

Modelo para un Servicio de Mensajería Instantánea sobre Redes Oportunistas con Dispositivos Móviles

Francisco E. Páez^{1,3}, Gabriela Olguín¹, José M. Urriza¹, Javier D. Orozco^{2,3} y Rodrigo Santos^{2,3}

¹Universidad Nacional de la Patagonia San Juan Bosco, Puerto Madryn Argentina
²Universidad Nacional del Sur, Bahía Blanca, Argentina, ³CONICET

Resumen. Este trabajo presenta un modelo para una aplicación de mensajería instantánea sobre redes oportunistas formada por dispositivos móviles. El servicio admite el envío y recepción de mensajes, en ausencia de una infraestructura de red, o cuando la conexión es intermitente o limitada. Para ello, se emplean algoritmos de ruteo para redes tolerantes a demoras, algoritmos de descubrimiento de nodos vecinos, y se estudian las interfaces de comunicación de dispositivos móviles actuales. Se identifican problemas y dificultades para una implementación de los métodos y diseños propuestos en la literatura.

Palabras clave: Redes oportunistas, comunicación, mensajería instantánea.

1 Introducción

En la actualidad, la gran difusión de dispositivos móviles con amplia capacidad de cómputo y conectividad por *Bluetooth*, *Wi-Fi* o *Wi-Fi Direct*, abre nuevas posibilidades de comunicación. En particular, es posible establecer espontáneamente redes oportunistas (*RO*), conformadas por dichos dispositivos, complementado las redes de comunicación con infraestructura. Ante la saturación o ausencia de estas últimas, las *RO* ofrece canales de comunicación alternativos. El presente trabajo propone un modelo de mensajería instantánea (*MI*), sobre *RO* compuestas por dispositivos móviles, denominado *MIRO*. Se describe la arquitectura de la aplicación, la técnica de ruteo a emplear, y se exploran cuestiones de implementación. Actualmente varias aplicaciones de *MI* como *Windows Live Messenger*, *Yahoo! Messenger*, *Skype*, *Messenger* de Facebook, *Hangouts* de Google, etc., que emplean el modelo *cliente-servidor*. Estas no se benefician de la presencia una posible *RO*. Ante condiciones de conectividad intermitente, limitada o nula, su funcionalidad se ve limitada. Aplicaciones para dispositivos móviles, como *WhatsApp*, *Telegram* o *Line*, operan también con el modelo *cliente-servidor*, y presentan limitaciones similares.

El trabajo se organiza de la siguiente manera. En la sección 2 se hace una introducción a las *RO*. En la sección 3 se presentan trabajos previos. Luego, en la sección 4 se presenta la arquitectura del modelo propuesto. En la sección 5 se

comentan cuestiones de implementación sobre dispositivos móviles. Finalmente en la sección 6 se plantean trabajos futuros y se presentan las conclusiones.

2 Introducción a las Redes Oportunistas

En una *RO* el número y distribución de los nodos participantes es variable ([1, 2]). Debido a su movilidad, estos pueden unirse o abandonar la red arbitrariamente, sin ningún protocolo de unión ni restricciones temporales. Una *RO* no presenta una topología fija, y puede no existir una ruta entre el emisor y el destinatario de un mensaje. Los mensajes que no pueden ser entregados directamente al destino, se propagan mediante su copia a nodos intermedios. Esto requiere de mecanismos de ruteo tolerantes a demoras y ausencias de conexión, con almacenamiento y reenvío (*store and forward*) posterior de los mensajes. En una *RO* la infraestructura es provista por los mismos dispositivos participantes, aprovechando sus recursos ociosos. Las *RO* se encuadran dentro de las *DTNs* (*Delay Tolerant Networks*), cuyo protocolo estándar es el *Bundle Protocol* (*BP*). Una implementación de *BP* para dispositivos móviles y de propósito dedicado es *IBR-DTN* [3].

El ruteo de mensajes en este tipo de redes es tratado en varios trabajos ([4-9]). Los métodos epidémicos ([6, 10]) permiten una mayor cobertura y probabilidad de entrega de los mensajes. Sin embargo, emplean mayor cantidad de recursos de los nodos y la tasa de entrega, respecto de los recursos utilizados, es por lo general baja. El método *spray and wait* ([4]) propone una distribución limitada de los mensajes para mejorar esa relación. Los enfoques de ruteo pueden clasificarse como independientes del contexto (algoritmo epidémico y sus variantes), basados en la movilidad (predictivos) o como sensibles al contexto social. Cuando la interacción entre los usuarios influye en la conformación y configuración de la *RO*, esta se denomina *red oportunista social* (*Opportunistic Social Network*) ([11]). En este tipo de redes, se utilizan algoritmos de ruteo que consideran la popularidad de un nodo, contactos comunes entre nodos vecinos, perfiles de intereses, patrones de movimiento y densidad de nodos, etc., propuestos en [12-17], entre otros. Aspectos como escalabilidad, seguridad, incentivo para participación de usuarios, minimización del costo energético y de almacenamiento, entre otros, aún no cuentan con soluciones satisfactorias ([18]).

3 Trabajos Previos

Varios trabajos presentan aplicaciones de *MI* y distribución de contenidos en *RO*. En [19], se presenta la aplicación *OneChat*, que permite salas privadas, grupales y transferencia de archivos. Está limitado a *RO* donde los nodos puedan comunicarse directamente (*single-hop*). En [20], se describe el sistema *PassItOn*, que basa el ruteo en información social y espacial. La aplicación *whisper* para *Android* ofrece un servicio de mensajería, empleando la implementación *IBR-DTN* del protocolo *BP*. Otra aplicación para comunicación oportunista, para la plataforma *Android*, se describe en [21]. En [22], se presenta el *framework MobiClique* para establecer redes sociales en base a interacciones entre usuarios de dispositivos móviles. En [23], se

describe *TinyBox*, un *framework* para compartir datos multimedia, entre usuarios o por grupos. Emplea políticas de intereses para seleccionar qué datos intercambiar y técnicas *p2p* para sincronizar las copias. En [24], se describe el *framework CoCam*, para generación y difusión de contenido multimedia en un mismo espacio físico, por medio de teléfonos móviles, con perfiles de usuarios basados en datos de redes sociales. Estos trabajos plantean escenarios de uso acotados, donde los nodos tienen una alta probabilidad de encontrarse.

4 Modelo *MIRO*

Se propone el modelo *MIRO* de mensajería instantánea, para *RO* constituidas por dispositivos móviles. El modelo permite el envío y recepción de mensajes de texto a uno o más destinatarios.

Un usuario cuenta con un identificador único (*ID*), un nombre o pseudónimo, un *perfil de interés*, una lista de contactos y una lista de grupos a los que pertenece.

El *ID* de usuario se genera mediante una función *hash* (de 6 bytes), aplicada a un identificador único global, como el correo electrónico o el número de teléfono del usuario. El *perfil de interés* es una lista predefinida de tópicos, con un valor binario que indica si el usuario tiene interés o no en el mismo. La lista de contactos es un conjunto de *IDs*, nombres y *perfiles de interés* de otros usuarios. El usuario puede agregar un nuevo contacto manualmente mediante su *ID* o seleccionándolo de entre los nodos cercanos. Los grupos facilitan el envío de mensajes a múltiples destinatarios. Un grupo tiene un *ID*, un nombre, un *perfil de interés* y la lista de usuarios que lo componen. El usuario puede crear un grupo a partir de un conjunto de sus contactos, y enviar mensajes a uno o más destinatarios y/o grupos.

Además, existe un *perfil de interés de la vecindad*, que es la suma de los *perfiles de interés* de los nodos vecinos. Éste ayuda en el proceso de ruteo de mensajes.

4.1 Escenarios de Uso

Un servicio de *MI* debe entregar los mensajes al destino en un lapso adecuado a las expectativas del usuario. Sin embargo, las *RO* operan bajo una estrategia de mejor esfuerzo. Suponiendo que el entorno donde opera la red (sea un espacio abierto o cerrado) tiene dimensiones estables y límites conocidos, se puede alcanzar un cierto nivel de *calidad de servicio* en el envío y recepción de los mensajes, si los nodos se encuentran dentro del mismo. Se supone que estos nodos son principalmente dispositivos móviles (*smartphones*, *tablets*, etc.), aunque también pueden ser *notebooks* o PCs. Un ejemplo es un congreso especializado, donde la organización puede emplear la aplicación para coordinar sus actividades. Para los asistentes sería un medio práctico para recibir novedades e intercambiar opiniones o ideas. Una característica de este tipo de congresos es que las actividades generalmente se realizan en espacios delimitados, y la mayoría de los participantes cuenta con dispositivos móviles con varias opciones de conectividad. Debido a la cercanía entre los nodos, la comunicación mediante una *RO* puede tener una tasa de éxito aceptable.

4.2 Arquitectura General de MIRO

El modelo cuenta con cinco módulos, presentados en la Fig. 1. El *módulo de gestión de mensajes* prepara los mensajes a enviar, y administra los mensajes recibidos para el usuario. Procesa y almacena las copias de mensajes en curso para otros destinatarios. El *módulo de gestión de perfiles* administra la funcionalidad de *perfiles de interés*. El *módulo de ruteo* selecciona, de entre los nodos vecinos, los destinatarios siguientes para el reenvío de mensajes. El *módulo de gestión de vecinos* regula el proceso de descubrimiento y registra los nodos vecinos. El *módulo de gestión de conexiones* realiza el envío y recepción de los mensajes y el descubrimiento de vecinos, empleando la *API (Application Programming Interface)* que provee el Sistema Operativo (*SO*) para alternar entre las interfaces de comunicación disponibles. Este módulo abstrae la interfaz de comunicación empleada, y el protocolo subyacente utilizado por la misma para la comunicación oportunista.

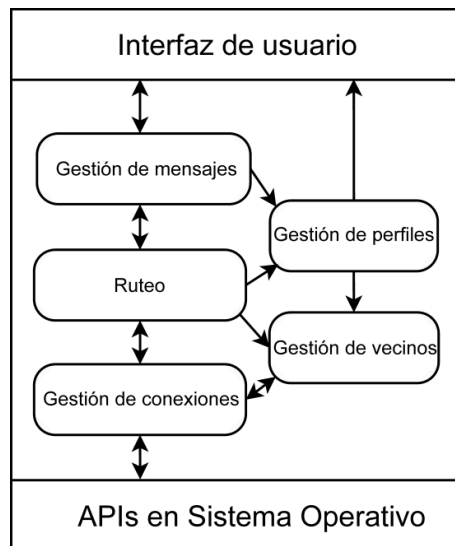


Fig. 1. Arquitectura general y módulos de MIRO.

4.3 Tipos y Estructura de los Mensajes

Se definen cinco tipos de mensajes: de usuario, aviso de mensaje recibido (*ACK*), de invitación a un grupo, de *ACK* de invitación al grupo y de anuncio de presencia. Un campo de tipo permite diferenciarlos. Los cuatro primeros constan de una cabecera común y una sección de datos. La cabecera indica la versión del protocolo, el tipo de mensaje, *ID* del mensaje, *ID* del emisor intermedio, *ID* del emisor original, *ID* de destinatario, el *perfil de interés* del emisor, *perfil de interés* del destinatario,

timestamp con tiempo de envío original del mensaje, máximo número de copias a reenviar y máximo número de saltos antes de descartar el mensaje (*HOP*).

La sección de datos del mensaje de usuario contiene el texto del mensaje, codificado mediante *ASCII* (*American Standard Code for Information Interchange*) extendido. La sección de datos del mensaje de invitación a un grupo contiene el *ID* de grupo, el nombre del mismo, el *perfil de interés* del grupo y el nombre del usuario emisor. La sección de datos del mensaje de *ACK* de invitación a un grupo contiene el nombre del usuario que responde, y el *perfil de interés* de su vecindad, obtenido mediante la suma de los *perfiles de interés* de sus vecinos. Si un usuario no acepta la invitación al grupo se envía el *ACK* al usuario emisor original, en lugar de al grupo.

Los mensajes de *ACK* de recepción de mensajes permiten confirmar la recepción del mensaje por parte del destinatario. Para la creación de grupos permiten conocer cuales usuarios han confirmado su unión al mismo. La recepción del *ACK* tiene asociado un *timeout* configurable, pasado el cual la aplicación informará al usuario de la ausencia de confirmación.

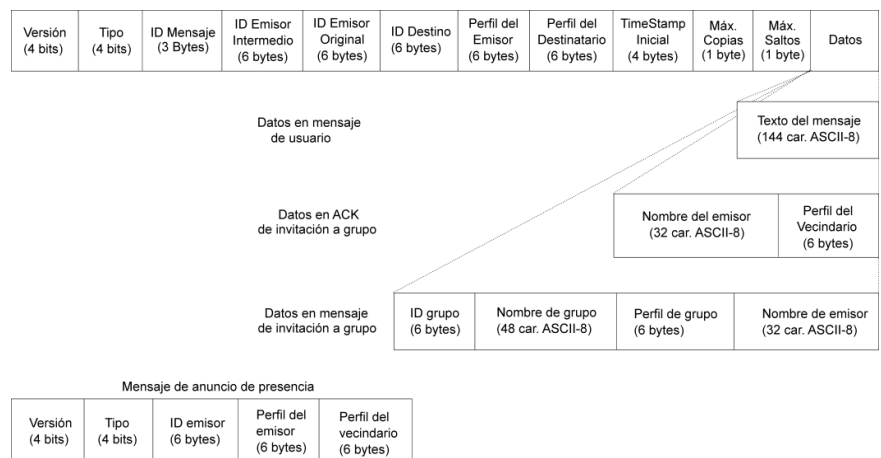


Fig. 2. Formato de mensajes.

El mensaje de anuncio de presencia contiene la versión del protocolo, el tipo de mensaje, *ID* del usuario emisor, su *perfil de interés* y el *perfil de interés* de su vecindad. Se emplea para el intercambio de *perfiles de interés* entre nodos vecinos. Como la estructura del *perfil de interés* es fija, este se codifica como un vector de 48 bits (6 bytes). Esto simplifica la operación de comparación de perfiles en el ruteo, y el armado de la cabecera del mensaje.

4.4 Algoritmo de Ruteo

El módulo de ruteo selecciona los nodos intermedios a los cuales reenviar un mensaje, realizando una comparación de *perfiles de interés*, seleccionando aquellos nodos que

compartan el mayor número de intereses en común. En primer lugar genera un vector con los intereses comunes entre el *perfil de interés* del emisor original y del destinatario, mediante un *AND*. Luego inspecciona la lista de nodos vecinos, comparando los *perfiles de interés* de estos con este vector combinado, también mediante una operación *AND*. Por cada nodo vecino se registra el número de tópicos en común (el número de *bits* en 1). Aquellos nodos vecinos con un mayor número de tópicos en común serán seleccionados como nodos intermedios. Si no se puede cumplir con el número de copias a enviar, se vuelve a realizar el proceso de comparación, primero contra el *perfil de interés* de la vecindad de cada nodo vecino, y luego empleando el *perfil de interés* del nodo destino, y el *perfil de interés* del nodo emisor original. Si aún la cantidad de nodos seleccionados así no fuera suficiente, el mensaje se almacena para reenviarlo posteriormente, evitando repetir el envío de un mensaje a un nodo intermedio ya empleado. El proceso de ruteo evita enviar nuevamente el mensaje al nodo emisor original, o al nodo intermedio que reenvió el mensaje, evitando así posibles ciclos en la ruta del mismo. Este proceso de ruteo está basado en el método *SANE (Social Aware Networking, [13])*.

4.5 Envío de Mensajes

A continuación se describe el proceso de envío de un mensaje (Fig. 3).

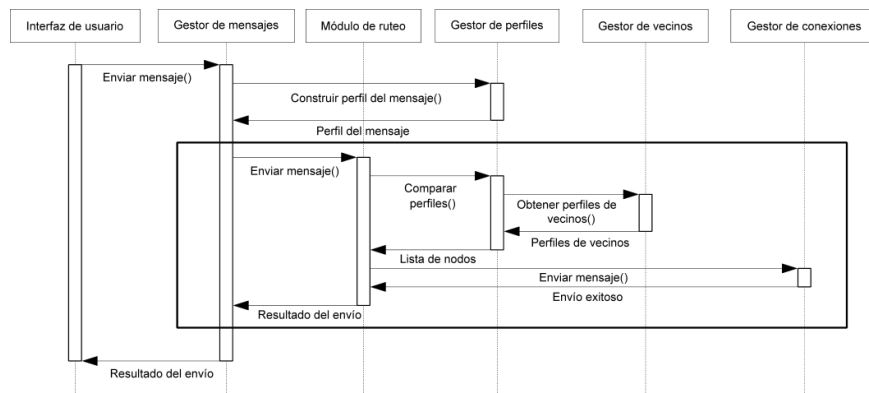


Fig. 3. Envío de mensaje a un usuario o grupo.

El usuario selecciona uno o más usuarios o grupos, e ingresa el texto del mensaje a enviar. Estos datos son pasados al *módulo de gestión de mensajes*, que construye el mensaje de usuario (ver sección 4.3) para cada destinatario. El *perfil de interés* del usuario y del destinatario se obtiene por medio del *gestor de perfiles*. En caso de no contar con el *perfil de interés* del destinatario, se emplea un *perfil de interés* donde todos los tópicos están activados (denominado *neutro*). Luego, el mensaje es pasado al *módulo de ruteo*. Este selecciona los nodos a los cuales reenviar copias del mensaje, haciendo uso del algoritmo de ruteo descrito en la sección 4.4. El *gestor de conexiones* finalmente envía una copia del mensaje a cada nodo vecino seleccionado.

Si al menos una copia del mensaje fue transmitida a un nodo vecino, se informa al usuario que el mensaje fue enviado. El *módulo de ruteo* verifica con el *gestor de vecinos* si el destinatario del mensaje se encuentra entre los nodos vecinos. De ser así, realiza la entrega directa con el *gestor de conexiones*, y no reenvía el mensaje a otros nodos. Luego elimina la copia del mensaje.

Al recibir un *ACK*, el *gestor de mensajes* informa al usuario que el mensaje fue entregado. Las copias del mensaje que tenga el nodo para su envío son eliminadas. Si expira el *timeout* asociado al *ACK*, la aplicación informa de la falta de confirmación en la entrega del mensaje. Un *ACK* puede recibirse luego de vencer el *timeout*, en dicho caso, se actualiza la confirmación de entrega del mensaje.

4.6 Recepción y Reenvío de Mensajes

A continuación se presenta el proceso de recepción y reenvío de mensajes (Fig. 4). El *módulo de ruteo* determina, en primer lugar, si es el nodo destinatario del mensaje. Esto se comprueba verificando el *ID* del destinatario con el *ID* del usuario y con los *IDs* de los grupos a los que el nodo pertenece. Si el mensaje no está destinado al usuario, ni fue recibido anteriormente, este es reenviado. En caso que el destinatario sea un grupo el mensaje debe reenviarse, para propagarlo al resto de los participantes.

La recepción de mensajes la realiza el *módulo de gestión de conexiones*. Este módulo realiza una escucha periódica de mensajes entrantes, ocultando al resto del modelo la interfaz de comunicación empleada. Los mensajes recibidos son entregados al *módulo de ruteo*. Este determina si el mensaje debe ser entregado al usuario o reenviado. En el primer caso el mensaje se entrega al *gestor de mensajes*, se alerta al usuario y se genera un *ACK* de recepción del mensaje. Si el mensaje debe ser reenviado, el *módulo de ruteo* escoge a que nodos reenviar copias. Emplea el mismo proceso de envío de mensajes descrito en la sección 4.5.

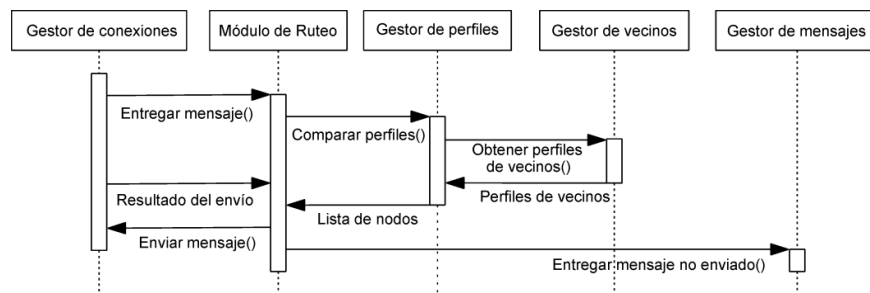


Fig. 4. Recepción y reenvío de mensajes.

Si el mensaje a reenviar utiliza un *perfil de interés neutro*, y el nodo intermedio cuenta con el *perfil de interés* del destinatario, lo reemplaza con éste. Los mensajes pendientes de reenvío duplican su periodo de espera por cada intento de envío no exitoso, hasta una cota máxima. El *gestor de mensajes* define un tiempo máximo de

reenvío, expirado el cual el mensaje es descartado. El módulo de ruteo decrementa el *HOP* con cada reenvío. Si éste llega a cero el mensaje es descartado.

En caso de tener que reenviar un *ACK*, el nodo intermedio verifica si tiene copias pendientes de envío del mensaje al que el *ACK* hace referencia. De ser así, elimina las copias almacenadas. De esta manera, se liberan recursos en la *RO* al descartar mensajes que ya han sido entregados al destinatario.

4.7 Descubrimiento de Vecinos

El *módulo de gestión de vecinos* mantiene una lista de nodos descubiertos durante el funcionamiento de la aplicación. Por cada nodo registra su *ID*, el *timestamp* de último contacto, si es un nodo vecino, su *perfil de interés* y el de su vecindario. Los nodos detectados más recientemente son considerados como vecinos. El *módulo gestor de vecinos* coordina la periodicidad del proceso de escucha y del proceso de envío del *mensaje de anuncio de presencia*. Los detalles específicos de cómo se realiza este proceso con las interfaces de comunicación disponibles, es responsabilidad del *gestor de conexiones*.

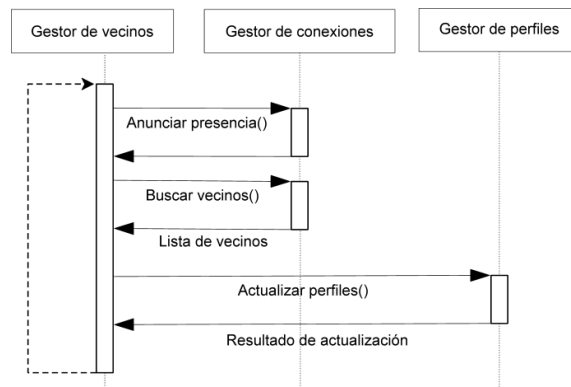


Fig. 5. Descubrimiento de nodos vecinos y anuncio de presencia.

El *gestor de conexiones*, realiza la escucha de los *anuncios de presencia* de nodos cercanos, y dependiendo de la interfaz de comunicación empleará el mecanismo más apto para tal fin (sección 5). Finalizada la escucha, el *gestor de vecinos* recibe una lista de nodos cercanos. Esta lista se compara con la lista de nodos conocidos. Por cada nodo ya registrado en esta última, se actualiza el *timestamp* de último contacto y se marca a dicho nodo como vecino. Si luego de verificar los nodos descubiertos, un nodo identificado previamente como vecino no estuviera entre estos, se lo desmarca como tal (se supone que el nodo se encuentra fuera del rango de alcance). Los nuevos nodos detectados se agregan, como nodos vecinos, a la lista de nodos conocidos. Los *perfiles de interés* recolectados son pasados al módulo de *gestión de perfiles* que actualiza con los nuevos datos los perfiles almacenados, y actualiza el *perfil de interés*

del vecindario propio. Un nodo es eliminado de la lista de vecinos si pasado un cierto número de búsquedas el mismo no aparece entre los nodos cercanos.

4.8 Creación y Gestión de Grupos

Los grupos facilitan al usuario el envío de mensajes a múltiples destinatarios. El usuario puede definir un nuevo grupo a partir de contactos en su lista. Para la creación de un grupo, un usuario selecciona los contactos que conformarán el mismo, además de él, un nombre para el grupo, y un *perfil de interés* para el mismo. A continuación, el *gestor de mensajes*, genera los *mensajes de invitación de unión al grupo*, destinados a los nodos que lo conformarán. Estos mensajes son enviados siguiendo el proceso descrito en la sección 4.5. Si un usuario acepta la invitación, agrega el *ID* del grupo y el *perfil de interés* de este a su lista de grupos. Los mensajes destinados a un grupo no requieren de un *ACK* de recepción.

La confirmación de unión a un grupo requiere del envío de un *ACK* al grupo. Contiene en su sección de datos el nombre del usuario que acepta la invitación y el *perfil de interés* de su vecindad. Los integrantes del grupo mantienen una lista de nombres de usuarios en el grupo mediante los *ACK* recibidos. Si un usuario no acepta la invitación a un grupo envía un *ACK* únicamente al nodo que envió la invitación. Si un usuario desea darse de baja de un grupo envía un mensaje de *ACK* al grupo con la cadena "REMOVE" en el campo de nombre de usuario, y el perfil de interés del vecindario en cero.

5 Gestión de Conexiones

Las interfaces de comunicación *Wi-Fi*, *Wi-Fi Direct* y *Bluetooth* no fueron diseñadas para establecer *RO*. Esto trae problemas y limitaciones. *Bluetooth* permite más flexibilidad para conexiones espontáneas con otros dispositivos y un menor consumo de batería, pero con un rango de alcance y tasa de transferencia limitados. *Wi-Fi* proporciona mayor ancho de banda y rango de alcance, con un mayor consumo de energía. Además, varias plataformas móviles (como *Android*) no permiten conexiones *ad hoc* sin modificaciones al sistema. *Wi-Fi Direct* permite la conexión de dispositivos sin requerir un punto de acceso (*Access Point (AP)*). Su soporte es relativamente reciente, mientras que *Wi-Fi 802.11* y *Bluetooth*, están disponibles en la mayoría de los *smartphones*.

En una *RO*, un nodo explora periódicamente el entorno para identificar nodos próximos. Un periodo menor mejora la oportunidad de encuentros oportunistas, incrementando el consumo de energía. La búsqueda de nodos vecinos con *Bluetooth* ha sido estudiada en [25-27]. El esquema de descubrimiento típico es asimétrico, donde un dispositivo toma el rol activo, enviando *balizas* periódicas, y el resto ocupa un rol pasivo de escucha. Para una *RO* los nodos deben alternar ambos roles, en intervalos de tiempo aleatorios. Esquemas adaptativos para cambiar entre mecanismos de descubrimiento se presentan en [25], a fin de mejorar el consumo de energía. Además, el *MTU (Maximum Transfer Unit, Máxima Unidad de Transferencia)* de

Bluetooth no es modificable en muchas plataformas.

Mediante *Wi-Fi* el dispositivo normalmente se conecta a una red por medio de un *AP*, que le asigna una dirección de red. Formará parte de la misma mientras se encuentre dentro del rango de alcance del mismo. En [28] se propone el protocolo *Wi-Fi Opp*, que aprovecha las capacidades *AP* de un dispositivo para establecer comunicaciones oportunistas. Para esto, un nodo toma dinámicamente el rol de *AP*, y los demás dispositivos actúan como clientes. El rol de *AP* rota entre los nodos, a fin de distribuir el costo de operación. Sin embargo, el nodo *AP* no puede conectarse a Internet por medio de *Wi-Fi*, lo que obliga en caso de requerirse conexión a cambiar entre un modo *oportunista* y otro de *infraestructura*. Además, el modo *AP* en los *smartphones* está diseñado para compartir acceso a la red de datos, lo que puede no ser deseado por el usuario en el caso de conexiones oportunistas.

Wi-Fi Direct es analizado en [29], y su uso en *RO* se discute en [30], donde se evalúa el costo temporal de la conformación de una red de dispositivos. Para grupos de seis nodos, en el peor de los casos, este proceso puede durar más de dos minutos. Sin embargo, en casos donde los dispositivos permanecen en contacto por periodos prolongados, estos tiempos pueden ser tolerados.

6 Trabajos Futuros y Conclusiones

En trabajos futuros se desarrollará una implementación del modelo, en principio mediante un simulador, para evaluar métricas de: rendimiento, consumo de energía, duración de los encuentros oportunistas, tasa de éxito de entrega de mensajes, tiempos promedio de entrega, número de saltos promedio para el envío de un mensaje, etc. Cuestiones como seguridad, privacidad y autenticación de los usuarios también se dejan para trabajos posteriores. La movilidad de los nodos podría permitir una eventual disponibilidad de acceso a una infraestructura de comunicación. La integración con ese tipo de redes permitiría un alcance global al servicio modelado.

En este trabajo, se ha presentado el diseño de un modelo de *MI* sobre redes oportunistas, *MIRO*. La propuesta define la arquitectura general del modelo, formato, envío y ruteo de mensajes, así como también descubrimiento y administración de nodos vecinos. Se plantearon ventajas y desventajas de las interfaces de comunicación disponibles en dispositivos móviles, para conformar una *RO*, en particular el modelo *MIRO*.

Referencias

- [1] L. Lilien, Z. H. Kamal, and A. Gupta, "Opportunistic Networks: Challenges in Specializing the P2P Paradigm," in *Database and Expert Systems Applications, 2006. DEXA '06. 17th International Workshop on*, 2006, pp. 722-726.
- [2] Huang Chung-Ming, Lan Kun-chan, and Tsai Chang-Zhou, "A Survey of Opportunistic Networks," in *Advanced Information Networking and Applications - Workshops, 2008. AINAW 2008. 22nd International Conference on*, 2008, pp. 1672-1677.

- [3] Michael Doering, Sven Lahde, Johannes Morgenroth, and Lars Wolf, "IBR-DTN: an efficient implementation for embedded systems," presented at the Proceedings of the third ACM workshop on Challenged networks, San Francisco, California, USA, 2008.
- [4] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," presented at the Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking, Philadelphia, Pennsylvania, USA, 2005.
- [5] Anders Lindgren, Avri Doria, Olov Schel, and #233, "Probabilistic routing in intermittently connected networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 7, pp. 19-20, 2003.
- [6] Amin Vahdat and David Becker, "Epidemic routing for partially connected ad hoc networks," Technical Report CS-200006, Duke University 2000.
- [7] Javier Orozco, Rodrigo Santos, Sergio F Ochoa, and Roc Meseguer, "Stochastic Performance Evaluation of Routing Strategies in Opportunistic Networks."
- [8] M. Musolesi and C. Mascolo, "CAR: Context-Aware Adaptive Routing for Delay-Tolerant Mobile Networks," *Mobile Computing, IEEE Transactions on*, vol. 8, pp. 246-260, 2009.
- [9] L. Pelusi, A. Passarella, and M. Conti, "Opportunistic networking: data forwarding in disconnected mobile ad hoc networks," *Communications Magazine, IEEE*, vol. 44, pp. 134-141, 2006.
- [10] Yoon Seung-Keun and Z. J. Haas, "Efficient Tradeoff of Restricted Epidemic Routing in Mobile Ad-Hoc Networks," in *Military Communications Conference, 2007. MILCOM 2007. IEEE*, 2007, pp. 1-7.
- [11] Sabrina Gaito, Elena Pagani, and Gian Paolo Rossi, "Strangers help friends to communicate in opportunistic networks," *Computer Networks*, vol. 55, pp. 374-385, 2011.
- [12] Chiara Boldrini, Marco Conti, and Andrea Passarella, "ContentPlace: social-aware data dissemination in opportunistic networks," presented at the Proceedings of the 11th international symposium on Modeling, analysis and simulation of wireless and mobile systems, Vancouver, British Columbia, Canada, 2008.
- [13] A. Mei, G. Morabito, P. Santi, and J. Stefa, "Social-aware stateless forwarding in pocket switched networks," in *INFOCOM, 2011 Proceedings IEEE*, 2011, pp. 251-255.
- [14] P. Costa, C. Mascolo, M. Musolesi, and G. P. Picco, "Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks," *Selected Areas in Communications, IEEE Journal on*, vol. 26, pp. 748-760, 2008.
- [15] A. Mtibaa, M. May, C. Diot, and M. Ammar, "PeopleRank: Social Opportunistic Forwarding," in *INFOCOM, 2010 Proceedings IEEE*, 2010, pp. 1-5.
- [16] Elizabeth M. Daly and Mads Haahr, "Social network analysis for routing in disconnected delay-tolerant MANETs," presented at the Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing, Montreal, Quebec, Canada, 2007.
- [17] Hui Pan, J. Crowcroft, and E. Yoneki, "BUBBLE Rap: Social-Based Forwarding in Delay-Tolerant Networks," *Mobile Computing, IEEE Transactions on*, vol. 10, pp. 1576-1589, 2011.
- [18] Wang Shengling, Liu Min, Cheng Xiuzhen, and Song Min, "Routing in pocket switched networks," *Wireless Communications, IEEE*, vol. 19, pp. 67-73, 2012.

- [19] Heming Cui, Suman Ramkumar Srinivasan, and Henning G. Schulzrinne, "ONEChat: Enabling Group Chat and Messaging in Opportunistic Networks," 2010.
- [20] Xing Bo, K. Seada, P. Boda, and N. Venkatasubramanian, "PassItOn: An Opportunistic Messaging Prototype on Mobile Devices," in *Consumer Communications and Networking Conference, 2009. CCNC 2009. 6th IEEE*, 2009, pp. 1-2.
- [21] Paolo Meroni, Elena Pagani, Gian Paolo Rossi, and Lorenzo Valerio, "An opportunistic platform for Android-based mobile devices," presented at the Proceedings of the Second International Workshop on Mobile Opportunistic Networking, Pisa, Italy, 2010.
- [22] Anna-Kaisa Pietiläinen, Earl Oliver, Jason LeBrun, George Varghese, and Christophe Diot, "MobiClique: middleware for mobile social networking," presented at the Proceedings of the 2nd ACM workshop on Online social networks, Barcelona, Spain, 2009.
- [23] C. Caprini, M. Zeni, N. Conci, and D. Miorandi, "TinyBox: Social, local, mobile content sharing," in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2013 IEEE International Conference on*, 2013, pp. 300-302.
- [24] E. Toledano, D. Sawada, A. Lippman, H. Holtzman, and F. Casalegno, "CoCam: A collaborative content sharing framework based on opportunistic P2P networking," in *Consumer Communications and Networking Conference (CCNC), 2013 IEEE*, 2013, pp. 158-163.
- [25] C. Drula, C. Amza, F. Rousseau, and A. Duda, "Adaptive energy conserving algorithms for neighbor discovery in opportunistic Bluetooth networks," *Selected Areas in Communications, IEEE Journal on*, vol. 25, pp. 96-107, 2007.
- [26] Diego Bohman, Matthias Frank, Peter Martini, and Christoph Scholz, "Performance of Symmetric Neighbor Discovery in Bluetooth Ad Hoc Networks," *GI Jahrestagung (I)*, vol. 4, pp. 138-142, 2004.
- [27] Theodoros Salonidis, Pravin Bhagwat, and Leandros Tassioulas, "Proximity awareness and fast connection establishment in Bluetooth," presented at the Proceedings of the 1st ACM international symposium on Mobile ad hoc networking & computing, Boston, Massachusetts, 2000.
- [28] Sacha Trifunovic, Bernhard Distl, Dominik Schatzmann, and Franck Legendre, "WiFi-Opp: ad-hoc-less opportunistic networking," presented at the Proceedings of the 6th ACM workshop on Challenged networks, Las Vegas, Nevada, USA, 2011.
- [29] D. Camps-Mur, A. Garcia-Saavedra, and P. Serrano, "Device-to-device communications with Wi-Fi Direct: overview and experimentation," *Wireless Communications, IEEE*, vol. 20, pp. 96-104, 2013.
- [30] M. Conti, F. Delmastro, G. Minutiello, and R. Paris, "Experimenting opportunistic networks with WiFi Direct," in *Wireless Days (WD), 2013 IFIP*, 2013, pp. 1-6.