



# Características dos Projetos em Camadas

Professor  
Wagner Gadêa Lorenz  
wagnerglorenz@gmail.com

Disciplina: Redes de Computadores I  
Curso de Sistemas de Informação



---

# Modelo OSI e TCP/IP

---

O modelo de referência **Open Systems Interconnection (OSI)** é uma representação abstrata em camadas criado como diretriz para o design de protocolos de rede.

O modelo OSI divide o processo de redes em sete camadas lógicas, cada uma com funcionalidades exclusivas e com serviços e protocolos específicos atribuídos.



---

# Modelo OSI e TCP/IP

---

Neste modelo, as informações são passadas de uma camada para a outra, começando na camada de **Aplicação** no host transmissor, continuando hierarquia abaixo, até a **camada Física**, passando para o **canal de comunicações** até o **host de destino**, onde a informação retorna hierarquia acima, terminando na camada de **Aplicação**.

# Modelo OSI e TCP/IP

Modelo OSI

7. Aplicação

6. Apresentação

5. Sessão

4. Transporte

3. Rede

2. Enlace de Dados

1. Física

Camadas de  
Aplicativos

Camadas de  
Fluxo de  
Dados

Modelo TCP/IP

Aplicação

Transporte

Internet

Acesso à  
Rede

DNS (Sistema de  
Resolução de  
domínios)

HTTP (Protocolo  
de Transferência  
de Hipertexto)

SMTP (Protocolo  
de Transferência  
de Correio  
Simples)  
POP (Protocolo dos  
Correios)

DHCP (Protocolo  
de Configuração  
Dinâmica de Host)



---

# Camada de Aplicação

---

**A camada de Aplicação, a número sete, é a camada superior dos modelos OSI e TCP/IP.**

É a camada que fornece a interface entre as aplicações que utilizamos para comunicação e a rede subjacente pela qual nossas mensagens são transmitidas.



---

# Camada de Aplicação

---

**Os protocolos da camada de Aplicação são utilizados para troca de dados entre programas executados nos hosts de origem e de destino.**

Há muitos protocolos da camada de Aplicação, e outros novos estão em constante desenvolvimento.



---

# Camada de Aplicação

---

Embora o conjunto de protocolos TCP/IP tenha sido desenvolvido antes da definição do modelo OSI, a **funcionalidade dos protocolos da camada de aplicação TCP/IP se ajusta à estrutura das três camadas superiores do modelo OSI: camadas de Aplicação, Apresentação e Sessão.**



---

# Camada de Aplicação

---

A maioria dos protocolos da camada de Aplicação TCP/IP foi **desenvolvida antes do surgimento de computadores pessoais, interfaces gráficas de usuário e objetos multimídia.**

Como resultado, esses protocolos implementam muito pouco da funcionalidade especificada nas camadas de **Apresentação e Sessão** do modelo OSI.



---

# Camada de Apresentação

---

A camada de **Apresentação** tem **três** funções **principais**:

- Codificação e conversão de dados da camada de Aplicação para garantir que os dados do dispositivo de origem possam ser interpretados pela aplicação adequada no dispositivo de destino.
- Compressão dos dados de forma que eles possam ser descomprimidos pelo dispositivo de destino.
- Criptografia dos dados para transmissão e decodificação de dados quando o destino os recebe.



---

# Camada de Apresentação

---

**As implementações da camada de Apresentação não são associadas normalmente a uma pilha de protocolos em particular. Os padrões para vídeo e gráficos são exemplos.**

Alguns padrões conhecidos para vídeo incluem QuickTime e Motion Picture Experts Group (MPEG).

QuickTime é uma especificação da Apple Computer para vídeo e áudio e MPEG é um padrão para compressão e codificação de vídeo.



---

# Camada de Apresentação

---

Entre os formatos de imagens gráficas conhecidos, há Graphics Interchange Format (GIF), Joint Photographic Experts Group (JPEG) e Tagged Image File Format (TIFF).

GIF e JPEG são padrões de compressão e codificação para imagens gráficas, e TIFF é um formato de codificação padrão para imagens gráficas.



---

# Camada de Sessão

---

**Como o próprio nome diz, as funções na camada de Sessão criam e mantêm diálogos entre as aplicações de origem e destino.**

**A camada de Sessão lida com a troca de informações para iniciar diálogos, mantê-los ativos e reiniciar sessões interrompidas ou ociosas por um longo período.**



---

# Modelo OSI e TCP/IP

---

**A maioria das aplicações, como navegadores Web ou clientes de e-mail, incorpora a funcionalidade das camadas OSI 5, 6 e 7. (Sessão, Apresentação e Aplicação)**



---

# Modelo OSI e TCP/IP

---

Os protocolos da camada de **Aplicação TCP/IP** mais conhecidos são aqueles que fornecem a troca de informações de usuários.

Esses protocolos especificam as informações de formato e controle necessários para muitas funções comuns de comunicação na Internet.



---

# Modelo OSI e TCP/IP

---

**Entre esses protocolos TCP/IP, há:**

- **O Protocolo de Serviço de Nome de Domínio** (Domain Name Service Protocol (**DNS**)) é utilizado para resolver nomes a endereços IP.
- **O Protocolo de Transferência de Hipertexto** (Hypertext Transfer Protocol (**HTTP**)) é utilizado para transferir arquivos que compõem as páginas Web da World Wide Web.



---

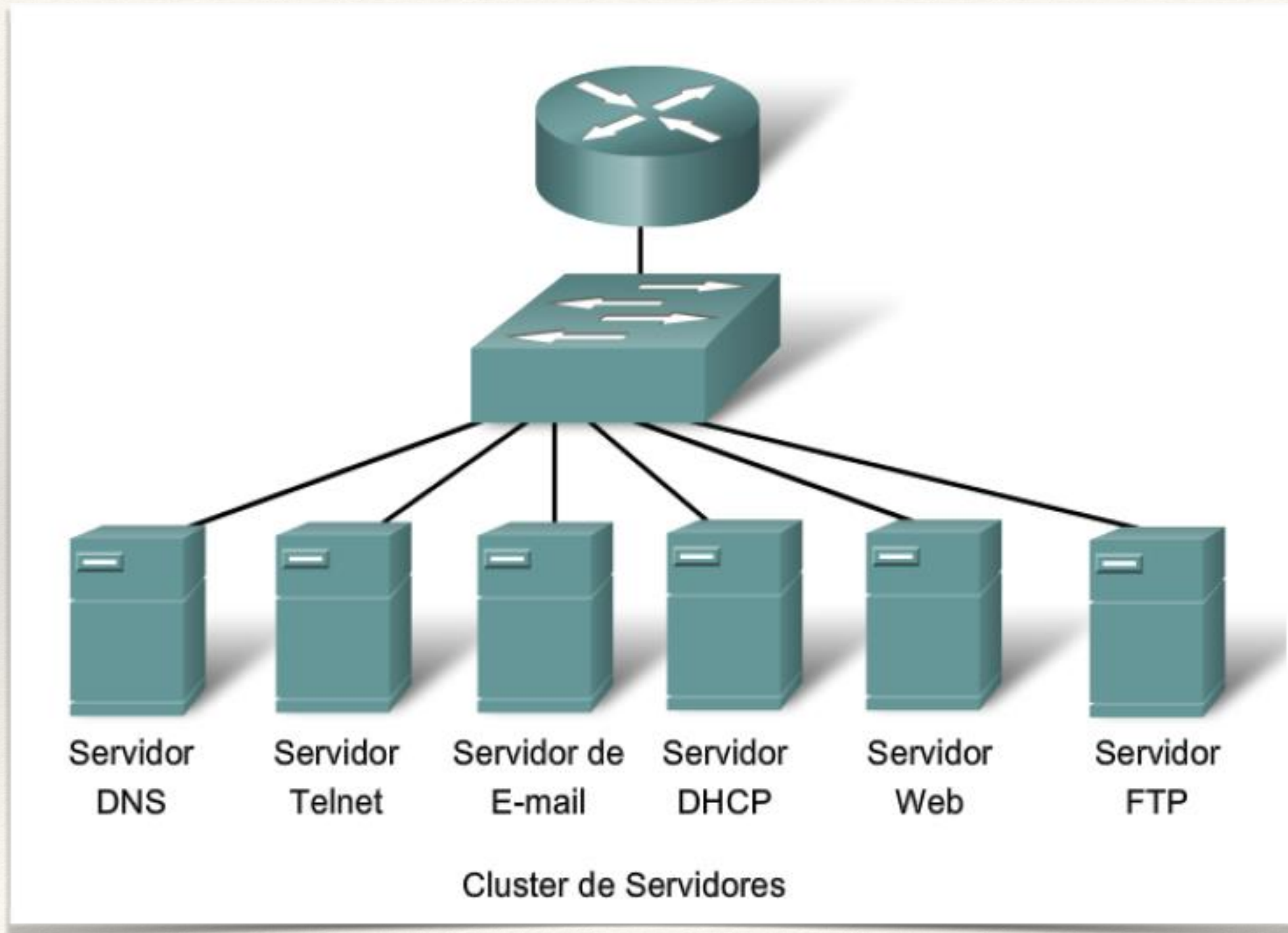
# Modelo OSI e TCP/IP

---

- O **Protocolo SMTP** é utilizado para transferência de mensagens e anexos de e-mail.
- **Telnet**, um protocolo de simulação de terminal, é utilizado para fornecer acesso remoto a servidores e dispositivos de rede.
- O **Protocolo de Transferência de Arquivos** (File Transfer Protocol (FTP)) é utilizado para transferência interativa de arquivos entre sistemas.



# Modelo OSI e TCP/IP





---

# Modelo OSI e TCP/IP

---

- **Servidor DNS (DNS)**

- Serviço que fornece o endereço IP de um site ou nome de domínio para que um host possa se conectar a ele.

- **Serviço Telnet**

- Serviço que permite que os administradores se conectem a um host em uma localização remota e controlem o host como se eles estivessem conectados localmente.

- **Servidor de e-Mail**

- Usa o protocolo SMTP, o protocolo POP3 ou protocolo MAP
- Usados para enviar mensagens de e-mail de clientes para servidores através da Internet
- Destinatários são especificados usando o formato usuário@domínio



---

# Modelo OSI e TCP/IP

---

- **Servidor DHCP**

- Serviço que designa gateway padrão da máscara de sub-rede do endereço IP e outras informações aos clientes

- **Servidor Web**

- Protocolo HTTP
- Usado para transmitir informações entre clientes web e servidores web
- A maioria das páginas web estão utilizando HTTP

- **Servidor FTP (File Transfer Protocol)**

- Serviço que permite o download e o upload de arquivos entre cliente e servidor.



---

# Software da Camada de Aplicação

---

As funções associadas aos **protocolos da camada de Aplicação** permitem que nossa rede humana faça interface com a rede de dados subjacente.

Quando abrimos um navegador Web ou uma janela de mensagem instantânea, uma aplicação é iniciada e o programa é colocado na memória do dispositivo quando executado.

Cada programa em execução carregado em um dispositivo é mencionado como um processo.



---

# Software da Camada de Aplicação

---

Dentro da camada de Aplicação, há duas formas de programa de software ou processos que fornecem acesso à rede: **aplicações e serviços**.



---

# Software da Camada de Aplicação

---

## **Aplicações que Detectam Redes**

Aplicações são programas de softwares usados por pessoas para se comunicarem pela rede.

Algumas aplicações de usuário final detectam redes, o que significa que elas implementam os protocolos da camada de Aplicação e conseguem se comunicar diretamente com as camadas inferiores da pilha de protocolos.

**Clientes de e-mail e navegadores Web são exemplos desses tipos de aplicação.**



---

# Software da Camada de Aplicação

---

## **Serviços da camada de Aplicação**

Outros programas podem precisar da assistência dos serviços da camada de Aplicação para utilizar recursos de rede, como transferência de arquivos ou spooling de impressão em rede.

**Embora transparentes ao usuário, esses serviços são os programas que fazem interface com a rede e preparam os dados para transferência.**

Diferentes tipos de dados - seja texto, gráficos ou vídeo - exigem serviços de rede diferentes para garantir que sejam preparados adequadamente para processamento pelas funções que ocorrem nas camadas inferiores do modelo OSI.



---

# Software da Camada de Aplicação

---

Cada aplicação ou serviço de rede utiliza protocolos que definem os padrões e formatos de dados a serem utilizados.

**Sem protocolos, a rede de dados não teria uma maneira comum de formatar e direcioná-los.**



---

# Aplicações de usuários, serviços e protocolos da camada de aplicação

---

Como já mencionado, a **camada de Aplicação** utiliza protocolos implementados dentro de **aplicações e serviços**.

Enquanto as aplicações oferecem uma maneira de criar mensagens e serviços da camada de aplicação estabelecem uma interface com a rede, **os protocolos fornecem as regras e formatos que regem como os dados são tratados**.

Todos os três componentes podem ser utilizados por um único programa executável e até mesmo usar o mesmo nome.

Por exemplo, "Telnet", podemos nos referir à aplicação, ao serviço ou ao protocolo.



---

# Aplicações de usuários, serviços e protocolos da camada de aplicação

---

No **modelo OSI**, as aplicações que interagem diretamente com pessoas são considerados como estando no topo da pilha, assim como as próprias pessoas.

**Como todas as camadas dentro do modelo OSI, a camada de Aplicação se fia nas funções nas camadas inferiores para completar o processo de comunicação.**



---

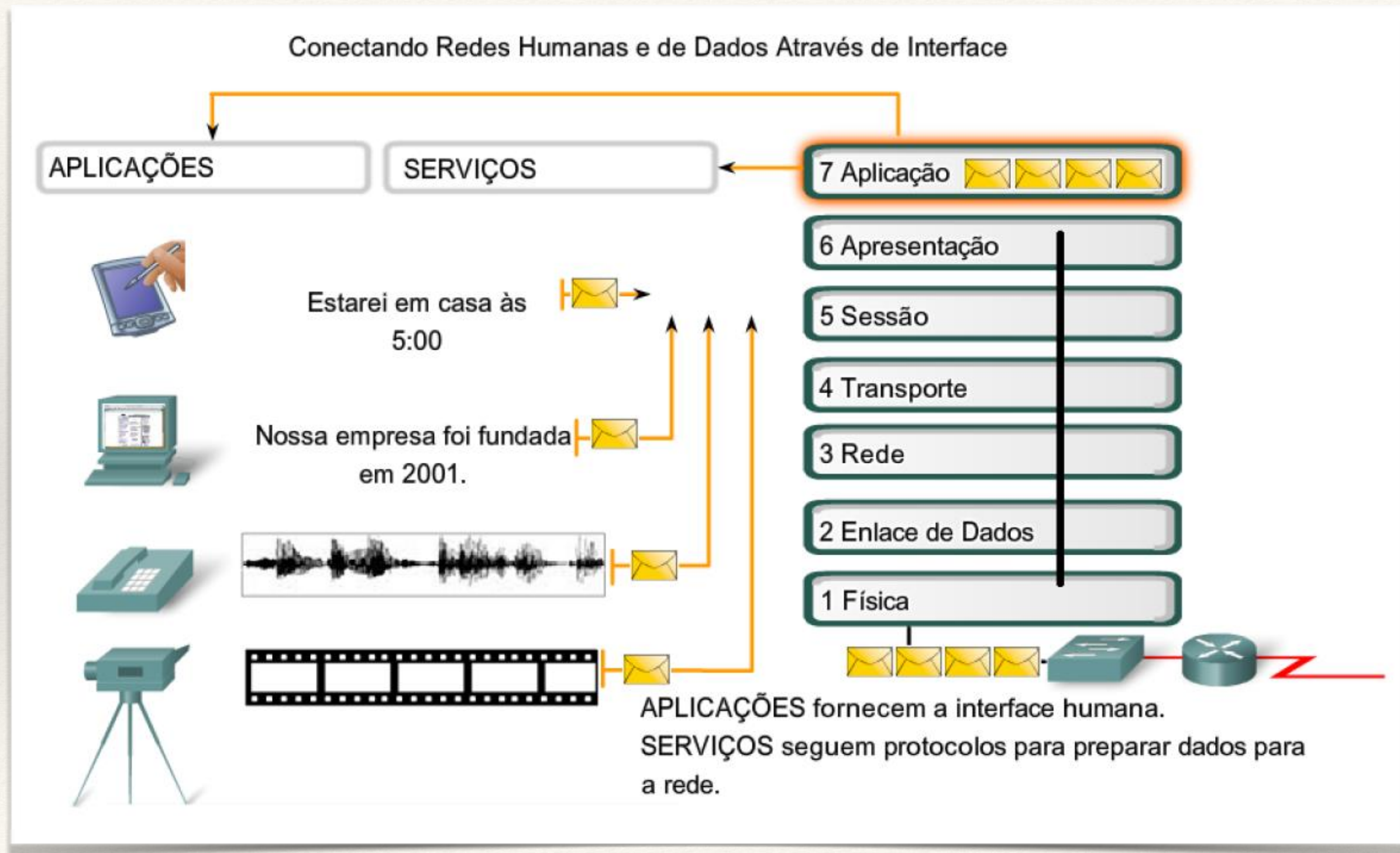
# Aplicações de usuários, serviços e protocolos da camada de aplicação

---

Dentro da camada de **Aplicação**, os protocolos especificam que mensagens são trocadas entre os **hosts** de **origem** e **destino**, a ***sintaxe*** dos comandos de controle, o **tipo** e **formato** dos **dados** sendo transmitidos e os métodos adequados para notificação de erros e recuperação.



# Aplicações de usuários, serviços e protocolos da camada de aplicação





---

# Função de Protocolo da camada de aplicação

---

Os **protocolos da camada de Aplicação** são utilizados pelos dispositivos de **origem** e **destino** durante uma **sessão de comunicação**.

Para que a comunicação tenha sucesso, os protocolos da camada de aplicação implementados nos hosts de origem e destino devem corresponder.



---

# Função de Protocolo da camada de aplicação

---

Os **protocolos** estabelecem **regras coerentes** para **troca de dados** entre **aplicações** e **serviços** carregados nos dispositivos participantes.

Os **protocolos** especificam como os **dados** dentro das **mensagens** são **estruturados** e os **tipos** de **mensagens** enviados entre **origem** e **destino**.

Tais mensagens podem ser solicitações de serviço, confirmações, mensagens de dados, de status ou de erro.



---

# Função de Protocolo da camada de aplicação

---

Os **protocolos** também definem **diálogos** de mensagem, garantindo que uma mensagem enviada seja **conhecida** pela **resposta esperada** e que os **serviços corretos** sejam **chamados** quando houver transferência de dados.



---

# Função de Protocolo da camada de aplicação

---

Muitos tipos diferentes de aplicações se comunicam via redes de dados.

Portanto, os serviços da camada de Aplicação devem implementar vários protocolos para fornecer a gama desejada de experiências de comunicação.



---

# Função de Protocolo da camada de aplicação

---

**Cada protocolo tem uma finalidade específica e contém as características necessárias para atender a tal finalidade.**

**Os detalhes do protocolo correto em cada camada devem ser seguidos para que as funções em uma camada façam interface adequadamente com os serviços na camada inferior.**



---

# Função de Protocolo da camada de aplicação

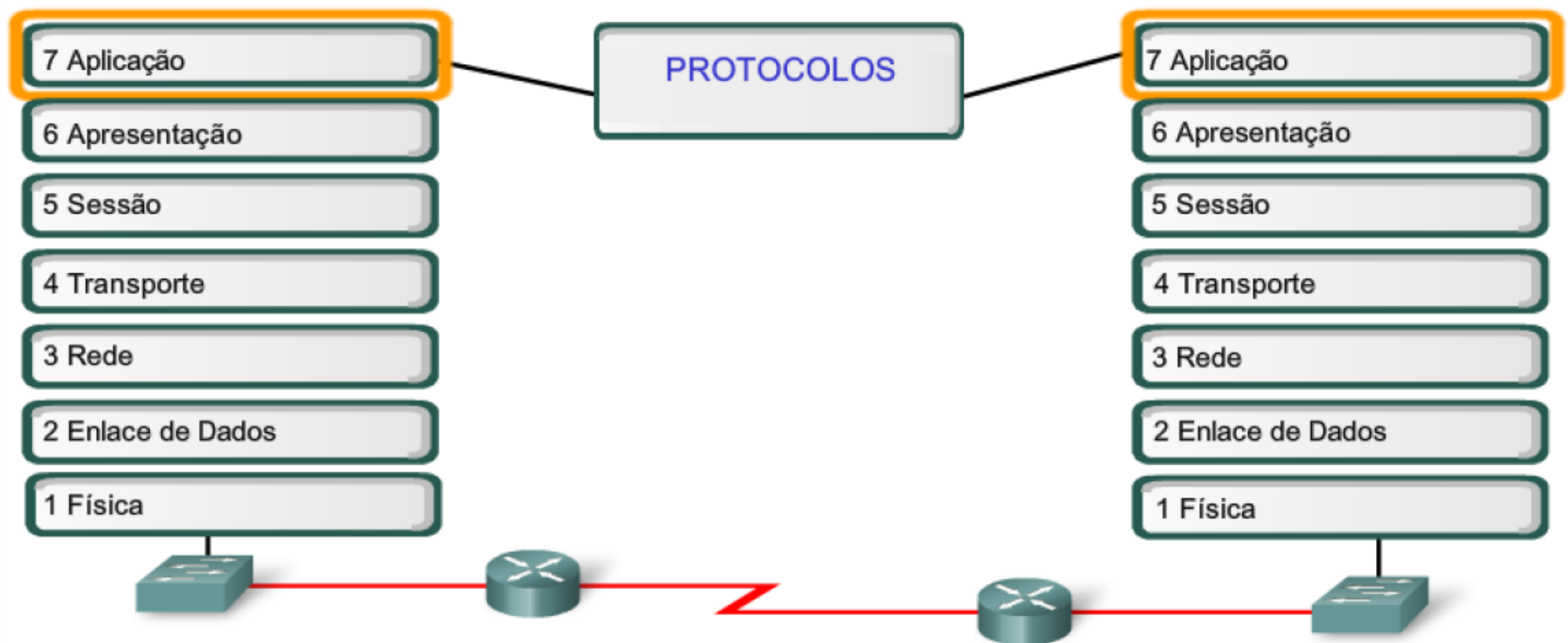
---

**Aplicações e serviços** também podem **utilizar diversos protocolos** no decorrer de **uma única conversa**.

Um **protocolo** pode **especificar** como **estabelecer** a **conexão** de rede e outro, descrever o **processo** para **transferência de dados** quando a mensagem passa para a camada imediatamente abaixo.



# Função de Protocolo da camada de aplicação



Os protocolos da camada de Aplicações fornecem as regras para a comunicação entre os Aplicações.



---

# Função de Protocolo da camada de aplicação

---

## **Protocolos:**

- Definem processos na outra extremidade da comunicação;
- Definem os tipos de mensagens;
- Definem a sintaxe das mensagens;
- Definem o significado de todos os campos informacionais;
- Definem como as mensagens são enviadas e resposta esperada;
- Definem a interação com a próxima camada mais baixa.



---

# Modelo cliente-servidor

---

Quando as pessoas tentam acessar informações em seus dispositivos, seja um PC, laptop, PDA, celular ou outro dispositivo conectado a uma rede, os dados podem não estar fisicamente armazenados neles.

**Se este for o caso, uma solicitação para acessar tais informações deve ser feita ao dispositivo onde os dados estão.**



---

# Modelo cliente-servidor

---

## O modelo Cliente/Servidor

No modelo *cliente/servidor*, o dispositivo que **solicita** as informações é chamado de **cliente**, e o que **responde** à solicitação é chamado de **servidor**.

Os processos de cliente e servidor são considerados como estando na camada de **Aplicação**.



---

# Modelo cliente-servidor

---

O **cliente** começa o **intercâmbio** ao **solicitar** dados do servidor, que responde enviando uma ou mais sequências de dados ao cliente.

Os protocolos da camada de **Aplicação** **descrevem o formato das solicitações e respostas entre clientes e servidores.**

Além da transferência real de dados, esse intercâmbio também pode exigir informações de controle, como autenticação de usuário e identificação de um arquivo de dados a ser transferido.



---

# Modelo cliente-servidor

---

Um **exemplo** de rede **cliente/servidor** é um ambiente corporativo no qual os funcionários utilizam um servidor de e-mail da empresa para enviar, receber e armazenar e-mails.

O cliente de e-mail no computador de um funcionário envia uma solicitação ao servidor de e-mail para qual-quer correspondência não lida.

O servidor responde enviando o e-mail solicitado ao cliente.



---

# Modelo cliente-servidor

---

Embora os dados sejam normalmente descritos como fluindo do servidor ao cliente, alguns sempre vão do cliente ao servidor.

O fluxo de dados pode ser igual em ambas as direções, ou mesmo maior na direção do cliente ao servidor.



---

# Modelo cliente-servidor

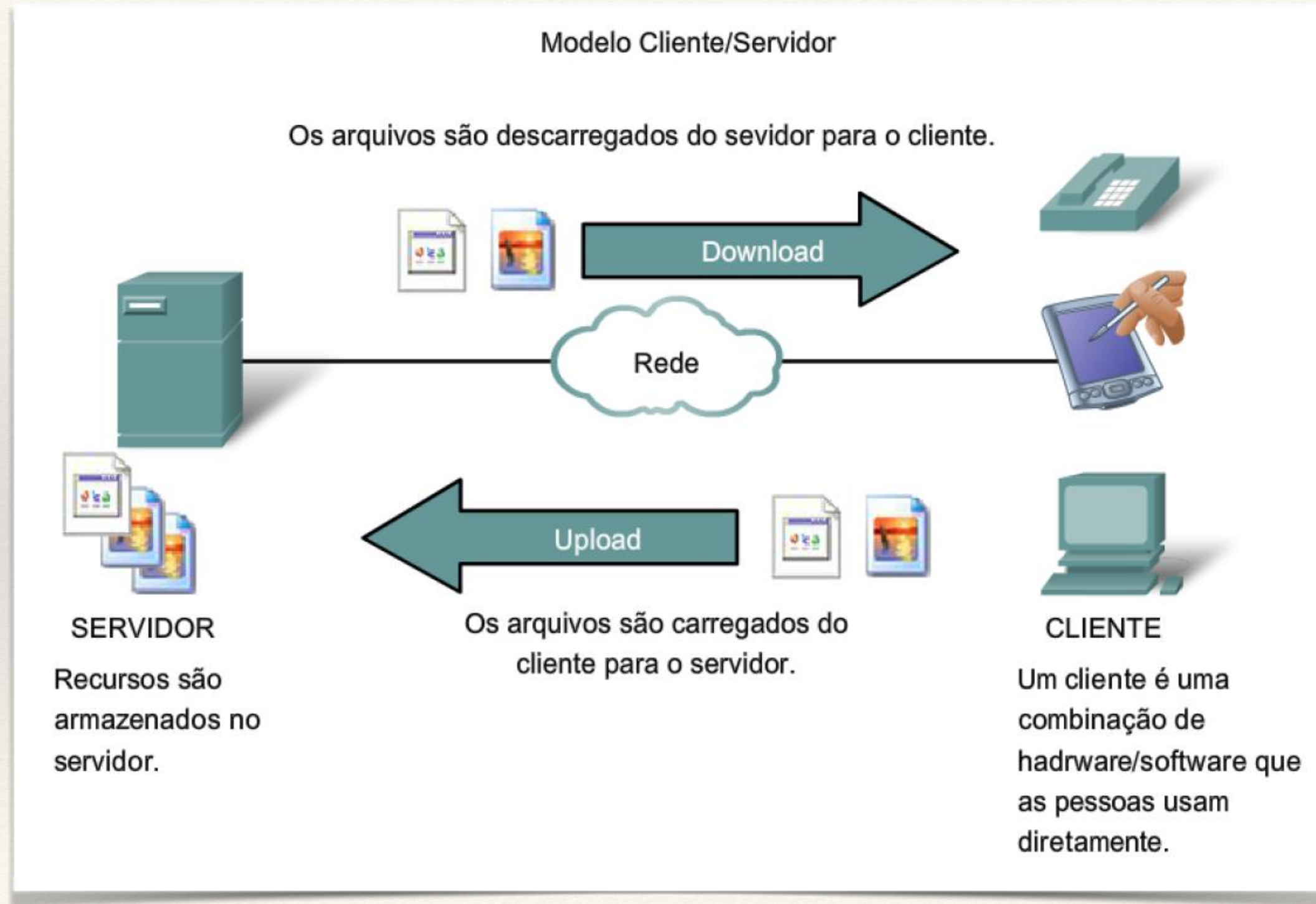
---

Por **exemplo**, um cliente pode transferir um arquivo ao servidor para armazenamento.

A transferência de dados de um cliente para um servidor é mencionada como um upload, e de um servidor para um cliente, como um download.



# Modelo cliente-servidor





---

# Servidores

---

Em um contexto geral de rede, **qualquer dispositivo que responda a solicitações de aplicações de clientes funciona como um servidor.**

Um **servidor** normalmente é um computador que contém informações a serem compartilhadas com muitos sistemas cliente.



---

# Servidores

---

Por **exemplo**, páginas Web, documentos, bancos de dados, imagens e arquivos de áudio e vídeo podem ser armazenados em um servidor e entregues aos clientes solicitantes.

Em outros casos, como uma impressora em rede, o servidor de impressão fornece as solicitações de impressão do cliente à impressora especificada.



---

# Servidores

---

Diferentes tipos de aplicações de servidor podem ter exigências diferentes para acesso a clientes.

Alguns servidores podem exigir **autenticação** de **informações** da **conta** do **usuário** para verificar se este tem **permissão** de **acesso** aos **dados solicitados** ou de **uso** de uma **operação** em particular.



---

# Servidores

---

Tais servidores se fiam em uma lista **central de contas de usuário e às autorizações**, ou **permissões** (para acesso a dados e operações), concedidas a cada usuário.

Ao utilizar um cliente FTP, por exemplo, se você solicitar um upload de dados ao servidor FTP, poderá ter permissão para gravar em sua pasta individual, mas não para ler outros arquivos no site X.



---

# Servidores

---

Em uma **rede cliente/servidor**, o servidor executa um serviço, ou processo, às vezes chamado de ***daemon*** de **servidor**.

Como a maioria dos serviços, daemons normalmente são executados em segundo plano e não estão sob o controle direto de um usuário final.



---

# Servidores

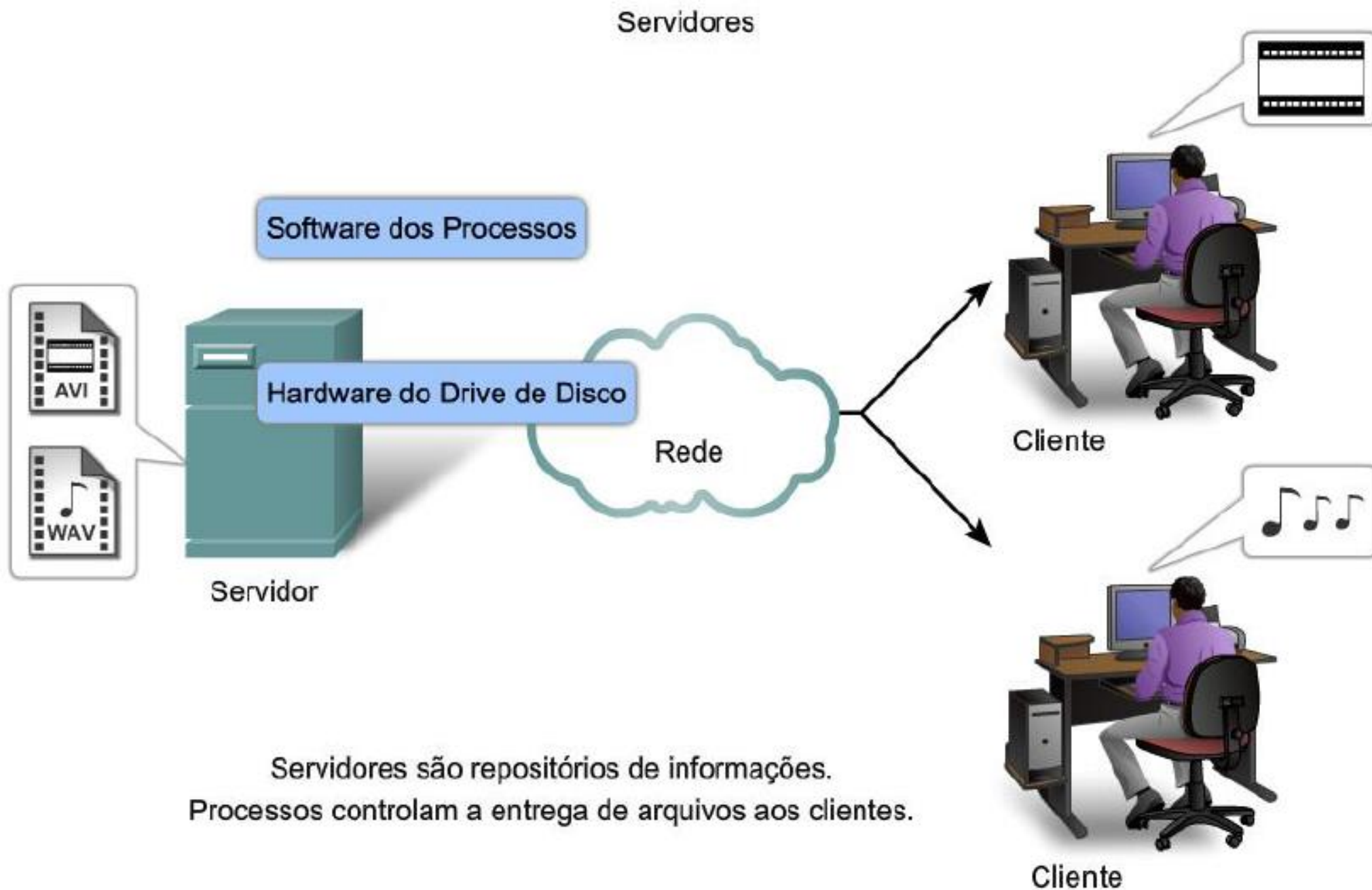
---

**Daemons** são descritos como "**ouvintes**" de uma solicitação de um cliente, porque são programados para responder sempre que o servidor recebe uma solicitação para o serviço fornecido pelo daemon.

**Quando um daemon "ouve" uma solicitação de um cliente, troca as mensagens correspondentes com o cliente, como exigido por seu protocolo, e envia os dados solicitados ao cliente no formato adequado.**



# Servidores





---

# Protocolos e Serviços da Camada de Aplicação

---

Uma única aplicação pode utilizar **vários serviços** diferentes de suporte da camada de Aplicação.

Assim, o que parece para o usuário como uma solicitação para página Web pode totalizar, na verdade, dezenas de solicitações individuais.



---

# Protocolos e Serviços da Camada de Aplicação

---

Além disso, para cada solicitação, diversos processos podem ser executados.

Por **exemplo**, um cliente pode exigir diversos processos individuais para formular apenas uma solicitação a um servidor.



---

# Protocolos e Serviços da Camada de Aplicação

---

Além disso, os servidores normalmente têm diversos clientes solicitando informações ao mesmo tempo.

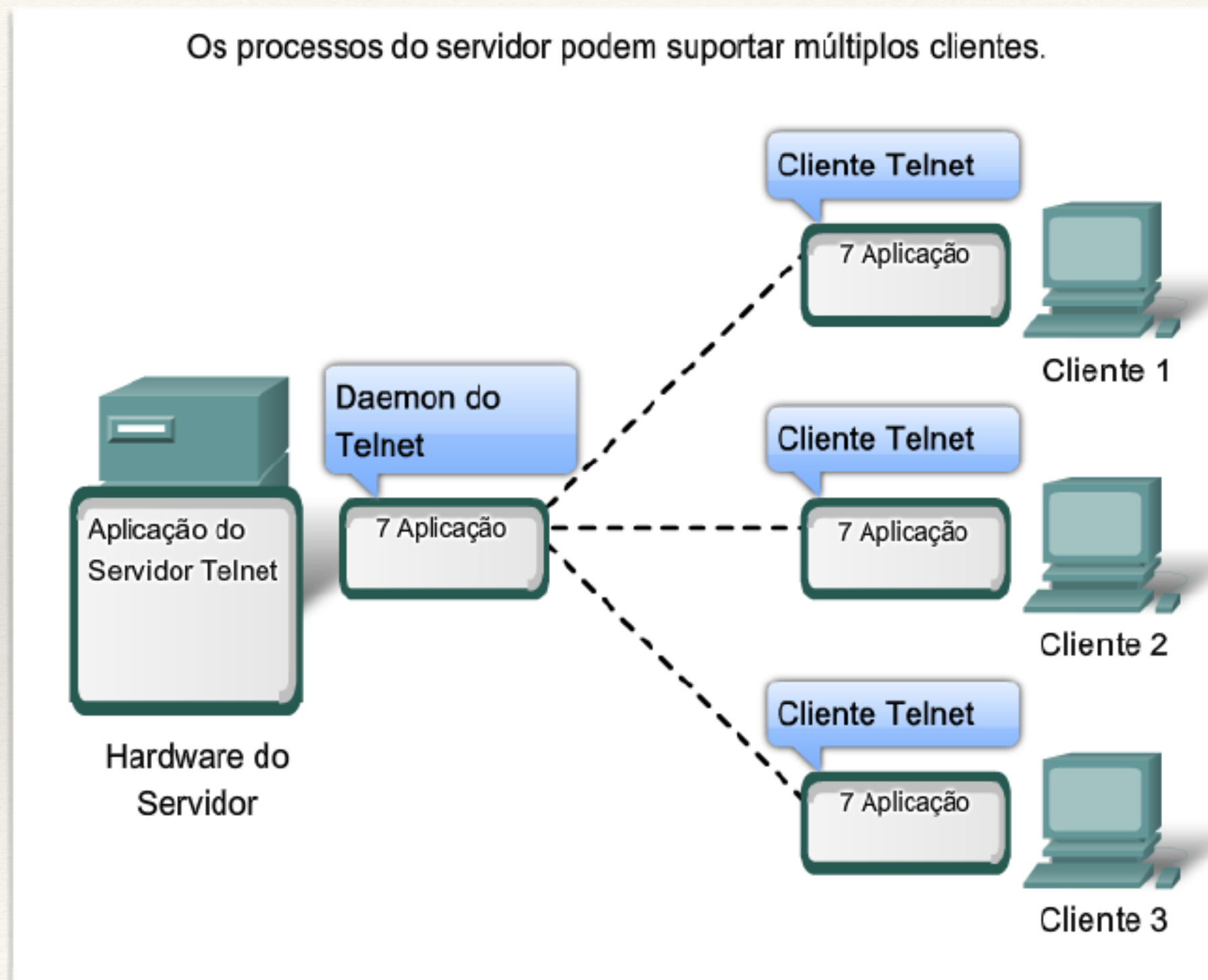
Por **exemplo**, um servidor Telnet pode ter muitos clientes solicitando conexões a ele.

**Tais solicitações individuais devem ser tratadas simultaneamente e separadamente para que a rede tenha sucesso.**

Os processos e serviços da camada de Aplicação se fiam no suporte de funções da camada inferior para gerenciar com sucesso as diferentes conversas.



# Protocolos e Serviços da Camada de Aplicação





---

# Redes e Aplicações Não-Hierárquicas (P2P)

---

## O Modelo Não-Hierárquico

Além do modelo cliente/servidor para a rede, também há o **modelo não-hierárquico**.

A rede **não-hierárquica** envolve duas formas diferentes: **design de rede não-hierárquica** e **aplicações não-hierárquicas (P2P)**.

A duas formas têm características semelhantes, mas, na prática, funcionam de maneira bastante diferente.



---

# Redes e Aplicações Não-Hierárquicas (P2P)

---

## Redes Não-Hierárquicas

Em uma **rede não-hierárquica**, **dois ou mais** computadores são **conectados** via **rede** e podem **compartilhar recursos** (como impressoras e arquivos) sem ter um servidor dedicado.

Cada dispositivo final conectado (conhecido como par (**peer**)) pode funcionar **como cliente ou servidor**. Um computador pode assumir o papel de servidor para uma transação ao mesmo tempo em que é o cliente de outra.

As funções de cliente e servidor são definidas de acordo com a solicitação.



---

# Redes e Aplicações Não-Hierárquicas (P2P)

---

**Uma rede residencial simples, com dois computadores conectados e compartilhando uma impressora, é um exemplo de rede não-hierárquica.**

Cada pessoa pode configurar o seu computador para compartilhar arquivos, possibilitar jogos em rede ou compartilhar uma conexão à Internet.



---

# Redes e Aplicações Não-Hierárquicas (P2P)

---

Outro **exemplo** de funcionalidade de rede **não-hierárquica** é ter dois computadores conectados a uma rede ampla que utilizam aplicações de software para compartilhar recursos entre si através da rede.



---

# Redes e Aplicações Não-Hierárquicas (P2P)

---

**Diferentemente do modelo cliente/servidor, que utiliza servidores dedicados, as redes não-hierárquicas descentralizam os recursos em uma rede.**

Em vez de localizar informações para compartilhar em servidores dedicados, as informações podem ser encontradas em qualquer lugar em um dispositivo conectado.

A maioria dos sistemas operacionais atuais suporta compartilhamento de arquivos e impressão sem exigir um software de servidor adicional.



---

# Redes e Aplicações Não-Hierárquicas (P2P)

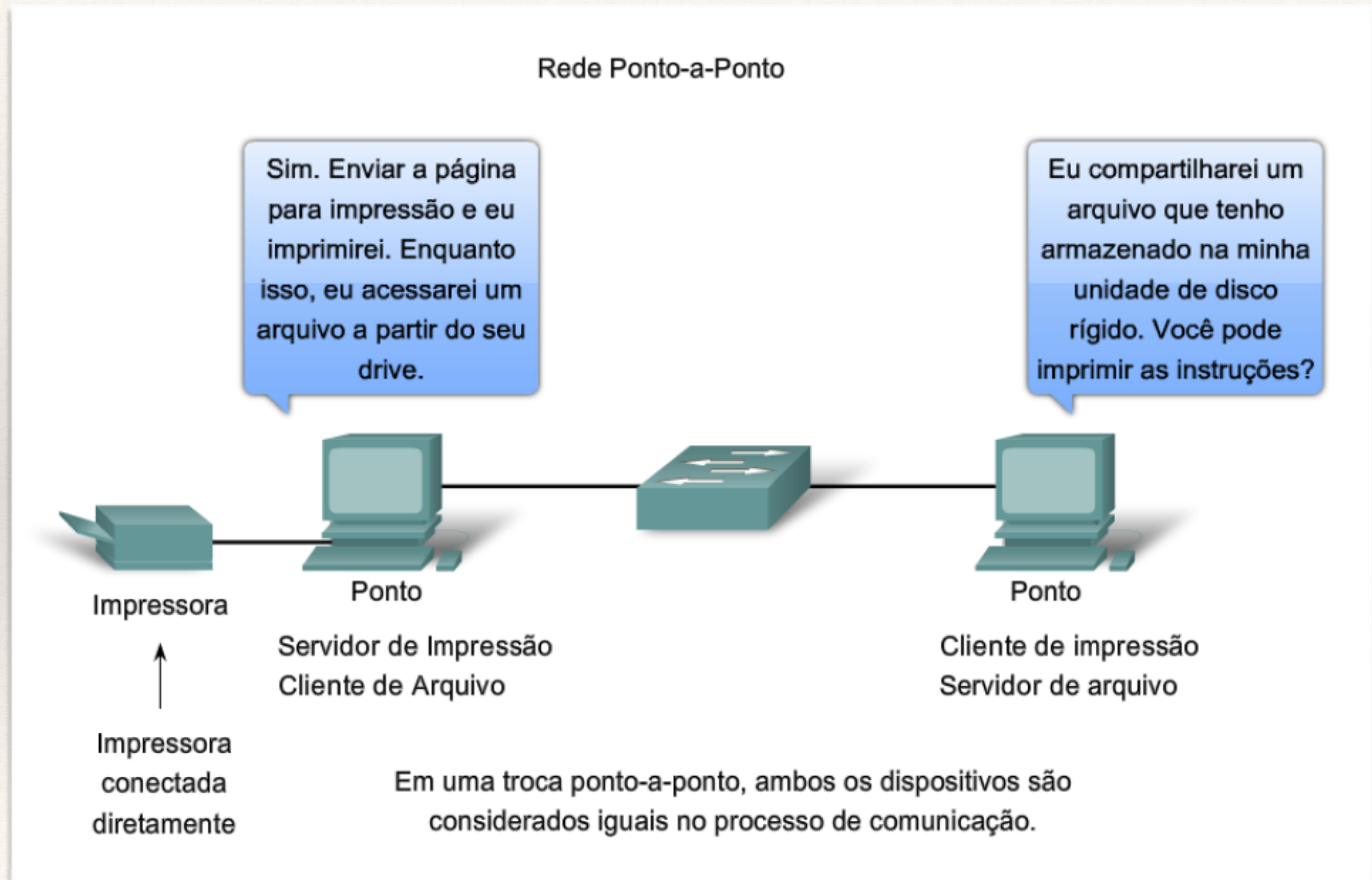
---

**Como redes não-hierárquicas normalmente não utilizam contas de usuário centralizadas, permissões ou monitoramento, é difícil executar políticas de segurança e acesso a redes com mais de alguns poucos computadores.**

**As contas de usuário e direitos de acesso devem ser definidos individualmente em cada dispositivo.**



# Redes e Aplicações Não-Hierárquicas (P2P)





---

# Redes e Aplicações Não-Hierárquicas (P2P)

---

## Aplicações Não-Hierárquicas

Uma **aplicação não-hierárquica (P2P)**, diferentemente de uma rede hierárquica, **permite que um dispositivo aja como cliente e servidor na mesma comunicação.**

**Neste modelo, cada cliente é um servidor e cada servidor é um cliente.**



---

# Redes e Aplicações Não-Hierárquicas (P2P)

---

Ambos podem iniciar uma comunicação e são considerados iguais no processo de comunicação.

No entanto, aplicações não-hierárquicas exigem que cada **dispositivo final forneça uma interface de usuário e execute um serviço de segundo plano.**

Ao iniciar uma aplicação não-hierárquica específica, ela chama a interface de usuário e serviços de segundo plano exigidos. Depois disso, os dispositivos podem se comunicar diretamente.



---

# Redes e Aplicações Não-Hierárquicas (P2P)

---

Algumas **aplicações P2P** utilizam um **sistema híbrido** no qual o **compartilhamento de recursos é descentralizado**, mas os **índices que apontam para as localizações de recursos são armazenados em um diretório centralizado**.

Em um sistema híbrido, cada par (**peer**) acessa um servidor de índice para obter a localização de um recurso armazenado em outro par (peer).

O servidor de índice também pode ajudar a conectar dois pares, mas quando conectado, a comunicação ocorre entre os dois pares sem comunicação adicional ao servidor de índice.



---

# Redes e Aplicações Não-Hierárquicas (P2P)

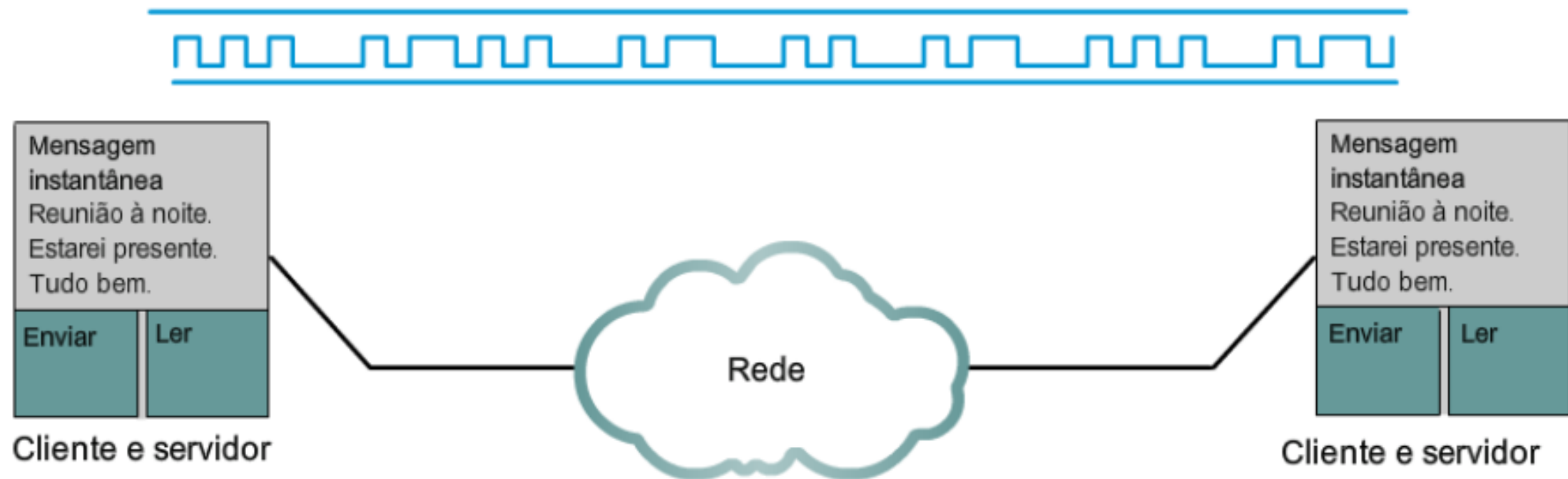
---

Aplicações não-hierárquicas podem ser utilizadas em redes não-hierárquicas, redes de cliente/servidor e pela Internet.



# Redes e Aplicações Não-Hierárquicas (P2P)

Aplicações ponto-a-ponto  
Cliente e servidor na mesma comunicação



Ambos os clientes:

- Iniciam uma mensagem
- Recebem uma mensagem

Ambos os clientes  
simultaneamente:

- Envia
- Recebem



---

# Serviços e Protocolo DNS

---

A camada de **Transporte** utiliza um *esquema* de **endereçamento** chamado *número de porta*.

Os **números de porta** identificam **aplicações e serviços** da camada de **Aplicação** que são a **origem** e o **destino** dos dados.



---

# Serviços e Protocolo DNS

---

## Alguns protocolos são:

- Domain Name System (**DNS**)- Porta TCP/UDP **53**
- Hypertext Transfer Protocol (**HTTP**) - Porta TCP **80**
- Simple Mail Transfer Protocol (**SMTP**) - Porta TCP **25**
- *Protocolo* **POP**<sub>8</sub>- Porta UDP **110**
- **Telnet** - Porta TCP **23**
- Dynamic Host Configuration Protocol (**DHCP**) - Porta UDP **67**
- File Transfer Protocol (**FTP**) - Portas TCP **20** e **21**



---

# Protocolo DNS

---

Em redes de dados, os dispositivos são rotulados com endereços IP numéricos, para que possam participar do envio e recebimento de mensagens pela rede.

No entanto, a maioria das pessoas tem dificuldade em lembrar esse endereço numérico.

**Assim, os nomes de domínio foram criados para converter o endereço numérico em um nome simples e reconhecível.**



---

# Protocolo DNS

---

Na Internet, tais **nomes** de **domínio**, como **www.cisco.com**, são muito mais fáceis de lembrar do que **198.133.219.25**, que é o endereço numérico real desse servidor.

Além disso, se a Cisco decidir alterar o endereço numérico, isso será transparente para o usuário, já que o nome de domínio continuará sendo **www.cisco.com**.

O novo endereço simplesmente será vinculado ao nome de domínio existente e a conectividade será mantida.



---

# Protocolo DNS

---

Quando as redes eram pequenas, era simples manter o mapeamento entre os nomes de domínio e os endereços que eles representavam. No entanto, à medida que as redes começaram a crescer e o número de dispositivos aumentou, este sistema manual ficou inviável.



---

# Protocolo DNS

---

O **Domain Name System (DNS)** foi criado para **resolução de nomes de domínio para endereço** para tais redes.

O DNS utiliza um conjunto distribuído de servidores para definir os nomes associados a tais endereços numerados.



---

# Protocolo DNS

---

O **protocolo DNS** define um **serviço automatizado** que alia os nomes de recursos com o *endereço de rede* numérico necessário.

Ele inclui o formato para consultas, respostas e formatos de dados.



---

# Protocolo DNS

---

**As comunicações do protocolo DNS utilizam um único formato, chamado de mensagem.**

Este formato de mensagem é utilizado para todos os tipos de consultas de cliente e respostas de servidor, mensagens de erro e transferência de informações de *registro de recursos* entre servidores.



---

# Protocolo DNS

---

O **DNS** é um serviço **cliente/servidor**. No entanto, é diferente dos outros serviços cliente/servidor que examinamos.

Enquanto outros serviços utilizam um cliente que é uma aplicação (como navegador Web, cliente de e-mail), o cliente DNS é executado como um serviço.

O cliente DNS, às vezes chamado de *resolvedor DNS*, suporta a resolução de nome para outras aplicações de rede e outros serviços que precisam dele.



---

# Protocolo DNS

---

**Ao configurar um dispositivo de rede, geralmente fornecemos um ou mais endereços de Servidor DNS que o cliente DNS pode utilizar para resolução de nome.**

Normalmente, o provedor de serviço de Internet fornece os endereços a serem utilizados para os servidores DNS.



---

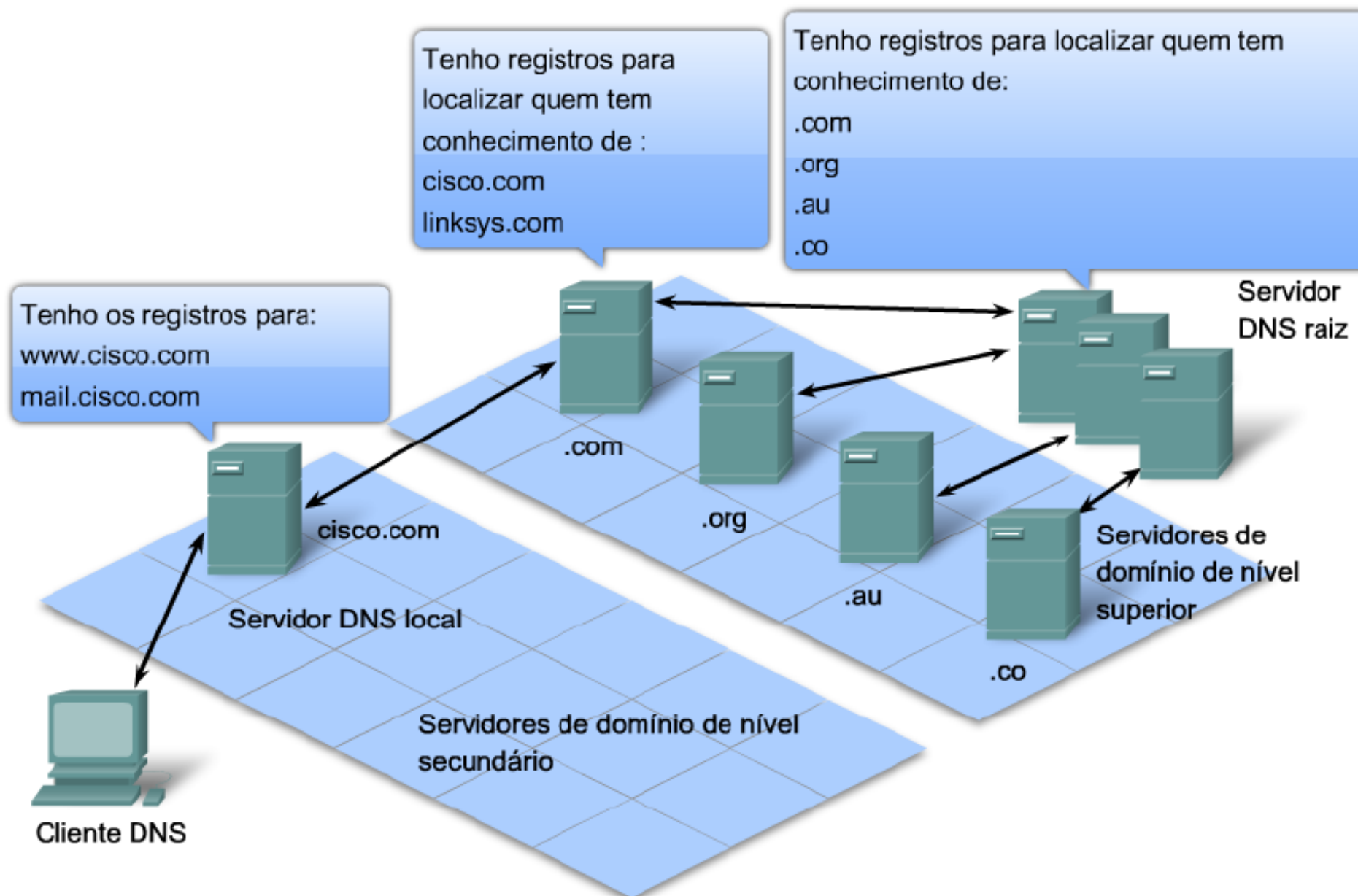
# Protocolo DNS

---

Quando a aplicação de um usuário solicita uma conexão a um dispositivo pelo nome, o cliente DNS solicitante consulta um desses servidores de nome para atribuir o nome a um endereço numérico.

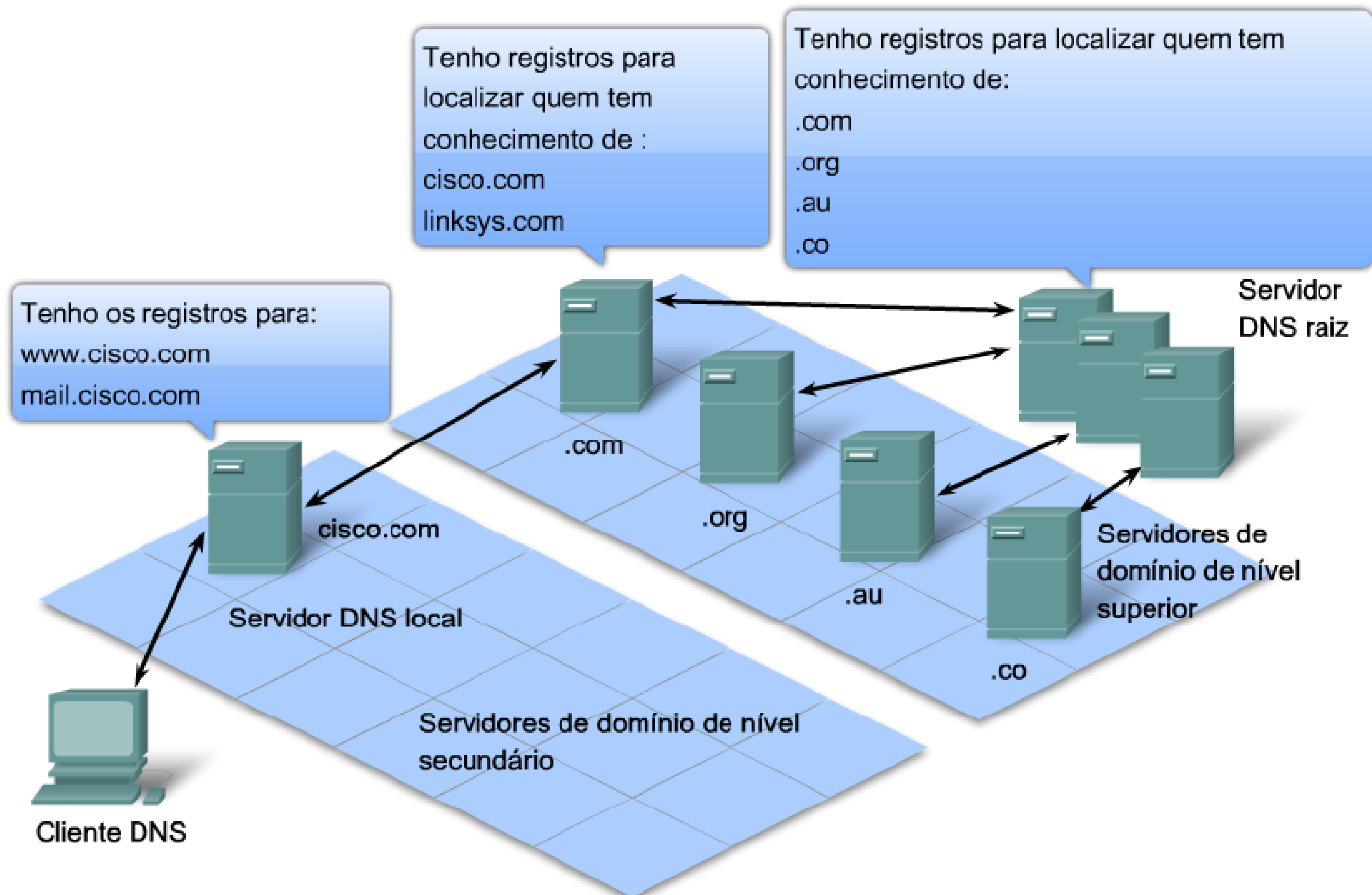


# Protocolo DNS



Hierarquia de servidores DNS que contêm os registros dos recursos correspondentes aos nomes com endereços.





Hierarquia de servidores DNS que contêm os registros dos recursos correspondentes aos nomes com endereços.



---

# Serviços WWW e HTTP

---

Quando um endereço **Web** (ou **URL**) é digitado em um navegador Web, este estabelece uma conexão com o serviço Web executado no servidor utilizando o protocolo **HTTP**.

**URLs** (ou Uniform Resource Locator) e **URIs** (Uniform Resource Identifier) são os nomes que a maioria das pessoas associa a endereços Web.



---

# Serviços WWW e HTTP

---

A URL **http://www.cisco.com/index.html** é um exemplo de URL que se refere a um recurso específico - uma página Web nomeada **index.html** em um servidor identificado como **cisco.com**.



---

# Serviços WWW e HTTP

---

Os navegadores Web são as aplicações cliente que nossos computadores utilizam para se conectar à World Wide Web e acessar recursos armazenados em um servidor Web.

**Semelhantemente à maioria de processos de servidor, o servidor Web é executado como um serviço de segundo plano e disponibiliza diferentes tipos de arquivo.**



---

# Serviços WWW e HTTP

---

Para acessar o conteúdo, os **clientes Web** fazem **conexões** ao **servidor** e **solicitam** os **recursos** desejados.

O **servidor** **responde** com os **recursos** e, no recebimento, o **navegador** **interpreta** os dados e os **apresenta** ao **usuário**.



---

# Serviços WWW e HTTP

---

Os navegadores podem interpretar e apresentar muitos tipos de dados, como texto simples ou **Hypertext Markup Language (HTML)**, linguagem na qual as páginas Web são construídas).

No entanto, outros tipos de dados podem exigir outro serviço ou programa, normalmente mencionado como plugins ou add-ons.

Para ajudar o navegador a determinar que tipos de arquivo ele está recebendo, o servidor especifica o tipo de dados incluído no arquivo.



---

# Serviços WWW e HTTP

---

Para entender melhor como o navegador Web e o cliente Web interagem, podemos examinar como uma página Web é aberta em um navegador.

Para este exemplo, utilizaremos a URL: `http://www.cisco.com/web-server.htm`.

Primeiro, o navegador interpreta as três partes da URL:

1. **http** (protocolo ou esquema)
2. `www.cisco.com` (nome do servidor)
3. **web-server.htm** (nome do arquivo específico solicitado).



---

# Serviços WWW e HTTP

---

Então, o navegador consulta um servidor de nomes para converter **www.cisco.com** em um endereço numérico, que utiliza para se conectar ao servidor.

Utilizando os requerimentos do **protocolo HTTP**, o navegador envia uma **solicitação GET** ao **servidor** e pede o **arquivo web-server.htm**.

O servidor, por sua vez, **envia o código<sup>17</sup> HTML** para esta **página Web** ao navegador.

Por fim, o **navegador decifra o código HTML** e **formata a página** para a janela do navegador.



---

# Serviços WWW e HTTP

---

O **Protocolo de Transferência de Hipertexto (HTTP)**, um dos protocolos do conjunto **TCP/IP**, foi originalmente desenvolvido para publicar e recuperar páginas e agora é utilizado para sistemas de informação distribuídos e colaborativos.

O **HTTP** é utilizado na **World Wide Web** para transferência de dados e é um dos protocolos de aplicação mais usados.



---

# Serviços WWW e HTTP

---

O **HTTP** especifica um **protocolo** de **solicitação/resposta**. Quando um cliente, normalmente um navegador Web, envia uma mensagem de **solicitação** a um **servidor**, o **protocolo HTTP** define os **tipos de mensagem** que o **cliente** utiliza para **solicitar** a **página Web** e também os **tipos de mensagem** que o **servidor** usa para **responder**.

**Os três tipos de mensagem comuns são GET, POST e PUT.**



---

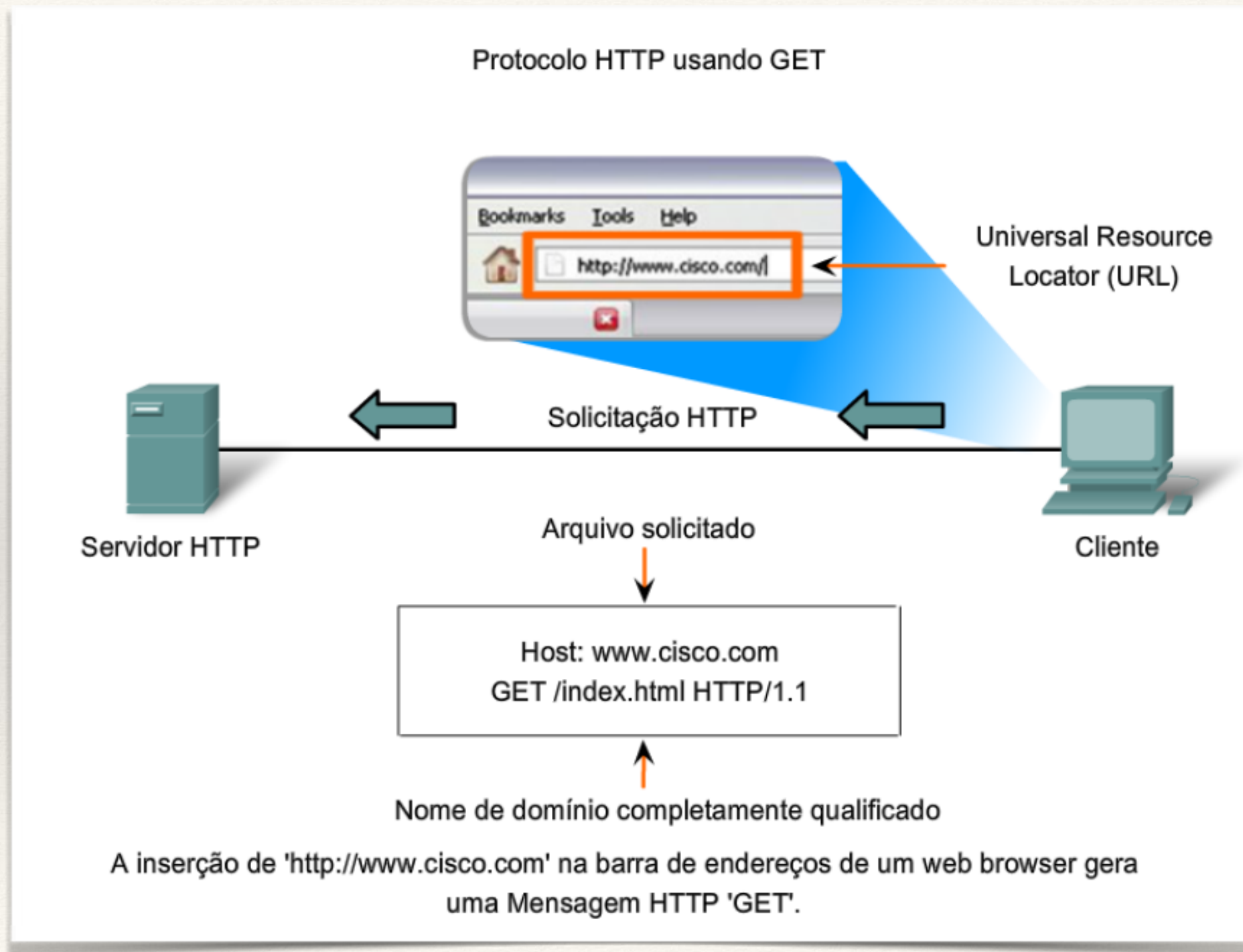
# Serviços WWW e HTTP

---

**GET** é uma solicitação de cliente para dados. Um navegador Web envia a mensagem GET para solicitar páginas de um servidor Web.

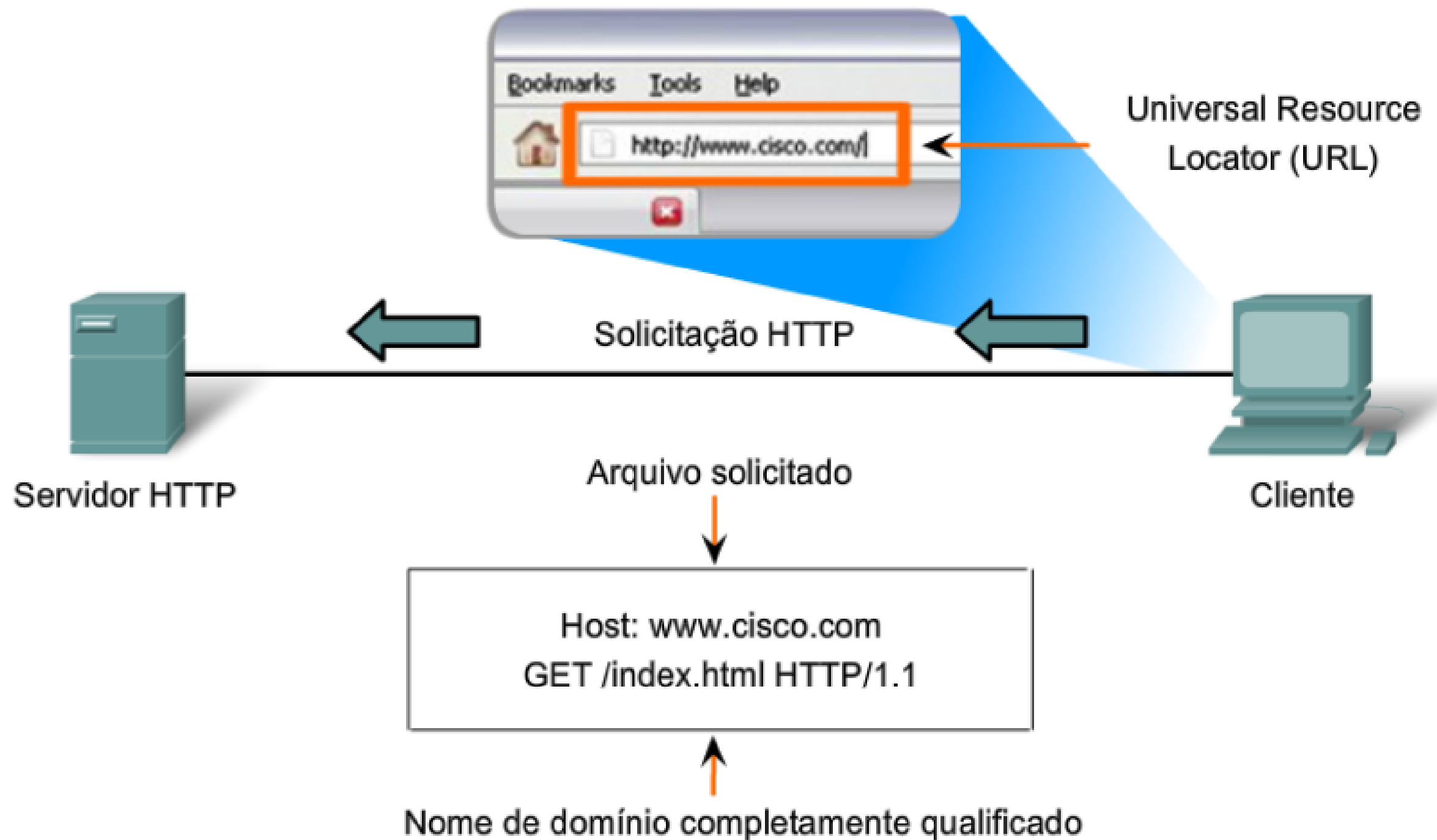


# Serviços WWW e HTTP





## Protocolo HTTP usando GET



A inserção de 'http://www.cisco.com' na barra de endereços de um web browser gera uma Mensagem HTTP 'GET'.



---

# Serviços WWW e HTTP

---

Como mostrado na figura, quando o servidor recebe a solicitação GET, responde com uma linha de status, como HTTP/1.1 200 OK, e uma mensagem própria, cujo corpo pode ser o arquivo solicitado, uma mensagem de erro ou alguma outra informação.



---

# Serviços WWW e HTTP

---

**POST** e **PUT** são utilizados para enviar mensagens que fazem upload de dados ao servidor Web.

Por **exemplo**, quando o usuário insere dados em um formulário incluído em uma página Web, **POST inclui os dados na mensagem enviada ao servidor.**

**PUT faz upload dos recursos ou conteúdo para o servidor Web.**



---

# Serviços WWW e HTTP

---

Embora seja notavelmente flexível, o HTTP **não** é um protocolo seguro.

As mensagens **POST** fazem **upload** de informações ao servidor em texto simples que podem ser **interceptadas** e lidas.

Da mesma forma, as respostas do servidor, normalmente páginas HTML, também são não-criptografadas.



---

# Serviços WWW e HTTP

---

Para comunicação segura pela Internet, o protocolo **HTTP** Seguro (**HTTPS**) é utilizado para acessar ou enviar informações do servidor Web.

O **HTTPS** pode utilizar **autenticação** e *criptografia* para **proteger** os dados que trafegam entre o **cliente** e o **servidor**.

O **HTTPS** especifica regras adicionais para a passagem de dados entre a camada de Aplicação e a de Transporte.



---

# Serviços e-mail e Protocolo SMTP/POP

---

O **e-mail**, o serviço de rede mais popular, revolucionou a forma como as pessoas se comunicam graças a sua simplicidade e velocidade.

Ainda assim, para ser executado em um computador ou outro dispositivo final, o e-mail precisa de várias aplicações e serviços.



---

# Serviços e-mail e Protocolo SMTP/POP

---

Dois exemplos de protocolo da camada de **Aplicação** são **Post Office Protocol (POP)** e **Simple Mail Transfer Protocol (SMTP)**.

Como o HTTP, tais protocolos definem processos de **cliente/servidor**.



---

# Serviços e-mail e Protocolo SMTP/POP

---

Quando as pessoas compõem mensagens de e-mail, elas normalmente utilizam uma aplicação chamada ***Mail User Agent (MUA)***, ou cliente de e-mail.

**O MUA permite que mensagens sejam enviadas e coloca as mensagens recebidas na caixa de correio do cliente, ambos sendo processos diferentes.**



---

# Serviços e-mail e Protocolo SMTP/POP

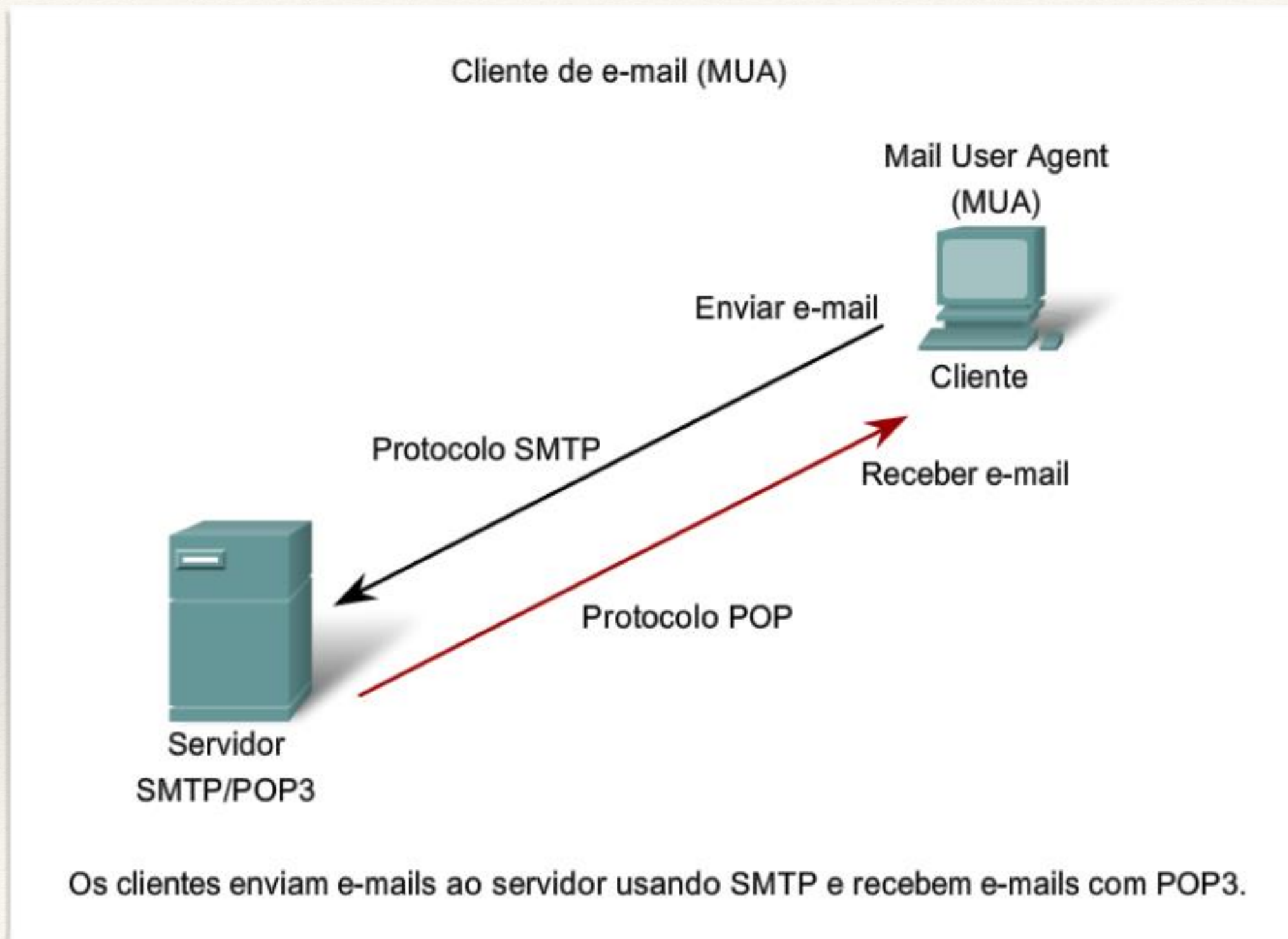
---

Para **receber mensagens** de e-mail de um servidor de e-mail, o cliente de e-mail pode utilizar o **POP**.

Enviar e-mail de um cliente ou servidor utiliza formatos de mensagem e cadeias de comandos definidos pelo protocolo **SMTP**. Normalmente, um cliente de e-mail fornece a funcionalidade dos dois protocolos em uma aplicação.



# Serviços e-mail e Protocolo SMTP/POP





---

# FTP

---

O **File Transfer Protocol (FTP)** é outro protocolo da camada de **Aplicação** comumente utilizado.

O **FTP** foi desenvolvido para possibilitar transferências de arquivos entre um cliente e um servidor.

Um cliente FTP é uma aplicação que roda em um computador e utilizado para carregar e baixar arquivos de um servidor que executa o daemon FTP (FTPD).



---

# FTP

---

Para transferir os arquivos com sucesso, o FTP precisa de **duas conexões** entre o **cliente** e o **servidor**: **uma para comandos e respostas** e **outra para a real transferência do arquivo**.



---

# FTP

---

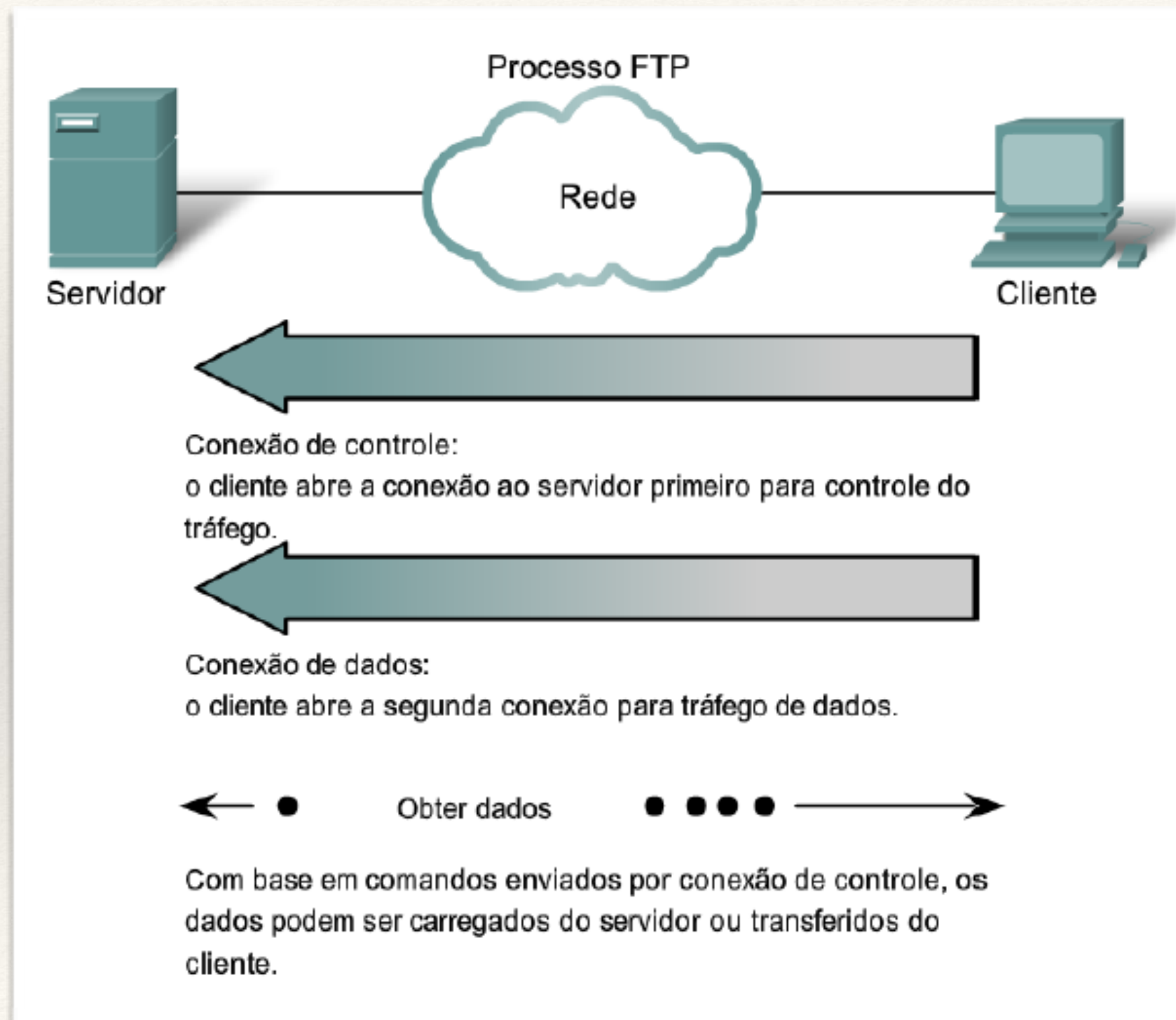
**O cliente estabelece a primeira conexão com o servidor na porta TCP 21. Tal conexão é utilizada para controlar o tráfego, consistindo de comandos do cliente e respostas do servidor.**

**O cliente estabelece a segunda conexão com o servidor pela porta TCP 20. Essa conexão é para a transferência real de arquivo e criada toda vez que houver um arquivo transferido.**

**A transferência de arquivos pode acontecer em ambas as direções. O cliente pode baixar um arquivo do servidor, ou o cliente pode fazer carregar um arquivo ao servidor.**



# FTP





---

# DHCP

---

O serviço do **Protocolo DHCP** permite que os dispositivos em uma rede **obtenham endereços IP e outras informações de um servidor DHCP**.

Este serviço **automatiza a atribuição de endereços IP, máscaras de sub-rede, gateway e outros parâmetros de rede IP**.



---

# DHCP

---

O **DHCP** permite que um **host** obtenha um **endereço IP** quando se conecta à rede.

O **servidor DHCP** é **contatado** e um **endereço** é **solicitado**.

O **servidor DHCP** escolhe um **endereço** de uma **lista configurada de endereços** chamada **pool** e o **atribui** ("aluga") ao **host** por um **período determinado**.



---

# DHCP

---

Em redes locais maiores, ou onde a população de usuários muda frequentemente, o DHCP é preferido.

Novos usuários podem chegar com laptops e precisar de uma conexão.

Outros têm novas estações de trabalho que precisam ser conectadas.

**Em vez de fazer com que o administrador de rede atribua endereços IP para cada estação de trabalho, é mais eficiente ter endereços IP atribuídos automaticamente usando o DHCP.**



---

# DHCP

---

**Os endereços distribuídos pelo DHCP não são atribuídos permanentemente aos hosts, mas apenas alugados por um certo tempo.**

Se o **host** for **desativado** ou **removido** da rede, o endereço volta ao pool para reutilização. Isso é especialmente útil com usuários móveis que vêm e vão em uma rede.

Os usuários podem se mover livremente de local a local e restabelecer conexões de rede. O host pode obter um endereço IP quando a conexão ao hardware for feita, via LAN, com ou sem fio.



---

# DHCP

---

O DHCP possibilita que você acesse a Internet utilizando **hotspots** sem fio em aeroportos ou cafés.

**Ao entrar na área, o cliente DHCP de seu laptop entra em contato com o servidor DHCP local via conexão sem fio. O servidor DHCP atribui um endereço IP a seu laptop.**



---

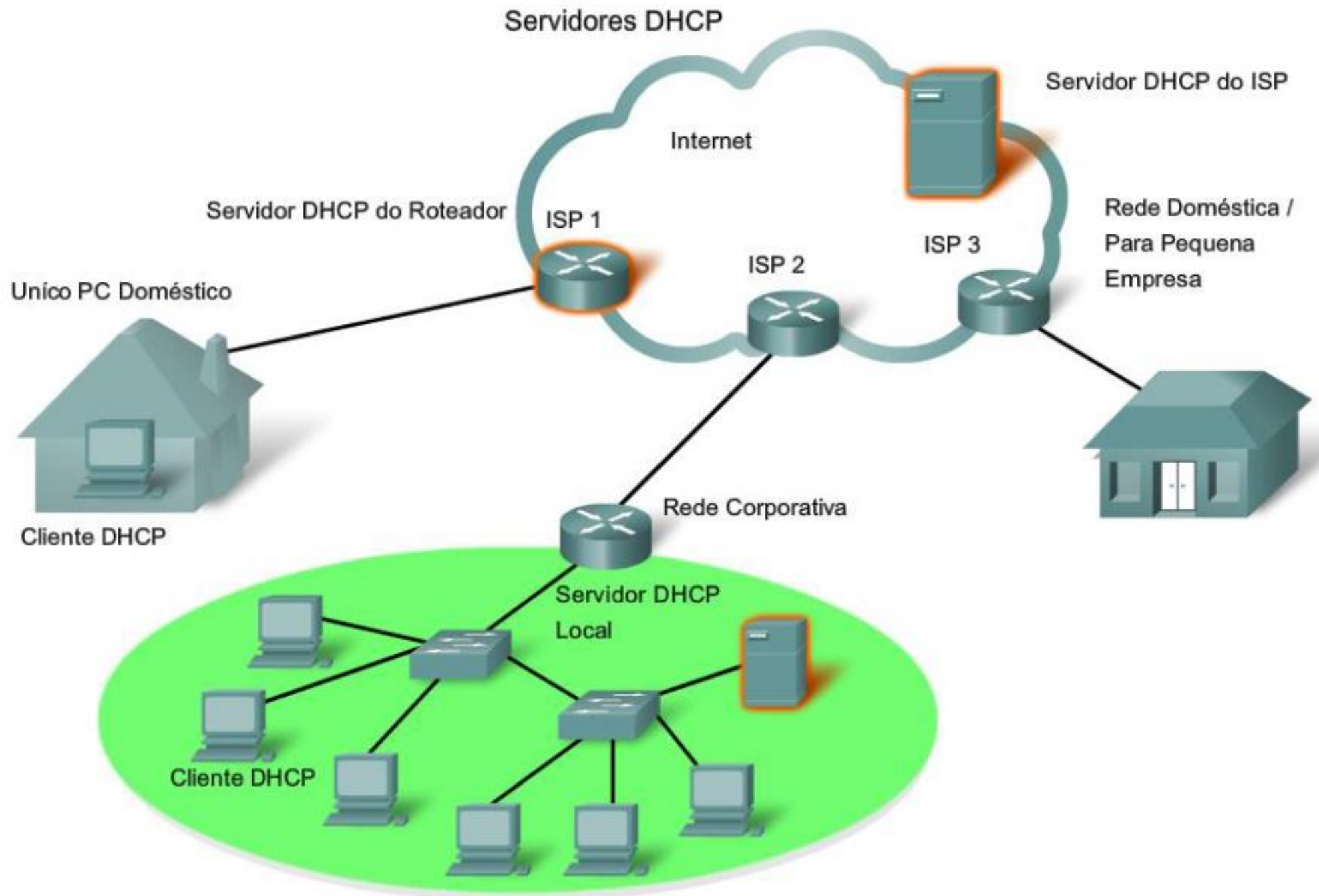
# DHCP

---

Como a figura mostra, diversos tipos de dispositivos podem ser servidores DHCP quando executam softwares de serviço DHCP. O servidor DHCP na maioria de redes médias a grandes normalmente é um servidor local dedicado com base em PC.



# DHCP





---

# DHCP

---

Com redes residenciais, o servidor DHCP normalmente fica localizado no **ISP** e um host na rede residencial recebe sua configuração IP diretamente do ISP.

O DHCP pode apresentar um risco à segurança, pois qualquer dispositivo conectado à rede pode receber um endereço. **O risco torna a segurança física um fator importante ao determinar se o endereçamento dinâmico ou manual será utilizado.**



---

# DHCP

---

O **endereço dinâmico** e o **estático** têm seus lugares no projeto de redes. Muitas redes utilizam DHCP e endereçamento estático.

O DHCP é utilizado para hosts de finalidade geral, como dispositivos de usuário final, e endereços fixos são usados para dispositivos de rede, como gateways, switches, servidores e impressoras.



---

# DHCP

---

Sem o DHCP, os usuários devem inserir o endereço IP, *máscara de sub-rede* e outras configurações de rede manualmente para entrar na rede.

O servidor DHCP mantém um pool de endereços IP e aluga um endereço a qualquer cliente habilitado por DHCP quando o cliente é ativado.

Como os endereços IP são dinâmicos (alugados) em vez de estáticos (atribuídos permanentemente), os endereços em desuso são automaticamente retornados ao pool para realocação.



---

# DHCP

---

Quando um dispositivo configurado com DHCP se inicializa ou conecta à rede, o cliente transmite um pacote DHCP DISCOVER para identificar qualquer servidor DHCP disponível na rede.

Um servidor DHCP responde com um DHCP OFFER, que é uma mensagem de oferta de aluguel com informações de endereço IP atribuído, máscara de sub-rede, servidor DNS e *gateway padrão*, além da duração do aluguel.



---

# DHCP

---

O cliente pode receber diversos pacotes DHCP OFFER se houver mais de um servidor DHCP na rede local, portanto, deve escolher entre eles e transmitir um pacote DHCP REQUEST que identifique o servidor explícito e a oferta de aluguel que o cliente está aceitando.

Um cliente pode decidir solicitar um endereço que já havia sido alocado pelo servidor.



---

# DHCP

---

Presumindo que o endereço IP **solicitado** pelo **cliente**, ou **oferecido** pelo **servidor**, ainda seja **válido**, o servidor retornará uma mensagem DHCP **ACK** que confirma ao cliente que o aluguel foi finalizado.

Se a oferta **não** for **mais válida** - talvez devido a um **encerramento** ou **alocação** do **aluguel** por **outro** b - o servidor selecionado responderá com uma mensagem DHCP **NAK** (Negative Acknowledgement - confirmação negativo).

**Se uma mensagem DHCP NAK for retornada, o processo de seleção deverá recomençar com uma nova mensagem DHCP DISCOVER sendo transmitida.**



---

# DHCP

---

**Quando o cliente tiver o aluguel, este deverá ser renovado antes do vencimento por outra mensagem DHCP REQUEST.**



---

# DHCP

---

O servidor **DHCP** garante que todos os **endereços IP sejam exclusivos** (um endereço IP não pode ser atribuído a dois dispositivos de rede diferentes simultaneamente).



---

# DHCP

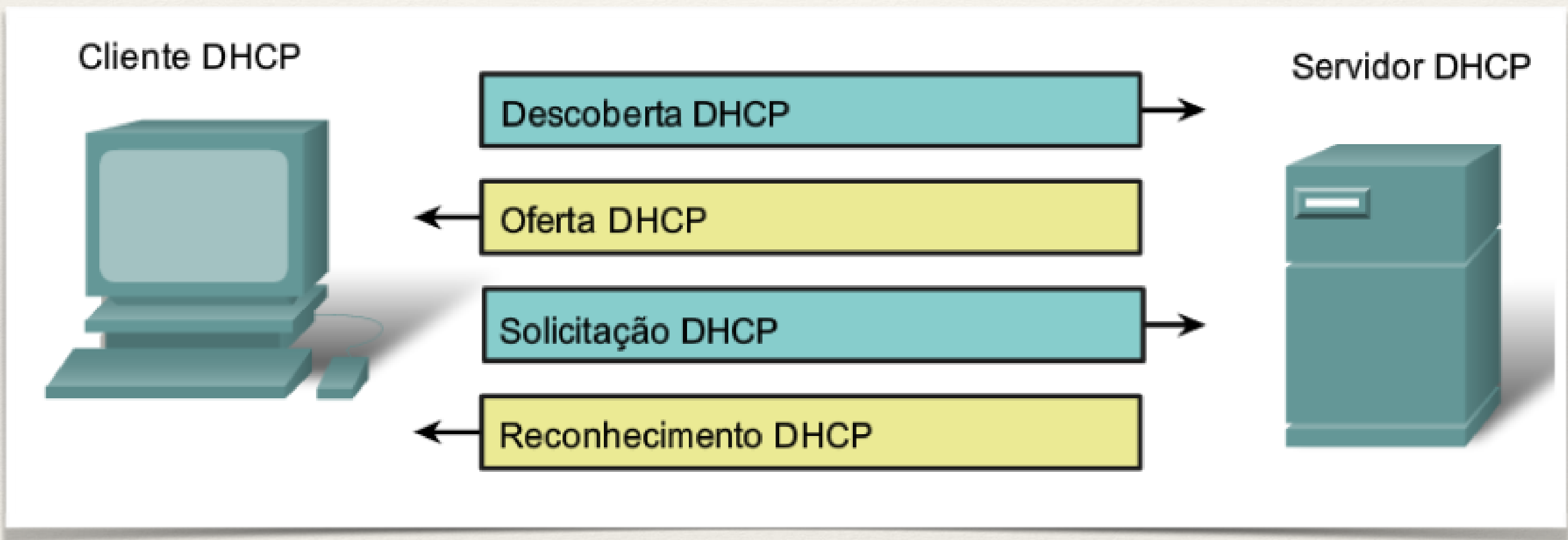
---

Utilizar o DHCP permite que os administradores de rede facilmente reconfigurem endereços IP de clientes sem ter que fazer alterações nos clientes manualmente.

A maioria dos provedores de Internet utiliza o DHCP para alocar endereços a seus clientes que não precisam de um endereço estático.



# DHCP





---

# Resumo

---

A camada de **Aplicação** é responsável pelo **acesso direto a processos subjacentes** que **gerenciam** e fornecem a **comunicação** à rede humana. Esta camada serve como **origem e destino de comunicações** em **redes de dados**.



---

# Resumo

---

**As aplicações, protocolos e serviços da camada de Aplicação permitem que os usuários interajam com a rede de maneira significativa e efetiva.**

**Aplicações são programas de computador com os quais o usuário interage e que iniciam processos de transferência de dados mediante solicitação do usuário.**



---

# Resumo

---

**Serviços** são programas de **segundo plano** que fornecem a **conexão** entre a **camada de Aplicação** e as camadas **inferiores** do **modelo de rede**.

**Protocolos** fornecem uma **estrutura de regras** e **processos acordados** que **garantem** que os serviços executados em um dispositivo em particular possam **enviar e receber** dados de vários **dispositivos de rede diferentes**.



---

# Resumo

---

O fornecimento de dados pela rede pode ser solicitado de um servidor por um cliente, ou entre dispositivos que operam em um acordo **não-hierárquico**, onde a relação **cliente/servidor** é estabelecida de acordo com o dispositivo de origem e destino no momento.

Mensagens são trocadas entre os serviços da camada de Aplicação em cada dispositivo final de acordo com as especificações do protocolo para estabelecimento e uso dessas relações.



---

# Resumo

---

Protocolos como **HTTP**, por exemplo, suportam a entrega de páginas Web a dispositivos finais.

Protocolos **SMTP/POP** suportam o envio e recebimento de e-mail.

**DNS** resolve nomes legíveis utilizados para se referir a recursos de rede em endereços numéricos da rede.



---

# Próxima Aula

---

- Apresentação Trabalho I





---

# Dúvidas

---

- Conteúdo
  - Classroom
- Dúvidas
  - [wagnerglorenz@gmail.com](mailto:wagnerglorenz@gmail.com)





---

# Referências Bibliográficas

---

- KUROSE, James F.; ROSS, Keith W. Redes de Computadores e a Internet: uma abordagem top-down. 3. ed. São Paulo: Person Addison Wesley, Brasil, 2006.
- Tanenbaum, A. S. Redes de Computadores, Tradução da 4ª Edição. Rio de Janeiro: Campus, 2003.
- Tanenbaum, A. S. Redes de Computadores, Tradução da 5ª Edição. Rio de Janeiro: Pearson, 2011.  
<http://ulbra.bv3.digitalpages.com.br/users/publications/9788576059240/pages/-18>
- COMER, Douglas E. Redes de Computadores e Internet. Editora Prentice Hall, New Jersey 1999.
- MENDES, Douglas Rocha. Redes de Computadores: Teoria e Prática. Editora Novatec, São Paulo, 2010.
- Material Cisco Network Academy.