

PROTOTIPO PARA POSICIONAMIENTO REMOTO DE UNA ANTENA DE TELECOMUNICACIONES POR MEDIO DE UNA CONEXIÓN SEGURA VPN

Aragón Banderas Osbaldo; Arce Valdez Jesús Leonel; Hernández Salazar Flabio Cesar; Campos Vázquez Leonardo; Rivas Fernández Brenda
Instituto Tecnológico Superior de la Región de los Llanos
División de Ingeniería Mecatrónica
C. Tecnológico # 200 ote. C.P. 34700 Cd. Gpe. Victoria Dgo. Mex.
Tel. (676)8823712
osbaldo.ab@regionllanos.tecnm.mx

RESUMEN.

Los sistemas de inclinación remota (RET) son una herramienta que permite reposicionar antenas de telecomunicaciones desde una ubicación remota. En el presente proyecto se propone el diseño e implementación de un sistema de este tipo, al que se puede acceder desde internet por medio de un servidor VPN (Virtual Private Network), de esta manera la interfaz de conexión entre el operador y el sistema se establece bajo un esquema de seguridad confiable, puesto que las redes VPN son bastantes seguras al solo permitir el acceso a la red si se tiene una determinada credencial. Se propone el diseño mecánico para el funcionamiento del sistema, así como el proceso de creación del servidor VPN en una Raspberry Pi y la implementación de un servidor web en una tarjeta NodeMCU que funge como interfaz entre el sistema y el usuario. Por último, se evaluaron los tiempos de respuesta del sistema y el rango en el cual se puede posicionar la antena.

Palabras Clave: RET, VPN, antenas, posicionamiento remoto.

ABSTRACT.

Remote electrical tilt (RET) systems are a tool that allows the repositioning of telecommunication antennas from a remote location. In this project, the design and implementation of a system of this type is proposed, which can be accessed from the internet through a VPN (Virtual Private Network) server, in this way the connection interface between the operator and the system is established under a reliable security scheme, since VPN networks are quite secure by only allowing access to the network if you have a certain credential.

The mechanical design for the operation of the system is proposed, as well as the process of creating the VPN server on a Raspberry Pi and the implementation of a web server on a NodeMCU card that acts as an interface between the system and the user. Finally, the response times of the system and the range in which the antenna can be positioned were evaluated.

Keywords: RET, VPN, antennas, remote positioning.

1. INTRODUCCIÓN

El prototipo está basado en el estándar de sistemas de inclinación remota (RET). Se definirá el problema del cual surge la necesidad de implementar este tipo de tecnología en la infraestructura de telecomunicaciones de nuestro país, haciendo hincapié en la necesidad de un sistema de este tipo en la red de cobertura móvil nacional para poder hacer frente a la alta demanda de usuarios en constante crecimiento. Será importante mencionar los diversos conceptos relacionados con las formas

de recepción y emisión de las antenas de telecomunicaciones, así como la manera en que distribuyen su cobertura dependiendo de diversos factores, tales como diseño, dimensiones, orientación, energía, etc. Y posteriormente se describen las herramientas y tecnologías en materia de software y hardware involucradas en el desarrollo del prototipo. Para concluir con los resultados obtenidos al realizar mediciones de tiempos de latencia desde la instrucción hasta la ejecución además de rangos y características del prototipo en cuestión.

1.1. Problema

En México el uso de redes de telefonía móvil se ha incrementado tal como se observa en datos del Instituto Nacional de Estadística y Geografía (INEGI) [1], pasando de un 71.5 % de la población total del país con acceso a un teléfono móvil en el año 2015 hasta un 75.1 % en el año 2019. Esto revela una necesidad por parte del sector de telecomunicaciones para aumentar la cobertura de sus redes y optimizar el ajuste de estas, pues la alta demanda de los servicios provoca que la infraestructura de redes se incremente y pueda sufrir interferencias.

Para aumentar la cobertura demandada se necesita realizar el ajuste de posición de la antena con respecto a un satélite de comunicación, para ello, suelen utilizarse los sistemas de inclinación remota (RET) que incrementan la cobertura de la estación base mientras reduce la interferencia [2]. Los sistemas (RET) presentan ventajas como alta precisión y amplia gama de ajustes [3]. Para lograr el ajuste se necesita posicionar la antena en base a dos magnitudes: la elevación y el azimut. La elevación corresponde al ángulo de elevación formado entre la dirección de onda radiada y la horizontal de la base terrena, el azimut corresponde al ángulo de apuntamiento horizontal de una antena [4].

El ajuste puede realizarse de forma mecánica con motores especializados como el YAESU 6550 [5], sin embargo, estos sistemas son costosos y presentan poca flexibilidad para poder integrarse con otro tipo de tecnologías como son el control de forma remota dentro de una red segura VPN [6].

1.2. Objetivos.

Diseñar un prototipo de sistema de inclinación remota controlado por medio de una conexión segura tipo VPN para

manipular la posición de una antena de telecomunicaciones de manera remota a través internet.

2. SISTEMA DE INCLINACIÓN ELÉCTRICA REMOTA (RET)

Los RET se utilizan principalmente para monitorear y controlar dispositivos de línea de antena (ALD) como la unidad de control remoto (RCU) y el amplificador montado en la torre (TMA). También se trata de un estándar producido por Antena Interface Standards Group para garantizar la compatibilidad e interoperabilidad de los dispositivos ALD y la infraestructura de control [3]. El sistema puede aumentar la cobertura de la estación base de manera efectiva al tiempo que reduce la interferencia en la región adyacente.

Los RET también tienen ventajas como ajuste en tiempo real, alta precisión y una amplia gama de ajustes. Para ello debe ajustarse la dirección de la antena en base a dos coordenadas, el ángulo de inclinación de la antena con respecto al eje z y el ángulo de azimut, que se corresponde al ángulo entre la antena y el eje x. En la figura 1, se tiene la estructura básica de un sistema RET se aprecian los dos ajustes, las antenas y los amplificadores de torre. [4]

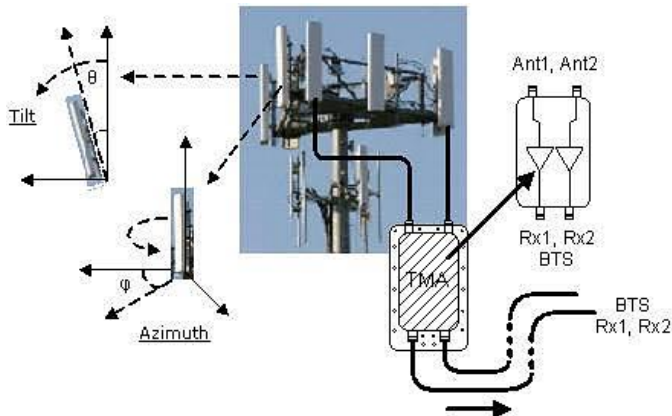


Figura 1. Estructura de un sistema RET

3. DESARROLLO DEL PROTOTIPO.

3.1. Estructura General

Se realizó un diagrama esquemático del prototipo con la distribución de los componentes del sistema, con un microordenador Raspberry Pi 3b+ que funciona como servidor VPN y permite a equipos de cómputo conectados a Internet acceder a la misma red local que la Raspberry. En esta red local se crea un servidor web utilizando un microcontrolador NodeMCU con conectividad WiFi. Dicho servidor es utilizado para establecer los ángulos a los que se debe trasladar el sistema de posicionamiento, y en base a la información que se introduce al servidor, esta información se procesa para el driver controlador de los servomotores desde el NodeMCU, por medio de un protocolo de comunicación serie tipo I2C.

La energía necesaria para operar los motores se obtiene de una fuente conmutada de 5 V y 3 A. Cada servomotor se conecta a las salidas del controlador y así un usuario conectado a internet puede acceder a la red local mediante servidor VPN de la Raspberry Pi, sin embargo, para ello necesita de las credenciales válidas con lo que se logra una operación a distancia del sistema de posicionamiento de forma segura. El esquema general del prototipo se tiene en la figura 2.

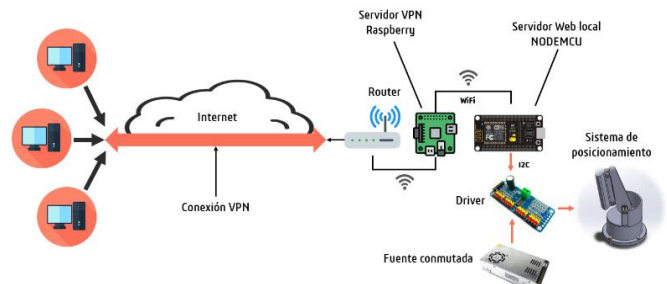


Figura 2. Diagrama esquemático del prototipo

3.2. Diseño del modelo mecánico.

En la figura 3 se tiene el modelo del sistema de posicionamiento diseñado y simulado en SolidWorks, como se puede observar es un mecanismo de dos grados de libertad, consta de una base para fijar el dispositivo a la estructura (bancada) de una torre, dentro de la base se encuentra fijo un servomotor que permite modificar el ángulo de azimut de la antena en un rango de 180°.

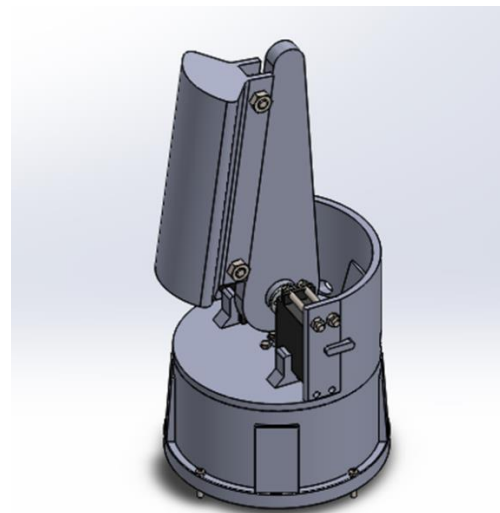


Figura 3. Sistema de posicionamiento

En la Figura 4 se pueden observar distintas vistas de la base del mecanismo, así como las dimensiones de cada una de sus partes, se tiene una ranura para ajustar el servomotor, algunos orificios para la salida de los cables y unos nervios que

distribuyen la carga que soporta la base evitando deformaciones en el material.

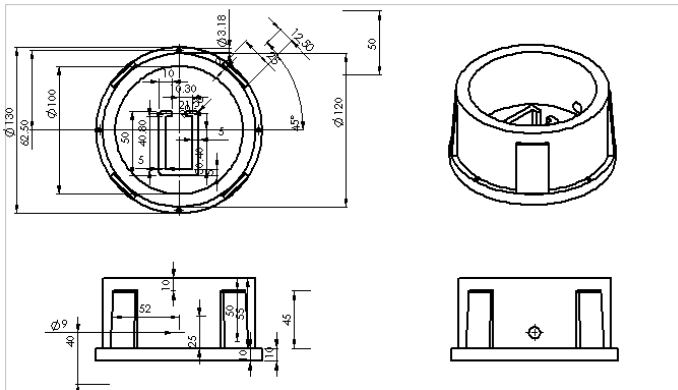


Figura 4. Plano de la base fija

La estructura se elaboró con impresión 3D en PLA.

3.3. Implementación del servidor VPN.

Utilizando una Raspberry Pi 3b+ con el sistema operativo Raspbian se creó el servidor VPN [6] de la siguiente manera:

1. Se configuró una dirección IP fija para la Raspberry, debido a que al encenderse el router asigna una dirección diferente en cada ocasión, en base a los dispositivos conectados, la dirección usada fue 192.168.1.204 con una máscara de subred 24, en la opción Router se introdujo la dirección de la puerta de acceso del router utilizado para la conexión de internet, 192.168.1.254 y como servidor DNS se utilizó una propiedad de Google con dirección 8.8.8.8.
2. Se procedió a establecer una dirección IP externa fija para poder conectarse al servidor VPN desde fuera de la conexión local de internet, para ello se utilizó un servicio de DNS dinámico como NO-IP. Este permite asignar un nombre de dominio a una dirección IP pública que cambia continuamente, permitiendo su conexión a esta. Para crear el servicio DNS en la página oficial de NO-IP se creó una nueva cuenta, con el nombre del servidor DNS [7], un correo electrónico y una contraseña. Luego de validar la dirección de correo electrónico se procedió a configurar el servidor DNS, para ello se le asignó el nombre de sisrep.ddns.net, el cual se corresponde a un servicio gratuito con vencimiento de un mes. A este nombre de dominio se le asignó la dirección pública del modem: 189.169.104.74.
3. Una vez realizada la configuración del nombre de dominio, se instaló el cliente No-IP en la Raspberry Pi con los comandos:

```
wget http://www.no-ip.com/client/linux/noip-duc-linux.tar.gz
tar -zxvf noip-duc-linux.tar.gz
```

```
cd noip-2.1.9-1
sudo make install
```

Comandos encargados de descargar y descomprimir los archivos de instalación necesarios para posteriormente instalar el cliente. Al teclear el último comando se abrió una interfaz en la cual se introdujeron los datos de validación de la cuenta de No-IP utilizada, tales como el correo electrónico y la contraseña, además de que se seleccionó el nombre de dominios a utilizar por el cliente, que para este caso fue sisrep.ddns.net.

4. El siguiente paso fue abrir los puertos del router para que redirija las peticiones del cliente al servidor VPN. Para realizar esto en el navegador se tecleó la dirección de la puerta de acceso del router, que para este caso fue: 192.168.1.254, y en el formulario se introdujo la contraseña del modem. Una vez que se accedió al router se modifica la opción NAT del panel, para poder cambiar la configuración de los puertos del modem. En NAT se accedió al muestreo de puertos para poder abrir uno nuevo que redireccionará al servidor VPN de la Raspberry, el puerto creado fue del tipo VPN cuya dirección es la misma que la de la Raspberry, el tipo de protocolo es UDP, este es el que se utiliza en la mayoría de los servicios de VPN, el puerto es el número 1194 que a su vez apuntará al puerto público 1194. Una vez habilitado el puerto se agregó a la configuración del router.
5. Posteriormente se procedió a instalar y configurar el servidor OPENVPN. Se estableció la dirección IP del servidor, así como dirección de la puerta de acceso del router. A continuación, se procedió a generar el servidor bajo la configuración establecida. Finalmente, se aceptaron las opciones siguientes para poder actualizar la seguridad del servidor.

3.4. Implementación de un servidor web local a través de un NodeMCU.

En la Figura 5, se muestra una captura del código del sketch principal programado en la IDLE de Arduino, como se puede observar está muy simplificado puesto que la mayoría de la programación se hizo en ficheros separados dependiendo de la acción que realizan y se incluyeron simplemente en la cabecera del programa principal. Mas adelante se mostrará el código de las dependencias del programa.

Las primeras siete dependencias incluidas son ficheros 'header' correspondientes a las librerías usadas. Entre estas librerías se encuentran la que posibilita la conexión 'Wi Fi', la que permite crear un servidor asíncrono con uso de 'web sockets' para la comunicación cliente/servidor, la que permite crear y gestionar archivos Json, y la gestión de ficheros tipo JavaScript y html de forma comprimida desde la memoria flash del dispositivo

NodeMCU en lugar de hacerlo por medio de la memoria dinámica cada vez que el cliente solicite hacer uso de estos

```
#include <ESP8266WiFi.h>
#include <ESPAsyncWebServer.h>
#include <FS.h>
#include <WebSocketsServer.h>
#include <ArduinoJson.h>
#include <SoftwareSerial.h>
SoftwareSerial P1(D3, D4);
#include "config.h"
#include "API.hpp"
#include "WebSockets.hpp"
#include "set_connection.hpp"
#include "server.hpp"
#include "set_AWS.hpp"

void setup(void)
{
    Serial.begin(115200);
    P1.begin(115200);
    SPIFFS.begin();

    Connect_STA(true);

    InitServer();
    InitWebSockets();
}

void loop(void)
{
}
```

Figura 5. Programa principal del NodeMCU

3.5. Implementación de web sockets con archivos Json en el servidor web local

Se usaron Web Sockets para el envío de datos cliente/servidor y servidor/cliente, esto con el fin de evitar el uso de formularios, el manejo de 'endpoints' y de recargar la página cada vez que se hiciera una petición que solo hacen lento el proceso. Para ello se usó la librería 'WebSocketsServer', y una vez que se incluyó ésta en el sketch principal, se creó la dependencia 'set_AWS' para establecer las configuraciones y funciones a usar para el manejo de web sockets en el servidor asíncrono.



Figura 6. Interfaz de la página web

3.6. Aplicación web

En la Figura 6, se muestra la interfaz de la aplicación web, la interfaz es muy sencilla pero práctica, en el lado izquierdo se tiene la opción de habilitar o deshabilitar la opción de poder variar la posición de los motores, esto quiere decir que en caso de que, por ejemplo, el 'switch' del eje Azimut se encuentre apagado, no se podrá modificar la orientación de este, aun cuando el 'slider' correspondiente en el lado derecho sea manipulado.

Es importante indicar que el 'slider' del eje Azimut tiene un rango de 0° a 160°, mientras que el eje Tilt tiene un rango de 55° a 125°. En ambos ejes, los 'sliders' cuentan con una posición por defecto de 80° para el Azimut y 90° para el Tilt, este mismo valor se establecerá en ellos, y por tanto en los ejes si se presiona el botón inferior con la leyenda 'Default'.

4. RESULTADOS

En la figura 7 se tiene el prototipo armado, tomando en cuenta que la parte primordial del prototipo es avalar que se puede utilizar una conexión a internet para controlar un RET a través de un servidor VPN.

Un parámetro importante que permite hacer la afirmación de si los objetivos del proyecto se cumplieron o no, es la latencia entre el cliente y el servidor. Es decir, el tiempo que transcurre entre que un usuario realiza una petición de ajuste de cualquiera de los ejes de orientación de la antena (Azimut y Tilt) a través de la interfaz de la página web, hasta que el servidor, o sea el Node MCU recibe la petición y la procesa



Figura 7. Sistema de posicionamiento armado

Uno de los objetivos que se busca es que el cambio o ajuste de los ejes de orientación sea efectuado en tiempo real, tanto desde la red local como desde la red externa. El hecho de que se requiera una respuesta en tiempo real no quiere decir que instantáneamente el NodeMCU pueda procesar la petición tan pronto como el usuario la solicitó, pero sí que el tiempo de duración de este proceso sea tan pequeño que llegue a ser lo suficientemente imperceptible para verlo como algo instantáneo.

El cliente interactúa con los elementos de la interfaz web, al hacerlo envía peticiones al servidor correspondientes con el elemento que esté manipulando. Al momento que el usuario realiza esto, se toma el tiempo en milisegundos transcurrido desde que éste cargó por primera vez la página. El servidor recibe la petición y devuelve al cliente un aviso de 'recibido', el backend detecta el mensaje del servidor, y guarda el tiempo en que sucedió este evento en la variable 'end'. Si ambas variables de tiempo (start y end) son enviadas al servidor y se imprimen en el monitor serial para conocer su valor, solo queda restar ambos tiempos y se tiene el tiempo de respuesta o latencia de la comunicación cliente-servidor con una precisión de ± 5 microsegundos.

Tabla 1. Pruebas de latencia para un cliente conectado en la red local

PRUEBA	LATENCIA (ms)
1	89.845
2	61.835
3	97.79

4	258.985
5	20.685
6	9.29
7	55.32
8	44.975
9	80.37
10	57.565
Promedio	77.366 ms

Tabla 2. Pruebas de latencia para un cliente conectado en la red externa

PRUEBA	LATENCIA (ms)
1	198.715
2	113.945
3	131.35
4	148.4
5	190.655
6	110.14
7	514.545
8	109.485
9	134.75
10	128.055
Promedio	178.004 ms

En la tabla 1 se muestran 10 pruebas realizadas con este método para un cliente conectado dentro de la red local, en la misma tabla se presenta el promedio del tiempo de latencia basado en las diez pruebas efectuadas. Correspondiente a la tabla 2, posee la misma estructura de información que la de la tabla 1, salvo que las pruebas se realizaron para un cliente conectado al servidor desde una red externa a través de la conexión VPN segura. Pruebas de latencia para un cliente conectado en la red local.

5. CONCLUSIONES

Los objetivos iniciales del proyecto se cumplieron de forma satisfactoria, debido a que mediante una conexión VPN fue posible enlazarse a la interfaz de control del sistema, representada por medio de un servidor web local implementado con la tarjeta NodeMCU desde una conexión remota, garantizando así el posicionamiento adecuado de la antena a distancia y de forma segura, pues al estar alojado el servidor en una red local, no corre el riesgo de sufrir ataques de terceros a lo largo de internet. Por medio de este sistema se garantiza que un operador, ubicado en un centro de operaciones pueda modificar la inclinación y el azimut de una antena de telecomunicaciones para aumentar la cobertura de red de dicha antena de forma remota y segura.

Al realizar el proyecto se pudo observar que las redes privadas virtuales (VPN) son una herramienta muy poderosa en la ingeniería pues permite extender una red local a través de internet, logrando que se pueda acceder a interfaces de monitores e instrumentación de forma remota, todo bajo un esquema de seguridad que es producto de una conexión punto a

punto entre un servidor y un cliente VPN. De esta manera una persona en otra parte del mundo puede observar el funcionamiento de un proceso y manipularlo de forma segura sin la posibilidad de la irrupción de terceros mal intencionados. Se puede aumentar la robustez del sistema propuesto en este proyecto, mediante la incorporación de actuadores de mayor precisión o bien utilizando un driver dedicado como el PCA9685, pues, aunque al inicio se planteó utilizar éste, por las condiciones en las que se realizó el proyecto, se optó por implementar una tarjeta Arduino Uno como driver para controlar los motores.

Otro aspecto que mejorar es el servidor web, que puede implementarse en las nuevas tarjetas ESP32, cuya velocidad de procesamiento permite reducir la latencia, permitiendo incluso tener más dispositivos conectados al mismo tiempo.

6. BIBLIOGRAFÍA

- [1] Instituto Nacional de Estadística y Geografía, «INEGI,» 22 junio 2021. [En línea]. Available: https://www.inegi.org.mx/contenidos/saladeprensa/boletines/2021/OtrTemEcon/ENDUTIH_2020.pdf. G.M. Cotty, junio 2021.
- [2] L. Wensheng, L. Yi y Z. Yanming, «The Design of RET Control System Based on AISG2.0,» IEEE, 2010.
- [3] G. A. E. Jiménez, «Proceso de implementación de una radiobase para la tecnología de LTE,» Universidad Nacional Autónoma de México, Ciudad de México, 2016.
- [4] J. M. Huidobro, «Antenas de telecomunicaciones,» Revista digital de ACTA, pp. 1 - 11, 2013.
- [5] L.-P. Shen, H. Wang, N. Hojjat, W. Lotz y H. Jamali, «Dual-Polarized Wideband Remote Electrical Tilt Multi-Beam Antennas» 2018 IEEE International Symposium on Antennas and Propagation & USNC/URSI National Radio Science Meeting, 13 Julio 2018
- [6] OpenVPN INC, «VPN Software Solutions & Services for Business» 2020. [En línea]. Available: <https://openvpn.net/>. [Último acceso: 25 Junio 2020].
- [7] NO-IP, «Dynamic DNS,» 25 Junio 2020. [En línea]. Available: <https://www.noip.com/remote-access>. [Último acceso: 26 Junio 2020].
- [8] A. A. A. Cancio y R. R. Rodríguez, «Optimización de la cobertura celular para las estaciones GSM Terrena, Matute, Club campestre de Claro Soluciones Móviles S.A.,» Universidad Tecnológica de Bolívar, Cartagena, 2015.