Adapting a Captive Portal to enable SMS-based Micropayment for Wireless Internet Access

Jaume Barceló, Miquel Oliver, and Jorge Infante

Network Technologies and Strategies Research Group, Universitat Pompeu Fabra,
Passeig de Circumval.lació 8,
08003 Barcelona, Spain
Tel: (+34) 93 542 29 42 Fax: (+34) 93 542 24 51
{jaume.barcelo,miquel.oliver,jorge.infante}@upf.edu

Abstract. This paper introduces a micropayment mechanism suitable for Wireless Internet Access Providers. It is proposed that the users obtain the credentials that allow them to surf the web after sending a Premium-rated SMS, thus avoiding a direct payment relationship between the user and the WISP. Mobile users are familiar with Premium SMS and consider them a secure and convenient payment method, because of the existing trust relationship between users and Mobile Network Operators. A third party named SMSBroker that acts as intermediary between the MNO and the WISP is also required in practice. The concept has been implemented and tested in a real wireless access network.

1 Introduction

With the development of low cost hardware for wireless networking based on IEEE 802.11b/g technology, many access points are deployed in cities around the world and wireless access has already become an ubiquitous way to connect to the Internet.

At present, second and third generation mobile networks offer connectivity with well-defined authentication and authorization procedures for a large customer base, but at lower speed and higher costs than WiFi Networks. In this article we propose a Wireless Internet Service Provider (WISP) that takes advantage of the billing relationship between a user and a Mobile Network Operator (MNO) to charge for the offered service. Our scheme is based on the use of Premium-rated Short Message Service (PSMS).

2 Micropayment

A micropayment system has to allow the payment of small quantities (up to 1 Euro) for digital goods and services. [1] identifies the key characteristics determining the success of a micropayment schemes as trust, ease of use, pervasiveness and transaction speed.

Many Internet-based micropayment solutions exist and [2] provides an extensible way to embed in a web page all the information necessary to initiate a micropayment (amounts and currencies, payment systems, etc).

Although this solution will allow content providers to charge for their Internet digital goods, it might not not be the best solution when the service to be sold is the Internet access, since an Internet-based micropayment requires an already working Internet connection to succeed.

The mechanism that we have adopted in our solution involves the sending of a *premium rate* (overpriced) SMS by the customer. *Premium rate* SMS (PSMS) payments started appearing in 2001 and have since then become a norm for spontaneous payments in conjunction with TV and Radio broadcasts. Recently they have been incorporated as an additional marketing channel that permits user interaction and instantaneous feedback.

The success of phone-based payments is based on the following factors:

- No additional registration required for the user: any mobile phone owner with access to a telecom network with premium services could make payments through this medium. Users already have trusted billing relationship with their MNO, either using a prepayment card or a monthly bill.
- Ease of use, the user needs only to send an SMS.
- Pervasive. Almost everyone has a mobile phone handy and is familiar with the sending of SMS.
- The transaction speed, usually between one an two seconds, is perceived as fast by human users.

The main drawback of SMS-micropayments is the high transaction cost. If a customer pays about 1 Euro for a good or service, the merchant receives less than the half of it. This price overhead makes this solution useful only for occasional or impulse users. Other market segments, including frequent intensive users might be charged using different payment schemes, such as a flat monthly bill.

3 Money and Message Flow

Fig. 1 introduces the actors involved in our proposal. The first actor is the user and needs both a GSM/UMTS terminal and a WiFi enabled terminal. The second element is the Mobile Network Operator (MNO) that owns the Radio Access Network and a Core Network.

The SMSBroker acts as a gateway between the MNO network and the Internet. Actually it is connected to all the MNO operating in the country, thus making it possible to send and receive messages from any user. The SMSBroker has a SMSGateway that parses SMS coming from the MNO interfaces and resends them as HTTP requests to the Access Server of the WISP. The WISP has deployed a number of Access Points that form the Access Network, separated from the Internet by a dynamic firewall known as Access Server.

Fig. 1 also shows how the money is divided between the actors. The values have been taken from the Spanish case and might differ in other countries. However, they can be used as a reference for the European market.

PSMSs have fixed prices:0.15, 0.30, 0.60 and 0.90 Euro. Conversations with local WISP lead us to the conclusion that they were willing to offer 20 minutes of Internet access for 0.40 Euro. This means that the final user has to be charged 0.90 Euro, since less than 50% of the PSMS price reaches the WISP.

The user pays 0.90 Euro by sending a PSMS. This amount of money plus VAT (16% in our country) is going to be charged to the user's prepaid card or in a monthly bill. The MNO retains about 0.30 Euro and pays the rest to the SMSBroker which in turn pays about 0.40 to the WISP.

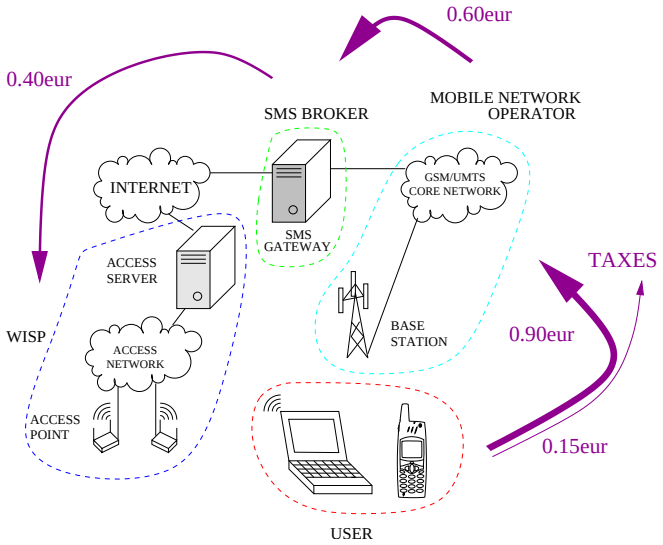

Fig. 1. Actors involved in a SMS-Enabled Internet access and the money flow among them.

The purpose of the next figure (Fig. 2) is to clarify the message interchange between the different entities that finally grants user access to the Internet.

Message labeled one (1) in the figure is the initial PSMS that the user sends. The MNO forwards it to the SMSGateway. The SMSGateway generates a HTTP request including the user's mobile phone number (Mobile Subscriber Integrated Services Digital Network number, MSISDN) as a parameter.

A servlet in the Access Server receives the HTTP request. It stores the MSISDN – or an MD5 hash of the MSISDN if privacy is a concern – together with a randomly generated 4-digit numerical password. A welcome message and the password are sent as an answer to the HTTP request.

This answer is labeled as two (2) in the figure and reaches the SMSGateway that converts it into an SMS and sends it to the user through the MNO.

Now the user has the password and turns on a laptop that gets networking configuration using DHCP [3] and opens a browser. A form appears on the screen

asking for credentials and the user fills it in using the mobile's phone number and the received password. This is labeled as three (3).

After the authentication and authorization procedure, the user is allowed to browse the web for 20 minutes, labeled as four (4).

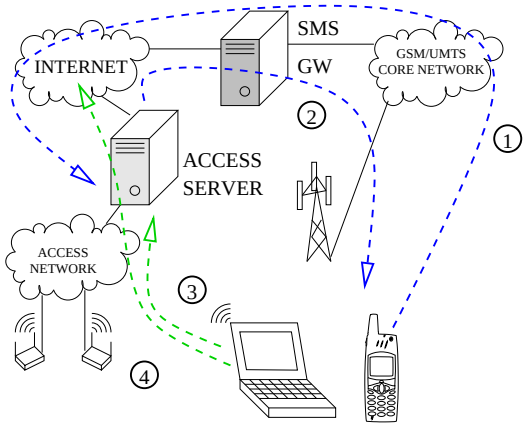

Fig. 2. Messages interchanged to grant Internet access to the user.

The user does not need to install any special software, since the configuration and authentication process is performed using well known protocols such as DHCP and HTTPS.

4 PSMS and SMSBroker

Premium rate SMS is offered by many MNO in Europe. The operator has a monthly billing relationship with millions of users. A merchant can purchase a phone number from the operator worth, for example, 0.90 Euro. Every time a user sends an SMS to that number, 0.90 Euro will be added to that user's phone bill and the MNO will pass through a subset of that money to the merchant.

The merchant probably would need to obtain *the same* number from all MNO operating in the country, and additionally would need direct connection to every MNO network. This can be unaffordable for a medium or small business.

The solution to this situation is to take advantage of a third party called SMSBroker. This actor obtains a number associated with given PSMS price from the different MNO, the same number from all MNO. This number can be used to offer different services from different providers, sharing the costs. In our case, the number is *7212* and the first parameter (word) in the message has to be *upf*. The number together with the first parameter identify uniquely a service provider.

Using a SMSBroker removes the requirement of the direct connection of the service provider with all the MNO networks. Actually, the server provider only

needs a connection to the Internet. This is because the SMSBroker connects to the MNOs and provides a SMSGateway that converts PSMS messages in HTTP requests that then are forwarded through the Internet.

As explained before, the destination number of the message together with the first parameter identifies a service provider, that is, a host name and port number where the PSMS are going to be forwarded.

The answer to the HTTP request originated by the SMSGateway can be a text SMS, but also a logo or a ringtone. The special code *zero vertical slash* (0|) have to be attached at the beginning of any answer to indicate to the SMSGateway that the content should be interpreted and resent as a text message.

5 Access Server

The Access Server performs mainly two tasks: Answering PSMS messages forwarded by the SMSGateway and authenticating the users controlling the access to the Internet. To accomplish the second task, the *NoCat captive portal* has been used.

5.1 Captive Portal

When providing public Internet access, users must be securely identified when they connect, and then allocate only the resources they are entitled to. The built-in security features of 802.11b are designed to create a private network with trusted clients but they aren't well suited for public-access networks.

According to [4] a captive portal is a router or a gateway host that will not allow traffic to pass before the user is authenticated. It is essentially a mechanism to prevent users from accessing network resources (usually Internet access) until they have authenticated with a server. Typically a captive portal is used at wireless hotspots. allowing the user to log in, authenticate and use the network according their privileges. The users do not need to know a particular address to authenticate. Whenever unauthenticated users attempt to browse, they are transparently redirected to the authentication page.

Two of the most well known open source implementations of the captive portal concept are WifiDog [5] and NoCat [6] [7]. The first one is a fully embeddable solution that can run on the Access Point. The second is written in Perl and needs two Linux servers to run: NoCatGateway and NoCatAuth

In our approach, the NoCat captive portal have been modified to host the desired functionality of allowing Internet access only to those users that send a PSMS message to pay for the service.

The NoCatAuth is implemented as CGIs running on the Apache web server and is in charge of the following tasks:

- Presents the user with a network login prompt via an SSL-protected Web page.
- Verifies user credentials.

– Securely notifies the wireless gateway of the user's status, and authorizes further access.

On the NoCatGateway side, the software

– Manages local connections.
– Sets bandwidth-throttling and firewall rules modifying the Linux *iptables*.
– Times out old logins after a user specified time limit.

The system was designed to preserve trust. The gateways and end users only need to trust the Auth system, which is secured with a registered SSL certificate. Passwords are never given to the wireless gateway (thus protecting the users from any malicious node owners), and gateway rules are modified only by a cryptographically-signed message from the Auth system, protecting the gateway from users or upstream sites trying to spoof the Auth system. To make this possible, the Auth Server's public PGP has to be distributed among the gateways.

The connection process The connection process involves several phases that are detailed in Fig. 3:

1. Redirect. The users in the WISP coverage area are immediately issued a DHCP lease.All access beyond contacting the Auth service is denied by default. When users try to browse the Web, they are immediately redirected to the gateway service, which then redirects them to the Auth system's SSL login page (after appending a random token and some other information to the URL line).
2. Connect Back. Once the user has logged in correctly, the Auth system then prepares an authorization message, signs it with PGP, and sends it back to the wireless gateway. The gateway has a copy of the Auth service's public PGP key, and can verify the authenticity of the message. Since part of the data included in the response is the random token that the gateway originally issued to the client, it's very difficult to fake out the gateway with a replay attack. The digital signature prevents the possibility of other machines posing as the Auth service and sending bogus messages to the wireless gateway.
3. Pass Through. After message verification, the NoCatGateway modifies its firewall rules to grant further access, and redirects the users back to the site they were originally trying to browse to.

To keep the connection open, a small window is opened on the client side (via JavaScript) that refreshes the login page every few minutes. Once the user moves out of range or closes the *renewal* pop-up window, the connection is reset and requires another manual login.

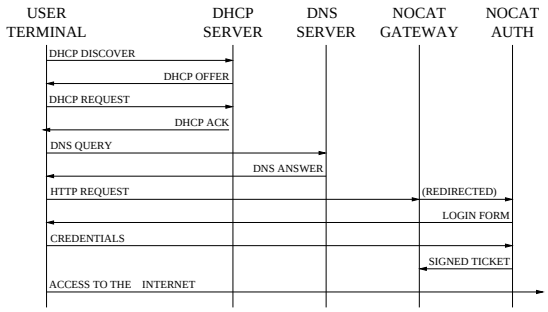

Fig. 3. Connection process for the NoCat captive portal

Authorization Sources NoCatAuth admits several authentication sources, including password files, *Radius*, *MySQL* database and the flexible *Pluggable Authentication Modules* (PAM). We decided that the database method was the most convenient for our purposes because it allowed the inclusion of time-management information and could be easily accessed both by NoCat and by the servlet that attended the messages sent by the users. We used the database scheme included in NoCat as a starting point.

The most relevant table is called *member* and stores the user information. It contains the following fields: *url*, *description*, *created*, *modified*, *status*, *login*, *pass* and *name*. The most important are *login* and *password*, that contain the credentials that a user has to present for identification.

5.2 Modifications to the original NoCat

In order to implement the time-management information to control that the user receives 20 minutes of Internet access for each PSMS sent, the NoCat software and database scheme required some modifications. Two more fields were added to the *member* table mentioned above.

Changes to the database One field called *MinutesLeft*, that stores the credit (in minutes) of each user was included. When sending a PSMS, the user gets 20 minutes credit, and can increase this credit in 20 more minutes by sending another PSMS.

Another new field *pass_clear* contains the password in clear text and it is used to remember the password to the users whenever they send a PSMS to increase their credit. This passwords consist on four-digits numbers. It is obvious that these are not strong passwords, but convenience prevailed over security. It more important that these passwords are easy to remember and type. Therefore a design decision was made to make them similar to the Personal Identification Numbers (PIN) already used in mobile phones.

A table called *history* was added to the database. A user that exhausts the credit is deleted from the *member* table and all the information related to that

user is moved to the *history* table. Finally, another table was included to store all the PSMS received together with the sender's number.

Renewal window NoCat requires that every user maintains the session (and thus the firewall) open by means of a *renewal window* (See Fig. 4). This is actually a browser window opened using JavaScript at login time that contains a form with the user information and refreshes automatically periodically, notifying the gateway that the user is still active.

The renewal period was chosen to be 60 seconds, and the event is used to decrease in 1 the credit in minutes of the user. The renewal window is used to inform the user of the credit left, using a JavaScript counter that shows minutes and seconds. The same window provides information to the user about how to increase the credit, basically by means of sending another PSMS.

Every minute the information of the pop-up window is updated. After sending a PSMS to increase the credit, the database is updated immediately, but the user has to wait until the seconds counter of the pop-up window reaches the value zero and the renewal actually happens to see the updated credit in the pop-up. This is because HTTP is a client initiated protocol, and therefore the server has to wait until the client initiates the transaction to provide the updated information.

To sum up, every renewal implies reading the value in the field *MinutesLeft*, subtracting one to that value and re-sending the updated value to the user's pop-up window.

The user can actively close the session by clicking on the *logout* button of the pop-up window. At this moment a bye-bye message remembering the password is presented to the user, that will retain the minutes left for the next session.

Conversely, the user can close the browser window, leave the coverage area or turn off the computer without actually logging out. In this case, the gateway detects that the renewal has not occurred and closes the session for that user. As before, the user retains the minutes left for the next session.

When renewal occurs and the field *MinutesLeft* reaches the zero value, the pop-up window does not contain the counter any more, but a message informing that the connection is being closed. At this time the user is removed from the *member* table in the database and included in the *history* table. A minute later, the firewall is actually closed for that user.

5.3 Processing Requests

The PSMS are received by the Access Server in the form of HTTP GET messages at the host and port specified at the SMSGateway. The SMSGateway can be configured to provide some information in addition to the actual the content of the PSMS, such as date, time, MSISDN and MNO identifier. In our case, the user's mobile number was required.

Listening at port 8080 in the Access Server's Internet interface there is an Apache Tomcat 5.5 running to attend the HTTP requests. The Web Application takes the form of a servlet that checks if the request contains the keyword *wifi*.

If so, it takes the sender's MSISDN and compares it to the field *login* of the existing entries in the table *member* described above.

If the user is not yet in the database, the servlet generates the four-digit password, creates a new entry in the *member* table with that user information, and answers the user with a welcome message and the password.

If the user already exists, it adds 20 to the *MinutesLeft* field that contains the credit. The answer to the requests informs the user that the credit has been successfully updated and sends the password as a reminder. This answer is sent to the SMSGateway that converts it into the answer of the original PSMS sent by the user.

The users will probably remember the passwords or otherwise save the answer SMS containing the password for later use. If not, they will have to send another PSMS to receive the password again, and increase their credit in 20 minutes as a side-effect.

All messages received at the Access Server are stored in the database for accounting purposes.

6 The user experience

This section describes, step by step, the user perception of the service. He or she arrives at the WISP coverage area and turns a computer that receives DHCP configuration. The Access Server is set as the default gateway. The user types any URL and the DNS request is resolved transparently. The HTTP request is intercepted and redirected by the *captive portal*. What the user can actually see in the browser is a login form. It consists on a logo identifying the service, a login and password fields and the terms of use.

The user is informed that is required to send a PSMS to number *7212* with the keywords *upf* and *wifi*, and how much is going to be charged for that message. The user sends the message and immediately receives the welcome message and the four-digit password.

Following the instructions, the user introduces the mobile phone number as a login and the password. After clicking on the login button, a message informing about the success of the authentication is presented for 5 seconds. After that, the user is redirected to the originally requested URL. The renew window pops-up, showing the credit in the form of a count-down counter (Fig. 4).

Now the user can surf the Net as long as the pop-up window is kept open. When the user does not need Internet access anymore, the logout button can be clicked to close the connection and save the remaining credit for another occasion (Fig. 5). If the Internet is needed again, the login and password have to be reintroduced in the login form.

The connection is closed also in the case that the pop-up window can not be refreshed. Possible causes are lack of wireless network coverage or the user closing the window or turning the computer off. As in the previous case, the session is closed and new session has to be opened if the user wants to surf again.

At any moment, normally when the credit is about to expire (the countdown timer approaching to zero), the user can send another message exactly equal to the first one to increase the credit in 20 minutes. In this case the answer says that the credit has been successfully increased and contains the password as a reminder. After some seconds (between 1 and 60) the pop-up window updates to show the increased credit.

Fig. 4. The pop-up window to keep the connection open. A count-down timer indicates the credit.

Fig. 5. Bye-bye window that appears after clicking the logout button.

6.1 Deployment of the service in a multiple WISP Access Network

The service has been deployed in an Wireless Open Access Network (OAN) [8] [9] [10], that is an access network shared by different providers. It was not feasible to create a large wireless access network only to test the new service, but it was easy to integrate the new service into existing infrastructure of the OAN. The coverage includes the University campus distributed in five different locations across the city, and is going to grow and merge with a municipal wireless access network and other universities.

A user connecting to the OAN and trying to browse is offered a list with all the different providers, nine in our OAN deployment. Some of this are well-established services with more than one thousand registered users. Other providers are research-oriented or under development and have only a few registered users. This is the case of the SMSMicropayment service. Since the OAN is supported by the University, it can be used only for academic and research purposes. Therefore, only a small group of selected volunteers where invited to test the service.

6.2 The test

The test was run from December 1st to December 20th. Seven of the invited testers actually participated sending a total of 30 PSMS with the key words *UPF WIFI* to the number *7212*. Each of this messages had a cost of 0.30 Euro (plus VAT), much lower than 0.90 planned in the business case. The reason of using a lower fare is that it was only a test and only the MNO expenses had to be covered. Seven volunteers opened a total of 54 Internet-access sessions during their tests. The results obtained by laptop users where positive, however PDA users were not able to benefit from the service because the PDA browser did not open the *renewal* pop-up window. The result was that the Internet session closed every minute and the user had to re-introduce the telephone number and password every time.

The test was performed at the university premises using the university OAN. Therefore the test had to be necessarily bounded and limited, to avoid raising suspicion on someone obtaining economic profit exploiting the University facilities.

7 Future work

7.1 Combining payment methods

It is simple to extend the presented PSMS-Micropayment approach to include other payment methods. Being the wireless access network a metropolitan network with coverage inside cafes and in terraces, the cafe owner could buy a large number of usernames and passwords to the WISP at a reasonable price.

Then this usernames and passwords would be delivered to the cafe customers together with their drinks. The customers might choose to enjoy their Wireless Internet Access immediately or save it to use it later in any other coverage area. The customer that finishes the 20 minutes received with the drink and wants to keep browsing can either order another drink or send a PSMS to increase the credit.

Another possibility would consist in selling prepayment scratch-cards at kiosks with a considerably larger credit and lower price per minute.

Once the user is on-line, after making the first micropayment, subsequent browsing-time extensions purchases could be done using a myriad of Internet-based payment methods (e.g. paypal) that are much more flexible and do not have the economical overhead associated to PSMS.

The recent apparition of new PSMS worth 1,20 Euro offers a new alternative for users that plan to use the net longer than 20 minutes. Probably the WISP would be able to provide 40 to 60 minutes of Internet browsing to the user and still obtain a profit.

7.2 A PDA-friendly captive portal

Most of nomadic users that are potential clients for public wireless Internet access use *Personal Digital Assistant* (PDA) with small screen and limited web browser.

The NoCat captive portal requires that the browser opens a pop-up *renewal* window, something that is not in the capabilities of many PDAs. Therefore, one of the most urgent lines of work is the substitution or modification of NoCat to make the system PDA-friendly.

8 Conclusions

This paper describes PSMS-based micropayments and why they are convenient to charge the users for Wireless Internet Access. A model that involves the Mobile Network Operator and the SMSBroker, in addition to the user and the WISP, is offered as a solution. The cash flow between the actors and the message interchange that allow the user to connect to the Internet are analyzed and exemplified.

An open source captive portal have been modified to obtain a Wireless Access Server with the desired functionality. Then, the overall proposal has been implemented and a WISP have been deployed in a real Wireless Access Network to test the solution, and has been positively validated by users.

References

1. Kniberg, H.: What makes a micropayment solution succeed. M. eng. thesis, KTH/Applied IT, Sweden (2002)
2. Michel, T.: Common markup for micropayment per-fee-links. W3C Working Draft (1999)
3. Droms, R.: Dynamic Host Configuration Protocol. RFC 2131 (Draft Standard) (1997) Updated by RFC 3396.
4. Potter, B., Fleck, B. In: 802.11 Security. O'Reilly (2003)
5. Fils, I.S.: Wifidog. Downloadable open source captive portal (2005)
6. Flickenger, R.: Nocatauth: Authentication for wireless networks. O'Reilly Network (2001)
7. Erle, S.: Nocatgw and nocatauth. Downloadable open source captive portal (2003)
8. Battiti, R., Cigno, R.L., Sabel, M., Orava, F., Pehrson, B.: Wireless lans: From warchalking to open access networks. MONET **10**(3) (2005) 275–287
9. Infante, J., Oliver, M., Macian, C.: Wi-fi neutral operator: Promoting cooperation for network and service growth. In: ITS Conference on Regional Economic Development, Pontevedra, Spain (2005)
10. Barcelo, J., Macian, C., Infante, J., Oliver, M., Sfairopoulou, A.: Barcelona's open access network testbed. In: IEEE Tridentcom, Barcelona, Spain (2006)