Comunicações

por Computador

Módulo 3: HTTP (HyperText Transfer Protocol)

1. Introdução

* Protocolo da **camada de aplicação** da *Web*
* Implementado em dois programas **-> Cliente e Servidor**
* **Cliente:** browser envia ao Servidor mensagens de requisição HTTP para os objetos dessa página
* **Servidor:** recebe as requisições e responde ao Cliente com mensagens de resposta HTTP que contêm esses objetos pedidos

**Estes dois programas, executados em *hosts* diferentes, conversam por meio de troca de mensagens HTTP**

* Uma página *Web* é constituída por um conjunto de objetos (imagens, arquivo HTML, vídeo, etc…)

**HTTP UTILIZA PROTOCOLO TCP COMO PROTOCOLO DE TRANSPORTE**

1. O Cliente HTTP inicia primeiro uma conexão TCP com o Servidor
2. Estando a conexão estabelecida, os processos do *browser* e do Servidor acedem o TCP por meio das *interfaces* dos seus *sockets*. (o mesmo para o Cliente em si)

Assim, o Cliente envia mensagens de requisição HTTP para o seu *socket* e o Servidor recebe as mensagens HTTP pela *interface* do seu *socket* e envia mensagens de resposta para essa mesma interface.

**No fundo, quando um Cliente envia uma mensagem para o *socket*, a mensagem sai, passando para as mãos do TCP.**

**Como o HTTP usa o TCP, o TCP dá-lhe o serviço confiável de transferência de dados.**

* Todas as mensagens de requisição HTTP emitidas pelo Cliente chegam intactas ao Servidor
* Todas as mensagens de resposta HTTP emitidas pelo Servidor chegam intactas ao Cliente

NÃO HÁ NECESSIDADE DE PREOCUPAR COM DADOS PERDIDOS DADO QUE O TCP JÁ O FAZ.

1. Tipos de Conexões

Nas aplicações da Internet, o Cliente e o Servidor comunicam-se por um período prolongado de tempo. Neste período, o Cliente faz uma série de requisições e o Servidor vai respondendo a cada uma delas.

**Tendo em conta que a interação Cliente-Servidor acontece por meio de uma conexão TCP, quando se cria uma aplicação é preciso tomar uma decisão.**

* Cada par de requisição/resposta deve ser enviado através de uma conexão distinta para cada par?
* Cada par de requisição/resposta deve ser enviado através da mesma conexão TCP?

HTTP USA CONEXÕES PERSISTENTES (MESMA CONEXÃO TCP PARA VÁRIOS PARES) NO MODO PADRÃO.

CLIENTES E SERVIDORES HTTP PPDEM CONFIGURAR PARA CONEXÃO NÃO PERSISTENTE.

----------------------------------------------------------

2.1 Conexões Não Persistentes

* **Só pode ser enviado no máximo um objeto *Web* por cada conexão estabelecida. ->** HTTP/1.0 usa HTTP não persistente.
* Para cada objeto que exista na página *Web* é criada/encerrada uma conexão TCP
* Por cada conexão TCP que é estabelecida, essa mesma é encerrada logo após o Servidor enviar o objeto **-> Conexão não persiste para outros objetos.**

Rapidamente se percebe que, este método de conexão é mais lento, dado que criar várias conexões TCP gera atrasos. Esta ideia fica mais clara se calcularmos o **RTT (*Round Trip Time*)**, ou seja, o tempo que leva para que um pacote vá do Cliente ao Servidor e volte para o Cliente.

2.2 Conexões Persistentes

* O Servidor deixa a sua ligação aberta, ou seja, o *socket*, mesmo depois de enviar a mensagem de resposta ao Cliente
* Os pedidos HTTP posteriores (pedido/resposta) são enviados através dessa mesma ligação

**Desta forma, uma página *Web* completa pode ser enviada mediante uma única conexão TCP Persistente.**

1. Formato Mensagem HTTP

Existem dois tipos de mensagens HTTP que irão ser estudadas e parametrizadas consoante as suas especificações.

----------------------------------------------------------

3.1 *Request Message* (Pedido)

3.2 *Reponse Message* (Resposta)

----------------------------------------------------------

1. Cookies (informação do Estado)

* HTTP não tem estado o que simplifica o projeto do servidor
* Necessidade de identificar utilizadores num site

**Com os cookies, os *websites* conseguem monitorizar os vários utilizadores.**