Redes de Computadores II

Aula 02 - Web: Protocolo HTTP e servidor Apache - Parte I

Apresentação

Veja aqui uma introdução a esta aula.

Vídeo 01 - Introdução

Nas últimas décadas, foram criadas diversas aplicações para a Internet que se tornaram bastante populares, sendo utilizadas até nos dias atuais, como por exemplo, o correio eletrônico (SMTP), acesso remoto (Telnet e SSH), a transferência de arquivos (FTP), entre outros. Contudo, a Internet como hoje a conhecemos, se deve em grande parte ao imenso sucesso alcançado por um protocolo em particular criado na década de 1990: o HTTP, pois ele está por trás do funcionamento da Web. É ele, juntamente com os programas *navegadores (browsers)* e *servidores Web*, que nos permitem acessar diariamente nossos *sites* preferidos para trabalhar, nos divertir, ler notícias, realizar compras, etc.

Nesta primeira parte estudaremos o protocolo responsável por permitir a comunicação entre clientes e servidores, chamado de **HTTP** (*HyperText Transfer Protocol*). Analisaremos formato das mensagens de requisição e resposta, funcionamento das suas diferentes versões, entre outros aspectos do protocolo.

Vídeo 02 - Apresentação

Objetivos

Após estudar o conteúdo desta aula, você será capaz de:

- Entender o funcionamento básico do protocolo HTTP nas versões 1.0 e 1.1.
- Entender o formato básico das mensagens HTTP trocadas entre clientes (ex: Firefox) e servidores (Ex: Apache).
- Entender o funcionamento de alguns mecanismos presentes no protocolo HTTP, como cookies.

A Web e o HTTP

Até meados da década de 1990, a Internet se caracterizava como uma rede de uso acadêmico e científico. Seus usuários a utilizavam para enviar e receber e-mails, acessar e trocar arquivos entre computadores remotos etc. Apenas em 1991, com a criação do serviço *World Wide Web* (teia de amplitude mundial), ou simplesmente Web, iniciou-se a fase de popularização da Internet.

Esse sucesso todo da Web veio do fato de que as pessoas desejam informações e a Web permite que pessoas as disponibilizem na Internet de modo muito fácil e que outras pessoas acessem essas informações também de modo muito simples.

Essa forma simples de disponibilizar as informações consiste em utilizar um formato padrão para representá-las, que é o HTML. A forma simples de acessar a informação é atribuir a cada arquivo, que contém informações em HTML, um endereço ou URL (*Uniform Resourc Locator*). Além disso, existem programas servidores, que executam nas máquinas que armazenam os arquivos, programas clientes, que utilizam os endereços atribuídos aos arquivos para solicitá-los aos servidores, e o protocolo HTTP, que é o protocolo utilizado pelos dois programas para se comunicarem.

Atividade 01

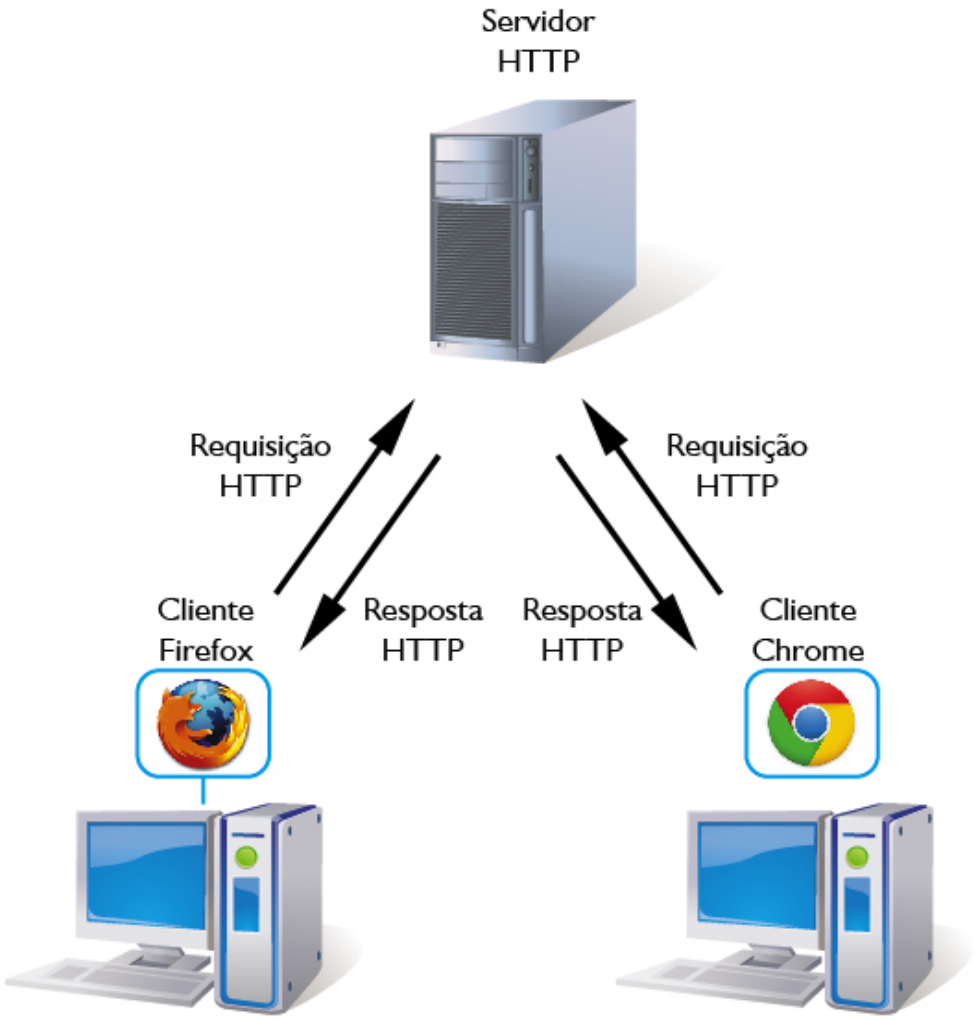
1. Utilizando a própria Web, pesquise sobre a sua história e importância para o desenvolvimento da Internet.

Características do protocolo HTTP

O protocolo HTTP é um dos componentes principais do serviço que comumente chamamos de Web. Esse serviço possui uma arquitetura bastante simples, baseada no modelo cliente-servidor. Conforme pode ser visualizado na Figura 1, apenas três componentes estão diretamente ligados ao seu funcionamento, a saber:

- Clientes: Firefox, Internet Explorer, Chrome, Opera, ...
- Servidor: Apache, IIS, ...
- Protocolo de comunicação: HTTP

Figura 01 - Componentes da aplicação HTTP.



O protocolo HTTP é executado em clientes e servidores, sendo sua principal função a definição do modo como eles trocam mensagens, bem como a estrutura dessas mensagens. Dessa forma, clientes e servidores desenvolvidos de forma independente conseguem se comunicar, bastando, para isso, que implementem o protocolo HTTP.

Boa parte do protocolo controla a transferência de páginas web entre servidores e clientes. Os clientes são chamados *browsers* ou *navegadores.* Uma página Web é formada por uma série de objetos, tais como arquivos texto codificados na linguagem HTML, figuras (JPG, GIF, PNG...), arquivos de áudio, arquivos de vídeo etc. Normalmente toda página web possui um arquivo HTML principal, e dentro dele existem referências para outros objetos, como imagens.

O arquivo HTML principal de uma página web, ou qualquer um de seus objetos, pode ser acessado por um cliente por meio de uma URL (*UniformResourceLocator*). Normalmente acessamos o arquivo HTML principal e o *browser* obtém automaticamente todos os objetos aos quais o arquivo faz referência.

Além disso, toda URL possui várias partes. Na Figura 2, ressaltamos o nome do servidor e o caminho do objeto (incluindo seu nome). A rigor, podemos separar a identificação do objeto em duas partes: o caminho e o seu nome propriamente dito. Na Figura 2, por exemplo, "/9.10/" seria o caminho, e "index.html" o nome do objeto. Além disso, temos "HTTP://" que identifica o protocolo HTTP.

Figura 02 - URL utilizada para acessar uma página web

http://nome_servidor/caminho_objeto
http://start.ubuntu.com/9.10/index.html

Atividade 02

1. Na URL http://www.moodle.imd.ufrn.br/login/index.php, qual o nome do servidor e qual a identificação do objeto?

Características do protocolo HTTP

O protocolo HTTP exige confiabilidade, utilizando, portanto, o protocolo de transporte TCP. Por padrão, servidores HTTP irão aguardar por requisições de clientes na porta TCP/80.

Há duas versões padronizadas do protocolo: HTTP/1.0 (padronizado em maio de 1996, pela RFC 1945) e HTTP/1.1 (padronizado em junho de 1999, pela RFC 2616). As duas versões do protocolo são compatíveis, de forma que um cliente que

implementa apenas a versão 1.0 consegue acessar um servidor que implementa a versão 1.1 e vice-versa. Algumas das principais diferenças entre as duas versões do protocolo serão explicadas nas seções seguintes.

Quando um *browser* obtém um arquivo HTML que contém referências a imagens, o próprio *browser* obtém essas imagens e as mostra para você junto com as informações contidas no arquivo HTML. Além disso, o *browser* obtém as informações do servidor usando o protocolo HTTP.

Interação entre cliente e servidores HTTP

Para entendermos como de fato ocorre a troca de informação entre um servidor e um cliente HTTP, vamos inicialmente descrever, de forma genérica, como ocorre a transferência de uma página web. Dependendo da versão do protocolo em uso, haverá uma importante diferença na forma como cliente e servidor interagem. Por padrão, o HTTP/1.0 transfere páginas em um modo conhecido como *não persistente*, enquanto que o HTTP/1.1 utiliza um modo conhecido como *persistente*.

Apesar de haver diferenças na **forma** como clientes e servidores trocam as mensagens no HTTP/1.0 e HTTP/1.1, o **conteúdo** dessas mensagens pode ser idêntico.

HTTP/1.0 – conexões não persistentes

Para um cliente acessar a página web indicada no quadro abaixo, no modo não persistente, ele realiza a sequência de passos descritos a seguir. Imagine que o arquivo HTML dessa página faz referência a três imagens.

http://start.ubuntu.com/9.10/index.htm

1. Cliente inicia uma conexão TCP com o servidor start.ubuntu.com na porta 80;
2. Cliente envia uma mensagem de requisição HTTP ao servidor solicitando a página web /9.10/index.html;
3. Servidor recebe a requisição e responde com o objeto solicitado (a página) através de uma mensagem de resposta HTTP;
4. Servidor encerra a conexão TCP após a confirmação (acks do TCP) de que o cliente recebeu corretamente sua resposta;
5. Cliente recebe a mensagem de resposta HTTP. Essa mensagem indica que o objeto "encapsulado" é um arquivo HTML. O cliente extrai o arquivo HTML da resposta, analisa seu conteúdo, e encontra referências para 3 outros objetos;
6. Os passos 1 até 4 são repetidos para cada um dos objetos referenciados. Ou seja, novas conexões TCP são estabelecidas;

Quando um cliente ou servidor utiliza conexões não persistentes, para cada objeto existente em uma página irá se repetir o processo de abertura de conexão; solicitação e recebimento do objeto; encerramento de conexão. Assim sendo, esse modo de operação não é apropriado para páginas que possuam um grande número de objetos (como está ficando cada vez mais comum). Abrir diversas conexões sequencialmente é muito lento, e mesmo que fossem abertas em paralelo, isso iria consumir muitos recursos (memória, processador) de clientes e servidores.

Vídeo 03 - HTTP

HTTP/1.1 – conexões persistentes

Um dos principais avanços do HTTP/1.1 é permitir que vários objetos sejam solicitados dentro de uma mesma conexão.

Dessa forma se atinge uma alta velocidade de transferência, sem implicar em um consumo exagerado de recursos em clientes e servidores. Para um cliente acessar a página web(indicada no quadro abaixo, no modo persistente) ele realiza a sequência de passos descritos a seguir. Novamente, imagine que o arquivo HTML dessa página faz referência a três imagens.

```
http://start.ubuntu.com/9.10/index.htm
```

1. Cliente inicia uma conexão TCP com o servidor start.ubuntu.com na porta 80;
2. Cliente envia uma mensagem de requisição HTTP ao servidor solicitando a página web/9.10/index.html;
3. Servidor recebe a requisição e responde com o objeto solicitado (a página) através de uma mensagem de resposta HTTP;
4. Cliente recebe a mensagem de resposta HTTP. Essa mensagem indica que o objeto “encapsulado” é um arquivo HTML. O cliente extrai o arquivo HTML da resposta; analisa seu conteúdo; encontra referências para 3 outros objetos e os requisita imediatamente, em paralelo, pela mesma conexão TCP.
5. Cliente encerra as conexões TCP após obter todos os objetos referenciados na página.

Esse esquema é bem mais eficiente que o não persistente, pois evita o atraso gerado pela abertura de várias conexões e o consumo de recursos que isso gera.

À medida que um cliente recebe os objetos que formam uma página web, ele os exibe na tela. Vale salientar que dois clientes diferentes podem exibir uma mesma página web de forma ligeiramente diferente. A forma como uma página web será “desenhada” na tela do cliente nada tem a ver com o protocolo HTTP.

Veja aqui a explicação em vídeo sobre conexões *não persistentes e* conexões *persistentes*

Vídeo 04 - Persistência

Mensagens de requisição HTTP

Mensagens de requisição HTTP são aquelas enviadas pelo cliente e que contém algum tipo de solicitação a ser atendida por um servidor. No protocolo HTTP essas mensagens são legíveis (podemos entender) possuindo uma ou mais linhas de texto, que são:

- A primeira linha é chamada de **linha de requisição**, que é obrigatória e possui três campos;
- As linhas seguintes são chamadas de **linhas de cabeçalho**. Elas são opcionais (apesar de quase sempre presentes) e possuem dois ou mais campos. Indicam opções relacionadas a cada requisição. Há cerca de 50 opções distintas definidas para o protocolo HTTP/1.1.

Vamos a seguir analisar em detalhes uma mensagem de requisição HTTP que foi enviada de um *browser* para um servidor web. Primeiro, vejamos a requisição completa:

```
GET /12.04/index.html HTTP/1.1
Host: start.ubuntu.com
User-agent: Mozilla/4.0
Accept-language: pt-br
```

Agora vamos nos deter a linha de requisição "**GET /12.04/index.html HTTP/1.1**", que indica:

- O Método (tipo da requisição). “**GET**” é o mais comum. Usado quando o cliente solicita um objeto do servidor;
- O objeto solicitado incluindo o caminho até ele,“**/12.04/index.html**”;
- A versão do protocolo utilizada pelo navegador: “**HTTP/1.1**”;

Já as linhas de cabeçalho utilizadas, nessa requisição em particular, foram as seguintes:

- **Host: start.ubuntu.com**: Indica o nome do servidor no qual se deve buscar o objeto;
- **User-agent: Mozilla/4.0**: Indica o tipo de cliente, ou seja, o navegador;
- **Accept-language: pt-br**: Indica a linguagem preferencial do objeto requisitado.Você poderia ter várias versões do mesmo arquivo (cada uma em uma língua diferente).

Mensagens de resposta HTTP

Mensagens de resposta HTTP são aquelas enviadas pelo servidor em resposta a uma mensagem de requisição de um cliente. Também são legíveis e possuem o seguinte formato:

- A primeira linha é chamada de **linha de estado**, que é obrigatória e indica o *status* da resposta (sucesso, erro, etc);
- A seguir, podem vir uma ou mais **linhas de cabeçalho**, que contém informações sobre o servidor, os dados existentes na resposta etc.
- Dependendo do tipo de resposta, ao seu final virão os **dados** (arquivo HTML, imagem JPG etc...) solicitados pelo cliente.

Vamos, a seguir, analisar em detalhes uma mensagem de resposta HTTP. Primeiro, vejamos a resposta completa:

```
HTTP/1.1 200 OK
Date: Tue, 06 Apr 2012 15:06:06 GMT
Server: Apache/2.2.8 (Ubuntu)
Last-Modified: Fri, 05 Feb 2012 17:24:22 GMT
Content-Length: 2908
Content-Type: text/html; charset=UTF-8
Content-Language: pt-br

<html lang="pt_BR">
```

Agora vamos no deter a linha de *status* "**HTTP/1.1 200 OK**". Ela indica que:

- O servidor está utilizando a versão 1.1 do protocolo HTTP: "**HTTP/1.1**";
- A solicitação pode ser atendida com sucesso: "**200 OK**". Na terceira parte da mensagem de resposta virá o objeto solicitado.

Já as linhas de cabeçalho utilizadas nessa resposta em particular foram as seguintes:

- **Date: Tue, 06 Apr 2010 15:06:06 GMT**: Indica a data e hora no servidor;
- **Server: Apache/2.2.8 (Ubuntu)**: *Software* que está sendo executado no servidor;
- **Last-Modified: Fri, 05 Feb 2010 17:24:22 GMT**: Data e hora de modificação do objeto existente no servidor;
- **Content-Length: 2908**: Tamanho do objeto em bytes;
- **Content-Type: text/html; charset=UTF-8**: Tipo e codificação do objeto; Nesse caso indica que é um arquivo texto cujo conteúdo é HTML. Existem valores para outros tipos (e suas respectivas codificações), como arquivos comprimidos, imagens, áudio etc.
- **Content-Language: pt-br**: Linguagem do objeto.

Atividade 03

1. Pesquise sobre as diversas opções definidas pelo protocolo HTTP que podem estar presentes nas mensagens de requisição e de resposta. Cite duas delas.

2. Pesquise sobre os diversos valores de linhas de estado definidos para o protocolo HTTP, bem como seu significado. Cite dois deles.

3. Simulando um cliente HTTP

   - Abra um terminal Linux e digite:
     telnet start.ubuntu.com 80
   - Digite uma requisição HTTP:
     GET /12.04/index.html HTTP/1.1
     host: start.ubuntu.com
   - tecle<enter> duas vezes e analise a resposta do servidor. O que você acha que foi mostrado?

Interação entre clientes e servidores HTTP: *cookies*

Ao longo da evolução da web, diversos mecanismos foram desenvolvidos ou aprimorados para permitir uma interação mais complexa entre clientes e servidores. Os *cookies* são uma ferramenta bastante utilizada, que permite aos servidores HTTP identificarem os seus usuários, sem que para isso as pessoas tenham que digitar um usuário e senha em um formulário a cada página que acessam, por exemplo. Para seu funcionamento, eles utilizam:

- Linhas de cabeçalho inseridas nas mensagens de requisição e resposta HTTP;

- Arquivos, armazenados na máquina do usuário, e gerenciados pelo navegador (cliente);

- Um banco de dados, mantido no servidor HTTP.

Quando um usuário acessa um site que usa *cookies* pela primeira vez, o servidor responde incluindo o cabeçalho Set-cookie seguido de alguma identificação única. Por exemplo:

```
Set-cookie: 1678453
```

O navegador armazena essa informação em um arquivo texto. Todas as vezes que o usuário voltar a acessar esse site, o navegador irá incluir em suas requisições a linha:

```
Cookie: 1678453
```

Dessa forma se identifica unicamente esse usuário do site, e servidores podem saber, por exemplo, que páginas do site cada usuário visita, a que horas, por quanto tempo etc. Pode-se, também, apresentar páginas “personalizadas” com propaganda direcionada a cada usuário.

Cookies sempre representaram um ponto polêmico da web. Se por um lado um site bem intencionado pode utilizá-los para oferecer uma série de funcionalidades interessantes, por outro, um site mal intencionado pode utilizá-los para “monitorar” os seus usuários.

Veja aqui a explicação em vídeo sobre *cookies*

Vídeo 05 - Cookies

Atividade 04

1. Acesse o site http://www.imd.ufrn.br e veja se ele enviou algum *cookie* para seu navegador. Dica: No Firefox vá em “Ferramentas/ Propriedades da Página/Segurança/Exibir Cookies”.

Resumo

Nesta aula, estudamos a teoria de funcionamento de um dos serviços mais importantes da Internet, a web, e do protocolo de aplicação a ela relacionado, o HTTP. Estudamos como ocorre a comunicação entre o seu navegador (como o Firefox) e um servidor(como o Apache).

Autoavaliação

Inicie o Wireshark em seu computador e realize os procedimentos descritos a seguir.

- Inicie a captura dos pacotes no Wireshark.
- Abra um navegador e acesse a página http://moodle.imd.ufrn.br.
- Encerre a captura de pacotes.

Analisando os pacotes capturados pelo Wireshark, responda as questões abaixo:

1. Qual versão do HTTP está sendo executada pelo seu navegador e pelo servidor?
2. Quais linguagens o seu navegador informa aceitar ao servidor?
3. Qual foi a linha de status retornada pelo servidor para seu navegador?
4. Quando o arquivo HTML baixado foi modificado no servidor?
5. Qual o tamanho em bytes desse arquivo HTML?

Referências

APACHE HTTP Server. Disponível em: <http://httpd.apache.org/>.Acesso em: 29 set. 2010.

KUROSE, J.; ROSS, K. **Redes de computadores e a internet**. 5. ed. São Paulo: Addison Wesley, 2010.

NETCRAFT.Disponível em: <http://netcraft.com>. Acesso em: 29 set. 2010.

RFC 1945. Disponível em: <http://www.ietf.org/rfc/rfc1945.txt>.Acesso em: 29 set. 2010.

RFC 2616. Disponível em: <http://www.ietf.org/rfc/rfc2616.txt>.Acesso em: 29 set. 2010.