

Sincronización de Datos entre Aplicaciones sobre Redes Móviles

Borda Luciani, Fabricio Damián; Gagliardi, Edilma Olinda

Departamento de Informática
Facultad de Ciencias Físico Matemáticas y Naturales - Universidad Nacional de San Luis
Argentina
e-mail: {fborda,oli}@unsl.edu.ar
Fax: 54-2652-430224

Y

Hernández Peñalver, Gregorio

Departamento de Matemática Aplicada
Facultad de Informática - Universidad Politécnica de Madrid
España
e-mail: gregorio@fi.upm.es
Fax: 34-91-3367426

RESUMEN

Las redes móviles están en pleno auge desde no hace más de una década. Las redes móviles están compuestas por dispositivos que no siempre poseen conexión con la red. Sobre los dispositivos pueden correr aplicaciones que, en caso de trabajar con otras aplicaciones de la red, tienen la dificultad de mantener sus datos actualizados y operar la aplicación cuando el dispositivo pierda la conexión. Para facilitar estos inconvenientes se desarrolló una herramienta que simplifique las tareas de la sincronización de los datos entre las aplicaciones y resolver inconvenientes que puedan surgir en la sincronización. Esta herramienta está dividida en dos componentes específicas: una que reside en cada dispositivo móvil que tiene la aplicación y la otra en un dispositivo de la red que centraliza los datos de las aplicaciones.

Presentamos la descripción de la herramienta que facilita las tareas de sincronización entre las aplicaciones sobre los dispositivos de la red móvil y cómo trata los problemas que estas tareas implican.

Palabras Claves: Sincronización de datos, Red móvil, Red MANET, Computadoras Ocasionalmente Conectadas.

1. INTRODUCCIÓN

En la década de los '70 con el surgimiento de los sistemas distribuidos y las redes, comenzó a aparecer una necesidad: la sincronización de datos. A medida que nos acercamos a los tiempos actuales, este concepto fue adquiriendo cada vez más importancia y, a su vez, complejidad. Un ejemplo claro son las bases de datos distribuidas, donde los sistemas acceden y operan en la base de datos simultáneamente.

Con la llegada de los dispositivos móviles, como teléfonos celulares o computadoras portátiles, nace la necesidad de integrar estos dispositivos a las redes tradicionales. Como consecuencia de esto, surgen las redes móviles o redes MANET¹.

En las redes MANETs, los usuarios pueden estar temporalmente fuera del alcance de la

¹ Del inglés Mobile Ad-hoc NETWORK, son redes computadoras auto-configurables de dispositivos móviles con conexión sin cables

red y luego volver a conectarse. Esta desconexión de la red puede darse por varios factores. En este trabajo nos dedicamos al problema que surge cuando hay falta de conexión y se debe a que movimientos geográficos de los dispositivos escapan del alcance de la red.

Cuando un dispositivo cae fuera del alcance de la red, tendría que poder seguir operando su aplicación hasta que retome su conexión. En ese lapso de pérdida de conexión, el usuario del dispositivo móvil y el resto de los usuarios de la red, realizan modificaciones en sus respectivas bases de datos. En el momento que el usuario del dispositivo móvil retoma la conexión, el sistema debe actualizar tanto los datos del cliente móvil hacia el resto de los usuarios, como los datos de los usuarios de la red hacia el cliente móvil. Esta operación recibe el nombre de sincronización de datos.

Formalmente, el término “sincronizar” denota que un evento ocurre al mismo tiempo que otro. En ambientes de Computación, la sincronización de datos representa el proceso de hacer que dos conjuntos de datos sean aparentemente idénticos.

Según el sistema, existen distintos alcances para la sincronización: los más utilizados son a nivel de archivo o a nivel base de datos. A nivel de archivo se logra comparándolo contra otro del sistema de archivos, comparando fechas y tamaños. CVS (Concurrent Version System) es un ejemplo de esto ya que sincroniza las líneas de texto dentro de los archivos. Dentro de un sistema, se puede pensar en sincronizar base de datos en lugar de archivos. Por ejemplo, el sistema consiste en llevar la libreta de contactos sobre un dispositivo móvil. La libreta de contactos puede constar de campos como nombre, compañía, dirección, teléfono, e-mail, etc. Todos estos campos forman una entrada en su libreta de contactos y pueden ser considerados como un registro en la base de datos. Después de la ejecución de la sincronización, las libretas de contactos sincronizadas deben quedar idénticas.

Dado que estos dispositivos pueden estar distribuidos geográficamente, puede darse el caso de una eventual pérdida de la conexión

con la red y al momento de retomar dicha conexión, es posible que no se encuentre en la misma ubicación [6]. Estos dispositivos reciben el nombre de *computadoras ocasionalmente conectadas* [5].

Esto acarrea varios problemas: uno de ellos y principal enfoque en este trabajo es cómo permitir que las aplicaciones que requieren conexión en este dispositivo puedan seguir operando sus sistemas sin conexión con la red y al momento de retomar la conexión se actualicen los cambios en los datos de las aplicaciones.

Sobre el análisis y estudio realizado, surgió el diseño de una herramienta que facilite la sincronización de datos entre aplicaciones sobre redes móviles, de manera tal que cuando las aplicaciones sufran desconexiones, puedan reinsertarse a la red adecuadamente.

2. TAREAS EN PROGRESO

Las líneas de investigación que se están desarrollando están enfocadas a resolver los problemas antes mencionados.

Cuando un dispositivo de la red pierde la conexión al resto de la red por algún motivo, la aplicación pierde la posibilidad de seguir trabajando con las demás aplicaciones [6]. Es decir, no podrá recibir los cambios realizados en las demás aplicaciones y las restantes aplicaciones no tendrán a su vez sus cambios. Además, cuando el dispositivo retome la conectividad a la red, deberá actualizar sus datos con los cambios surgidos en las demás aplicaciones e informar sus cambios al resto de las aplicaciones de la red.

Ya que este es un problema inevitable debido al problema físico de pérdida de conectividad, nos propusimos implementar una herramienta que simplifique la sincronización de datos entre aplicaciones sobre dispositivos móviles. Esta herramienta se diseñó para poseer, entre otras características, operaciones para agregar, eliminar, confirmar, actualizar o sincronizar datos surgidos en las aplicaciones de los dispositivos cuando estos no poseen conexión con la red [2].

Herramienta para la sincronización

La herramienta que administra la sincronización entre las aplicaciones está dividida en dos componentes o módulos principales: uno que centraliza todos los datos de las aplicaciones sobre los dispositivos móviles denominado *Motor de Sincronización Servidor*; y el otro que reside en el dispositivo móvil, encargado de resolver los problemas de actualización de los datos de las aplicaciones llamado *Motor de Sincronización Cliente*.

Las características del *Motor de Sincronización Servidor* son [3]:

- Almacenar los últimos datos modificados de las aplicaciones.
- Identificar globalmente a los datos de las aplicaciones
- Recibir y actualizar los datos modificados en las aplicaciones de la red

El componente *Motor de Sincronización Cliente* que reside en cada dispositivo móvil tiene las siguientes características [1] [3]:

- Manejar la conectividad con el servidor de la aplicación
- Confirmar los cambios de la aplicación local al resto de las aplicaciones
- Actualizar los cambios realizados en otras aplicaciones de la red a la aplicación local.
- Manejar y resolver conflictos surgidos

Identificación de los datos

Unos de los problemas a resolver cuando las aplicaciones están dispersas por la red, es como identificar los datos surgidos en cada aplicación y que se puedan diferenciar con los datos de otras aplicaciones.

Debido a que los datos pueden surgir de la aplicación de cualquier dispositivo móvil de la red, puede darse que los identificadores que se le asignan estén duplicados con los de otros.

Para resolver este problema, se utilizará en la herramienta un identificador único local (LUID) para identificar a los datos de la aplicación de cada dispositivo. Es decir,

cada una de las aplicaciones administrará los identificadores de los datos locales. A su vez, la aplicación central asignará a cada dato que recolecte de las aplicaciones locales un identificador único global (GUID) para identificarlo únicamente.

Con esta disposición, un mismo dato en dos aplicaciones locales pueden no tener el mismo LUID pero obligatoriamente van a poseer el mismo GUID. Pero, para que la aplicación local pueda compartir sus datos con las aplicaciones, llevará un mapeo de identificadores GUID – LUID, por el cual la aplicación local va a identificar y operar los datos en su aplicación con el identificador LUID, pero cuando la comunicación sea con el sistema central se hará a través del identificador GUID.

Conflicto de datos

El gran desafío de los sistemas móviles es como asegurar que los cambios en los datos hechos en las aplicaciones locales sobre el mismo conjunto de datos sean propagados al resto de las aplicaciones.

Por mas precauciones que se tomen en el manejo de los identificadores de los datos y la sincronización de los mismos, hay casos donde las colisiones de datos son inevitables y los problemas surgidos son llamados Conflicto de Datos: una situación que debe ser resuelta durante la sincronización para evitar que los datos estén corruptos.

En [3] se nombran los distintos tipos de conflicto de datos que pueden darse en aplicaciones sobre dispositivos móviles:

Conflictos de Inserción

Supongamos que se dan de alta al mismo recurso en sus respectivas aplicaciones locales. La primera aplicación en confirmar su alta lo realizará sin inconvenientes. El problema lo posee la segunda aplicación al intentar sincronizar sus datos, ya que el registro ya fue dado de alta por la otra aplicación y genera problemas para determinar que registro va a prevalecer.

Para resolver este tipo de conflicto hay 2 opciones: falla la inserción del dato de los 2 usuarios o se aplica lógica del negocio que resuelve el conflicto. Esta última opción, puede hacer que la segunda inserción se

transforme de una inserción en una actualización o rechazar la inserción mas reciente.

Conflictos de Actualización

Es similar al conflicto de inserción. Dos sistemas modifican el mismo dato en sus aplicaciones locales y al intentar sincronizar sus datos con la aplicación central se producen los conflictos.

Para resolver este tipo de conflicto, también contamos con 2 opciones: falla la actualización de las dos aplicaciones o se aplica lógica del negocio que resuelve el conflicto. Esta última puede inspeccionar qué parte del dato cambió cada aplicación y realizar una mezcla con ambos datos modificados de forma de obtener un dato final híbrido.

Conflictos de Actualización vs. Eliminación

Un problema frecuente en aplicaciones sobre dispositivos móviles es resolver las modificaciones de un dato que ha sido eliminada por una aplicación pero modificada por otro. Para resolver estos problemas se puede optar o por hacer prevalecer el cambio mas reciente sobre el restante, o estableciendo una regla “modificación antes que eliminación” o “eliminación antes que modificación”. Modificación antes que eliminación, indica que ante un conflicto de este tipo, primero se ejecuta la modificación del mismo y luego la eliminación del dato. Eliminación antes que modificación, indica lo inverso: se elimina el dato y la modificación carece de sentido.

3. TRABAJOS FUTUROS Y CONSIDERACIONES FINALES

A partir de la investigación realizada en distintos medios, se detectó que no existe aún una herramienta de libre acceso que facilite la sincronización de datos entre aplicaciones de dispositivos móviles. También, debido al gran auge que está teniendo el desarrollo de aplicaciones sobre la web, varias compañías ofrecen sistemas bajo algún propósito específico que manejan determinadas aplicaciones, como por ejemplo libretas de contactos en teléfonos

celulares, correos electrónicos accesibles a través de la web, entre otros. Por ejemplo, el estándar de SyncML que es avalado por grandes empresas de telefonía celular, ofrece un conjunto de operaciones uniformes para tratar el tema de sincronización de datos, aunque en la actualidad sólo se utiliza para sincronizar libreta de contactos y calendarios web entre un dispositivo móvil y una computadora. Por todo ello, la contribución de este trabajo consiste en proveer una herramienta que permite facilitar la sincronización de datos entre aplicaciones sobre redes móviles.

Actualmente, la herramienta de sincronización se encuentra en una etapa de verificación donde se evalúan experimentalmente los distintos casos. Los casos de prueba están enfocados a los distintos estados que pueden tener las aplicaciones y a cómo se desempeña la sincronización de datos ante la eventual pérdida de conexión. En este sentido, la herramienta está llegando a su término, dado que con la evaluación de todos los casos de prueba planteados, se espera un resultado exitoso.

Estos trabajos están enmarcados dentro de la Línea de Investigación Geometría Computacional y Bases de Datos Espacio-Temporales, perteneciente al Proyecto Tecnologías Avanzadas de Bases de Datos 22/F314, Departamento de Informática, Universidad Nacional de San Luis; en el Proyecto AL06_PF_013 Geometría Computacional, subvencionado por a Universidad Politécnica de Madrid; y en el marco de la Red Iberoamericana de Tecnologías del Software (RITOS2), subvencionado por CYTED.

4. REFERENCIAS

- [1] Beron Mario, Gagliardi Edilma Olinda, Hernández Peñalver Gregorio - “Un evaluador de algoritmos de ruteo” – Tesis de Maestría en Ingeniería del Software, Universidad Nacional de San Luis – 2006.
- [2] David Buchmann – “SyncML (Synchronization Markup Language)” - Diploma Thesis in Informatics, University of Fribourg, Switzerland.

[3] Chris Kleisath, Ojas Rege. “The Developer’s Roadmap to Mobile Database Management and Synchronization” - <http://www.mobilizedsoftware.com/webbooks/>

[4] Chris Kleisath, Ojas Rege. “The Business Value Roadmap to Mobilized Software Solutions” - <http://www.mobilizedsoftware.com/webbooks/>

[5] Alan Zeichick, “Mobile Applications Developer FAQ” -

<http://www.intel.com/cd/ids/developer/asmo-na/eng/20387.htm>

[6] Fabian Kuhn, Roger Wattenhofer, Aaron Zollinger – “Ad-Hoc Networks Beyond Unit Disk Graphs” – 2003.

[7] Beron Mario, Gagliardi Edilma Olinda, Hernández Peñalver Gregorio - “Estrategias de ruteo alternativas para redes móviles” – XI Congreso Argentino de Ciencias de la Computación, Concordia, Entre Ríos – Argentina – 2005.