Definitions, Definitions, Definitions

Lead to Understanding

Background

TCP/IP from Lecture 1

Sources

Webopedia

Web Monkey

Etc.

Want to define

- CGI
- HTTP
- Comparison of HTTP/1.0 and HTTP1.1

We'll do some preliminary defintions first

➔CGI

Client/Server Architecture

A network architecture in which each computer or process on the network is either a client or a server

Client

An application that runs on a personal computer or workstation. Relies on a server.

Server

A computer or device on a network that manages network resources (e.g., file server, web server, print server etc.)

➔CGI

Protocol

An agreed-upon format for transmitting data between two devices. The protocol determines the following:

- **the type of error checking to be used**
- **data compression method, if any**
- **how the sending device will indicate that it has finished sending a message**
- **how the receiving device will indicate that it has received a message**

➔CGI

Web Server

A computer that delivers (serves up) Web Pages

Servlet

Small program that runs on a server.

Usually refers to a Java applet that runs within a Web server environment.

Analogous to a Java applet that runs within a Web browser environment

➔CGI

API

application program interface: set of routines, protocols, and tools for building software applications

→CGI←

Common Gateway Interface

Specification for transferring information between a WWW Server and a CGI program.

CGI program

any program designed to accept and return data that conforms to the CGI specification.

→CGI←

CGI program

Most common way (still – even with Javascript & PHP in increasing use!) for web servers to interact dynamically with users.

HTML pages with forms often use CGI to process the form's data.

Server-side!

➔CGI ➔

Drawbacks

Each time CGI script executed, a new process is started – can slow down the server.

Servlets, API faster

➔HTTP

Communications protocol.

Defines the following:

- ***rate of transmission (in baud or bps)***
- ***whether transmission is to be synchronous or asynchronous***
- ***whether data is to be transmitted in half-duplex or full-duplex mode***

...and others ...

➔HTTP

State

The last-known or current status of an application or a process.

Maintaining state and/or managing state refer to keeping track of the condition of the process.

➔HTTP

Stateless (Sounds lonely!)

Having no information about what occurred previously.

Most modern applications maintain state, which means that they remember what you were doing last time you ran the application, and they remember all your configuration settings.

➔HTTP

Stateless CONT.

The Internet is intrinsically stateless because each request for a new Web page is processed without any knowledge of previous pages requested..

Because maintaining state is extremely useful, programmers have developed a number of techniques to add state to the World Wide Web.

These include server APIs, and the use of cookies.

➔HTTP

Socket

BSD method for accomplishing interprocess communication (IPC). What this means is a socket is used to allow one process to speak to another, very much like the telephone is used to allow one person to speak to another.

➔HTTP ➔

***H*ypertext *T*ransfer *P*rotocol**

- ***Underlying protocol used by WWW***
- ***Defines how messages are formatted and transmitted, and what actions Web servers and browsers should take in response to various commands***
- ***Entering a URL sends an HTTP command to the Web server***
- ***Uses TCP***

➔HTTP ➔

- ***Also Stateless***
- ***Considered a Shortcoming***
- ***New technologies (Javascript, cookies, etc.) address this shortcoming***

➔HTTP ➔

Used to transmit **resources** *(file, output of a CGI script, ...)*

Steps:

- *Client opens a connection and sends a*

Request *message to an* ***HTTP server***

2. Server returns a ***Response*** *message, often containing resource requested.*

3. Server closes connection (HTTP stateless)

➔HTTP ➔

Format of Requests and Responses similar:

- ***Initial line***
- ***Zero or more header lines***
- ***Blank line (<CR><LF>)***
- ***An optional message***
  - ***A file, output, ...***

➔HTTP ➔

Header Lines

- ***Provide information about the request or response or about object in message body***
- ***Header1: Something, something else...***

Examples:

From:

User-Agent:

Server:

Last Modified:

.....

➔HTTP ➔

HTTP message:

- ***Initial line : different for Request and Response***
- ***Header1: Value1***

  Header2: Value2

  Header3: Value3

- ***Optional message***
  - ***File contents, output: Can be many lines long***

➔HTTP ➔

Initial line: Requests:

- ***GET /path/to/file/filename HTTP/1.0***

Says "Get me this resource

Also could be POST (sends data to server)

or HEAD (tells server to return response headers only)

Initial line: Response

- ***HTTP/1.0 200 OK***
- ***HTTP/1.0 404 File Not Found***
- ***HTTP/1.0 500 Server Error***

➔HTTP➔

***Proxy*: Program that acts as an intermediary between a client and a server**

- **Receives requests from clients**
- **Forwards requests to servers**
- **Passes back the same way**

Often used for firewalls: Requests sent to proxy, not directly to server

With a proxy, GET includes the complete URL

➔HTTP ➔

Example *To retrieve file at: http://www.somehost.com/path/file.html*

Step 1 a socket opened to www.somehost.com, port 80

Step 2 Through the socket is sent:

Get path/file.html HTTP/1.0

From: someuser@hi.is

User-Agent: HTTPTool/1.0

[Blank line]

Step 3 (Server Response) Through the socket is sent:

HTTP/1.0 200 OK

Date Tues, 11 Feb 2003 23:59:59

Content-Type: text/html

Content-Length 1354

<HTML>

<BODY>

...

</HTML>

➔HTTP ➔

To see some HTTP, use the non-graphical browser lynx:

Type

lynx -mime_header someURL | more

at the command prompt

➔CGI Revisted ➔

CGI *Used in order for information passed from the client to the server using HTTP to be processed by a program running on the Server.*

CGI *a protocol that defines how data is to be passed between a web server and a CGI application.*

CGI *defines a set of environment variables used to pass the data.*

CGI *variables may be set on the cient side using Javascript.*

➔HTTP/1.0 & HTTP 1.1 ➔

Differences

HTTP/1.1 ***Maintains client-server state***

by allowing persistent connections.

One TCP connection can be reused for multiple

HTTP transactions

HTTP/1.0 ***Stateless. Each HTTP transaction creates its own TCP connection.***

HTTP/1.1 ***Requires Host: in Header***

HTTP/1.0 ***No Header Fields required***

➔HTTP/1.0 & HTTP 1.1 ➔

Differences

HTTP/1.1 ***New Features:***

Improved caching:Original server (if resources passed along) decides what can be cached

Persistent Connections

Chunked encoding: Server breaks response into smaller chunks and sends them in series

Host: Required in Header of Request

... See http://www8.org/w8-papers/5c protocols/key/key.html