Redes de Computadoras II
HTTP

NOMBRE: Daniel Alberto Vinzia

HTTP Básico

1. ¿Cuál es la diferencia notoria entre el comportamiento de la versión 1.0 y 1.1 de HTTP?

La principal diferencia entre HTTP 1.0 y 1.1 es la gestión de las conexiones. HTTP 1.0 usa una nueva conexión para cada solicitud, mientras que HTTP 1.1 permite mantener la conexión activa para múltiples solicitudes, lo que reduce el tiempo de carga y el tráfico de red.

2. Eche una mirada más detenida a la respuesta de la solicitud del paso 5. ¿Cuál es el código de estado (status code), frase razón y versión del servidor HTTP? Sugiera una razón para tener el código de estado y la frase de razón en la respuesta.

El método **GET** solicita una representación de un recurso específico. Las peticiones que usan el método **GET** sólo deben recuperar datos. El método **HEAD** pide una **respuesta** idéntica a la de una petición **GET**, pero sin el cuerpo de la **respuesta**.

Devuelve un Status code 200 OK en HTTP 1.1

3. Luego de ejecutar el paso 6, ¿Cuál es la respuesta esta vez? ¿Por qué?

La interacción básica HTTP GET/response

1. ¿Su navegador utiliza HTTP versión 1.0 o 1.1? ¿Que versión de HTTP está corriendo el servidor?

Tanto el navegador como el servidor usan las versiones de HTTP 1.1

2. ¿Que lenguajes (si hay alguno) su navegador le ha indicado al servidor que puede aceptar?

El lenguaje aceptado es ingles de Estados Unidos (Accepted-Lenguage: en-us)

3. ¿Cuál is la dirección IP de su computadora? ¿Y la del servidor gaia.cs.umass.edu server?

La direccion IP de mi Computadora es: 192.168.1.102

La direccion IP del servidor es: 128.119.245.12

4. ¿Cuál es el código de status retornado por el servidor a su navegador?

El servidor responde con un código **200 OK**

5. ¿Cuándo el servidor modificó por última vez el archivo HTML recibido?

La ultima modificacion del servidor fue:

martes, 23 de Septiembre de 2003 a las 05:29:50 GMT

que es la misma que la fecha de la respuesta

6. ¿Cuantos bytes de contenido fueron retornados a su navegador?

la respuesta fue de 73 Bytes

7. ¿Inspeccionando los datos brutos (raw data) en la ventana packet-content, ve alguna cabezera en los datos que no es mostrada en la ventana packet-listing? Si es así, nombre uno.

la cabecera de los datos esta vacia, pero se incluye en el envio y se las ve vacia

La interacción básica HTTP GET/response

1.Inspeccione el contenido de la primera solicitud HTTP GET del navegador al servidor. ¿Puede ver alguna línea del tipo "IF-MODIFIED-SINCE" en el HTTP GET?

En la primera solicitud no se encuentra esta linea

2. Inspeccione el contenido de la respuesta del servidor. ¿El servidor, hizo explicito el contenido del archivo? ¿Qué puede decir?

El servidor envio toda la respuesta de lo que se ve en pantalla de manera de texto plano

3. Ahora fíjese el contenido de la segunda solicitud HTTP GET. ¿Puede ver alguna línea del tipo "IF-MODIFIED-SINCE:" en el HTTP GET? Si la hay, ¿qué información sigue después de la cabecera "IF-MODIFIED-SINCE:"?

La línea que se encuentra es la siguiente:

If-Modified-Since: Tue, 23 Sep 2003 05:35:00 GMT\r\n

4. ¿Cuál es el código de estado y la frase que devuelve el servidor en la segunda respuesta HTTP GET? ¿El servidor, hizo explicito el contenido del archivo? Explique.

El servidor devuelve un código 304 de que no fue modificado

Obteniendo documentos largos

1. ¿Cuántas mensajes de solicitud HTTP GET fueron enviados por el navegador?

El navegador solo envío un solo mensaje de solicitud HTTP al servidor

2. ¿Cuantás segmentos TCP fueron necesarios para enviar los datos de la respuesta HTTP completa?

TCP envia tres segmentos porque la respuesta HTTP es demasiado grande

3. ¿Cuál es el código de estado y la frase asociada a la respuesta de la solicitud HTTP GET?

el codigo sigue siendo 200 OK a la solicitud, como vemos HTTP no se entera de lo que realiza TCP

4. ¿Hay alguna línea de estado HTTP en los datos transmitidos asociada a TCP que mencionen "Continuation"?

no, en ningún se menciona un **continuation** o algún estado HTTP ya que el protocolo que se encarga de la fragmentación de los datos y de re ensamblarlos es TCP

Documentos HTML con objetos embebidos

1. ¿Cuántos mensajes de solicitud HTTP GET fueron enviados por el navegador? ¿A cuál direcciones fueron enviadas?

Fueron enviados 3 solicitudes a las siguientes IP:

128.119.245.12, 165.193.123.218 y por ultimo 134.241.6.82

2. ¿Se puede saber si su navegador ha descargado las dos imágenes en serie, o si se han descargado de los dos sitios web en paralelo? Explique.

las dos imágenes fueron descargadas en paralelo pero al ser muy pequeñas y muy rapido el internet cuando empezó a descargar una, ya la otra estaba descargada

Autenticación HTTP

1. ¿Cuál es el código de estado y la frase que devuelve el navegador en la respuesta a la solicitud del primer HTTP GET?

el estado es 401 Authorization Required, autorización requerida

2.Cuando el navegador envía el segundo mensaje HTTP GET ¿que campo nuevo se agrega en el mensaje HTTP GET?

envia lo siguiente

Authorization: Basic ZXRoLXN0dWRlbnRzOm5ldHdvcmtz\r\n

y debajo

Credentials: eth-students:networks

3.¿Cuál es el string que transporta el usuario y la password? Ayuda: busque el string que sigue a la cabecera "Authorization: Basic" en el mensaje GET del cliente. En el siguiente sitio https://www.base64decode.org/ hay herramientas para codificar y decodificar información en formato Base64. ¿En que otro tipo de aplicaciones se usa este código? Coloque el string obtenido anteriormente y decodifíquelo. ¿Qué opina sobre la autenticación HTTP?

en el sitio suministrado cuando le ingresamos **ZXRoLXN0dWRlbnRzOm5ldHdvcmtz** nos devuelve **eth-students:networks**

Base64 se utiliza para codificar datos binarios en un formato de texto legible por máquina, lo que permite su transmisión y almacenamiento en medios que solo soportan texto. Esto es útil en

escenarios como la incrustación de imágenes en HTML, el envío de archivos adjuntos en correo electrónico, o la transmisión de datos a través de protocolos que no admiten binarios.

Comunmente se usa en:

Desarrollo web, correo electronico, almacenamiento de datos, transmision de datos, etc.

la autenticacion HTTP no es mala y se usa para poder prohibir el ingreso de usuarios sin permisos