Development of a cooperative application for sending SMS on WiFi mobile phones

Pedro López, Diego García, Sergio Almagro, Juan J. Alcaraz, Fernando Cerdan
Dpto. Tecnologías de la Información y Comunicaciones
Universidad Politecnica de Cartagena
Plaza del Hospital, 1, 30202 CARTAGENA
pedro.lopez@upct.es, diego.gsanchez@upct.es, sergio.almagro@upct.es, juan.alcaraz@upct.es, fernando.cerdan@upct.es

Abstract

Most of current user mobile devices incorporate a WiFi interface. Peer communications are based on Ad-hoc network configuration, but the main handicap is the radio coverage. The success in deploying services and applications based on this concept needs the development of relay protocols and applications that allows intermediate users the reception and forwarding of information to destination users. The user may take the maximum benefit from the technology acquired inside the mobile phone for instance, since it is possible extend the radio coverage by several hops, and enjoy similar mobile's operator services free of charge. In this work we present the development of a cooperative or relay protocol integrated on a Java application to send SMS among user mobile phones through its WIFI interface.

1. Introduction

Wireless networks have experienced a big growing in the last years. From typical Internet services to hot multimedia applications, all of them need a big amount of bandwidth. For this reason the IEEE has paid much attention according to the improvement of new versions of standard 802.11 [1].

WLAN networks allow a good framework to develop much type of services and applications like:

- Multimedia messages applications, to send and receive sounds and video.
- Multicast diffusion of multimedia contents to a group of users depending on their geographical location.
- Voice over IP and telephony IP.
- Internet access shared by a community of mobile and wireless devices.
- Military communications in places or actions where is not possible to built some type of infrastructure.
- Provide connectivity in places where there are obstacles that interrupt the visibility's line among network nodes.

This service variety is of great interest due to the amount of functionalities as well as they may skip the infrastructure provided by a mobile operator.

As regard of hardware cost, building a WLAN is rather cheap, although some mobile phones or PDA models with WIFI interface may be expensive for some users. So that, the use of applications like the one presented in this paper, may contribute to expand market and reduce prices.

In a MANET (Mobile Ad-hoc Network) [2], mobile devices usually get connected self organized without any type of infrastructure. MANET deployment is in general temporary where users have an arbitrary mobility causing continuous topology changes. Routing algorithms in these networks are classified in general in two types:

- ***Single-hop:*** Information is straight send from sources to destination terminals. That's usually works an Ad-hoc network.
- ***Multi-hop***: Device terminal in addition to straight delivery may act forwarding information, making possible a communication between two users without line of vision. In this case we extend the radio coverage, using the connectivity of

978-0-7695-3367-4/08 $25.00 © 2008 IEEE
DOI 10.1109/UBICOMM.2008.66

intermediate close terminals for receive and forwarding information to destination terminals or other intermediate ones.

Traditional routing algorithms have been developed assuming that there is routers providing the needed connectivity in the network. Nevertheless, these protocols are not suitable for wireless networks where nodes may move and packets may be long time delayed. An added difficult arise in a MANET since some nodes may appear or disappear at any time. So that, routing information is almost unknown.

The MANET IETF working group [2] has defined a set of properties that should fulfil protocols in these networks [3] and show a list of protocols that incorporate those recommendations.

In a MANET becomes necessary that routing algorithms be efficient and provide many valid routes to avoid waste of energy consumption in route discovering task due to blocking situations.

To obtain routes we may use proactive algorithms PMRP (Proactive MANET Routing Protocols) or reactive algorithms RMRP (Reactive MANET Routing Protocols) also called On Demand [4]. For each type of algorithm we have advantages and drawbacks.

A proactive algorithm provides a biggest probability to find a route, while a reactive one optimizes network performance and gets bigger energy efficiency, but the protocol behaviour is worst. So, seems clear that the best solution is a compromise between both methods.

Another important consideration is the type of terminal's network used, as regard of energy consumption, due to the different battery capacities. While the device is sending or receiving information it consumes energy, so is very important that terminals be able to save energy remaining in a minimal consumption state when possible. This desired state is *idle state* where there aren't transmissions or receptions, but the terminal is listening to the channel and ready for a message's transmission.

Mobility also influences the protocol behaviour and the energy's consumption. Relative mobility defines the mobility of a terminal respect to others and determines the network performance. So that, the implemented routing protocol must be adaptive for several reasons: The first is mobility that makes network topology dynamic.

The second important reason is the devices' battery dependence.

When batteries of some terminal get run out the algorithm must correct the routing information to optimize the lifetime of the network nodes. Finally the third reason takes into account the packet priority.

In summary, is not trivial to choose a routing algorithm from the literature, to deploy in a real working scenario. Each one has advantages and drawbacks and most of them have never been implemented in real user devices like PDAs or mobile phones. We only have theorical information, with the added difficult that layer *MAC* IEEE 802.11 has shown high degree of inefficiency in scenarios with mobility and multihop routing. [5].

In this work we face up the challenge to build an application of great interest, to send SMS between mobile commercial phones with the possibility of peer communication without direct line of vision. We specify and implement a communication protocol with multihop capacity that allows the deployment of WIFI ad-hoc and spontaneous networks, without theoretical coverage limits. The rest of the word is organized as follows: In section two we show to the readers which are the purposes of this project. In section three we enumerate the statements to develop both protocol and the application that make possible extend coverage thanks to relay or multihop technique. In section four we specify the developed multihop protocol. In section five we describe de user characteristics of the client software application terminal. Finally we conclude the paper in section six.

2. Application and protocol purposes

With this application we pretend to offer end users connected to a WLAN network a software solution within can be possible send and receive text messages using user's mobiles and their WiFi interfaces autonomously without the need of an operator network or infrastructure. Obviously these messages can be sent and received free of charge as often as desired.

To achieve the communication between the most separated sender and receiver mobiles in each transmission is necessary that the network devices can receive, store and forward datagrams to other devices acting as intermediate nodes recursively. With this technique all involved devices can define a route between sender and receiver dinamically.

For these reasons the application requires a communication's protocol that implements a multi-hop routing algorithm previously described using the desired conditions that a MANET protocol must have.

Analyzing mobile's operating systems, APIs available and the existing restrictions we may determine an optimal solution to reach the applications' purposes.

The application's environment is characterized by the existence of a users' group that want to use a simple way of communication based on text messages. They have mobiles with a WiFi interface that bring them the possibility of create a WLAN ad-hoc network to send and receive messages. It's necessary that these users assign previously a correct IP address to their devices' WiFi interfaces and therefore can create an ad-hoc network successfully. When this network is operative then they can execute the developed application.

One interesting idea in this environment is the fact that if the number of mobiles in a network grows then theorically the messages can be delivered to their destinations more safely.

3. Relay protocol and application development starting points.

As network technology we will use IEEE 802.11 in ad-hoc mode. The transfer rate can be different depending if used devices can support 802.11g (54 Mbps) or previous 802.11b (11 Mbps) standard.

Referring to the type of packages that the application must send, text messages, the conditions necessary are fulfilled to use a slight transport protocol like UDP and therefore datagrams. Later will be implemented a transmitted message's recognition's mechanism (ACKs) that lets to the sender to know if there were errors in sending operations.

The environment's development has been based on J2ME and CLDC/MIDP [7]. Due to the important communications' restrictions of this platform, the routing algorithm will be integrated as part of the application. So that, the tasks of routing, information's storing and forwarding will be done from applications' levels, based on the use of sockets and other MIDP API's mechanisms without being able to reach low levels stack's protocols implemented in the mobile's operating system.

The protocol and the application have been designed optimizing the energy's efficiency. For it, one of the premises to consider is the minimization of the transmissions' number and the size of the datagrams without losing effectiveness at the time of delivering datagrams from the source node to the destination one.

4. Relay protocol specification

The communication's protocol that we have implemented is based on a routing algorithm that uses techniques of selective flooding. The majority aspects are proactive although also incorporates certain reactive shades in this implementation.

In the implemented protocol we have defined the following states:

- ***Idle state:*** This is the state in which begins the protocol and in which normally remains waiting for an event (send or receive datagrams). When the protocol is in this state consumes the lowest possible energy. Sleep mode is not possible to define since the classes provided for the API of J2ME communications operate under an active state of execution.

  In consequence the application, as soon as possible, must return to this state to try to optimize the energy consumption.
- ***Send a datagram:*** Includes all tasks related to send a datagram.
- ***Receive a datagram:*** Includes all tasks related to receive/discard a datagram.
- ***Forward a datagram:*** This state is quite similar to the state "send a datagram". Differences are explained afterwards.

Figure 2 shows the complete protocol flowchart, including the above mentioned states, as well as the decision paths.

4.1 Messages identifiers list

Before explain the flowchart it's very important to remark and understand the meaning of the messages' identifiers list implemented in this application.

All messages have a field called message's identifier that contains a different and exclusive number for each message sent to the device's network. This field is generated when the user wants to send a message and it remains invariant in receiving and forwarding tasks. Moreover the identifier is introduced in the sender's identifiers list. With this information, when the device receives a message, the protocol's implementation can check if it has been received previously or not and process o discard the packet.

If all devices apply this technique then congestion is prevented and the traffic's growth is controlled because unnecessary datagrams aren't sent to the network.

4.2 Initial state

Our protocol starts at "*Idle*" state. It remains in this state until a datagram's sending or receiving event happens. The flowchart includes a three path chart to move from this initial state, in order to facilitate protocol's understanding, but really the protocol doesn't check continuously if an event has happened or not. Events are asynchronous in this protocol.

4.3 Datagram sending

When the developed application interacts with the protocol to send a SMS, it goes to the "*Send a datagram from application*" state. Here, first of all, the message's identifier is created and inserted into the sender's identifiers list mentioned previously.

Next, the datagram is sent to all network's devices reachable, by using broadcast IP addressing and the protocol goes back again to idle state. This flooding action maximizes the datagram probability of reaching its destination, while the messages identifiers technique applied can minimize the amount of necessary traffic in the network.

4.4 Datagram reception

If a datagram has been received the protocol switches to *"Receive a datagram"* state. In this state the message's identifier of the datagram is decoded to check if exists in the mobile's identifiers list. If the identifier exists in the list then the message was received and processed at least once, so it's immediately discarded because forward it once more is unnecessary.

If the message's identifier is new in the list, it means that is a new datagram to process. The identifiers list is updated with the new one. Later the destination IP address is checked, so if the datagram is for the receiver then it's all decoded and the application can process its fields and show the content and relevant message's information to the user.

4.5 Datagram forwarding

Continuing with the last state, in the other hand, when the message's addressee isn't the receiver then it must be forwarded to others mobiles. In this case the protocol switches to the *"Forward datagram"* state.

First the TTL's (Time to Live) field is checked to determine the maximum allowed hops for this datagram. If this field is zero then the datagram has been forwarded the maximum number of times so a next forwarding will be unnecessary. In this case the datagram is discarded. It's the complementary technique applied to avoid congestion.

When the TTL's field is bigger than zero then it can be forwarded, therefore TTL is decremented one unit and is sent to the network using broadcast addressing as a normal datagram send. Last, the protocol returns to the idle state.

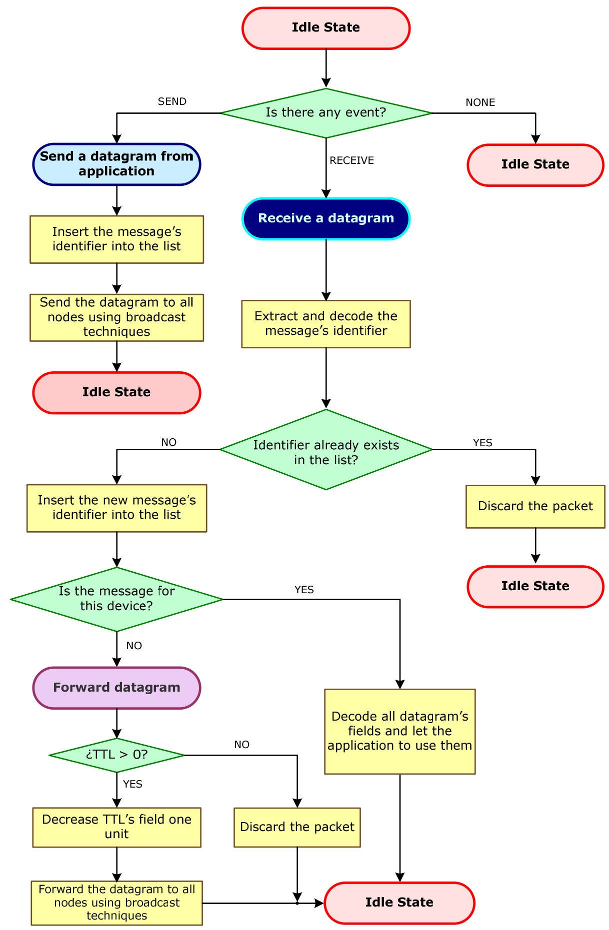

Figure 2: Protocol flowchart

5. Client Software application terminal

Client software application terminal is based on a MIDlet with the graphic components LCDUI from the operating system, to allow the user interact with the application with the system keyboard using menus.

Addressing is user predefined and is based on IPv4. This software is a MIDlet developed in Java 2ME with the following functionalities:

– Ability to send SMS up to 250 characters length, although this length can be changed.
– ACK messages can be sent from destination to source. ACK messages are automatically generated by the protocol designed, and use the same technique described to usual messages.
– Warning information is showed when the memory of buffer received messages is full.
– Provide information about date and time of message reception, route from source to destination and number of hops done.
– Numbering of messages and user contacts in the device's persistent memory.
– Address book with a capacity of 100 contacts, including error control's mechanisms to avoid duplicity of them.
– Storage up to 50 received messages.
– Own IP query and for each contact in the address notebook.
– Warning sounds depending on reception or forwarding messages with volume control allowed.
– Warning errors when using application.
– Capacity to keep user configuration after closing the application.

6. Real Test, conclusions and Future work

We obtained the first working protocol and software application from the emulator incorporated in Netbeans IDE and the standard exit. To deploy a real cooperative network we install the software in 5 Nokia N80 mobile phone terminals. This application is intended to provide SMS communications for short groups of people like hikers, friends in a shopping centre, confidential warnings to VIP people,…

We found some executing differences to solve between the emulator and the commercial phone. The most important are related to RMS registers and images screen shows.

During test, apart from the correct working of the protocol and the user application, we test how depending of the user mobility the route message was different from the ACK message, due to the protocol nature design. In some cases, the communication has happened with any hops despite of being inside the radio coverage, due to temporary low quality signal between source and destination.

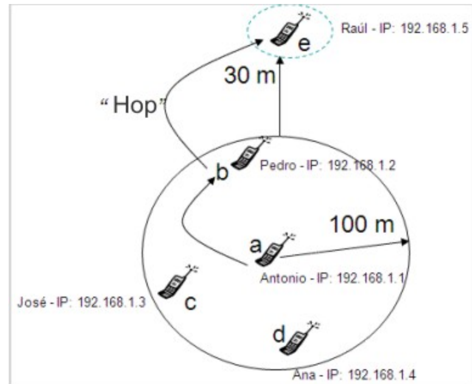

Figure 3: Example of a real test deployment

Figure 3 shows an example of a real test deployment, where IP addressing belongs to a private class C. In this figure we can see as Raul receives the message from Antonio thanks to the relay work of the intermediate Pedro's phone.

The default WLAN phone's configuration allows transmitting with a 100 mW power, reaching up to 100 m in open places. Relay technique make possible to extend this theorical distance without limit, if we can put always intermediate terminals to receive and forward the information to next terminal until destination one. In close places, the maximum distance may vary between 5m. and 15m. It depends of the numbers of walls and other existing obstacles, although in general there is a great dependence of the used hardware.

As a general conclusion the application works perfectly in the environment and for the purpose as it was designed. Many test show a lack of blocking tasks, when handling communications and storing data in RMS registers.

7. References

[1] http://www.ieee802.org/15/ Institute of Electrical Engineers. The Working Group for Wireless Personal Area Networks (WPAN)
[2] http://www.ietf.org/html.charters/manet-charter.html Mobile Ad-hoc Networks IETF workgroup.
[3] http://www.ietf.org/rfc/rfc2501.txt MANET Perfomance Issues, Corson & Macker, (1999)
[4] Taxonomy for Routing Protocols in Mobile Ad Hoc Networks. SICS Tech. Report, Feeney, Laura M., (1999).
[5] DUCHA: A new dual-channel MAC protocol for multihop ad hoc networks. IEEE Transactions on Wireless Communications, HQ Zhai, HF Wang, YG Fang, (2006).
[6]http://www.nokia.es/A4429084.
[7] http://java.sun.com/javame/index.jsp Java ME Platform: the Most Ubiquitous Application Platform for Mobile Devices.