

Protocolos y tecnologías para la transmisión de datos en tiempo real por la red.

Ing. Liester Cruz Castro.

Especialista en bases de datos espaciales y profesor de la Facultad 6, Universidad de las Ciencias Informáticas, Carretera San Antonio km 2½, La Habana, Cuba.

lcruz@uci.cu

Ing. Miosotis Aida Troche Rodríguez.

Especialista en Gestión de Configuración, Facultad 6, Universidad de las Ciencias Informáticas, Carretera San Antonio km 2½, La Habana, Cuba.

miosotisaída@uci.cu

Ing. Javier Rodríguez Ramírez.

Profesor del Instituto Superior Politécnico "José A. Echeverría", Calle 114 entre 119 y 129, Marianao, La Habana, Cuba.

jramirez@cemat.cujae.edu.cu

Ing. Arianna Guerra Medrano.

Especialista en la Empresa Cubana de Software para la Salud, Universidad de las Ciencias Informáticas, Carretera San Antonio km 2½, La Habana, Cuba.

amedrano@softel.sld.cu

Resumen:

El Protocolo de Transferencia de Hipertexto (HTTP) tiene amplia difusión como mecanismo de transmisión de datos por la red. Sin embargo, debido a su complejidad existen sistemas que necesitan funcionar en tiempo real, por lo que el protocolo HTTP no permite su correcta implementación debido a las características de su funcionamiento. Existen nuevas tecnologías que intentan solucionar los problemas de latencia, pero aún resultan insuficientes porque están implementadas con HTTP. El protocolo WebSocket surge a partir de la implementación de HTML 5 que propone una forma novedosa de transmitir información por la Web, representando mejoras en cuanto a velocidades de interacción entre los componentes de software de un sistema informático. El presente trabajo está orientado a describir y comparar tecnologías de forma tal que constituya una herramienta a la hora de seleccionar una para la implementación de un sistema informático en tiempo real basado en tecnologías Web.

Palabras claves: protocolo; tiempo; velocidad; latencia; tecnología.

Introducción

En informática, la latencia se refiere a los retardos temporales que se registran en una red. Estos retardos se producen por la demora en la propagación y en la

transmisión de los paquetes de datos. Al sumarse todos estos retardos, se obtiene la demora en que se transmite una información (Definicion.de). Por tanto, se infiere que a menor grado de latencia, mayor precisión tendrá la información transmitida. *Un sistema en tiempo real es aquel en el que para que las operaciones computacionales sean correctas no solo es necesario que la lógica e implementación de los programas computacionales sean correctas, sino también el tiempo en el que dicha operación entregó su resultado* (Gillies, 1995). La precisión de un sistema en tiempo real está dada por la latencia con que las notificaciones de eventos ocurridos en el sistema son entregadas a los usuarios del mismo.

Con el surgimiento de nuevos sistemas basados en la Web que simulan procesos que ocurren de forma simultánea existe la tendencia de abolir las viejas tecnologías y utilizar otras más actuales que permiten mayor grado de velocidad y por tanto mejoras en las funcionalidades de los sistemas. El presente artículo tiene como objetivo detallar y elegir protocolos y tecnologías existentes que permiten mejoras en la transmisión de datos, haciendo que las aplicaciones informáticas sobre la Web funcionen en tiempo real.

Desarrollo

En informática, un protocolo es un conjunto de reglas usadas por computadoras para comunicarse unas con otras a través de una red por medio de intercambio de mensajes. Puede ser definido como las reglas o el estándar que define la sintaxis, semántica y sincronización de la comunicación. Los protocolos pueden ser implementados por hardware, software, o una combinación de ambos (Slideshare.net).

Internacionalmente sobresalen tres principales tecnologías para la obtención de respuestas a solicitudes HTML de forma dinámica; estas son AJAX y Comet, soluciones que funcionan directamente sobre el protocolo HTTP, y WebSocket que funciona directamente sobre el protocolo TCP, todas con amplia difusión a nivel internacional. El conocimiento de las características del funcionamiento de dichas tecnologías se convierte en un punto fundamental a la hora de seleccionar el mecanismo de comunicación entre los componentes de software de un sistema en tiempo real. Cada tecnología presenta ventajas y restricciones que son determinantes en el desempeño final de la aplicación.

1. Ajax:

Es un conjunto heterogéneo de tecnologías Web utilizadas conjuntamente para permitir que las interacciones dinámicas cliente-servidor comunes en las aplicaciones Web, se realicen sin que estas requieran recargar o refrescar la página. AJAX es el acrónimo de Asynchronous JavaScript and XML (JavaScript Asíncrono y XML), no es una tecnología en sí misma, es una combinación de tecnologías usada de una manera concreta. Utiliza distintas tecnologías de la siguiente forma:

- XHTML (o HTML) y CSS para crear la interfaz de usuario y la información de estilo. Estos elementos pueden ser modificados dinámicamente para mostrar información nueva, cambiar el estilo de la página o alterar partes de ella. Para realizar estos cambios usualmente se emplea el DOM (Document Object Model) (Universidad de Alicante).

- JavaScript se utiliza para manipular los elementos de la interfaz a través del DOM y para establecer la comunicación con el servidor Web a través del objeto XMLHttpRequest. Este objeto permite transmitir datos (normalmente en formato XML) entre el cliente y el servidor Web sin necesidad de recargar la página o refrescarla.

Utilizando AJAX, las aplicaciones Web pueden permitir altos niveles de interactividad y de procesamiento, sin requerir las interacciones estándar entre cliente-servidor, que constan de los siguientes pasos:

- Se presenta una interfaz de la página Web.
- Se recoge información del usuario (a través de formularios, enlaces, etc.).
- El usuario envía la información (envía el formulario, se activa un enlace, etc.).
- El servidor procesa la información enviada.
- El servidor responde con una nueva interfaz de la página web que es enviada al usuario.

Con AJAX, todos estos pasos se pueden realizar en la misma interfaz. Algunos ejemplos son las aplicaciones de mapas, búsquedas en tiempo real, chat, validación de formularios en tiempo real y procesamiento en segundo plano, aplicaciones de carro de compras y aplicaciones de correo electrónico. Las aplicaciones Web y la interacción mediante el uso de AJAX han dado origen al término "Web 2.0", término que sin duda indica una gran revolución en el modo en que se hace uso de la Web. Si bien ninguna de las tecnologías que componen AJAX son nuevas o innovadoras, la creciente popularidad de su uso conlleva a la necesidad de que los desarrolladores conozcan las implicaciones para la accesibilidad que estas tecnologías acarrearán.

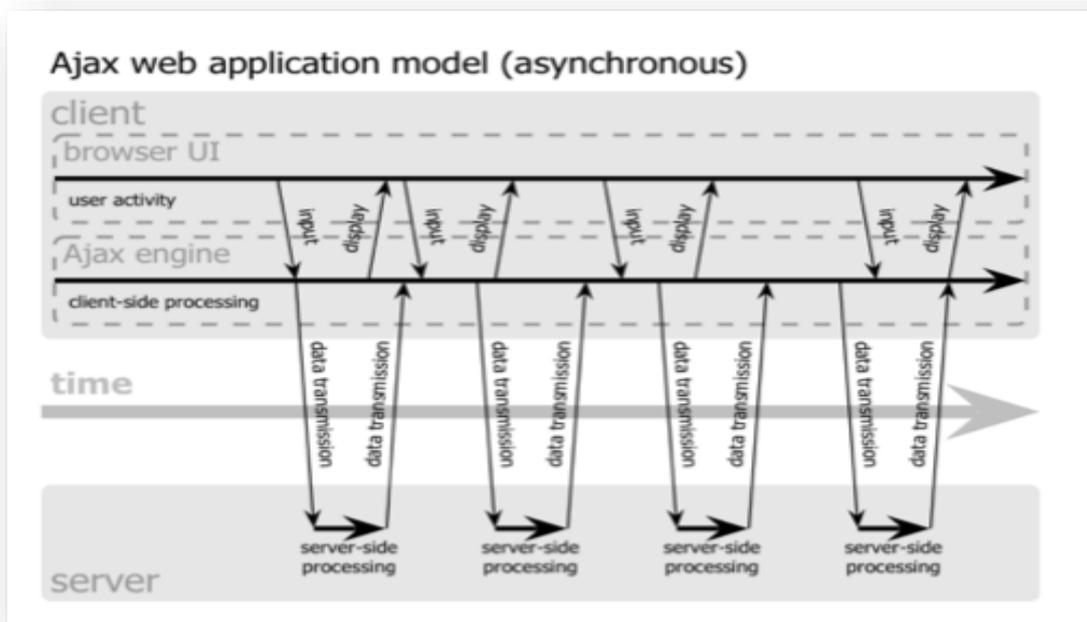


Figura 1. Esquema de funcionamiento de AJAX.

Ventajas:

- Utiliza tecnologías ya existentes.
- Soportada por la mayoría de los navegadores modernos.
- Interactividad. El usuario no tiene que esperar hasta que lleguen los datos del servidor.

- Portabilidad (no requiere plug-in como Flash y Applet de Java).
- Mayor velocidad, esto debido que no hay que retornar toda la página nuevamente.
- La página se asemeja a una aplicación de escritorio (AJAX Ya).

Desventajas:

- Se pierde el concepto de volver a la página anterior.
- La existencia de páginas con AJAX y otras sin esta tecnología hace que el usuario se desoriente.
- Problemas con navegadores antiguos que no implementan esta tecnología.
- No funciona si el usuario tiene desactivado el JavaScript en su navegador.
- Requiere programadores que conozcan todas las tecnologías que intervienen en AJAX.
- Dependiendo de la carga del servidor se pueden experimentar tiempos tardíos de respuesta que desconciertan al visitante (AJAX Ya).

Para implementar aplicaciones en tiempo real utilizando Ajax, es necesario aplicar una técnica conocida como encuestado ("polling"), la cual consiste en realizar peticiones periódicas al servidor en busca de notificaciones recientes. En la mayoría de los casos no se conoce con exactitud cuando el servidor produce nuevos resultados que puedan ser de interés para el cliente, por tanto, al estar realizando peticiones constantemente, es posible que el cliente obtenga resultados en blanco. Si al contrario, las peticiones se espacian en tiempo, es posible entonces que las respuestas que se obtienen no sean lo más actuales posible. El cliente debe decidir el tiempo con el cual estará realizando las peticiones ya que éstas tienen un costo notable en cuanto a tráfico de red debido a las características de HTTP. Sin embargo, en el caso de que el servidor produzca actualizaciones de forma periódica, se puede sincronizar el cliente Ajax con el servidor de forma tal que no se obtengan respuestas en blanco.

2. Comet

En el modelo estándar HTTP, el servidor no puede iniciar la conexión ni enviar una respuesta no solicitada al cliente, por tanto, para recibir eventos asincrónicos tan pronto como sea posible, el cliente debe encuestar al servidor periódicamente en busca de nuevo contenido. Sin embargo, el encuestado continuo consume un significativo ancho de banda al forzar los paquetes de solicitudes/respuestas cuando no hay datos disponibles. También puede ser ineficiente porque reduce la capacidad de dar respuesta de la aplicación ya que la información es encolada hasta que el servidor recibe la próxima solicitud de encuesta del cliente. Para mejorar esta situación varios mecanismos de programación han sido implementados recientemente y son frecuentemente agrupados bajo la etiqueta común de "Comet". Tales mecanismos pueden enviar datos a los clientes de manera más actualizada mientras se evita la latencia experimentada por las aplicaciones clientes debido al frecuente abrir y cerrar de conexiones para el encuestado periódico del servidor en busca de información.

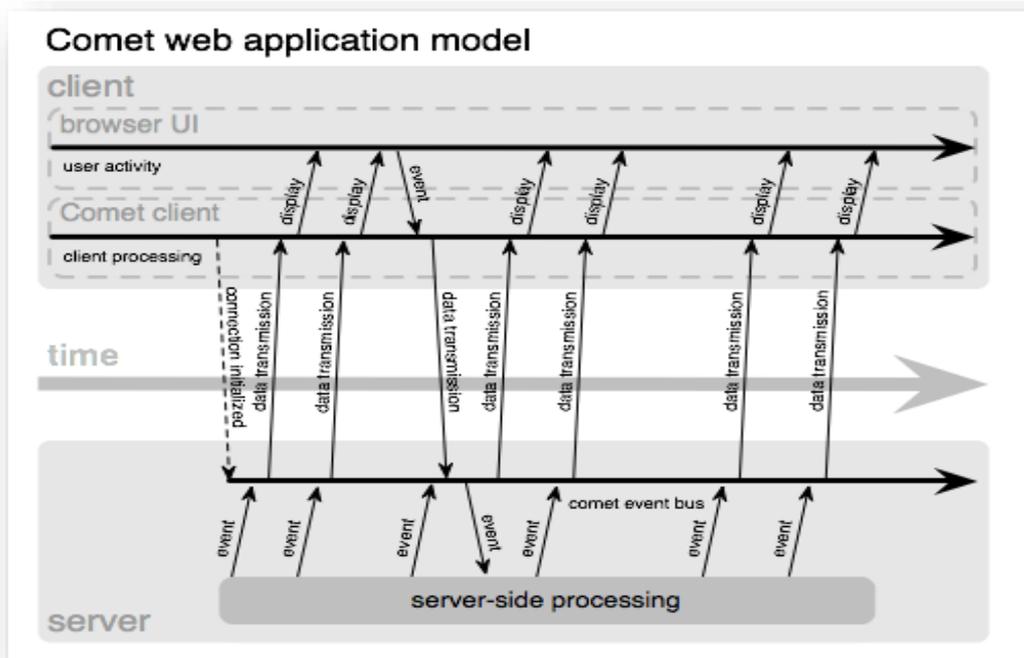


Figura 2. Esquema de funcionamiento de Comet.

Los dos mecanismos más comunes de envío de información por parte de los servidores son "Long Polling" y "HTTP Streaming":

Long Polling (Encuestado Largo)

El servidor intenta mantener abierta (no responder inmediatamente a) cada solicitud HTTP, respondiendo solamente cuando hay eventos que enviar. De esta manera, siempre hay una solicitud pendiente disponible para usar en el envío de información de eventos al ocurrir, minimizando así la latencia en el envío de mensajes.

El ciclo de vida básico del mecanismo Long Polling es el siguiente:

1. El cliente realiza una solicitud inicial y queda a la espera de una respuesta.
2. El servidor demora la respuesta hasta que una actualización esté disponible, o un estado o "timeout" ocurra.
3. Cuando una actualización esté disponible, el servidor envía una respuesta completa al cliente.
4. El cliente típicamente envía una nueva solicitud de encuestado largo.

El mecanismo Long Polling puede ser aplicado tanto a conexiones HTTP persistentes como a no persistentes. El uso de conexiones HTTP persistentes evita el sobre-encabezado adicional de establecer una conexión TCP/IP para cada encuestado largo.

Deficiencia del mecanismo Long Polling:

- Sobre-encabezamiento: Con la técnica Long Polling, cada solicitud y cada respuesta es un mensaje HTTP completo y en consecuencia contiene un conjunto completo de cabeceras HTTP en el empaquetado del mensaje. Para mensajes pequeños e infrecuentes, los encabezados pueden representar un alto porcentaje de la información transmitida. Esto no introduce problemas técnicos significativos si el MTU de la red permite que toda la información incluyendo el encabezado HTTP, quepa en un solo paquete IP. Por otro lado puede introducir problemas de negocio relacionado con el costo de la información

tales como la cantidad de información transmitida que puede ser significativamente mayor que la carga útil real transmitida por HTTP.

XMPP

XMPP (Extensible Messaging and Presence Protocol) es un conjunto de tecnologías para mensajería instantánea, presencia, salones de chat, llamadas de audio y video, y ruteo genérico de información XML. Fue desarrollado originalmente en la comunidad de código abierto de Jabber para proveer una alternativa abierta, segura, libre de spam y descentralizada para los servicios de mensajería instantánea que eran privativos en aquellos momentos (MSN Messenger de Microsoft y AIM de AOL, entre otros). XMPP ofrece varias ventajas claves sobre otros servicios:

- **Abierto:** Los protocolos XMPP son libres, gratis, públicos y fácil de entender, además existen múltiples implementaciones en forma de cliente, servidor, componentes de servidor y librerías de códigos.
- **Estándar:** La IETF (Internet Engineering Task Force) formalizó el núcleo de los protocolos de flujo de información XML como una tecnología aprobada para mensajería instantánea. Las especificaciones de XMPP fueron publicadas como RFC 3920 y RFC 3921 en 2004, y la XMPP Standards Foundation continúa publicando muchas series XEP.
- **Probado:** La primera tecnología Jabber/XMPP fue desarrollada por Jeremie Miller en 1998 y ahora son bastante estables; cientos de desarrolladores están trabajando en estas tecnologías, hay cientos de miles de servidores XMPP corriendo en Internet hoy en día, y millones de personas utilizan XMPP para la mensajería instantánea a través de servicios públicos tales como Google Talk.
- **Descentralizado:** La arquitectura de la red XMPP es similar a la del correo electrónico; como resultado, cualquiera puede correr su propio servidor XMPP, permitiendo a individuos y a organizaciones tomar el control de su experiencia en comunicaciones.
- **Seguro:** Cualquier servidor XMPP puede ser aislado de la red pública y presenta una seguridad robusta usando SASL y TLS que están incorporados en el funcionamiento base de las especificaciones XMPP, y las redes XMPP están virtualmente libres de spam.
- **Extensible:** Usando el poder de XML, cualquiera puede construir funcionalidades personalizadas sobre los protocolos base; para así mantener la interoperabilidad. Extensiones comunes son publicadas en las series XEP, pero tales publicaciones no son requeridas y las organizaciones pueden mantener sus extensiones privadas si así lo desean.
- **Flexible:** Las aplicaciones XMPP más allá de la mensajería instantánea incluyen manejo de red, sindicación de contenido, herramientas de colaboración, compartición de archivos, juegos, monitoreo de sistemas remotos, servicios Web, computación en la nube y mucho más.
- **Diverso:** Un amplio rango de compañías y proyectos de código abierto usan XMPP para construir y desarrollar aplicaciones en tiempo real.

HTTP Streaming

El servidor mantiene una solicitud abierta indefinidamente; eso significa que nunca termina la solicitud o cierra la conexión, incluso después de enviada la información al cliente. Este mecanismo reduce significativamente la latencia de la red debido a que el cliente y el servidor no necesitan abrir y cerrar conexiones.

El ciclo de vida básico de una aplicación que utiliza “HTTP Streaming” es la siguiente:

1. El cliente hace una solicitud inicial y queda a la espera de una respuesta.
2. El servidor demora la respuesta hasta que una actualización esté disponible, o un estado o “timeout” ocurra.

3. Cuando una actualización esté disponible, el servidor la envía hacia el cliente como parte de la respuesta.
4. Los datos enviados por el servidor no finalizan la solicitud o la conexión. El servidor retorna al paso 3.

El mecanismo HTTP Streaming está basado en la capacidad del servidor de enviar varios pedazos de información en la misma respuesta sin terminar una solicitud o la conexión. Este resultado puede ser obtenido en ambos servidores HTTP/1.0 y HTTP/1.1.

Problemas de HTTP Streaming:

- Intermediarios de red: El protocolo HTTP permite intermediarios (proxies, gateways, etc.) que se involucren en la transmisión de la respuesta del servidor al cliente. No es requisito para un intermediario el reenviar inmediatamente una respuesta parcial y es legal para él acumular la respuesta completa antes de enviar algún dato al cliente. HTTP Streaming no funcionará con tales intermediarios.
- Client Buffering: No hay ningún requisito en las especificaciones existentes de HTTP para una librería cliente que haga que la información de una respuesta parcial de HTTP esté disponible en una aplicación cliente. Por ejemplo, si cada pedazo de respuesta contiene llamados a funciones JavaScript, no hay requisitos para que el navegador las ejecute hasta que sea recibida la respuesta completa. Sin embargo, en la práctica la mayoría de los navegadores ejecutan JavaScript recibido en respuestas parciales, pero algunos requieren un desbordamiento de pila que inicie la ejecución, así que bloques de espacios en blanco pueden ser enviados para lograr un desbordamiento de pila.
- Técnicas de empaquetado: Usando HTTP Streaming los mensajes de algunas aplicaciones pueden ser enviados dentro de una sola respuesta HTTP. La reparación del flujo de la respuesta en varios mensajes necesita ser realizada a nivel de aplicación y no a nivel de HTTP. En particular no es posible usar trozos HTTP como delimitadores de los mensajes de aplicación ya que los proxies intermediarios pueden rearmar el mensaje (por ejemplo al combinar diferentes trozos en uno más grande). Este problema no afecta la técnica de Long Polling la cual provee una técnica canónica de empaquetado: cada mensaje de aplicación puede ser enviado en una respuesta HTTP diferente.

3. WebSocket

WebSocket permite establecer un canal de comunicaciones bidireccional. En contraste con los eventos enviados por el servidor (SSE), el protocolo WebSocket no está construido sobre HTTP. Sin embargo, define un comportamiento de saludo para cambiar la existente conexión HTTP para una conexión WebSocket de nivel más bajo. Este protocolo no trata de simular un canal de envío de información a partir del servidor sobre HTTP. Solamente define un protocolo de empaquetado sobre TCP. De esta manera WebSocket permite la comunicación en dos sentidos de forma nativa.

El protocolo tiene dos partes: el saludo y la transferencia de datos. El saludo de apertura está orientado a ser compatible con software del lado del servidor e intermediarios basados en HTTP, de forma tal que un solo puerto puede ser usado por los clientes HTTP y por los clientes WebSocket comunicándose con el mismo servidor.

Después de recibir el encabezado de respuesta HTTP, la información será transmitida de acuerdo con el protocolo WebSocket. Esto significa en este punto que

solamente serán transmitidos paquetes WebSocket por la red. Un paquete puede ser enviado en cualquier momento y en cualquier dirección.

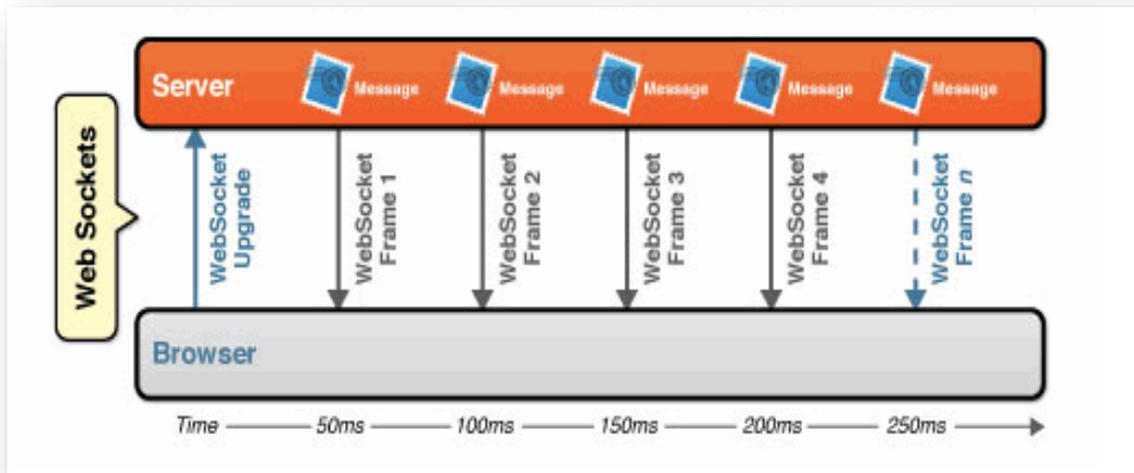


Figura 3. Esquema de funcionamiento de WebSocket.

Para establecer una conexión WebSocket, el cliente y el servidor realizan un “upgrade” del protocolo HTTP al protocolo WebSocket durante el saludo inicial. Una vez establecido, los paquetes de datos WebSocket son enviados en ambos sentidos a través de un canal full-duplex a la misma vez.

Ventajas:

- Elimina el problema del sobre-encabezado de los paquetes, transmitiéndose por la red solamente la información necesaria. En el protocolo WebSocket los mensajes tienen como encabezado entre 4 y 14 bytes, mientras que en HTTP oscilan aproximadamente entre 200 bytes y 2 Kb.
- El gasto involucrado en el manejo de WebSocket es mínimo. Protocolos Comet tales como Bayeux y BOSH, por ejemplo, usan algunos trucos para romper la barrera de HTTP de Petición-Respuesta. Esto fuerza a ambos protocolos a implementar un completo manejo de sesiones y conexiones. Debido al hecho de que WebSocket no está implementado sobre HTTP, nunca caerá en los problemas causados por las limitaciones de dicho protocolo.
- Los mensajes pueden viajar en ambos sentidos a la misma vez.
- WebSockets está diseñado para funcionar sobre la infraestructura de red existente, por tanto no tiene problemas con proxies intermediarios.

Conclusiones

Con Websocket escribir aplicaciones Web en tiempo real altamente interactivas se convierte en una tarea sencilla. La API de Websocket es fácil de entender y de usar, y el gasto involucrado en el manejo es mínimo. Debido al hecho de que

dicho protocolo funciona sobre el protocolo TCP no tiene que recurrir a los trucos que realizan protocolos Comet populares tales como Bayeux o BOSH. Simular un canal bidireccional sobre HTTP lleva a protocolos ser más complicados y menos eficientes, especialmente si solo hay una pequeña cantidad de datos a transmitir, tales como pequeños eventos de notificaciones; el gasto en cuanto a tráfico de red asociado a protocolos clásicos de Comet es muy alto con respecto a Websocket. Además, Websocket usa los mismos puertos que usan las conexiones bajo el estándar HTTP, que es ampliamente utilizado, por lo que se adapta a las condiciones del mercado y promete ser el sustituto de este último. Se recomienda entonces tener en cuenta en su selección al protocolo Websockets cuando se trata de garantizar comunicación en tiempo real sobre la Web.

Referencias

Definición.de. Internet. [En línea] [Citado el: 22 de Noviembre de 2012.] <http://definicion.de/internet/>.

Gillies, Donald. Real-Time System References and Reading Materials. Vancouver, British Columbia, Canada : s.n., 1995.

Slideshare.net. [En línea] [Citado el 2 de Abril de 2014.] <http://www.slideshare.net/000kmi000/conceptos-bsicos-de-internet-y-sus-aplicaciones>.

Universidad de Alicante. Accesibilidad Web. [En línea] [Citado el 2 de Abril de 2014.] <http://accesibilidadweb.dlsi.ua.es/?menu=accesibilidad-aplicaciones-ajax>.

AJAX Ya. [En línea] [Citado el: 13 de Noviembre de 2012.] <http://www.ajaxya.com.ar/temarios/descripcion.PHP?cod=34&punto=2>.



www.sociedadelainformacion.com

Edita:



Director: José Ángel Ruiz Felipe
Jefe de publicaciones: Antero Soria Luján
D.L.: AB 293-2001
ISSN: 1578-326x