



UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA

DEPARTAMENTO DE ELECTRÓNICA



Estructura de Computadores I

“Protocolo XMPP”

Nombres	Rol
Cristian Acuña S.	2921006-3
Manuel Almendra V.	2921019-5
Hugo Herrera L.	2921020-9
Profesor: Agustín González	
Ayudante: Evandry Ramos	
Fecha: 06 de Septiembre 2013	

RESUMEN

El presente trabajo pretende mostrar los aspectos generales del protocolo XMPP, protocolo de la capa de aplicación originalmente diseñado para mensajería instantánea; esto, mediante las herramientas de análisis entregadas en el presente curso, y haciendo uso a referencias oficiales (RFCs producidos por la IETF) y herramientas de software para el análisis de protocolos (Wireshark).

INTRODUCCIÓN

XMPP (Extensible Messaging and Presence Protocol) es un protocolo de la capa de aplicación diseñado originalmente por Jeremie Miller en 1998 para la mensajería instantánea (originalmente llamado *Jabber*). El protocolo consiste, a grandes rasgos, en un sistema de mensajería presencial el cual usa datagramas en XML para implementar las funcionalidades necesarias en este tipo de mensajería. Es un protocolo libre y abierto (según el standard de software libre). Éste fue creado con la intención de unificar los servicios de mensajería instantánea, dado que para su época no existía un protocolo que pudiese satisfacer dicha necesidad, lo que llevaba a situaciones incómodas como el uso de múltiples clientes (uno por cada servidor/servicio) lo cual era altamente ineficiente. Lo anterior motiva a su creador a desarrollar un protocolo el cual pudiese convertirse en el standard ‘*de facto*’, siendo éste un protocolo abierto y libre de implementación; cualquier persona podía (y puede) realizar un ‘*fork*’ del cliente que se adecuase a las necesidades del desarrollador/usuario.

1. CÓMO FUNCIONA XMPP

Arquitectura

Como se ha mencionado el protocolo XMPP tiene como una de sus funciones hacer la mensajería instantánea (IM) entre los usuarios de manera accesible y menos compleja posible para ellos. De este modo esto posee una arquitectura Cliente-Servidor descentralizada, es decir, la comunicación entre clientes no hay necesidad que pase por un servidor central, sino que cualquier cliente tiene la capacidad de crear su propio servidor XMPP y adherirse a la red, esto ayuda a que no haya saturación de la conexión (pensando en que XMPP está pensada para una gran cantidad de

usuarios que requieren de la conexión sea rápida y confiable, como debe ser la mensajería instantánea) y también hace que no exista un único punto de falla.

Cuando un “Cliente A” envía un mensaje XMPP a algún otro cliente “Cliente B” que posee un servidor XMPP distinto al del primero, el cliente inicialmente se contacta con su servidor XMPP y este servidor se contacta directamente con el servidor del cliente B (ver figura 1), a diferencia de otros protocolos como la de “Email network” (ver figura 2) que no va necesariamente directo desde el servidor de A hasta el servidor de B, sino que se va haciendo una ruta consultando en los distintos servidores por el servidor B y más a diferencia aún que el protocolo World Wide Web que, a grandes rasgos, al requerir una página consulta a nuestro servidor Local por el servidor que posee nuestra información requerida, si éste no se encuentra en su base de datos, el servidor no se contacta con otros servidores, sino que es el navegador el que se conecta con otros servidores a través de sus direcciones que posee en su base de datos (ver figura 3).

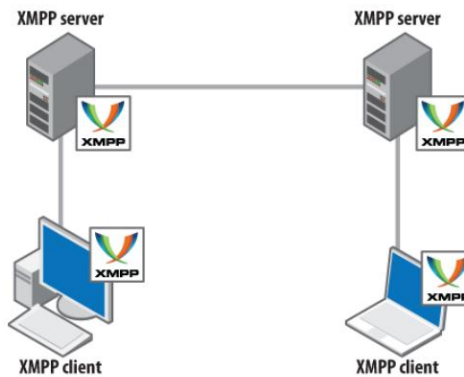


Figura 1 Arquitectura XMPP, conexión directa entre servidores.

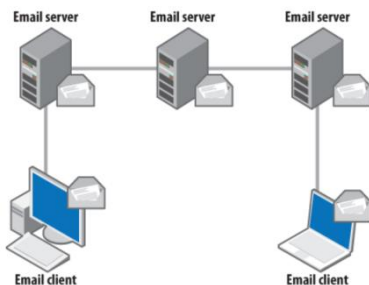


Figura 2 Arquitectura Email Network, enrutamiento al servidor destino

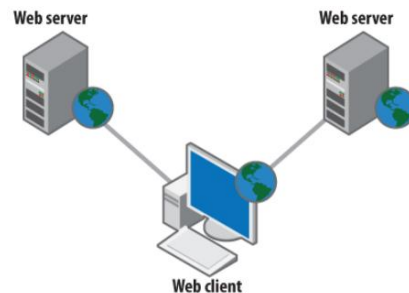


Figura 3 Arquitectura World Wide web, sin conexión entre servidores

Direcciones

XMPP se basa normalmente en el sistema de Servidor de nombres de dominio (DNS) para proporcionar la dirección a un usuario, en lugar de usar una dirección IP (por ser menos complejo de recordar). El puerto destino estándar utilizado para el recibimiento de mensajes XMPP es el 5222.

En XMPP se establece el concepto Jabber ID como toda “entidad alcanzable en cualquier punto final de una red XMPP”. Éste está compuesto por un nombre de **usuario**, un **dominio** y un **recurso** (“Full JID” si es que posee recurso), de la siguiente forma:

JID = < nombredeusuario@dominio >/recurso

Como por ejemplo:

Usuario Whatsapp:

JID: 56987654321@whatsapp.net/Whatsapp

En Whatsapp el **nombre de usuario** no es más que el código del país y el número del usuario , el **dominio** es el servidor en el que reside el usuario y es generalmente otorgado por el software haciendo alusión al nombre de éste , en este caso “whatsapp.net”, el **recurso** no es más que una cadena de referencia que identifica al cliente que está utilizando el usuario, este recurso se utiliza para el enrutamiento a esa conexión en lugar de cualquier otra conexión que se podría tener abierta en ese momento, éste es a menudo el nombre de su ordenador, su ubicación o el software de cliente que está usando.

Envío de la información

Cuando se presenta la pregunta: “¿Cómo funciona un protocolo de la capa de aplicación?”, es necesario responderse la pregunta: ¿Qué es lo que se envía y cómo se envía?

Lo que se envía es información como texto XML (*eXtensible Markup Language*, lenguaje utilizado para almacenar datos en forma legible que mayormente se usa para la lectura de datos en diferentes protocolos de internet).

Cuando se inicia una sesión con un servidor XMPP, se genera una conexión TCP persistente (“Always on”) y luego se negocia un **stream** XML con el servidor. (stream es la distribución de

multimedia en la red en donde el usuario tiene acceso al producto al mismo tiempo que se descarga, almacenándolo en un búfer de datos para esto.)

Una vez establecido el stream XML con el servidor, éste último también abre un stream de retorno con el cliente, es decir, hay una corriente en ambos sentidos. De este modo tanto el cliente como el servidor pueden intercambiar de manera asincrónica y en ambos sentidos tres fragmentos especiales XML por el stream de manera infinita, llamados stanzas XML, que son las unidades básicas para entender el protocolo XMPP. Estos son:

1. Message (<message/>): Es un mecanismo de envío de un mensaje de una entidad a otra habiendo 5 tipos: normal , chat, groupchat, headline, error.

2. Presence (<presence/>): Es un mecanismo para mostrar la presencia de un usuario a múltiples otros usuarios, ya sea como mostrar estado en línea, cambio de estado, de nickname, o algún otro dependiendo del software.

3. Iq (<iq/>): (Info/Query) Es una interacción de solicitud-respuesta, con el cual una entidad es capaz de solicitar alguna información a otra entidad pero que no interfieren directamente con la interfaz de la comunicación, como informes de errores. Hay 4 tipos: get, set, result y error. En el ejemplo del **Anexo 1** se ilustra una sesión simplificada en XML de XMPP, incluyendo la interacción entre stream y stanzas, desde el cliente al servidor como viceversa.

Pasarelas

XMPP permite el uso de paralelas o gateway que son ciertos servidores XMPP que logran “traducir” un mensaje XMPP a otros protocolos de mensajería instantánea tales como Windows Live Messenger (programa ya discontinuado), ICQ, Yahoo! Messenger, SMS, Email entre otros. Esto se logra registrándose en un servidor XMPP que ofrezca este servicio, esto no hace necesario el disponer de una cuenta JID (user@dominio) en este servidor, ya que generalmente estos servicios permiten el ingreso mediante cualquier JID de algún otro servicio XMPP, ejemplo: Pidgin.

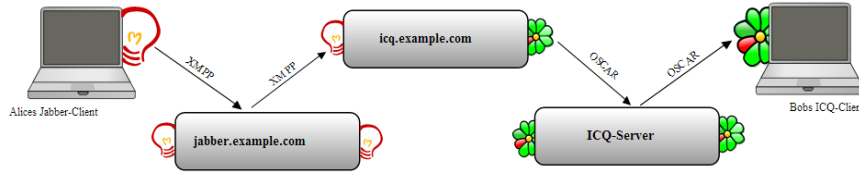


Figura 4 Representación esquemática de una Pasarela XMPP

2. VENTAJAS Y DESVENTAJAS

Ventajas

Abierto: es un protocolo abierto y sencillo de comprender, por lo cual han surgido una gran cantidad de implementaciones de clientes, servidores y librerías para los desarrolladores. La lista de algunos clientes se pueden ver en **Anexo 2**.

Estándar y experiencia: al ser un estándar, todas las implementaciones basadas en el protocolo son compatibles unas con otras. Además se trata de un protocolo lanzado hace 15 años lo que conlleva una vasta experiencia en el campo tecnológico.

Descentralizado: Posee una arquitectura cliente-servidor descentralizada, la cual permite a las compañías montar su propio servidor con el control total de sus comunicaciones, con la posibilidad de conectarse con otros servidores. También permite la comunicación a través de otros servidores (protocolos) a través de *gateways*.

Seguro: soporta seguridad en la capa de transporte, además puede ser instalada en una red privada y complementarse con otros protocolos de seguridad como SALS.

Extensible: ya se han desarrollado un gran número de extensiones, a los cuales se pueden adicionar nuevas de parte de cualquier interesado. Ver en **Anexo 3** para algunas características incorporadas y nuevas aplicaciones del protocolo XMPP.

Flexible: Aunque en un principio las aplicaciones eran orientadas a la mensajería instantánea, en la actualidad el abanico de necesidades a satisfacer ha crecido. Las aplicaciones también se pueden enfocar en compartimiento de archivo, monitoreo de sistemas remotos, servicios web, nubes virtuales, juegos, etc.

Proyección a Futuro: En el tema IoT (Internet of Things), XMPP toma una importante función. . Las fortalezas mencionadas anteriormente dejan a XMPP federarse sobre dominios y proveer de

suscripciones públicas en conjunto con una identificación precisa y una alta seguridad sin comprometer su adaptabilidad es crucial para los sistemas IoT emergentes. Ver las extensiones de XMPP dedicados a este fin en **Anexos 4**.

Desventajas

XMPP presenta una sobrecarga y redundancia de datos de presencia. Alrededor de un 70% del tráfico entre servidores corresponde a datos de presencia, siendo un 60% de estos usados para redundancia.

Además dado que los datos binarios son codificados en un único documento XML, hay poca posibilidad de entregar datos binarios sin modificar. Esto es debido a que al ser codificado en un único documento XML, se hace altamente improbable que la transmisión de datos binarios permanezca en su estado original.

CONCLUSIONES

XMPP es un protocolo que funciona a base de enviar streams de texto encriptado en XML, por lo anterior es que se cuenta con una flexibilidad que permite una constante ampliación. Constantemente se desarrollan nuevas extensiones que son utilizadas en, a su vez, nuevos servicios. Además es de gran utilidad que sea un protocolo abierto (aunque también permite desarrollos privativos, lo cual lo hace atractivo para la industria) y estándar, por lo que cualquier desarrollador puede aportar a otros interesados o bien recibir aportes para complementar su trabajo.

Una de las proyecciones a futuro más importantes es ser la base fundacional para el IoT, concepto revolucionario donde se puede tener un control remoto de cualquier aparato “inteligente” remotamente (similar al concepto de ‘impresora inalámbrica’, pero extendiendo el concepto a otros aparatos y no necesariamente en una LAN), así como también el seguimiento de cualquier objeto con una “etiqueta inteligente” alrededor del mundo. XMPP se muestra como el ideal para sostener este nuevo paradigma de la red.

REFERENCIAS

- “XMPP: The Definitive Guide” Peter Saint-Andre, Kevin Smith, and Remko Tronçon
- Documentos RFC del grupo de trabajo IETF XMPP
- wiki.xmpp.org

ANEXOS

Anexo 1: Ejemplo de sesión simplificada en XML de XMPP

```
Cliente: <stream:stream>
Cliente: <presence/>
Cliente: <iq type="get">
Cliente: <query xmlns="jabber:iq:roster"/>
Cliente: </iq>
Servidor: <iq type="result">
Servidor: <query xmlns="jabber:iq:roster">
Servidor: <item jid="fscavone@uca.edu.py"/>
Servidor: <item jid="jeuzzarru@jeuzzarru.com"/>
Servidor: </query>
Servidor: </iq>
Cliente: <message from="fscavone@uca.edu.py"
to="jeuzzarru@jeuzzarru.com"
type="normal">
Cliente: <body>Esto es un mensaje !</body>
Cliente: </message>
...
...
...
Cliente: <presence type="unavailable"/>
Cliente: </stream:stream>
```


Anexo 2: Lista de clientes de mensajería instantánea (IM) que implementan el protocolo XMPP

- iChat: Cliente de Mensajería instantánea incluido en el Sistema operativo OSX de Apple.
- Jabbear: Cliente de Mensajería instantánea incluido en el Sistema operativo Windows, Web.
- LJTalk: Cliente de Mensajería instantánea utilizado por el producto LiveJournal.
- Psi: Un cliente con licencia GNU GPL que se puede utilizar bajo Windows, Linux, u OSX.
- Gajim: Cliente multiplataforma disponible para Windows y GNU/Linux. Se encuentra programado en Python y para la interfaz utiliza PyGTK. Al igual que Psi es un cliente solo enfocado a Jabber.
- Exodus: Cliente con licencia GNU GPL que se puede utilizar bajo Windows.
- Coccinella: Cliente que proporciona VoIP mediante Jigle y que se puede utilizar bajo Windows, Linux, u OSX.
- Tkabber: Cliente con licencia GNU GPL escrito con Tcl/Tk que se puede utilizar bajo Windows y Linux.
- Champoo Chat: Cliente con licencia GNU GPL escrito en Java, que provee una experiencia de chat tipo IRC, con soporte para canales, conferencias y chats privados. Posee además un cliente de Twitter muy simple, totalmente integrado con el sistema.
- Pidgin (software): Uno de los clientes más usados en GNU/Linux, soporta otros protocolos como el de MSN, Yahoo!, etc.
- Telepathy: La infraestructura de Freedesktop.org para comunicación en tiempo real. Es la predeterminada para Ubuntu, Maemo y Meego.

Anexo 3: Lista de algunas características incorporadas y nuevas aplicaciones del protocolo XMPP

- Encriptación del Canal: Este servicio provee la encriptación de la conexión entre cliente y servidor, o entre dos servidores. Esta característica es fundamental para la construcción de aplicaciones seguras.

- Autenticación: El servicio de autenticación asegura que cualquier entidad que intente comunicarse a través de la red sea primeramente identificado por un servidor, el cual actúa como guardián para el acceso a la red.
- Presencia: Permite averiguar sobre la presencia de algún ente en la red. En su forma más simple, permite conocer si una determinada identidad está disponible para la comunicación “instantánea”; no obstante, puede entregar información más detallada. El sistema se basa en una “suscripción de presencia” entre dos entidades, lo cual permite proteger la privacidad de la información del usuario.
- Listas de contactos: Permite al usuario guardar una “lista” de contactos en el servidor XMPP en uso. El uso más común para este servicio es una “lista de amigos” para mensajería instantánea, pero cualquier entidad que tenga una cuenta en el servidor puede utilizar el servicio para mantener una lista de entidades conocidas o de confianza (por ejemplo, que puede ser utilizado por ‘bots’).
- Mensajes Uno-a-uno: Permite al usuario el envío de mensajes a otra entidad. El uso clásico de este servicio es la mensajería instantánea persona-a-persona; mas los mensajes pueden ser texto XML arbitrario, y dos entidades cualquiera conectadas a la red pueden intercambiar mensajes - podrían ser bots, servidores, componentes, dispositivos, servicios web habilitados para XMPP, o cualquier otra entidad XMPP.
- Mensajería de multiconferencia: Este servicio, permite al usuario unirse a una sala de chat virtual para el intercambio de mensajes entre varios participantes, similar a Internet Relay Chat (IRC). Los mensajes pueden ser texto plano, o pueden contener extensiones XML para funciones más avanzadas, como la configuración de la sala, la votación en las habitaciones, y varios mensajes de control de sesión.
- Notificaciones: Este servicio, permite al usuario generar una notificación y se le entregue a varios suscriptores. Este servicio es similar a la mensajería multiconferencia, pero está optimizado para entrega de tipo “uno-a-muchos” con las suscripciones a canales explícitos o temas específicos (llamados "nodos").

Los anteriores son algunos de los servicios implementados en XMPP; sin embargo, se pueden encontrar muchos más revisando los *XEP (XMPP Extension Protocols)*, definidos en la [XMPP](#)

Standards Foundation (XSF). Con esto, se pueden implementar las siguientes aplicaciones (entre otras):

- **Mensajería Instantánea:** Los sistemas clásicos de mensajería instantánea que la mayoría de la gente está familiarizada combinan tres de los principales servicios: presencia, listas de contactos, y mensajería uno-a-uno. Tales sistemas pueden y a menudo incluyen más servicios y funciones, pero si usted tiene estos tres servicios, puede crear una aplicación de mensajería instantánea escueto.
- **Chat de Grupo:** El servicio de mensajes multiconferencia permite construir sistemas similares al grupo de chat IRC. A menudo, los sistemas de chat de grupo se utilizan para aplicaciones más específicas, tales como sistemas de tiempo real de negociación en el sector financiero, salas de información de socorristas y personal militar y clases virtuales.
- **Gaming:** Combinado con extensiones personalizadas, tanto mensajería uno-a-uno y mensajería multiconferencia permiten crear sistemas de juego simples. Por ejemplo, el servicio Chesspark (<http://www.chesspark.com/>) está construido en su totalidad con XMPP. Otros desarrolladores de juegos están usando XMPP para añadir presencia y contacto con las características de la lista para juegos multipartidistas existentes.
- **Sistemas de control:** La combinación de mensajería uno-a-uno y formularios de datos hace posible implementar sistemas ligeros para el control y la interacción con los sistemas remotos. Las aplicaciones desplegadas en este ámbito incluyen la gestión de red, telemetría científica y control robótico.
- **Geolocalización:** El servicio de notificación XMPP es independiente de carga. Un formato de carga definido es de geolocalización, lo que le permite construir fascinantes aplicaciones basadas en la localización, tales como el seguimiento de vehículos.
- **Middleware y Cloud Computing:** Un número de empresas y grupos de investigación están trabajando activamente en los sistemas basados en XMPP para los servicios de computación, middleware ligero, y la gestión de las infraestructuras de cloud computing. Aunque el uso de XMPP puede ser sorprendente aquí, porque este tipo de aplicaciones se han basado tradicionalmente en tecnologías de mensajería de peso pesado, se ha visto que XMPP comienza a picar lejos en el extremo inferior de este mercado. Esto se debe

principalmente a que las empresas que ya cuentan con una infraestructura de XMPP en funcionamiento se han percatado que puede ser también utilizado para casos de uso no-IM.

- Voz sobre IP (VoIP): La aplicación de Google Talk que puso en marcha en agosto de 2005 popularizó por primera vez el uso de XMPP para el chat de voz. Desde entonces, las extensiones XMPP para servicios de sesión de comunicación (llamadas ‘*Jingle*’) se han formalizado a través de la XSF, y se han implementado y desplegado de la talla de Nokia y el proyecto One Laptop Per Child. Las mismas extensiones también pueden ser utilizados para negociar una amplia gama de tipos de sesión de medios de comunicación, incluyendo vídeo, transferencia de archivos, pizarras, y la edición colaborativa.

Anexo 3: Lista de extensiones de XMPP para resolver problemas de IoT

- Efficient XML Interchange (EXI) Format
- Internet of Things - Sensor Data
- Internet of Things - Provisioning
- Internet of Things – Control
- Internet of Things – Concentrators

Para más detalles ir a la sección de IoT en el sitio [wiki XMPP](#).