Teoria de redes y comunicaciones 2

HTTP

Protocolo, buscadores de la web implementan la parte de cliente de este protocolo.
Elementos web:

- Recurso o objeto http.
- Referenciado por URI, URL, URN.
- Formato específico de url: protocol://[user:pass@]host:[port]/[path] (Navegadores ocultan mucho de esto).
- Páginas, applets, jpg, etc.

Http no tiene estado, se hace pedido y cierra conexión una vez terminado. (Una conexión TCP por cada pedido).
Trabaja sobre texto ASCII, permite enviar información binaria con encabezados MIME.

HTTP 1.0

Define formato, proceso basado en HTTP 0.9:

- Especificar versión en el requerimiento del cliente
- Metodos HTTP (GET, POST)
- Códigos de respuesta (200,400)
- Permite Ascii, utf-8, etc.
- MIME (archivos binarios).
- Sin conexiones persistentes.

Host virtual: Múltiples páginas web en un servidor, que serán mostrados dependiendo del host puesto la request.

Autenticación HTTP

HTTP-1.0 contempla autenticación con WWW-Authenticate Header, se usan encabezados para intercambiar información auth entre cliente y server.
El servidor indicará que necesita autenticarse para ver un documento con un 401, y el usuario ingresará los datos (o usará los que tenga cacheados), el servidor entonces dará acceso en base a esos valores.

Pipelining HTTP 1.1

- Paralelismo de peticiones de objetos HTTP.
- Solo se utiliza con conexiones persistentes.
- Mejores tiempos de rta.
- Sobre la conexión se debe mantener el orden de los objetos que se devuelven.

Otras cosas de HTTP 1.1

Nuevos verbos: Trace (para debugging), Connect (para generar conexiones y otros servicios montados por HTTP).

Cookies

Valores en el cliente asociado al servidor, cada vez que se mande una request el cliente va enviar el encabezado con las cookies que están asociadas a acciones pasadas del cliente, por ejemplo para armar sesiones.
Cookies como historia de la sesión

HTTPS

Utiliza puerto 443 por defecto, hay una etapa de negociación, se cifra toda la request y luego se autentica.

HTTP sobre TLS/SSL

Hay un handshake inicial donde se pasan qué algoritmos se pueden usar y además el server le pasa el certificado.
El cliente entonces verifica el certificado y extrae una clave (un cliente confiará en un certificado basado en si fue provisto por una de las autoridades confiadas).
El cliente le pasa al servidor una clave de sesión secreta cifrada con la clave pública de certificado del servidor. (Criptografía asimétrica).
Después se pasan los datos cifrados con estas claves (con algoritmos de criptografía simétrica).
Las claves secretas expiran y se deben crear y mandar devuelta.