

Plataforma de telecomunicaciones a la medida

Andrey Navarro-Monge
Escuela de Ing. Electrónica
Instituto Tecnológico de Costa Rica
San Carlos, Costa Rica
navarro08@estudiantec.cr

Límber Rodríguez-Rojas
Escuela de Ing. Electrónica
Instituto Tecnológico de Costa Rica
San Carlos, Costa Rica
lrodriguez@estudiantec.cr

Abstract—Prepaid communications platform that allows to send text messages and images through sockets and wirelessly to a cell phone using transceivers. The prepaid system has a database and signaling server in an AWS EC2 instance running RHEL8 that allows you to check your balance before making the communication to the end user.

Index Terms—Transceiver, Raspberry pi, sockets, Amazon Web Services, MariaDB.

I. INTRODUCTION

En este proyecto se desarrolló un sistema prepago de telecomunicaciones de envío de mensajes de texto e imágenes. Se crearon tres terminales de chat de texto; la primera corresponde a la terminal localizada en un teléfono celular y las otras dos en un sistema linux. Entre las terminales se encuentran un sistema empujado linux que realiza una comunicación inalámbrica con otro sistema linux; además de un servidor de señalización que se encarga de direccionar los paquetes de datos y de la logística del sistema prepago. Se realizaron pruebas de campo para comprobar el envío correcto de información entre terminales y funcionalidad de la base de datos prepago.

II. ESTRUCTURA DEL SISTEMA

La estructura general del sistema se muestra en la figura 1. El sistema se conforma por tres terminales, una en un teléfono celular y las otras dos en una consola de linux; además contiene una raspberry pi, un par de transceptores RF, un servidor de amazon y una base de datos MariaDB.

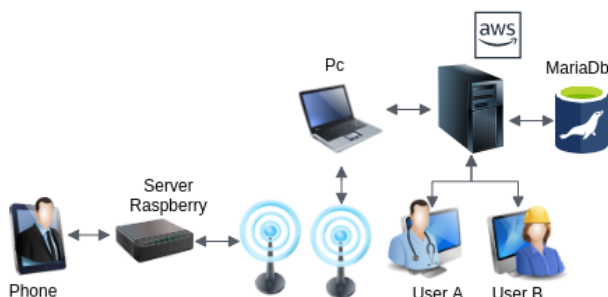


Fig. 1. Diagrama general del sistema.

A. Terminales

La terminal en el teléfono celular mantiene una conexión con la raspberry mediante sockets, en donde el socket servidor se encuentra en la raspberry y el socket cliente en el teléfono. El socket en la raspberry es el encargado de gestionar todas las operaciones desde y hacia el socket cliente; además, la raspberry contiene el módulo que se comunica con el otro sistema linux por medio de los transceptores.

En el teléfono se despliega un menú con las siguientes opciones disponibles:

- A-Send msj to A + your msj
- B-Send msj to B + your msj
- ?-How much do I have?
- 1-Top up your credit account
- 2-Check connection
- 3-Exit

En caso de que ninguno de los caracteres sea seleccionado se mostrará en consola un aviso de error y se mostrará nuevamente el menú.

Para enviar un mensaje de texto o una imagen hacia las terminales A o B, se escribe el caracter de destino más un espacio seguido del mensaje o del caracter especial '~' para indicar que se envíe una imagen.

Para consultar el saldo se envía el caracter '?' y para realizar una recarga de 10000 bytes se envía el caracter '1'.

Durante la transmisión de paquetes por los transceptores, se realiza una cuenta de la cantidad de bytes enviados y reenviados para posteriormente realizar una gráfica de barras estadística mostrando la eficiencia en la conexión y transmisión de datos por medio de los transceptores. Para mostrar esta gráfica de barras se envía el caracter '2'.

Finalmente, para cerrar la terminal se envía el caracter '3', se le indica al socket servidor que el socket cliente se ha cerrado y por ende el socket servidor también se cierra.

En el otro segmento del sistema, donde se encuentran las otras dos terminales que son iguales, con la diferencia de que la primera terminal abierta se le asignará una ruta con identificación 'A' y a la segunda 'B'. Ambas presentan las mismas opciones que la terminal del teléfono con la diferencia

de que no tiene la posibilidad de solicitar un estado de conexión de los transceptores.

Estas terminales contienen en su módulo un socket cliente que se comunica con el socket servidor localizado en una instancia de linux ejecutada en AWS.

B. Raspberry pi y sistema linux PC

Como se mencionó anteriormente, la raspberry pi ejecuta un socket servidor en su módulo. El subsistema identificado como Pc ejecuta un socket cliente que se comunica con el socket servidor localizado en AWS.

La raspberry pi y el subsistema Pc son los encargados de realizar la conexión inalámbrica y demás procesos de transmisión de datos usando los transceptores RF. Ambos son capaces de enviar y recibir paquetes, de determinar si un paquete se ha enviado correcta o incorrectamente, de confirmar la conexión y desconexión entre transceptores, y de informar a la terminal emisora si la información no ha sido entregada a la terminal receptora.

C. Transceptores RF

Estos son los módulos encargados de transmitir información de manera inalámbrica, para el desarrollo de este proyecto se utilizó un par de módulos **TLC1101**. Su máxima velocidad de transmisión (baud rate) es de 19200 bps y 20 bytes es la máxima cantidad de bytes que se puede transmitir en cada envío. Para asegurar una adecuada ganancia de transmisión es recomendable alimentar el transceptor al pin de 5V.

Uno de los módulos se conecta a la raspberry pi por medio de los pines externos que la misma incorpora, mientras que el otro módulo se conecta a la Pc por medio de un módulo de comunicación UART-USB. Todas las configuraciones y operaciones con los transceptores son realizadas por la raspberry pi y la Pc.

D. Servidor AWS y Base de datos

EL sistema cuenta con una máquina virtual EC2 utilizando RHEL8 como su sistema operativo. El servidor AWS, cuenta con dos funcionalidades principales: 1. Direccionamiento de paquetes a su destinatario final y 2. Verificación de saldo para realizar la transacción, de acuerdo a su base de datos.

Para esta primer función: El servidor de acuerdo al header con el que recibe el mensaje, elige el socket adecuado para el envío del mensaje. Además de esto, durante el proceso se incluye el identificador del emisor, de manera que al usuario destino, se le pueda notificar de cuál usuario es el origen del paquete.

Por último, se utilizó la base de datos relacional MariaDB en su versión 10.3. Esta base de datos incluye información de los usuarios registrados (ID del usuario) y su saldo actual. Nótese que este sistema funciona en conjunto con la primera funcionalidad, ya que el servidor primero hace la consulta a la base de datos y posterior a esto, redirige el mensaje si este cuenta con suficiente saldo.

Tabla 1. Formato de base de datos utilizado

Tabla de MariaDB utilizada	
nombre	saldo
A	722
B	53500
P	2400

III. DESCRIPCIÓN DE PROTOCOLOS DE COMUNICACIÓN USANDO TRANSCPTORES

En esta sección se explican los protocolos de conexión entre transceptores. Para obtener una percepción estadística del trasiego de información, se hace un conteo de la cantidad de bytes que se envían y se reenvían en caso de distorsión de la información.

A. Protocolo de inicio y de salida

Luego de que la raspberry o el Pc recibe un mensaje por socket, debe enviar el mensaje por los transceptores; sin embargo, antes de enviar el mensaje, debe realizar un protocolo de inicio para verificar el alcance de transmisión del otro transceptor.

Inicialmente, después de recibir el mensaje por socket, el primer transceptor envía un mensaje de cabecera ("EL5522TALLERDECOMU") hacia el segundo transceptor, el cuál está en estado de recepción de paquetes, y que cuando recibe el primer paquete lo primero que hace es compararlo con su mensaje cabecera, si ambos mensajes son correctos entonces le envía el código "InitCon" al primer transceptor y queda vinculado para una transmisión con el primer transceptor; si el primer transceptor reconoce el código entonces queda vinculado para transmisión con el segundo transmisor.

Si se diera el caso que el primer transceptor no recibe o no reconoce el código de respuesta del segundo transceptor, se procede a reenviar nuevamente el mensaje cabecera hasta alcanzar un número máximo de intentos. Si el segundo transceptor no reconociera el mensaje cabecera, simplemente lo ignora y vuelve a entrar en estado de recepción de paquetes.

B. Protocolo de envío de mensajes

En la figura 3 se muestra la lógica del protocolo de comunicación en el envío de texto.

Inicialmente, quien recibe la entrada del mensaje, debe enviar el paquete de datos por los transceptores, por lo que se debe implementar el protocolo de inicio mencionado en la sección III-A antes de enviar el paquete que contiene el mensaje.

Una vez el protocolo de inicio es exitoso se procede a enviar el paquete. El emisor envía el paquete de datos, el receptor lo recibe e inmediatamente lo envía de regreso al emisor, este último hace una comparación entre el paquete enviado y el recibido; si los paquetes son idénticos entonces el emisor envía una bandera (1 para paquetes correctos, 0 para incorrectos) de confirmación hacia el receptor para informarle que el paquete que había recibido anteriormente estaba correcto y por ende procede a realizar el protocolo de cierre de conexión.

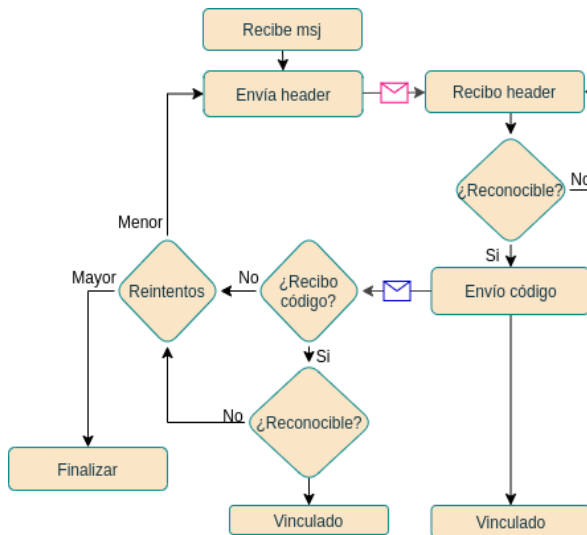


Fig. 2. Diagrama del flujo del protocolo de inicio.

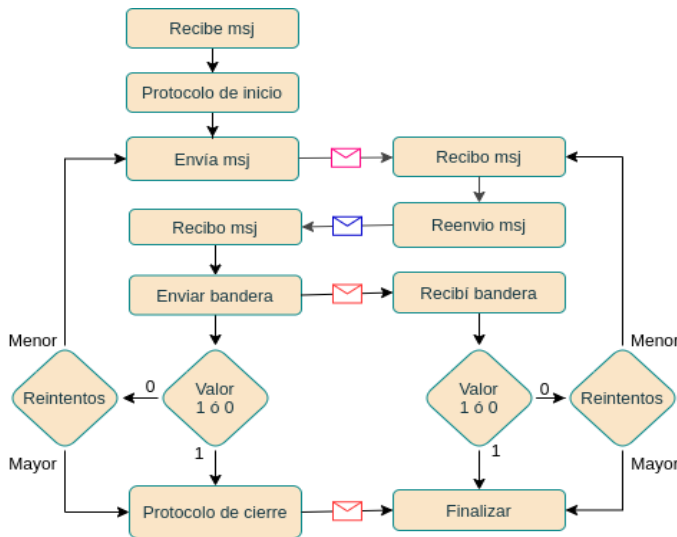


Fig. 3. Diagrama del flujo del protocolo en envío de mensajes de texto.

En caso de que el paquete fuese incorrecto, el emisor envía la bandera con el valor cero, el receptor la recibe y entiende que el paquete estaba mal, por lo tanto ambos vuelven a realizar el proceso de envío y recepción del paquete hasta que se verifique su integridad o de lo contrario si se alcanza un máximo número de reintentos se manda a cerrar la conexión entre los transceptores.

C. Protocolo de envío de imágenes

En la figura 4 se presenta la lógica implementada para el envío de la imagen entre transceptores.

El flujo en el protocolo de envío de imágenes es similar al de envío de mensajes de texto. Inicialmente el módulo que enviará la imagen recibe el caracter '~' como aviso de que se enviará una imagen. Seguidamente se procede con el protocolo de inicio para verificar alcance entre los transceptores; cuando

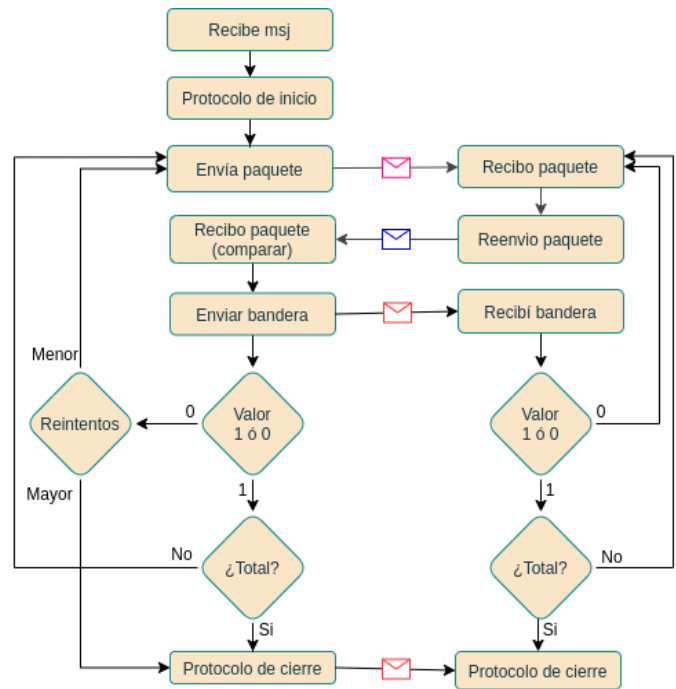


Fig. 4. Diagrama del flujo del protocolo en envío de imágenes.

se verifica la conexión se abre en formato de bytes el archivo de la imagen.

El archivo se irá leyendo de veinte en veinte debido a que esa es la mayor cantidad de bytes que puede enviar el transceptor por cada iteración. Una vez leído el paquete, se envía hacia el receptor, este lo recibe y lo reenvía inmediatamente hacia el emisor para que compare y luego confirme su integridad. Si la bandera es un uno, entonces se procede a leer y enviar el siguiente paquete hasta llegar al final del archivo de la imagen. Por otro lado, si la bandera es un cero entonces se vuelve a enviar el mismo paquete hasta que se entregue correctamente; en caso de que alcance el número máximo de reintentos se cancelará el envío y se cerrará la conexión.

En el envío de imágenes la conexión se cierra en dos posibles casos; el primero es que se intento enviar un paquete una determinada cantidad de veces y no se logró entregar correctamente y el segundo es que durante el envío de los paquetes los transceptores perdieron alcance en la transmisión RF. Cuando se presenta un error en la transmisión de imágenes se le informa al usuario quien estaba enviando la imagen.

IV. DESCRIPCIÓN DE FUNCIONALIDAD DEL SISTEMA PREPAGO

Como se mencionó en la estructura del sistema, la instancia AWS, se encarga de redirigir los mensajes a su usuario final. Sin embargo, primero es necesario realizar las consultas de saldo a la base de datos MariaDB. Si este usuario no cuenta con suficiente saldo, se debe evitar cobrar el monto del tamaño del mensaje. Caso contrario, se procede a descontar saldo y

redirigir a su destino. En la figura 5 se puede apreciar el flujo de envío con consulta de saldo.

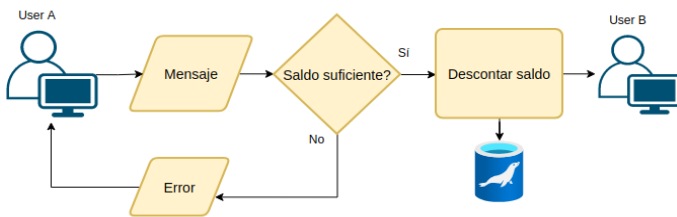


Fig. 5. Diagrama del flujo para consulta y envío de mensajes

El servidor AWS cuenta con 2 entradas principales para gestionar y consultar saldos. A continuación se detallan estas 2 opciones.

- **Opción ?:** Esta opción permite a los usuarios consultar su saldo. El servidor al recibir '?' detecta el usuario que envió el mensaje, y hace la consulta a MariaDB para posteriormente devolver el valor al usuario.
- **Opción 1:** Esta opción permite a los usuarios agregar 10000 a su saldo actual. De manera similar a la opción '?', el servidor al recibir '1' detecta el usuario que envió el mensaje, y hace la consulta a MariaDB para actualizar el saldo. Si la consulta se realizó exitosamente, se informa al usuario.

Nota: Cuando se envía una notificación o un error por saldo insuficiente desde el servidor AWS, aparece como destinatario del mensaje 'S', haciendo referencia al servidor.

V. RESULTADOS Y CONCLUSIONES

A. Envío de mensaje de texto entre las terminales

En el envío de mensajes de texto, el servidor de señalización permite que se envíen mensajes siempre que se tenga saldo, permitiendo que el saldo después de enviar el mensaje quede negativo. En caso de tener saldo negativo no se permite el envío de mensajes.

Desde el teléfono celular hacia usuario A o usuario B: En la terminal del teléfono se digita el carácter del destinatario (A o B) más el mensaje que se desea enviar, este mensaje se envía por sockets hacia la raspberry, en este punto la raspberry implementa el protocolo de inicio y de transmisión de texto con transceptores mencionado en la sección III. Después de que el subsistema Pc recibe correctamente el paquete, lo envía hacia el servidor AWS por sockets el cual direccionará el paquete y gestionará el sistema prepago. En caso de tener saldo se envía el paquete al destino final y se hace el cobro; caso contrario se le indica a la terminal emisora que no cuenta con saldo.

Cuando los transceptores no tienen alcance de transmisión se cancela la operación y se le informa a la terminal emisora que hubo error de conectividad.

Desde el usuario A o usuario B hacia el teléfono celular:

Cuando se envía un mensaje de texto desde el usuario A o B hacia el teléfono celular, se empieza por enviar el paquete de información por sockets hacia el servidor de señalización quien verificará si el usuario tiene saldo, en caso de que no tenga saldo se le informará y se finaliza el proceso de envío de ese mensaje. Si el usuario si tiene saldo entonces se procede a enviar el mensaje hacia el subsistema Pc el cual envía el paquete hacia la raspberry por los transceptores. Si se logra la transmisión de la información se le informa a la terminal emisora que se ha entregado el mensaje al destino, y en la terminal destino se muestra el mensaje recibido y el destinatario; dado que no se consiguiera la transmisión de la información se le informa a la terminal emisora que hubo un error de conectividad y por ende el mensaje no se entregó al destinatario.

Entre usuario A y usuario B: En el envío de mensajes de texto entre los usuarios A y B primeramente se envía el paquete al servidor de señalización que verifica el saldo del destinatario; si tiene saldo entonces procede a direccionar el paquete y además informa al destinatario que su mensaje ha sido entregado.

B. Envío de imágenes entre las terminales

Se envían las imágenes entre las terminales utilizando sockets y transceptores (para la comunicación inalámbrica). **Se descuenta el el saldo de acuerdo al tamaño de la imagen en bytes.**

Durante la transmisión de la imagen con los transceptores, se muestra en consola la cantidad de bytes leídos del archivo de la imagen, la cantidad de bytes que se envían y se reciben de vuelta (para verificar) , y un porcentaje del progreso de la transmisión. Si durante la transmisión se pierde un paquete o bien se envió incorrectamente se imprime el consola lo acontecido, en caso de que se pierda la conexión se notifica al usuario emisor que hubo un problema de conexión.

C. Recarga y consulta de saldo

Se permiten recargas utilizando la opción '1', **cuya recarga es equivalente a 1000.**

Para la consulta de saldo, se utiliza la opción '?', el cual devuelve el saldo actual al usuario.

D. Diagrama de barras de verificación de conexión

La terminal del teléfono presenta la opción de verificar la conexión, al seleccionarla se le solicita al módulo que está en la raspberry que le entregue (registro histórico) la cantidad total de bytes enviados y reenviados para mostrarle en la terminal una gráfica de barras con los datos del registro histórico.

VI. CONCLUSIONES

- El uso de una instancia AWS permitió tener acceso a un equipo virtual eficiente que realiza las mismas operaciones que un equipo físico.

- Termux facilita una terminal de ambiente Linux dónde se trabaja directamente sin requerimientos de rooting.
- El protocolo de inicio de transmisión permite detectar problemas cuando el destinatario es inalcanzable, evitando enviar paquetes de manera innecesaria.
- El transceptor utilizado puede enviar o recibir por cada iteración un máximo de 20 bytes.
- EL sistema de detección de errores evita que la comunicación entre el emisor y receptor pierda sincronización y que el mensaje se deba enviar de nuevo por completo.
- Utilizar el servicio de MariaDB dentro de la misma instancia AWS, permite hacer las consultas de manera rápida.
- Un aumento en la cantidad de baudios por segundo aumenta la velocidad de transmisión de datos.

VII. RECOMENDACIONES

- Utilizar un control de versiones de código de programación Gitlab.
- Comprender bien el uso de sockets servidor y cliente, y de la función select().
- Utilizar docker containers para facilitar la migración a AWS y evitar problemas de dependencias.
- Evitar utilizar etiquetas en los códigos de programación.
- Hacer el sistema escalable para más usuarios.
- Hacer una aplicación para el administrador de cuentas que permita modificar, agregar saldo y cancelar cuentas.
- Ofrecer más opciones para agregar saldo o para seleccionar distintas imágenes.
- Evitar utilizar librerías abstractas e incomprensibles para el manejo de las imágenes.
- Utilizar el software **PuTTY Configuration** para configurar los transceptores.

REFERENCES

- [1] Calvert, K. L., Donahoo, M. J. (2009). TCP/IP Sockets in C: Practical Guide for Programmers. Netherlands: Elsevier Science.
- [2] Van Winkle, L. (2019). Hands-On Network Programming with C: Learn Socket Programming in C and Write Secure and Optimized Network Code. United Kingdom: Packt Publishing.
- [3] Jorgensen, B. (2019). Beej's Guide Network Programming , Using Internet Sockets.
- [4] Linux man pages online. url: "<https://man7.org/linux/man-pages/index.html>"