

BlueNews: modelo para la propagación de información por medio de la tecnología Bluetooth

Oscar D. Ozaine, Alberto Urbina, Arnoldo Díaz-Ramírez, Heber S. Henández y Juan F. Ibáñez

Departamento de Sistemas y Computación

Instituto Tecnológico de Mexicali

{oscar.ozaine, aurbinae}@gmail.com, {adiaz, heberhdz, pacois}@itmexicali.edu.mx

Paper received on 23/07/12, Accepted on 06/09/12.

Resumen. La alta disponibilidad de dispositivos móviles y el bajo consumo de energía de la tecnología *Bluetooth* ha motivado la proliferación de aplicaciones de cómputo ubicuo. En este artículo se propone *BlueNews*, un modelo para la propagación de información (e.g., noticias) entre usuarios, basado en la tecnología *Bluetooth*. El modelo propuesto tiene dos modos de operación: modo *Cliente/Servidor* y modo *Servent*. En el primer modo, la aplicación Servidor busca periódicamente dispositivos móviles con el servicio *BlueNews* instalado (Clientes), para actualizarlos con la información más reciente que se ha publicado. En el modo *Servent*, los dispositivos pueden operar como clientes y servidores, intercambiando información entre sí. De esta manera, un dispositivo que no ha tenido conexión al servidor pueda recibir la información más reciente desde otro dispositivo cliente. Se presentan también los detalles de la implementación de un prototipo con el modelo propuesto.

Palabras Clave: Cómputo UbiCuo, Redes Ad hoc, Bluetooth.

1 Introducción.

Las computadoras personales y los dispositivos estacionarios de comunicación son cada vez menos usados y en algunos casos se han vuelto obsoletos. Esto se debe a que estamos entrando a una nueva era en el área de la computación: la era del *cómputo ubicuo*. El nombre de *cómputo ubicuo* fue utilizado por primera vez por Mark Weiser [13] para definir la existencia de dispositivos de cómputo distribuidos en un entorno físico, desplegados de manera *invisible* para los usuarios y que sin embargo, llevan a cabo tareas en beneficio de los usuarios.

Con la proliferación y disponibilidad de dispositivos móviles (e.g., teléfonos móviles, tabletas, etc), los usuarios pueden intercambiar información almacenada en sus dispositivos o disponible en la Web. Sin embargo, esta capacidad puede verse afectada por la estructura de la red de comunicación y su dependencia de un punto de acceso centralizado. En este escenario, las redes inalámbricas de propósito es-

pecífico o *Mobile Ad hoc Networks* (MANETs) tienen gran relevancia, ya que proporcionan un método de interconexión más flexible entre dispositivos [7]. Existen diversos estándares de comunicación con los que pueden implementarse redes MANET. Dos de ellos destacan: el estándar IEEE 802.11 para WLANs y la especificación Bluetooth [2] para comunicación inalámbrica de corto alcance. La velocidad de enlace, el rango de comunicación y el poder de transmisión de Bluetooth fueron elegidos para soportar bajo costo y eficiencia en el consumo de energía [9].

En este artículo se presenta *BlueNews*, un modelo para la propagación de mensajes por medio de dispositivos móviles, basado en el uso de Bluetooth. Con *BlueNews*, una organización puede difundir rápidamente información de interés para sus integrantes utilizando el paradigma de cómputo ubicuo. Además, en el artículo se presentan aspectos de implementación de un prototipo utilizando Android.

Este documento está organizado de la siguiente manera. En la Sección 2 se describen los conceptos básicos de las MANETs. Los aspectos más relevantes de tecnología Bluetooth se presentan en la Sección 3. En la Sección 4 se describe el modelo *BlueNews*, así como los protocolos para cada modo de operación. En la sección 5 se discute la implementación de un prototipo del modelo propuesto, utilizando dispositivos con el sistema operativo Android. Finalmente, en la Sección 6 se presentan las conclusiones y el trabajo futuro.

2 MANETs.

Una MANET, también conocida como red *Ad hoc*, es una colección de nodos inalámbricos que pueden organizarse dinámicamente en una topología arbitraria y temporal, para formar una red sin necesidad de usar una infraestructura predefinida. En la Fig. 1 (parte superior) puede observarse una red con infraestructura (e.g., Wi-Fi), que depende de la existencia de un nodo central o punto de acceso. En contraste, una MANET permite la comunicación entre los dispositivos sin la necesidad de un punto de acceso (Fig. 1, parte inferior). En una MANET, los nodos se pueden comunicar utilizando otros nodos como encaminadores si el destino está fuera del rango de comunicación.

La proliferación de dispositivos con capacidades de comunicación inalámbrica ha motivado el desarrollo de aplicaciones móviles (*mobile apps*), así como la necesidad de comunicar dispositivos entre sí de manera sencilla y eficiente. En este contexto, las MANETs tienen gran relevancia, ya que permiten la creación de una red sin la necesidad de una infraestructura predefinida.

Para implementar una red *Ad hoc* se han propuesto diversas tecnologías, entre las que destacan las definidas en los estándares Bluetooth, UWB Ultra-Wideband, HiperLAN/1, HiperLAN/2 y el IEEE 802.11 [7]. La elección de la tecnología de comunicación debe considerar diversos factores, como el costo del hardware, el consumo de energía, el rango de cobertura de la señal, por mencionar algunos. La tecnología Bluetooth representa una de las opciones más interesantes para la implementación de MANETs.

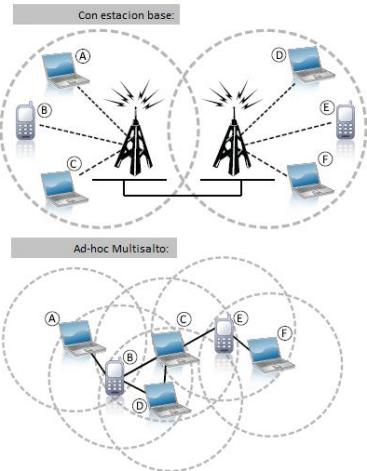


Figura 1. Comparación entre una red con infraestructura y una MANET.

3 Tecnología Bluetooth.

El estándar 802.15.1 [4], también conocido como Bluetooth, se creó para resolver un simple problema: reemplazar los cables usados en dispositivos móviles con comunicación de radio. Es un estándar abierto de conectividad inalámbrica para la transferencia de voz y datos entre dispositivos de comunicación y las computadoras personales, permitiendo a los usuarios la creación de *redes de área personal* (PANs) y de redes *Ad hoc*.

La tecnología Bluetooth opera en la banda sin licencia *Industrial, Científica y Médica* (ISM), que va desde los 2.4 a 2.485 GHz, usando el espectro de propagación, salto de frecuencias y la señal full-dúplex a un rango nominal de 1600 saltos/segundo. La banda ISM 2.4 GHz está disponible y sin licencia en la mayoría de los países. Su rango nominal es de 10 cm a 10 m en teoría, pero puede ser extendido a 100 m a través del incremento de poder de transmisión (con una mayor distorsión). Para reducir la interferencia al mínimo, Bluetooth utiliza saltos de frecuencia adaptados (AFH), que fueron diseñados para detectar otros dispositivos en el espectro y para evitar interferencias cuando muchos dispositivos usan el mismo canal. Esto se logra por saltos de señal entre 79 frecuencias en intervalos de 1 MHz.

Bluetooth soporta hasta 3 canales asíncronos de voz de 64 Kbyte/s cada uno, y datos asíncronos hasta 723.2 Kbyte/s (asimétricos) o 433.9 Kbyte/s (simétricos) [5]. También es un módulo de radio de bajo consumo, que consume hasta 97% menos energía que un teléfono móvil. Además, es *inteligente*, ya que cuando el tráfico de datos disminuye, el transmisor adopta modos de bajo consumo de energía.

La topología de comunicación Bluetooth se basa en la creación de *piconets*, que son subredes formadas por un máximo 8 dispositivos y en las que siempre debe existir un nodo *maestro* y hasta 7 *esclavos*. También se pueden formar *scatternets*, que son la unión de varias piconets por medio del dispositivo maestro, que es el único que se puede comunicar con otros piconets.

Las especificaciones definen todas las capas de la *Pila de Protocolos Bluetooth*. Las tres capas bajas: *Radio*, *Banda Base* y el *Manejador de Enlaces* están agrupados en un subsistema denominado *Controlador Bluetooth* y donde se encuentra la capa L2CAP. La capa SDP y las capas más altas se conocen como *anfitrión Bluetooth*. Esta agrupación de las capas requiere una interfaz de comunicación entre el controlador Bluetooth y el anfitrión Bluetooth. A esta interfaz se le conoce como HCI (*Host Controller Interface*). Las especificaciones Bluetooth hacen posible la compatibilidad entre diferentes sistemas Bluetooth por medio de la definición de protocolos de mensajes, que son intercambiados entre las capas equivalentes. Además, determinan una interfaz común entre los controladores y los anfitriones Bluetooth para hacer compatible los diferentes subsistemas.

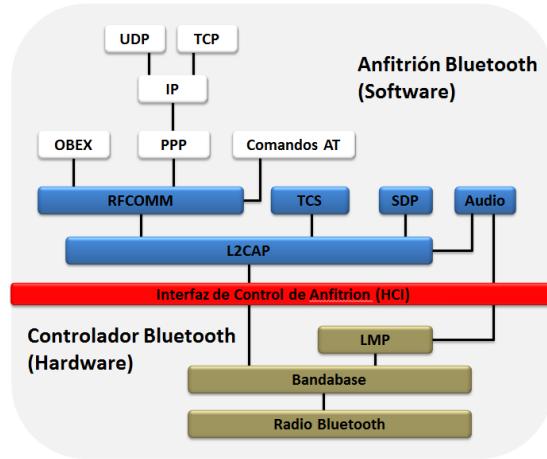


Figura 2. Pila de Protocolos Bluetooth.

Una característica atractiva es que los fabricantes pueden implementar sus propias capas de protocolos propietarios. Por lo tanto, la especificación abierta de Bluetooth incrementa el número de aplicaciones que pueden ser beneficiadas por sus capacidades. Debido a las muchas ventajas de este protocolo, se han desarrollado muchas *Interfaces de Programación de Aplicaciones* (APIs) para construir aplicaciones.

BlueZ [3] es la *Pila de Protocolos Bluetooth* oficial de Linux. Se desarrolló inicialmente por Qualcomm y se distribuye bajo la *Licencia Pública General* (GPL). BlueZ es parte del núcleo oficial de Linux desde la versión 2.4. BlueZ es una gran herramienta para desarrollar aplicaciones de Bluetooth eficientes [12]. La API de BlueZ fue elegida para implementar BlueNews debido a su gran desempeño. Android provee soporte a la pila de Bluetooth desde su versión 2.0 (Android SDK 5) y permite que una aplicación acceda a la funcionalidad Bluetooth por medio de la API de Bluetooth de Android, con el paquete *android.bluetooth*. Android SDK no proporciona funcionalidad para el protocolo L2CAP, que es requerido por la especificación Bluetooth HID. Sin embargo, para utilizar el protocolo L2CAP, que sí es pro-

porcionado por BlueZ, es posible que una aplicación Android se comunique directamente con la pila de protocolos BlueZ a través del *Kit de Desarrollo Nativo* (NDK) [1].

4 BlueNews.

BlueNews es un modelo basado en Bluetooth, que permite la difusión de información de interés de un conjunto de usuarios de una organización. El modelo contempla que las aplicaciones puedan operar en dos modos. En el modo *Cliente/Servidor*, existe un servidor que contiene y administra la información que se desea transmitir. Cada mensaje incluye parámetros como la fecha de inicio y final de propagación del mensaje, prioridad y perfil de usuario al que va dirigido, entre otros. Los dispositivos clientes que estén en el rango de alcance del servidor recibirán los mensajes recientes de manera *invisible* para el usuario.

El modo de operación *Servent*, utiliza el paradigma de comunicación *peer-to-peer* (P2P) [11], en el que los nodos pueden operar como clientes y servidores. En este modo, los dispositivos pueden intercambiar información vigente sin necesidad de conectarse al servidor. La Fig. 3 muestra la arquitectura del modelo BlueNews.

Para ilustrar los beneficios del modo *Servent*, puede considerarse el caso de una universidad. Si un estudiante o profesor no asistió al campus universitario el día en que una noticia importante fue publicada, tiene la posibilidad de recibirla si en algún sitio su dispositivo móvil entra en el rango de alcance de otro dispositivo con el servicio BlueNews y que contenga la información actualizada.

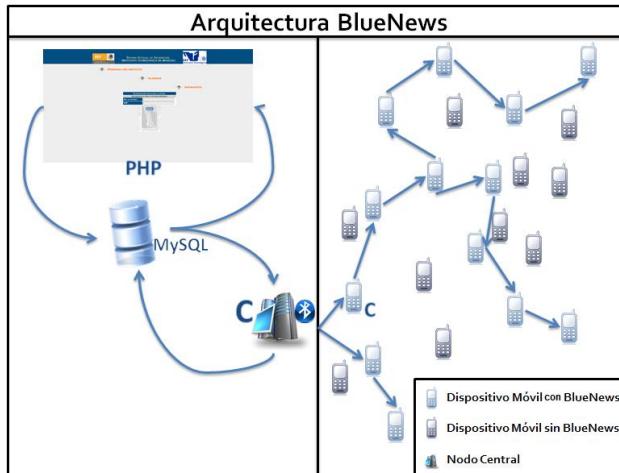


Figura 3. Arquitectura del modelo *BlueNews*.

4.1 Modo de operación Cliente/Servidor.

El algoritmo del modelo BlueNews cuando opera en modo *Cliente/Servidor*, como puede observarse en la Fig. 4, consiste de siete pasos:

1. El nodo central inicia la búsqueda de dispositivos (clientes) en el entorno.
2. Ya que encuentra un dispositivo remoto, se definen los roles *Cliente/Servidor*.
3. Se realiza la conexión L2CAP al dispositivo remoto.
4. El nodo central verifica si el dispositivo remoto tiene el servicio BlueNews. Si el dispositivo remoto no tiene el servicio BlueNews, ir a paso 7.
5. El dispositivo remoto envía al servidor el último mensaje que recibió.
6. Si existen nuevos mensajes, el servidor los envía al dispositivo remoto.
7. Termina la conexión L2CAP.



Figura 4. Diagrama de flujo del modo Cliente/Servidor.

4.2 Modo de Operación Servent.

El algoritmo utilizado por el modo *Servent* se muestra en la Fig. 5. El modo *Servent* considera la existencia de dos *hilos* (*threads*) de ejecución. El primer hilo de ejecución, denominado *hilo de investigación*, busca periódicamente en el entorno dispositivos con la aplicación BlueNews. El algoritmo de este hilo consta de seis pasos:

1. El nodo *Servent* inicia la búsqueda de dispositivos en el entorno.
2. Ya que encuentra un dispositivo remoto, define los roles Cliente/Servidor (e.g., el nodo con el mayor valor de su dirección MAC adquiere el rol de servidor).
3. Cuando los roles están definidos, se realiza la conexión al dispositivo remoto que encontró.
4. Se verifica si el dispositivo remoto tiene el servicio BlueNews. Si el dispositivo remoto no tiene el servicio BlueNews, ir a paso 6.
5. Se lleva a cabo la actualización.

6. Termina la conexión L2CAP.

El segundo hilo de ejecución, denominado hilo *listen*, espera por una conexión de algún dispositivo con la aplicación BlueNews, y consta de 6 pasos:

1. El nodo *Servent* siempre se encuentra en espera de nuevas conexiones entrantes.
2. Al momento de recibir una conexión, se definen los roles Cliente/Servidor.
3. Se lleva a cabo la conexión L2CAP al dispositivo remoto.
4. El dispositivo remoto envía una verificación de servicio, al que éste responde que si tiene el servicio BlueNews.
5. Se lleva a cabo la actualización.
6. Termina la conexión L2CAP.

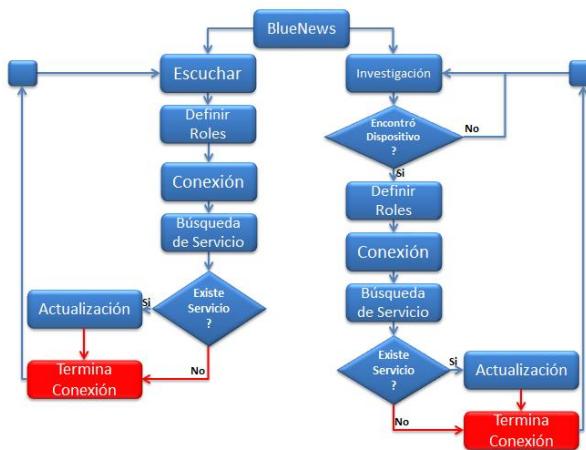


Figura 5. Diagrama de flujo del modo *Servent*.

Cuando un dispositivo Bluetooth ofrece un servicio, actúa como servidor. En esta situación es necesario especificar el tipo de servicio que se está ofreciendo y registrararlo. Los dispositivos Bluetooth utilizan el *Protocolo de Descubrimiento de Servicios* (SDP), en donde un cliente puede descubrir los servicios disponibles en otros dispositivos Bluetooth, así como los atributos de los servicios.

En contraste, el modelo BlueNews no registra un servicio Bluetooth. En su lugar, utiliza una aproximación similar al protocolo Gnutella [6]. En BlueNews, una vez que el *Servent* se ha identificado y ha establecido una conexión con el nodo central u otro *Servent* remoto, envía al *Servent* remoto una cadena de caracteres como petición de servicio: "BlueNews". El *Servent* con el servicio BlueNews debe responder con "BlueNewsOk". Este esquema reduce el consumo de recursos y tiempo ya que evita múltiples conexiones SDP para búsqueda de servicios.

Existen varios protocolos P2P, entre los que destacan Gnutella [10] y Freenet. En [8] se presentan los resultados de una evaluación comparativa de estas dos arquitecturas. Gnutella se desarrolló para el descubrimiento de nodos en la red y el intercambio de ficheros. Freenet tiene una topología de red similar a Gnutella, excepto

que su propósito principal es crear un sistema de almacenamiento de información global seguro. Para nuestro modelo se eligió implementar una aproximación del protocolo Gnutella ya que está orientado más al intercambio de información que al almacenamiento. Además, permite que la aplicación funcione de una manera más eficiente al hacer la búsqueda de servicios.

Para determinar el rol del *Servent* (servidor o cliente), la dirección MAC de cada dispositivo Bluetooth encontrada es utilizada. El *Servent* que tenga la dirección con menor valor actúa como cliente y debe hacer una petición de conexión al *Servent servidor*. Además, cada dispositivo debe ser capaz de atender múltiples conexiones entrantes y salientes. Esto puede hacerse implementando una aplicación multi-hilos.

Después de la petición de conexión, el *Servent cliente* verifica si la dirección remota pertenece a un dispositivo que ya se haya registrado en una lista *sin servicio BlueNews*. Esto se hace con el propósito de evitar conexiones repetidas con dispositivos que no tienen el servicio BlueNews. Si la dirección remota no es rechazada, una conexión L2CAP al *Servent* remoto es creada.

Ya que la conexión L2CAP se hizo con éxito, el *Servent servidor* envía el último número de mensaje insertado en su base de datos hacia el *Servent cliente*. Una vez que el cliente lo recibe, hace una resta del número de mensaje mas reciente menos el número de mensaje mas reciente del servidor. Con el resultado, que se envía al servidor, se toma una decisión. Si el resultado de la resta es igual a 0, termina la conexión L2CAP. Si es menor a 0, el *Servent servidor* actualiza al *Servent cliente* y termina la conexión L2CAP. Si es mayor a 0, el *Servent cliente* actualiza al *Servent servidor* y termina la conexión L2CAP.

Se eligió como protocolo de comunicación a L2CAP sobre RFCOMM debido a que con L2CAP la conexión se realiza de manera transparente al usuario, ya que trabaja en una capa por debajo de RFCOMM. El uso del protocolo L2CAP permite una mayor velocidad de conexión, evita la intervención del usuario, tanto como emparejamiento con otros dispositivos como para aceptar o denegar la conexión a estos. Es importante recordar que uno de los objetivos del modelo propuesto es que el usuario pueda llevar a cabo el intercambio de información con los demás dispositivos de manera automática, es decir, sin la intervención del usuario.

5 Implementación del Prototipo.

Se implementó un prototipo para comprobar la funcionalidad del modelo propuesto. La aplicación *Servidor BlueNews* se implementó en una computadora personal equipada con el sistema operativo Linux (Ubuntu 12.04). Se utilizó el lenguaje de programación C para el desarrollo de la aplicación y el API de BlueZ para la comunicación Bluetooth. Las aplicaciones *Cliente* y *Servent* se implementaron en dispositivos móviles con el sistema operativo Android 4.0. El lenguaje de programación utilizado fue Java y para el manejo de los aspectos de comunicación Bluetooth se utilizó el paquete *android.bluetooth*, así como la herramienta NDK para poder utilizar el protocolo L2CAP en Android. Para obtener un mejor desempeño, las aplicaciones *Servidor*, *Cliente* y *Servent* utilizan múltiples hilos.

Las aplicaciones *Cliente* y *Servent* tienen dos módulos principales, denominados *backend* y *frontend*. El módulo *backend* lleva a cabo las funciones relacionadas con

la recepción y envío de mensajes por medio de Bluetooth. El *frontend* es el encargado de mostrar los mensajes al usuario, configurar el funcionamiento de la aplicación, así como definir los perfiles de usuario, como puede observarse en la Fig. 6.

Las pruebas que se hicieron con el prototipo implementado mostraron que el modelo funciona correctamente en sus dos modos de operación: *Cliente/Servidor* y *Servent*. La Fig. 7 muestra un ejemplo del uso del prototipo.

Para el control y almacenamiento de los mensajes, la aplicación *Servent* utilizó una base de datos SQLite. Para que los usuarios autorizados puedan generar nuevos mensajes, además de la aplicación Android, se desarrolló una aplicación Web utilizando el lenguaje de programación PHP.



Figura 6. Módulo de configuración de la aplicación Cliente.



Figura 7. Ejemplo del módulo *frontend* de la aplicación Cliente.

6 Conclusiones y Trabajo Futuro.

En este artículo se presentó BlueNews, un modelo basado en la tecnología Bluetooth que permite la propagación de información utilizando el paradigma de cómputo ubicuo. El modelo propuesto tiene dos modos de operación: Cliente/Servidor y Servent, que permiten a los usuarios recibir información importante de manera automática y en diversos escenarios.

Para validar el modelo propuesto se implementó un prototipo utilizando herramientas de código abierto, entre las que destacan el API de BlueZ y el sistema operativo Android. La evaluación del prototipo mostró que el modelo propuesto funciona correctamente y que opera de manera invisible a los usuarios. En el artículo se presentan algunos aspectos importantes de la implementación del prototipo.

Como trabajo futuro se pretende implementar BlueNews en diferentes plataformas para dispositivos móviles.

Referencias

1. Android NDK official web site:<http://developer.android.com/tools/sdk/ndk/index.html>.
2. Bluetooth official web site. <http://www.bluetooth.com>.
3. Bluez official web site. <http://www.bluez.org>.
4. The IEEE 802.15.1 Standard. , June 2002..
5. Bluetooth SIG. Specification of the Bluetooth System. , 2001.
6. Woosik Kim Ho-Hyun Park. A Gnutella-based P2P System Using Cross-Layer Design for MANET. , 2007.
7. Chlamtac, M. Conti, J. Liu. Mobile ad hoc networking: imperatives and challenges. *Ad Hoc Networks*, 1(1):13 — 64, 2003.
8. Jem E. Berkes. Decentralized Peer-to-Peer Network Architecture: Gnutella and Freenet. 2003.
9. Pravin Bhagwat. Bluetooth: Technology for Short-Range Wireless Apps. *IEEE Inter*, 5(3):96—103, 2001.
10. M. Ripeanu. Peer-to-peer architecture case study: Gnutella network. *Proceedings of the 1st IEEE International Conference on Peer to Peer Computing*:99—100, 2001.
11. Stephanos Androultsellis-Theotokis, Diomidis Spinellis. A Survey of Peer-to-Peer Content Distribution Technologies. *ACM Computing Surveys*, 36(4):335—371, 2002.
12. Sukey Nakasima, Francisco Reyna, Arnoldo Diaz-Ramirez, Carlos T. Calafate. First Experiences With BlueZ. *Research in Computer Science*, 39:97—114, June 2008.
13. Mark Weiser. The computer for the 21st century. *Scientific American*, 265(3):94—104, 1991.