Implementación de una red de pruebas Wi-Fi para la asignación de recursos

Ignacio Mata[(1)], José Pulido[(1)], Sergio Fortes[(1)], Alfonso Fernández-Durán[(2)], Raquel Barco[(1)]
[(1)]Telecommunication Research Institute (TELMA), Universidad de Málaga E.T.S. Ingeniería de Telecomunicación, Bulevar Louis Pasteur 35, 29010 Málaga (Spain)
{josepa, sfr, ebm, sppulla, rbm}@ic.uma.es
[(2)]Nokia Spain, María Tubau 9, 28050 Madrid, Spain
alfonso.fernandez_duran@nokia.com

***Abstract*—This document presents the implementation of a Wi-Fi testbed network with multiple access points, facilitating the evaluation of various channel selection algorithms. Comprising multiple access points, a central controller, and a set of client devices, the network architecture mirrors a realistic scenario akin to an ISP managing several access points in densely populated urban areas. Leveraging MQTT and HTTP protocols for communication, alongside Docker for streamlined software deployment, the network allows for the assessment of channel selection algorithms, whether based on machine learning or heuristic rules.**

I. Introducción

La tecnología Wi-Fi basada en el estándar IEEE 802.11 ha sido ampliamente adoptada en los últimos años. Especialmente en entornos domésticos, proveyendo de conectividad a Internet a una amplia variedad de dispositivos, como teléfonos móviles, tabletas, ordenadores portátiles, televisores... [1]. Por tanto, actualmente cada hogar dispone de al menos un punto de acceso Wi-Fi, generando su propia WLAN (Wireless Local Area Network) [2].

La calidad de esta WLAN depende de varios factores, como la distancia entre el punto de acceso y el dispositivo, la presencia de obstáculos y sobre todo la interferencia de otras redes Wi-Fi. En entornos urbanos densamente poblados, la interferencia de otras redes Wi-Fi es un problema común que puede degradar significativamente la calidad de la conexión. [3]

Al tratarse de una banda no licenciada, los puntos de acceso (AP) Wi-Fi deben competir por el uso del espectro electromagnético, con técnicas como CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*) para evitar colisiones. Además, la selección adecuada de canales por parte de los puntos de acceso es crucial para minimizar las interferencias y optimizar el rendimiento de la red. Los puntos de acceso utilizan un proceso llamado "Selección de Canal Automática" (ACS) que se basa en la medición del uso de cada canal y en la elección del canal menos congestionado. Esta elección se realiza de forma independiente por cada punto de acceso, lo que puede llevar a una selección subóptima de canales y a una competencia innecesaria entre los puntos de acceso.

Una posible solución a este problema es la gestión centralizada de varios puntos de acceso pertenecientes al mismo ISP. [4] En este caso, los puntos de acceso enviarían reportes periódicos a un controlador central, en el que se haría el procesamiento de la información y se tomaría la decisión de selección de canal de cada punto de acceso. De esta forma, se podría evitar la competencia entre los puntos de acceso al tener la visión global de la red y se optimizaría la selección de canales para minimizar las interferencias.

El presente trabajo describe la implementación de una red de pruebas Wi-Fi con varios puntos de acceso, que permita evaluar diferentes algoritmos de selección de canal. La red de pruebas se compone de varios puntos de acceso, un controlador central y un conjunto de dispositivos cliente. Los puntos de acceso enviarán reportes periódicos al controlador, que tomará la decisión de selección de canal para cada punto de acceso. Los dispositivos cliente se conectarán a los puntos de acceso y se evaluará la calidad de la conexión en función de la selección de canal realizada por el controlador.

Así el trabajo se organiza como sigue: la Sección II presenta una visión general de la arquitectura de la red, la Sección III describe la implementación física de la red, la Sección IV detalla el desarrollo de la red Wi-Fi, la Sección V trata sobre la validación de la red y la Sección VI presenta las conclusiones y el trabajo futuro.

II. Arquitectura de la red

La arquitectura de la red se basa en el modelo de controlador-agente, es un patrón de diseño que divide una aplicación en dos componentes principales: el agente y el controlador.

- **Agente:** es el elemento gestionable de la red, en este caso, el punto de acceso. El agente es responsable de enviar reportes periódicos al controlador acerca del estado de la red. Además el agente expone al controlador una serie de parámetros que pueden ser leídos o escritos por el controlador así como una serie de comandos que pueden ser solicitados por el controlador.
- **Controlador:** es el elemento central de la red, que recibe los reportes de los agentes y toma decisiones en función de la información recibida. El controlador enviará la solicitud a los agentes para cambiar la configuración de la red.

Los agentes (puntos de acceso) serán los elementos gestionables de la red, que enviarán reportes periódicos al controlador acerca del estado de la red, incluyendo el *throughput*, el número de dispositivos conectados, la potencia de la señal de los clientes, etc. El controlador podrá solicitar información adicional a los agentes en cualquier momento, como el canal actualmente utilizado, un escaneo de los puntos de acceso vecinos, etc. En función de la información recibida, el controlador tomará la decisión utilizando un método de selección de

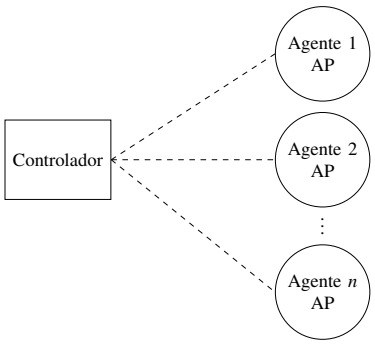

Fig. 1. Comunicación controlador-agente.

canal que no se abordará en este trabajo. El canal seleccionado se enviará a los agentes, que cambiarán su configuración para utilizar el nuevo canal.

Cada punto de acceso genera su propia WLAN con un SSID diferente y no tienen conexión directa entre ellos. Los puntos de acceso tampoco tendrán conocimiento de si los puntos de acceso cercanos, forman parte de la red de pruebas o no. Imitando por tanto un entorno realista en el que los puntos de acceso no tienen información sobre las redes vecinas. Es el controlador el que tiene la visión global de la red y puede tomar decisiones en función de la información recibida de los agentes.

A su vez, los dispositivos cliente se conectarán a los puntos de acceso y generarán tráfico de red, que será monitorizado por los agentes y reportado al controlador. En principio, los dispositivos cliente no enviarán información sobre la calidad de la conexión a nivel de aplicación, que puede ser un aspecto a considerar en futuras implementaciones.

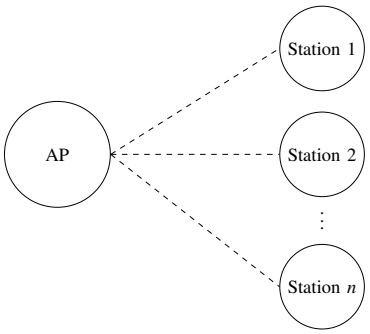

Fig. 2. Comunicación agente (AP)-Estación.

III. Implementación física

La implementación física de la red de pruebas se ha realizado en un entorno de laboratorio. Para la creación de los puntos de acceso se han utilizado dispositivos Radxa Rock 4C+, unos single-board computers similares a una Raspberry Pi. Estos dispositivos disponen de una interfaz Wi-Fi 802.11 b/g/n/ac (Wi-Fi 5) y un puerto Ethernet. Para la implementación del controlador se ha utilizado la misma placa para simplificar la implementación.

Los puntos de acceso cuentan con un puerto Ethernet para la conexión con el controlador y para proveer de conectividad a Internet a los dispositivos cliente. La conexión entre los puntos de acceso y el controlador se ha realizado mediante un switch Ethernet. La red de pruebas se ha conectado a un router que proporciona conectividad a Internet.

La antena interna de la placa no proporciona suficiente cobertura para un entorno realista, por lo que se ha instalado una antena de 2,4 GHz en cada punto de acceso. En futuras implementaciones se podría utilizar una antena de 5 GHz para poder estudiar la selección de canales en la banda de 5 GHz, que está menos congestionada que la banda de 2,4 GHz.

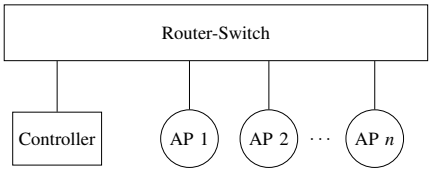

Fig. 3. Esquema de la implementación física.

A. *Capa de red: IP*

Al tratarse de una red de pruebas, esta se ha desplegado en un entorno de laboratorio de forma que solo se necesite una dirección IP de la red privada para el funcionamiento de la red. Sin embargo, también se puede desplegar utilizando diferentes direcciones IP de la red privada para cada uno de los elementos de la red, consiguiendo así una mayor flexibilidad y escalabilidad.

Para conseguirlo se ha utilizado un router-switch que proporciona conectividad a Internet a la red de pruebas, ejecutando NAT (*Network Address Translation*) creando así una red IP privada. A su vez, cada punto de acceso crea su propia red IP privada para los dispositivos cliente que se conectan a él. De esta forma se imita un entorno realista en el que los puntos de acceso no tienen visibilidad de las redes vecinas.

IV. Desarrollo de la red Wi-Fi

A. *Software*

Para la implementación de la red de pruebas se ha utilizado el sistema operativo Debian 11 tanto en los puntos de acceso como en el controlador. Se trata de una versión modificada por el fabricante de la placa Radxa Rock 4C+ para asegurar la compatibilidad con el hardware y la estabilidad del sistema.

Para la creación de los puntos de acceso se ha utilizado el software HostAPD (*Host Access Point Daemon*), que permite configurar la placa como un punto de acceso Wi-Fi. Este software se encarga de la gestión a nivel de enlace de la red Wi-Fi, incluyendo la autenticación de los dispositivos cliente, la gestión de la asociación de los dispositivos cliente a la red, el envío de beacons, etc.

Para gestionar la asignación de direcciones IP a los dispositivos cliente se ha utilizado DNSmasq, un software que implementa un pequeño servidor DHCP y un servidor DNS. DNSmasq se encarga de asignar direcciones IP a los dispositivos cliente. De esta forma se consigue una red Wi-Fi sencilla y funcional, como la que se puede encontrar en un entorno doméstico.

El software desarrollado para la comunicación entre los puntos de acceso y el controlador se ha implementado en Python, ya que es un lenguaje de programación de alto nivel que permite una rápida implementación y prototipado.

Además, Python cuenta con una amplia variedad de librerías que facilitan la implementación de aplicaciones de red, como Flask para la creación de una API REST de forma sencilla. De la misma forma, Python ofrece un acceso directo a herramientas de línea de comandos, como *iw*, que permite la configuración de la interfaz Wi-Fi de los puntos de acceso y la obtención de estadísticas de la red.

Al tratarse de una red con múltiples puntos de acceso, se necesita instalar el mismo software en varios dispositivos. Para facilitar la instalación del software en los puntos de acceso se ha utilizado Docker, un sistema de contenedores que permite empaquetar una aplicación y sus dependencias en un contenedor aislado. De esta forma, para desplegar un nuevo punto de acceso solo es necesario instalar Docker en el dispositivo y descargar la imagen del contenedor con el software preinstalado.

B. Implementación lógica y protocolos de comunicación

Existen diferentes tipos de comunicación entre los puntos de acceso y el controlador. Por un lado, está la recolección periódica de estadísticas de la red por parte de los puntos de acceso, que se envían al controlador. Por otro lado, el controlador puede solicitar información adicional de forma puntual a los puntos de acceso. Para implementar esta comunicación se han utilizado diferentes protocolos de red.

Para la recolección periódica de estadísticas se ha utilizado el protocolo MQTT (*Message Queuing Telemetry Transport*), un protocolo de mensajería ligero que se basa en el intercambio de mensajes entre un cliente y un servidor. En este caso, se ha instalado el *broker* (servidor) de MQTT en el controlador, utilizando el software Mosquitto. Los puntos de acceso actúan como clientes MQTT y envían mensajes al *broker* con las estadísticas de la red. El controlador se suscribe a los mensajes de los puntos de acceso y recibe las estadísticas de la red en tiempo real. De esta forma, el controlador tiene conocimiento de los puntos de acceso que forman parte de la red y su dirección IP. Cada punto de acceso debe conocer previamente la dirección IP del controlador para poder enviar los mensajes.

Para la comunicación puntual entre el controlador y los puntos de acceso se ha utilizado el protocolo HTTP (*Hypertext Transfer Protocol*). Cada punto de acceso expone una API REST que permite al controlador solicitar información adicional, como el canal actualmente utilizado, un escaneo de los puntos de acceso vecinos, etc. A su vez, esta API permite al controlador enviar comandos a los puntos de acceso, como la selección de canal, un cambio de SSID, etc.

A su vez, el controlador expone una API REST para poder gestionar la red al completo con una sola petición. Esta API permite al usuario obtener información sobre los puntos de acceso conectados a la red de forma unificada, así como enviar comandos a todos los puntos de acceso de forma simultánea. De esta forma, con una sola petición, el usuario puede cambiar todos los canales de los puntos de acceso sin tener que enviar la petición a cada uno de ellos de forma individual.

Para la definición de la estructura de datos que implementan los puntos de acceso, es decir, los parámetros Wi-Fi que están disponibles desde el punto de acceso para lectura o escritura, se ha utilizado la terminología del protocolo USP (*User Services Platform*). De esta forma se ha definido un conjunto de parámetros que son comunes a todos los puntos de acceso y que siguen la estructura de datos definida por el protocolo USP. De esta forma, se facilita la implementación de la comunicación entre los puntos de acceso y el controlador, ya que se utiliza una estructura de datos común y estandarizada.

C. Entorno de desarrollo

El desarrollo del software se ha realizado utilizando contenedores de desarrollo (*dev containers*). Esta tecnología permite definir un entorno de desarrollo a través de un archivo de configuración, que se ejecuta en un contenedor Docker. De esta forma, se garantiza que todos los desarrolladores trabajan en el mismo entorno, independientemente de su sistema operativo o de las herramientas que tengan instaladas en su máquina. Se utiliza el mismo software que tendrán instalado los puntos de acceso, lo que facilita la depuración y la resolución de problemas.

El despliegue del software en los puntos de acceso se ha automatizado haciendo uso de un servidor git (forgejo). Cada vez que se crea una nueva versión del código, se sube al servidor git y desde el controlador se puede ver qué versión del software está instalada en cada uno de los puntos de acceso. Además, se puede actualizar el software de forma remota desde el controlador. Este enviará una petición a los puntos de acceso para que descarguen la nueva versión del software y la instalen automáticamente. De esta forma, se facilita la gestión de los puntos de acceso y se garantiza que todos los puntos de acceso tienen la misma versión del software instalada, ya que controlarlo de forma manual sería inviable en una red de pruebas con múltiples puntos de acceso.

V. Validación de la red

A. Conexión con un algoritmo de selección de canal

Una vez la red de pruebas está desplegada, se podrá validar el rendimiento de un algoritmo de selección de canal. Este algoritmo puede estar basado en aprendizaje automático, en reglas heurísticas, o en cualquier otro enfoque. La red de pruebas permite evaluar el rendimiento de cualquier algoritmo. Para conectar este algoritmo a la red de pruebas, se utiliza la API REST que expone el controlador. De esta forma el algoritmo recibe la información de la visión global de la red ya procesada por el controlador y puede tomar decisiones en función de esta información. Una vez el proceso de optimización del algoritmo haya finalizado, enviará el resultado al controlador, que se encargará de enviar la nueva configuración a los puntos de acceso.

B. Interfaz web

Para facilitar la gestión de la red de pruebas se ha desarrollado una interfaz web que permite al usuario visualizar el estado de la red y realizar acciones sobre los puntos de acceso. La interfaz web se ha desarrollado utilizando el framework Flask, que permite la creación de aplicaciones web de forma sencilla en Python. La interfaz web muestra un resumen del estado de la red, incluyendo el número de puntos de acceso conectados, el número de dispositivos cliente, el canal actualmente utilizado por cada punto de acceso, etc. Además, permite al usuario enviar comandos a los puntos de acceso, como la selección de canal, un cambio de SSID, etc. De esta forma, el usuario puede gestionar la red de pruebas de forma sencilla y visual.

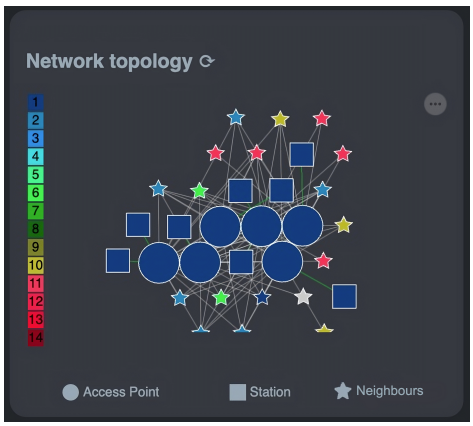

Fig. 4. Topología de la red en la interfaz web.

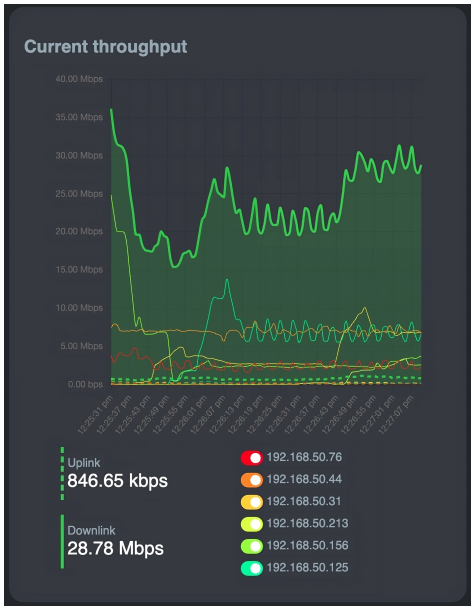

Fig. 5. Topología de la red en la interfaz web.

Como se puede ver en la Fig. 4, la interfaz web desarrollada muestra una gráfica representando la visión global de la red, incluyendo los puntos de acceso gestionables, los dispositivos cliente conectados a cada punto de acceso y los puntos de acceso no gestionables que se encuentran en el rango de cobertura de los puntos de acceso gestionables y pueden generar interferencias. Se han utilizado diferentes colores para representar los canales, proporcionando una visión clara para el usuario del estado de la red.

La Fig. 5 muestra el gráfico en tiempo real del tráfico de la red. Este gráfico permite al usuario visualizar la evolución de la tasa de transferencia de datos en función del tiempo y detectar posibles problemas de rendimiento en la red. Además, la gráfica permite filtrar por punto de acceso para facilitar la visualización del tráfico de cada punto de acceso de forma individual.

Desde la interfaz web también se puede comenzar el proceso de optimización del algoritmo seleccionado y ver el resultado de la optimización antes de aplicarlo a la red. En la Fig. 6 se muestra la interfaz web con las gráficas representando los puntos de acceso y los canales de cada uno, antes y después de la optimización.

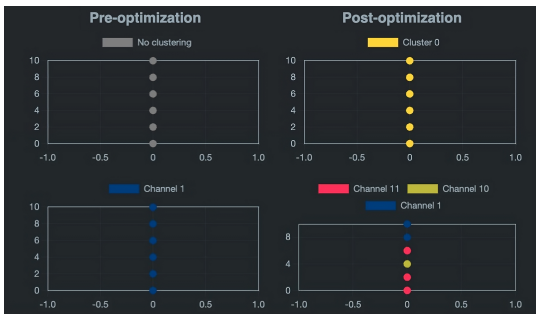

Fig. 6. Algoritmo de selección de canal en la interfaz web.

VI. Conclusiones

En este trabajo se ha presentado la implementación de una red de pruebas Wi-Fi con varios puntos de acceso, que permite evaluar diferentes algoritmos de selección de canal. La red de pruebas se compone de varios puntos de acceso, un controlador central y un conjunto de dispositivos cliente. La arquitectura de la red imita un entorno realista que podría ser el de un ISP que gestiona varios puntos de acceso en un entorno urbano densamente poblado.

En lo referente al trabajo futuro se plantea incluir a las estaciones como elementos gestionables de la red, de forma que el controlador disponga de la información de la calidad percibida por los dispositivos cliente a nivel de aplicación. De esta forma se podrían lanzar aplicaciones de prueba que generen tráfico de red de forma centralizada y evaluar la calidad de la conexión en función de la selección de canal realizada por el controlador.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el Centro para el Desarrollo Tecnológico Industrial (CDTI) del Ministerio de Ciencia e Innovación en el marco del proyecto MELODIC (Ref. IDI-20220551), el Ministerio de Asuntos Económicos y Transformación Digital y la Unión Europea - NextGenerationEU, en el marco del Plan de Recuperación, Transformación y Resiliencia y el Mecanismo de Recuperación y Resiliencia bajo el proyecto MAORI y la Junta de Andalucía a través de la Secretaría General de Universidades, Investigación y Tecnología con beca predoctoral (Ref. PREDOC 01712). También ha sido parcialmente financiado por la Universidad de Málaga a través del II Plan Propio de Investigación, Transferencia y Divulgación Científica.

Referencias

[1] B. Bellalta, "IEEE 802.11ax: High-efficiency WLANS," *IEEE Wireless Communications*, vol. 23, pp. 38–46, 2016.

[2] H. A. Omar, K. Abboud, N. Cheng, K. R. Malekshan, A. T. Gamage, and W. Zhuang, "A Survey on High Efficiency Wireless Local Area Networks: Next Generation WiFi," *IEEE Communications Surveys & Tutorials*, vol. 18, pp. 2315–2344, 2016.

[3] Z. Zhong, P. Kulkarni, F. Cao, Z. Fan, and S. Armour, "Issues and challenges in dense wifi networks," in *2015 International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2015, pp. 947–951.

[4] P. Gallo, K. Kosek-Szott, S. Szott, and I. Tinnirello, "Cadwan: A control architecture for dense wifi access networks," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 194–201, 2018.