (conexões persistentes)

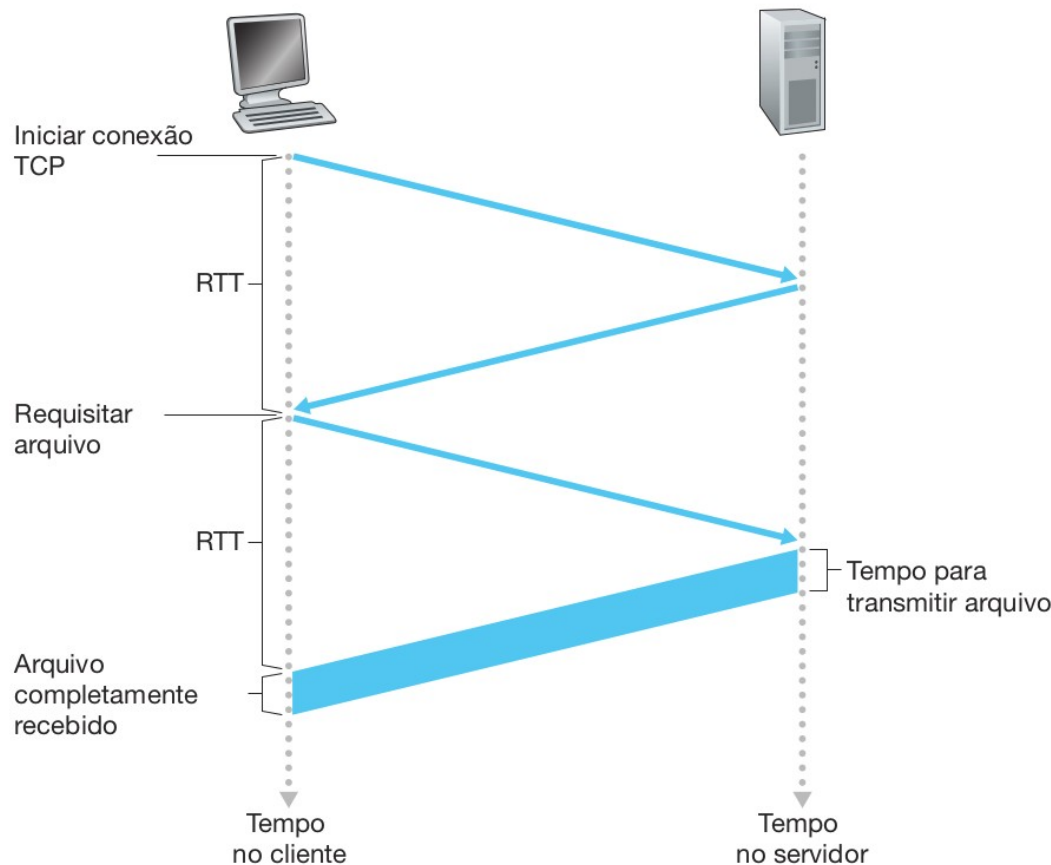
14

- A conexão TCP é **preservada** após a transferência de um arquivo;
- O servidor HTTP **fecha** a conexão após um **intervalo de tempo ocioso**.



# Transferência de uma página HTML

15



- Processo cliente HTTP inicia uma **conexão TCP** para o servidor (`www.someSchool.edu`) na porta 80;
- Conexão estabelecida, cliente envia uma mensagem de **requisição HTTP** de um objeto web (`someDepartment/home.index`);
- Servidor recebe a mensagem, extrai o objeto e o insere em uma **resposta HTTP**.
- Cliente recebe a resposta e indica que recebeu através de uma mensagem **TCP de confirmação** (ack).

<http://www.someSchool.edu/someDepartment/home.index>

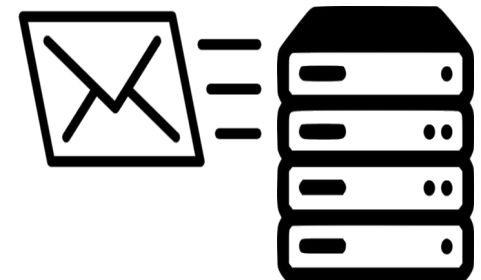
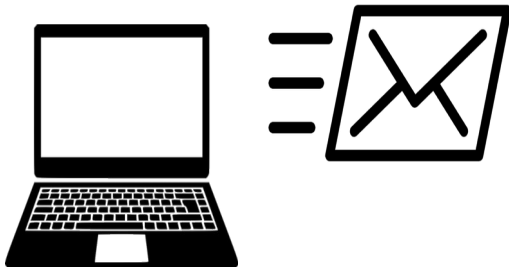
# Realizar conexão TCP

16

Cliente encaminha um pacote TCP com a flag SYN definida com valor 1

Servidor encaminha um pacote TCP com a flag SYN e ACK definidas com valor 1

Servidor encaminha um pacote TCP com a flag ACK definida com valor 1, estabelecendo uma conexão.





# Requisitar arquivo

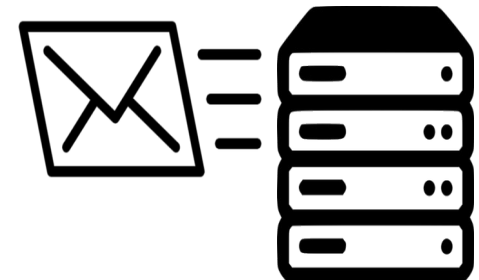
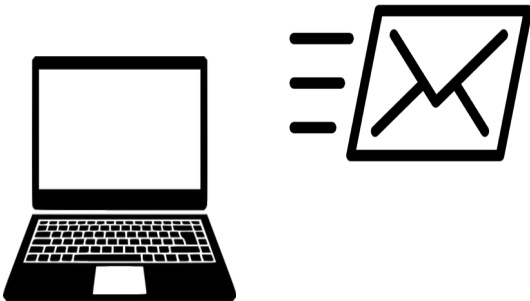
17

Cliente requisita o arquivo *home.index* ao servidor.

Servidor responde com o arquivo solicitado.

Cliente confirma com uma mensagem TCP ACK que recebeu todo o arquivo.

```
GET /home.index HTTP 1.1  
Host: www.someschool.edu  
Connection: keep-alive  
User-agent: Mozilla/5.0  
Accept-language:pt
```



- **Requisição HTTP**

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

- **Linha de requisição**

- » Método: Ação a realizar (GET, POST, HEAD, PUT, DELETE)
- » Objeto requisitado: /somedir/page.html
- » Versão do HTTP: http/1.1

- **Requisição HTTP**

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



**Linhas de cabeçalho**

- **Linhas de cabeçalho**

- » Host: endereço do servidor
- » Connection: persistente(*keep-alive*), não persistente (*closed*)
- » User-agent: browser/cliente
- » Accept-language: Linguagem preferencial (se houver) do conteúdo

- **Resposta HTTP**

```
HTTP/1.1 200 OK } Linha de estado
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 ...)

- **Linha de estado**

- » Versão: Versão do protocolo HTTP
- » Código de estado: Código de resposta da requisição HTTP
- » Frase: mensagem correspondente ao código de estado

- Agrupados em 5 classes (RFC 2616):
  - » **100-199**: Respostas de informação
  - » **200-299**: Respostas de sucesso
  - » **300-399**: Redirecionamentos
  - » **400-499**: Erros do cliente
  - » **500-599**: Erros do servidor
- Acessíveis através deste [link](#).

# Respostas de sucesso (200-299)

---

22

- **200 OK:** Requisição bem sucedida
- **201 Created:** Requisição bem sucedida e novo recurso criado.
- **202 Accepted:** Nenhuma ação foi tomada sobre a requisição
- **204 No Content:** Não existe um conteúdo para ser encaminhado ao cliente.
- **206 Partial Content:** Separa o conteúdo em vários fluxos.

- **301 Moved Permanently**: Objeto foi movido para outra URL (Definido no cabeçalho *Location* da mensagem de resposta)

# Resposta de erro do cliente (400-499)

---

24

- **400 Bad Request**: Erro genérico. Requisição do cliente não está como esperado pelo servidor.
- **401 Unauthorized**: O cliente não está autenticado.
- **403 Forbidden**: O cliente não tem direitos de acesso ao conteúdo (autenticado).
- **404 Not Found**: Recurso requisitado não existe no servidor.
- **405 Method Not Allowed**: Método utilizado pelo cliente não é permitido (PUT, POST, DELETE...)



# Resposta de erro do servidor (500-599)

---

25

- **500 Internal Server Error**: Erro genérico.
- **502 Bad Gateway**: Erro de comunicação com *gateway* do servidor.
- **503 Service Unavailable**: O servidor está sobrecarregado e não pode atender a requisição. Cabeçalho *Retry-After* deverá dizer o tempo para usuário retentar.
- **505 HTTP Version Not Supported**: Versão do protocolo HTTP não suportada pelo servidor.

- **Resposta HTTP**

```
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
```

**Linhas de cabeçalho**

(dados dados dados dados dados ...)

- **Linhas de cabeçalho**

- » Date: Data e hora que a resposta HTTP foi criada no servidor;
- » Server: Informações sobre o servidor;
- » Last-Modified: quando o objeto foi modificado (usado para cache)
- » Content-Length: Tamanho em bytes dos dados;
- » Content-Type: Tipo presente no corpo da resposta.

- Capítulo 2 do Livro do Kurose.



**Dúvidas?**