Universidad de Los Andes
Facultad de Ingeniería
División de Estudios de Postgrado
Postgrado en Computación

Escaneo de Redes IEEE 802.11: Un enfoque práctico para la Internet del Futuro

Autor: Antonio Gregorio Araujo Brett
Tutor: Dr. Andrés Arcia Moret — University of Cambridge, UK

Trabajo de grado presentado ante la ilustre Universidad de Los Andes como requisito parcial para optar al grado de *Magíster Scientiae* en Computación.

Mérida, Marzo de 2016

VEREDICTO DEL TRABAJO DE GRADO

Los suscritos, Miembros del Jurado designado por el Consejo Técnico del Postgrado en Computación para conocer y evaluar el Trabajo de Grado *"Escaneo de Redes IEEE 802.11: Un enfoque práctico para la Internet del Futuro"*, realizado por el Ingeniero ***Antonio Gregorio Araujo Brett***, cédula de identidad N° ***V-12.778.889***, acuerdan, según lo establecido en el Reglamento del referido Postgrado, el siguiente veredicto:

Trabajo de Grado: APROBADO

Observaciones:

Prof. Andrés Emilio Arcia Moret
Tutor

Prof. Francisco J. Hidrobo T.
Jurado

Prof. Eric Gamess
Jurado

UNIVERSIDAD DE LOS ANDES FACULTAD DE INGENIERÍA Div. de Estudios de Postgrado POSTGRADO EN COMPUTACIÓN MÉRIDA - VENEZUELA

Mérida, 08 de Marzo de 2016

Agradecimiento

A Moe por ser luz en mi vida e impulsarme siempre. A Juan Pablo por hacerme padre.

Al Profesor Andrés Arcia Moret por sus orientaciones durante mi paso por la maestría y en especial por ser guía en este trabajo.

Al Profesor José Aguilar por su apoyo en el campo de la inteligencia computacional.

Al Profesor Laudin Molina por su apoyo en el desarrollo del emulador de escaneo de redes IEEE 802.11.

A la Ilustre Universidad de Los Andes por mantener sus puertas abiertas al conocimiento.

Al Networking for Development Laboratory (N4D) de la Universidad de Cambridge y al Dr. Arjuna Sathiaseelan por el apoyo incondicional a este trabajo.

Al Centro Nacional de Desarrollo e Investigación en Tecnologías Libres (CENDITEL) por permitirme estudiar mientras trabajaba.

Resumen

Las redes IEEE 802.11 constituyen actualmente una de las opciones más populares para tener acceso inalámbrico a la Internet. Debido a su auge, los dispositivos en despliegues cada vez más amplios, heterogéneos y caóticos tienen que cumplir con un proceso de descubrimiento y conexión complejo, probando activamente todos los canales en la banda ISM. Esto con el fin de obtener información de todos los puntos de acceso que le permitan posteriormente asociarse a una red.

El estándar IEEE 802.11 no especifica ningún método para escanear los canales en la búsqueda de redes alcanzables. En este sentido, este trabajo describe un enfoque práctico para el descubrimiento de redes IEEE 802.11 en el contexto de la Internet del futuro. En primer lugar, se presenta el diseño y construcción de un *sniffer* de redes IEEE 802.11 que puede escuchar, simultáneamente, a los 11 canales de la banda ISM. Con este *sniffer*, hacemos ingeniería inversa de los algoritmos de escaneo implementados en dispositivos móviles con interfaces IEEE 802.11. En segundo lugar, se discute a partir de una técnica de inteligencia computacional, un enfoque para encontrar la configuración de secuencias de escaneo óptimas en redes IEEE 802.11 espontáneas. Hemos abordado el problema usando la optimización multi-objetivo, es decir, el compromiso entre una tasa alta de descubrimiento de puntos de acceso y la duración del proceso de escaneo. El algoritmo cultural se implementó en código abierto y se apoya en un emulador de escaneo basado en una extensa campaña de mediciones experimentales. Los resultados de la aplicación del algoritmo cultural arrojaron secuencias de escaneo que proporcionan mejoras en el desempeño de la búsqueda entre un 230 % y 600 % con respecto a una gama de dispositivos tipificados a partir del *sniffer* multicanal. Finalmente, y de manera prospectiva, se presenta un bosquejo de un sistema de descubrimiento asistido e inteligente. A través de este sistema, se derivan secuencias de escaneo óptimas en redes de acceso co-

munitario, que permitan a los dispositivos inalámbricos encontrar los puntos de acceso disponibles y eventualmente, asociarse a estas redes de una manera eficiente.

Índice general

Índice de figuras

Índice de tablas

Índice de algoritmos

Capítulo 1

Introducción

Las redes inalámbricas IEEE 802.11 [1] constituyen una de las principales tecnologías de acceso local y móvil a la Internet. Diferentes clientes inalámbricos, que van desde sensores hasta computadores personales, requieren que las redes IEEE 802.11 brinden soporte a las necesidades de usuarios que demandan movilidad y mejor calidad de servicio.

En la actualidad, han aparecido iniciativas de redes que constituyen una alternativa a los despliegues de operadores de red tradicionales y que utilizan principalmente tecnologías inalámbricas debido a la gratuidad del espectro no licenciado como la banda de 2.4GHz. Entre este tipo de redes se encuentran las redes comunitarias [2] que crecen orgánicamente ya que se forman con la agregación de puntos de acceso (AP por sus siglas en inglés) que pertenecen a diferentes usuarios. Los despliegues de redes comunitarias pueden representar un ejemplo de cómo en un futuro cercano, la creciente población de APs en distintos entornos será tal, que se podrían utilizar para abordar el problema de desconexión de personas a la Internet. En situaciones como esta, es necesario que los mecanismos utilizados por los dispositivos inalámbricos sean eficientes para escanear su entorno y encontrar un AP disponible, y eventualmente asociarse a una red. Sin embargo, el proceso de escaneo configurado en dispositivos móviles o de escritorio no sigue un patrón estándar o principio de diseño que funcione correctamente en todos los escenarios como lo describen Arcia-Moret et al. [3].

En este trabajo se describe un enfoque práctico para el escaneo de redes IEEE 802.11. En este enfoque se incluye una caracterización de la dinámica del proceso de escaneo basada en el enfoque de la ingeniería inversa para extraer conocimiento de dispositivos que poseen interfaces inalámbricas IEEE

802.11, y se aprovechan las ventajas de la inteligencia computacional para derivar secuencias de escaneo optimizadas que permitan a dispositivos móviles en redes comunitarias conectarse de una manera eficiente. Particularmente, en este trabajo nos concentramos en el escaneo activo en redes IEEE 802.11, pues es el proceso que puede controlarse para obtener una eventual reducción de los tiempos de conexión a una red IEEE 802.11.

El resto de este trabajo se estructura como sigue. En el capítulo 2 se describen los conceptos que conforman el marco teórico así como una revisión del estado del arte alrededor del problema de estudio.

En el capítulo 3 se presenta el diseño y construcción de un *sniffer* multicanal que permite movilidad, opera bajo estándares abiertos y utiliza componentes de bajo costo. Este dispositivo permite escuchar todos los canales de la banda 2.4GHz e identificar las estrategias de escaneo que siguen distintos dispositivos a través de un proceso de ingeniería inversa.

En el capítulo 4 se presenta un algoritmo cultural para mejorar el descubrimiento de topologías de redes IEEE 802.11. El problema de encontrar secuencias de escaneo óptimas se modela a través de un problema de optimización multiobjetivo y se resuelve a través del algoritmo cultural.

En el capítulo 5 se describe la plataforma experimental y las secuencias de escaneo óptimas resultantes de la aplicación del algoritmo cultural.

En el capítulo 6 se presenta un bosquejo de un sistema de descubrimiento asistido que incorpora inteligencia computacional para encontrar secuencias eficientes a dispositivos que se encuentran en despliegues de redes inalámbricas como las redes de comunitarias. Finalmente, en el capítulo 7 se presentan las conclusiones de este trabajo y se vislumbran posibles trabajos futuros.

1.1. Planteamiento del problema

Las redes inalámbricas IEEE 802.11 son consideradas una de las tecnologías de acceso inalámbrico con mayor despliegue en la actualidad. Para el año 2014 se habían transportado más de 5.5 Petabytes de datos por mes a través de Wi-Fi sólo en Estados Unidos, de acuerdo al Cisco Visual Networking Index (VNI) [1]. Cada día se están incorporando dispositivos con una interfaz inalámbrica IEEE 802.11 a redes de sensores inalámbricos, en los dominios del paradigma de Internet de las Cosas (IoT por sus siglas en inglés)

[1] http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.html

y en las redes comunitarias.

A medida que se incorporan más sensores inalámbricos y proliferan los despliegues de redes comunitarias, los dispositivos con interfaces Wi-Fi se enfrentan al problema de agilizar sus mecanismos de descubrimiento y conexión a redes 802.11.

El proceso de descubrimiento o escaneo ha sido estudiado desde diversos enfoques. Molina y Arcia-Moret [4] destacan dos tendencias: la disminución de la duración del escaneo y la optimización del proceso de escaneo tomando en cuenta el impacto en la asociación a las redes de los usuarios móviles. Desde la modificación de variables establecidas en el estándar IEEE 802.11 presentada por Castignani et al. [5], hasta la incorporación de información adicional a las tramas del estándar presentada por Kim y Kim [6], las distintas propuestas consideran maneras de mejorar el algoritmo de escaneo de los dispositivos. La identificación de secuencias de escaneo permite comprender el comportamiento de distintos dispositivos Wi-Fi de uso masivo, los que hasta hoy, no muestran el código fuente de la implementación del escaneo. La mejora del proceso de escaneo podría, a su vez, representar directamente una mejora en el proceso de *handover* (el cual ocupa hasta un 90 % del tiempo de reconexión como lo reporta Mishra et al. [7]), pues los clientes en movimiento buscan constantemente reconectarse.

En un futuro cercano, un escenario posible incluiría un entorno en el cual muchos puntos de acceso estarán dispuestos para brindar acceso a redes abiertas y los dispositivos inalámbricos necesitarán mecanismos para encontrarlos de manera eficiente y asociarse a ellos. Algunos trabajos, como el propuesto por Arcia-Moret et al. [8], vislumbran el uso de un sistema centralizado que proporcione a los dispositivos inalámbricos la información del mejor punto de acceso para conectarse.

Es por ello que el presente trabajo aborda la problemática descrita desde tres enfoques principales:

1. Permitir la identificación de las estrategias de escaneo que utilizan distintos dispositivos que poseen una interfaz IEEE 802.11. Este enfoque facilita estudios posteriores sobre el escaneo de redes IEEE 802.11 al proporcionar un mecanismo móvil y de bajo costo para la identificación de secuencias de escaneo.

2. Encontrar secuencias de escaneo óptimas con el uso de una técnica de inteligencia computacional. Este enfoque proporciona un marco de

trabajo con base en la inteligencia computacional para la optimización del proceso de escaneo en redes IEEE 802.11. El marco de trabajo podría ser reutilizado y adaptado para nuevos modelos del problema de escaneo.

3. Proponer un sistema de descubrimiento asistido para redes IEEE 802.11. En este enfoque se realiza una prospectiva sobre un sistema que asista al proceso de descubrimiento en redes IEEE 802.11 y que adapta un algoritmo cultural como un componente inteligente para calcular secuencias de escaneo óptimas.

1.2. Aportes

Adicionalmente, las siguientes publicaciones basadas en esta investigación se presentaron a la comunidad:

- **A. Araujo** y A. Arcia-Moret, "Identificación de Secuencias de Scanning en Redes 802.11" en Memorias de 1era Conferencia Nacional de Computación, Informática y Sistemas, CONCISA, 2013. ISBN: 978-980-7602-03-7.
- **A. Araujo** y A. Arcia-Moret, "Identificación de Secuencias de Scanning en Redes 802.11" en Revista Venezolana de Computación, volumen 1, número 1, 2014, ISSN: 2244-7040.
- A. Arcia-Moret, A. Sathiaseelan, **A. Araujo**, J. Aguilar, L. Molina, "Assisted Network Discovery for Next Generation Wireless Networks", aceptado como poster en 13th Anual IEEE Consumer Communications & Networking Conference, IEEE CCNC 2016, Las Vegas, Enero, 2016.
- A. Arcia-Moret, **A. Araujo**, J. Aguilar, L. Molina, A. Sathiaseelan, "Intelligent Network Discovery for Next Generation Community Wireless Networks", 12th Wireless On-demand Network Systems and Services Conference, IEEE WONS 2016, Cortina d'Ampezzo, Enero, 2016.

Capítulo 2

Marco teórico

2.1. Redes Comunitarias

En diferentes partes del mundo se han construido redes de gran escala como alternativas a las redes de los tradicionales proveedores de servicios de Internet (ISP por sus siglas en inglés) que requieren una suscripción especial para tener acceso a la Internet. Estas redes, denominadas redes comunitarias (CN por sus siglas en inglés), se encuentran en zonas tanto urbanas como rurales y pretenden brindar acceso universal y planes de servicio para integrar usuarios a la Internet; todo esto a través de tecnologías inalámbricas debido al bajo costo de utilizar el espectro no licenciado [2].

Las redes comunitarias suelen ser de gran escala, distribuidas y autogestionadas ya que están construidas y organizadas de una manera abierta y descentralizada. Una de las maneras en que las redes comunitarias se crean y crecen es gracias al despliegue espontáneo de nodos que pertenecen a diferentes usuarios.

En las redes comunitarias sus integrantes mantienen la propiedad de lo que hayan decidido compartir. Se sigue un modelo participativo que ha mostrado ser efectivo en conectar a personas geográficamente dispersas, extendiendo los derechos digitales de Internet.

En las redes comunitarias, el ISP entrega a los usuarios un dispositivo que contiene un modem de línea de abonado digital asimétrica (ADSL por sus siglas en inglés), un enrutador y un punto de acceso inalámbrico IEEE 802.11. Este dispositivo proporciona típicamente acceso a la Internet a través del proceso de traducción de direcciones de red (NAT por sus siglas en inglés).

Estos puntos de acceso son capaces de enviar múltiples identificadores de red, permitiendo a los usuarios compartir su acceso ADSL con otros clientes del mismo ISP usando un sistema de autenticación abierto de acceso a red, específicamente el identificador SSID de la red comunitaria que se describirá en la sección 2.2.2. En la Fig. 2.1 se muestra un bosquejo de un despliegue común de una red comunitaria, donde diferentes ISP proporcionan acceso a la red comunitaria a sus suscriptores, quienes también configuran un identificador para su red privada.

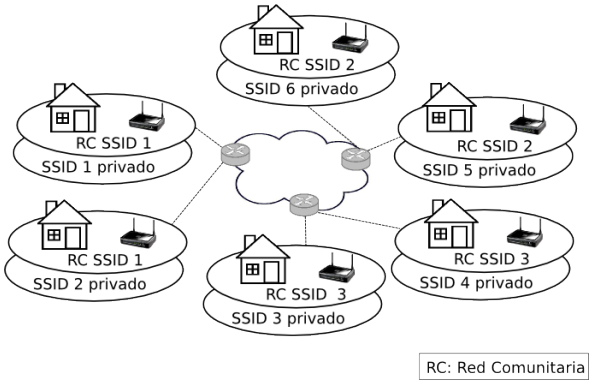


Fig. 2.1: Despliegue común de red comunitaria

Algunos ejemplos de redes comunitarias se encuentran en diversas partes del mundo como por ejemplo: Guifi.net[1] en España, la red metropolitana inalámbrica de Atenas (AWMN[2] por sus siglas en inglés) en Grecia, FunkFeuer[3] en Austria y Melbourne Wireless[4] en Australia, WasabiNet[5] en Estados Unidos, entre otras.

2.2. Redes IEEE 802.11

Los estándares inalámbricos constituyen la base para muchos productos inalámbricos, asegurando la interoperabilidad por aquellos que diseñan, despliegan y administran redes inalámbricas [9].

[1] http://guifi.net
[2] https://awmn.net
[3] http://www.funkfeuer.at
[4] http://melbourne.wireless.org.au
[5] http://gowasabi.net/

IEEE 802.11 [1] corresponde al estándar para redes inalámbricas de área local (WLAN por sus siglas en inglés) lanzado en 1997 para proporcionar conectividad inalámbrica de alta velocidad en la banda de frecuencia industrial, científica y médica (ISM por sus siglas en inglés). Desde su lanzamiento IEEE 802.11 ha tenido un conjunto de enmiendas a su versión original que definen distintos protocolos para incorporar mejoras en la comunicación inalámbrica. En la tabla 2.1 se presentan los protocolos con sus frecuencias y rangos aproximados de cobertura.

Tabla 2.1: Enmiendas de 802.11 con sus frecuencias y rangos aproximados.

Protocolo 802.11	Liberación	Frecuencia (GHz)	Ancho de banda (MHz)	Velocidad (Mbit/s)	Rango en interior (m)	Rango en exterior (m)
-	Jun 1997	2.4	20	1, 2	20	100
a	Sep 1999	5	20	6, 9, 12, 18, 24, 36, 48, 54	35	120
b	Sep 1999	2.4	20	1, 2, 5.5, 11	35	140
g	Jun 2003	2.4	20	6, 9, 12, 18, 24, 36, 48, 54	38	140
n	Oct 2009	2.4/5	20	1, 2, 5.5, 11	70	250
ac	Nov 2011	5	20	1, 2, 5.5, 11		

Las redes IEEE 802.11 están concentradas en las dos capas inferiores del modelo de referencia de Interconexión de Sistemas Abiertos (OSI por sus siglas en inglés) ya que incorpora tanto componentes físicos como de enlace de datos. En la Fig. 2.2 se muestra la relación entre varios componentes de la familia IEEE 802 y su lugar en el modelo OSI.

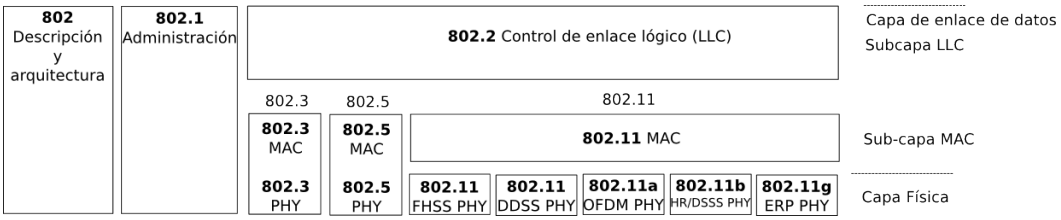


Fig. 2.2: IEEE 802 y su relación con el modelo OSI

2.2.1. Componentes de una red IEEE 802.11

Las redes IEEE 802.11 consisten de 4 componentes físicos principales que se muestran en la Fig. 2.3 y que se describen a continuación.

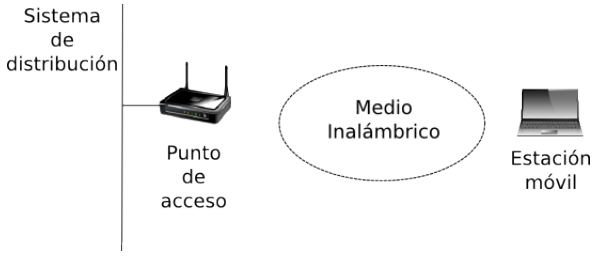


Fig. 2.3: Componentes principales de una red 802.11

- Estación móvil: (MS por sus siglas en inglés) es un dispositivo de computación que posee una o varias interfaces de red inalámbrica. Ejemplos comunes de estaciones son teléfonos inteligentes, tabletas, laptops, consola de video juegos, computadores de escritorio con interfaz inalámbrica, etc.
- Punto de Acceso: conocido como AP por sus siglas en inglés, es un dispositivo de red que interconecta equipos de comunicación para formar una red.
- Medio inalámbrico: es el medio físico a través del cual se intercambian tramas entre estaciones que se comunican.
- Sistema de distribución: es el componente lógico de 802.11 utilizado para enviar tramas a sus destinos. Se utiliza cuando se disponen varios puntos de acceso conectados para formar un área de cobertura amplia.

2.2.2. Despliegue de una red IEEE 802.11

El bloque principal de construcción de una red IEEE 802.11 está definido por el Conjunto de Servicio Básico (BSS por sus siglas en inglés) que consiste en un grupo de estaciones móviles que se comunican entre sí. El área donde se dan las comunicaciones es conocida como área de servicio básico y está condicionada por las características del medio inalámbrico.

Para comunicarse dentro de un BSS existen dos enfoques distintos: el modo Conjunto de Servicio Básico Independiente (IBSS por sus siglas en inglés) y el modo Infraestructura. Estos modos se ilustran en la Fig. 2.4 a) y b) respectivamente.

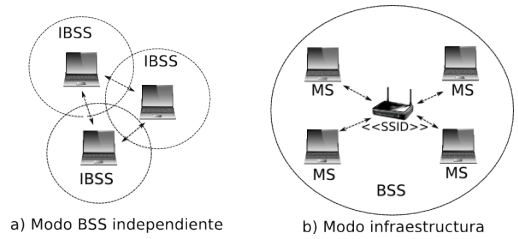


Fig. 2.4: Modos de comunicación dentro de un BSS

Modo BSS independiente

En este modo las estaciones se comunican directamente con cada una mientras están dentro del área de cobertura común. Generalmente se designa como modo *adhoc* debido a que este tipo de redes están diseñadas para propósitos específicos y tienen un uso esporádico. Un ejemplo de esto se encuentra cuando un una estación móvil tiene acceso a la Internet y otras estaciones móviles en la red sin conexión a la Internet pueden utilizar a la primera como un punto para alcanzar la red pública.

Modo Infraestructura

En este modo el AP funge como puente entre las estaciones móviles conectadas a través del medio inalámbrico con otros hosts conectados a un enlace Ethernet cableado llamado sistema de distribución. Este tipo de arquitectura proporciona varias ventajas. En primer lugar, permite extender la cobertura de una red cableada a un número mayor de estaciones móviles conectadas a diferentes AP. Para lograr una mayor área de cobertura, varios BSS se pueden conectar utilizando un *backbone* de red para formar un Conjunto de Servicio Extendido (ESS por sus siglas en inglés) como se muestra en la Fig. 2.5. Una de las principales características de un ESS es que permite que una estación móvil pueda enviar o recibir tramas hacia o desde otra estación móvil aunque

pertenezca a diferentes BSS. En un ESS todos los BSS se identifican con un Identificador de Conjunto de Servicio (SSID) común.

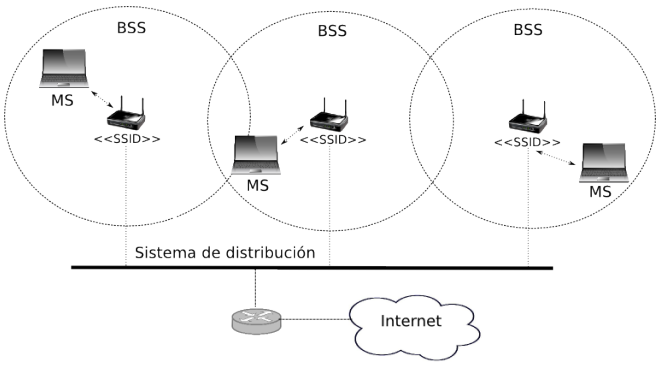


Fig. 2.5: Conjunto de servicio extendido

2.2.3. Soporte de movilidad IEEE 802.11

Uno de los principales objetivos de los despliegues de redes IEEE 802.11 es la movilidad. El estándar IEEE 802.11 proporciona movilidad entre áreas de servicio básico en la capa de enlace de datos. Si una estación móvil no se mueve o su movimiento ocurre dentro del área de servicio de su punto de acceso actual no ocurre ninguna transición. Existen dos tipos de transiciones entre puntos de acceso:

1. Transición de BSS: Esta transición requiere la cooperación entre puntos de acceso, al intercambiar información sobre la estación móvil que está en movimiento. La transición de BSS es iniciada por la estación e inicia el proceso de re-asociación con el nuevo punto de acceso. En la Fig. 2.6 se ilustra la transición de BSS. La estación móvil MS 1 se mueve en el tiempo t_1 del BSS 1 y llega en el tiempo t_2 al BSS 2.

2. Transición de ESS: Esta transición se refiere al movimiento de una estación móvil de un ESS a un segundo ESS. IEEE 802.11 no soporta este tipo de transición, excepto para permitir que la estación móvil se asocie con un punto de acceso en el segundo ESS una vez que ha dejado el primero. En la Fig. 2.7 se ilustra la transición de ESS. La estación

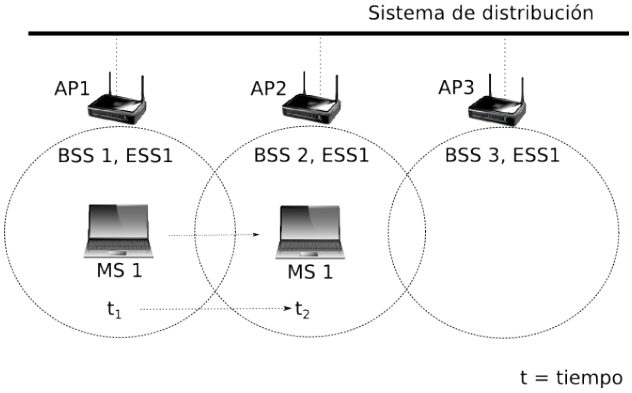


Fig. 2.6: Transición de BSS

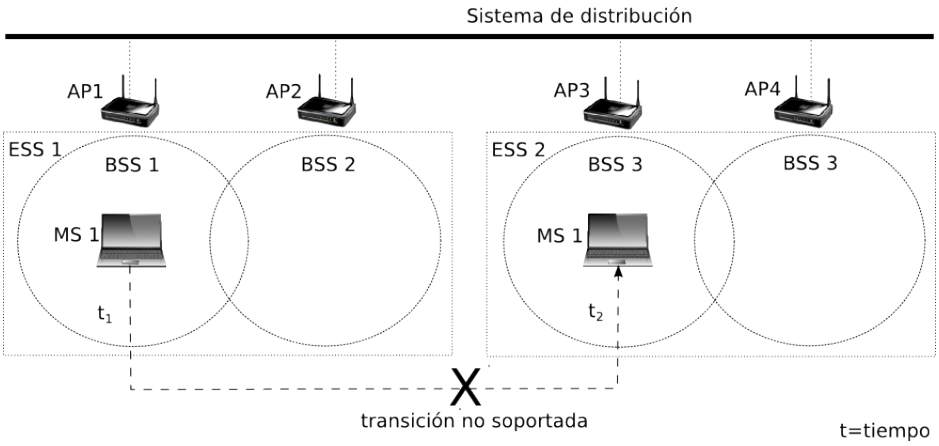


Fig. 2.7: Transición de ESS

móvil MS 1 se mueve en el tiempo t_1 del ESS 1 y llega en el tiempo t_2 al ESS 2.

Una situación común en las redes IEEE 802.11 ocurre cuando una estación móvil se acerca hacia los límites de la región de cobertura de un punto de acceso y entra a la región de otro; en este caso la estación móvil debe hacer una transición hacia el nuevo punto de acceso. El proceso que permite la transición se llama *handover*. Una vez que se realiza el *handover* la estación móvil envía las tramas hacia el nuevo punto de acceso para que este las dirija a su destino final.

El proceso de *handover* puede ser dividido en dos pasos lógicos distintos:

1. Descubrimiento: la estación móvil envía tramas de administración *Probe Request* para encontrar los puntos de acceso disponibles en su entorno. Este paso es conocido como escaneo y de acuerdo al estándar IEEE 802.11 [1] puede ser activo o pasivo. De acuerdo a Mishra et al. [7] el proceso de escaneo consume más del 90 % de la duración del *handover*. El proceso de escaneo se describirá con mayor detalle en § 2.3

2. Reautenticación: este paso involucra una reautenticación para que la estación móvil muestre su identidad antes de transmitir tramas a la red y una reasociación con el nuevo punto de acceso.

El proceso de *handover* inicia con la primera trama *Probe Request* de la estación móvil para escanear las redes disponibles y termina con la trama de respuesta de asociación del nuevo punto de acceso. Las tramas utilizadas en el proceso de *handover* se describirán con detalle en § 2.2.6.

2.2.4. Control de acceso al medio

El control de acceso al medio es un mecanismo que permite que varios estaciones móviles o nodos se comuniquen en una red que comparte un medio. En las redes IEEE 802.11 las estaciones móviles que intentan comunicarse comparten un medio inalámbrico no guiado en el cual las transmisiones son enviadas en forma de *broadcast*; esto implica que son escuchadas por todas las estaciones en el medio compartido. Un evento común en este entorno es la ocurrencia de colisiones entre transmisiones de estaciones móviles en un mismo instante de tiempo. El objetivo principal del control de acceso al medio es maximizar la utilización del medio y evitar que las estaciones móviles transmitan simultáneamente.

El control de acceso al medio fundamental de IEEE 802.11 es una función de coordinación distribuida (DCF por sus siglas en inglés) conocida como acceso múltiple con detección de portadora y evasión de colision (CSMA/CA por sus siglas en inglés) [1]. La DCF deberá estar implementada en todas las estaciones móviles con una interfaz IEEE 802.11.

Para que una estación móvil transmita datos, ésta debe detectar si el medio está ocupado por una transmisión de otra estación móvil. Si el medio no está ocupado, la transmisión se puede realizar. El algoritmo distribuido

CSMA/CA obliga que exista una cantidad de tiempo mínima entre secuencias de trama contiguas [1]. Una estación móvil verificará que el medio se encuentre desocupado por la cantidad de tiempo requerido antes de intentar transmitir. Si el medio está ocupado, la estación esperará hasta el final de la transmisión actual. Luego de la espera, o antes de intentar transmitir de nuevo inmediatamente después de una transmisión exitosa, la estación móvil seleccionará un intervalo de tiempo aleatorio denominado *backoff* y decrementará el contador de intervalo de *backoff* mientras el medio está desocupado. Una transmisión es exitosa cuando una trama de acuse de recibo positivo (ACK por sus siglas en inglés) se recibe desde la estación móvil o cuando una trama es transmitida completamente.

Existe una refinación del CSMA/CA que puede ser utilizada bajo ciertas circunstancias para minimizar las colisiones. Aquí las estaciones móviles que transmiten y reciben intercambian tramas de control pequeñas (RTS y CTS) después de determinar que el medio no está ocupado y luego de cualquier *backoff*, antes de transmitir datos.

Las tramas empleadas en el CSMA/CA y en las demás transmisiones de las estaciones móviles se describen en § 2.2.6.

2.2.5. Tramas de IEEE 802.11

El estándar IEEE 802.11 define tres tipos de tramas principales: datos, control y administración. Cada una de estas tramas definen a su vez subtipos de acuerdo a las funciones que cumple cada una. En la Fig. 2.8 se muestra una trama genérica de control de acceso al medio (MAC) de IEEE 802.11.

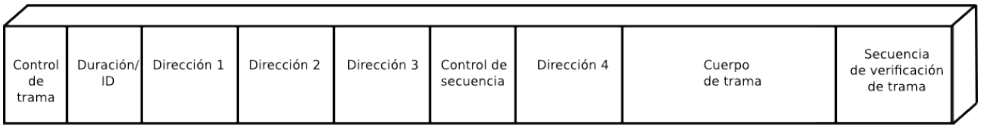


Fig. 2.8: Trama genérica de IEEE 802.11

Entre los campos de la trama genérica se encuentran:

- Control de trama: mantiene información de control utilizada para definir el tipo y subtipo de trama así como proporcionar información necesaria de los siguientes campos para procesarla.

- Duración/ID: se usa en tramas de control para indicar el tiempo restante necesario para recibir la próxima transmisión de trama.
- Dirección: una trama 802.11 puede contener hasta 4 campos de dirección. Estos campos son enumerados ya que son utilizados para diferentes propósitos de acuerdo al tipo de trama. En general la dirección 1 se usa para la estación destino y la dirección 2 para la estación origen.
- Control de secuencia: mantiene información del número de secuencia de cada trama y del número de cada trama enviada en el caso de ser una trama fragmentada.
- Cuerpo de trama: también se conoce como campo de datos y sirve para mover datos de capas superiores de una estación móvil a otra. Este campo depende del tipo de trama y subtipo (ver § 2.2.6).
- Secuencia de verificación de trama: este campo permite que una estación móvil verifique la integridad de las tramas recibidas; generalmente se define como chequeo de redundacia cíclica (CRC por sus siglas en inglés) debido a las operaciones matemáticas que se realizan para la verificación.

2.2.6. Tipos de tramas de IEEE 802.11

Tramas de datos

Estas tramas son las que mueven los datos de una estación móvil a otra.

Tramas de control

Estas tramas se utilizan en conjunto con las tramas de datos para las tareas de operación de limpieza de área, adquisición de canal, mantenimiento de prueba de portadora, el acuse de recibo positivo de datos recibido, entre otras. Entre las tramas de control se encuentran:

- RTS (*Request to send*): utilizadas para obtener control del medio para la transmisión de tramas.
- CTS (*Clear to send*): responden a una trama RTS y son utilizadas para reducir las colisiones entre tramas.

- ACK (*Acknowledgement*): utilizadas para enviar un acuse de recibo positivo requerido por el control de acceso al medio al recibir una trama de datos exitosamente.

Tramas de administración

Estas tramas permiten realizar funciones de administración del mantenimiento de la comunicación como descubrimiento de redes, asociación, autenticación, movilidad entre puntos de acceso, entre otras. Entre las tramas de administración se encuentran las siguientes:

- *Beacon*: anuncian la existencia de una red y constituyen una parte importante de muchas tareas de mantenimiento de la red. Se transmiten a intervalos regulares para permitir que una estación móvil pueda encontrar e identificar una red. En el modo infraestructura el punto de acceso es el responsable de transmitir las tramas *Beacon* dentro del área de servicio básico.

- *Probe Request*: En la Fig. 2.9 se muestran los campos de una trama *Probe Request*. Son utilizadas por las estaciones móviles para realizar un escaneo de un área y verificar si existen redes 802.11. Un *Probe Request* incluye en el cuerpo de la trama campos que identifican el SSID de la red y las velocidades de transmisión soportadas por la estación móvil.

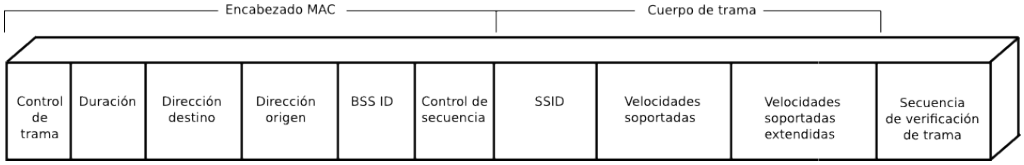


Fig. 2.9: Trama *Probe Request*

- *Probe Response*: En la Fig. 2.10 se muestran los campos de una trama *Probe Response*. Constituye la respuesta a una trama *Probe Request* recibida en una red con parámetros compatibles. En una red en modo infraestructura quien emite las tramas *Probe Response* es un punto de acceso. En esta trama se incluye información suficiente para que una estación móvil ajuste parámetros y pueda unirse a una red.

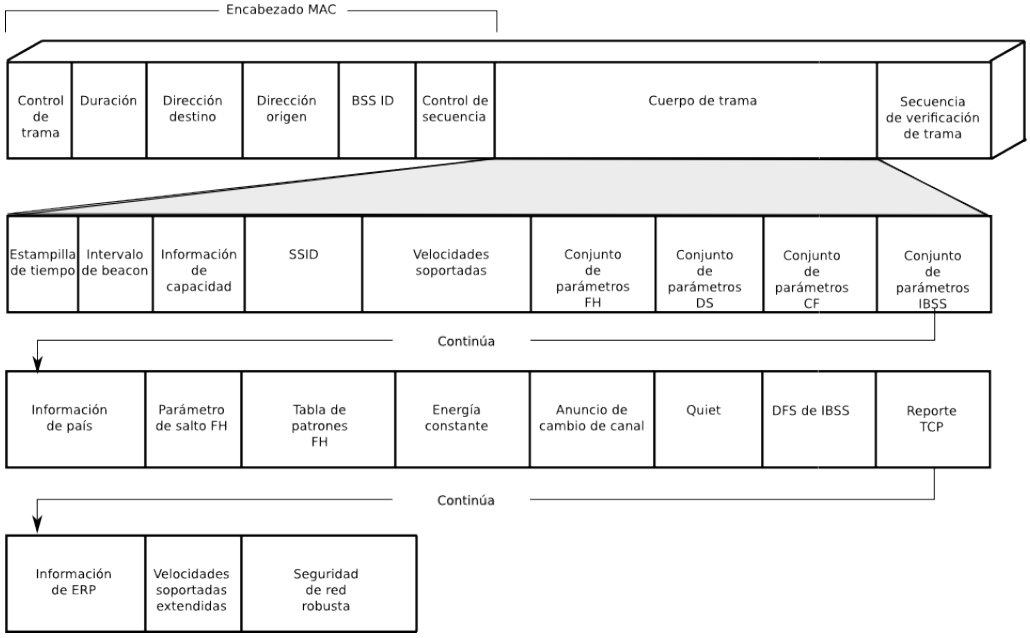


Fig. 2.10: Trama *Probe Response*

2.2.7. Proceso de conexión a una red IEEE 802.11

El estándar IEEE 802.11 propone los siguientes estados de conexión para una estación móvil [1]:

- no autenticada, no asociada: en este caso la estación móvil no puede transmitir porque no pertenece al BSS.
- autenticada pero no asociada: en este caso la estación móvil se autenticó a través de algún mecanismo soportado por el estándar IEEE 802.11. Entre los mecanismos de autenticación soportados están autenticación de sistema abierto (OSA por sus siglas en inglés), autenticación de clave compartida (SKA por sus siglas en inglés), autenticación de transición rápida (FTA por sus siglas en inglés) y autenticación simultánea de iguales (SAE por sus siglas en inglés); sin embargo el estándar también permite la definición de nuevos métodos de autenticación.
- autenticada y asociada: en este caso la estación móvil ha finalizado el

acuerdo de operación con un punto de acceso que incluye configuraciones para velocidades de transmisión y seguridad. En este estado la estación móvil ya puede enviar tramas de datos al punto de acceso.

En la Fig. 2.11 se muestra un bosquejo del proceso de conexión de una estación móvil a una red IEEE 802.11 con el mecanismo de autenticación OSA [10]. En el paso 1 la estación móvil envía tramas *Probe Request* para descubrir las redes que se encuentran cerca. En el paso 2 los puntos de acceso que reciben la trama *Probe Request* verifican que la estación soporte una velocidad de transferencia común y envían una trama *Probe Response* con el SSID, velocidades soportadas y otras capacidades del punto de acceso. Luego de recibir las tramas *Probe Response* la estación móvil escoge una red compatible. En el paso 3 la estación móvil envía una trama de autenticación al punto de acceso con el número de secuencia 1. El punto de acceso recibe la trama de autenticación y responde con otra trama cuyo número de secuencia es 2 como se muestra en el paso 4. Una vez autenticada la estación móvil, ésta envía una trama de solicitud de asociación al punto de acceso como se muestra en el paso 5. Si la solicitud de asociación incluye información que coincide con las capacidades del punto de acceso, éste creará una trama de respuesta de asociación con un identificador de asociación y la enviará a la estación móvil como se muestra en el paso 6. Luego de estos pasos la estación móvil se encuentra conectada a la red y puede iniciar la transmisión de datos.

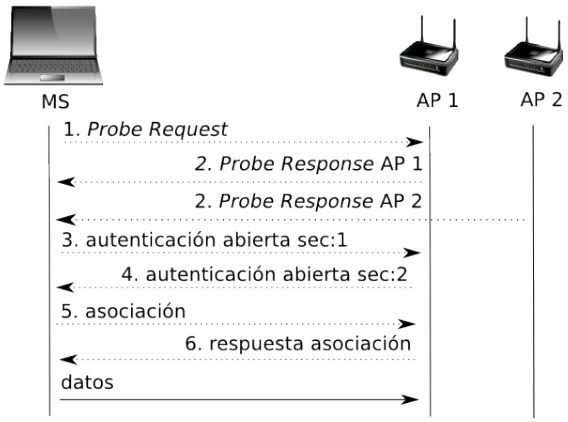


Fig. 2.11: Proceso de conexión a una red IEEE 802.11

2.3. Proceso de descubrimiento en redes IEEE 802.11

El proceso de descubrimiento o escaneo es una función del estándar IEEE 802.11 en la que las estaciones móviles buscan puntos de acceso disponibles para luego asociarse a las redes [1]. El proceso de escaneo se realiza sobre los canales de la banda 2.4GHz, compuesta por 11 canales en América y por 13 canales en Japón y en el resto del mundo. Aunque el escaneo tiene como objetivo encontrar las redes disponibles es un procedimiento costoso en términos de número de tramas *Beacons* y consumo de energía en la estación móvil como lo menciona Mishra et al. [7].

El estándar IEEE 802.11 [1] define dos temporizadores para ajustar el proceso de escaneo, específicamente el *MinChannelTime* (MinCT) y el *MaxChannelTime* (MaxCT) que determinan el tiempo que una estación móvil tiene que esperar en un canal particular. El estándar también define dos tipos de escaneo para encontrar puntos de acceso disponibles: pasivo y activo.

- Escaneo pasivo: la estación móvil escucha en cada canal las tramas *Beacons* emitidas por los puntos de acceso cercanos por una duración máxima establecida por el temporizador MaxCT.
- Escaneo activo: en este tipo de escaneo la estación móvil toma un rol más dinámico, ya que en lugar de esperar que los puntos de acceso se anuncien a si mismos, ésta intenta encontrarlos.

En este trabajo nos concentramos en el escaneo activo, pues de los dos procesos, es el que puede controlarse para obtener una eventual reducción de los tiempos de conexión a una red IEEE 802.11.

En la Fig. 2.12 se muestra un bosquejo del escaneo activo de IEEE 802.11 que recorre cada uno de los canales de la banda 2.4 GHz.

Un resumen del escaneo activo haciendo énfasis en los eventos de interés se muestra en el algoritmo 1. En este algoritmo, *ProbeDelay* corresponde al retardo esperado antes de enviar una trama *Probe Request* en un nuevo canal, MinCT corresponde a la mínima cantidad de tiempo a esperar en cada canal y MaxCT corresponde a la máxima cantidad de tiempo a esperar en un canal. El orden o secuencia en el cual se prueban los canales no está definido por el estándar IEEE 802.11; es por ello que los fabricantes de dispositivos inalámbricos implementan distintas estrategias de prueba de los canales. Una

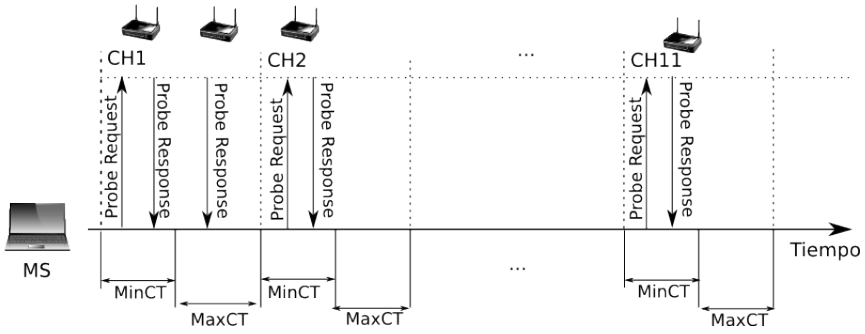


Fig. 2.12: Escaneo activo de IEEE 802.11

secuencia de prueba de todos los canales del espectro en un orden particular se conoce como escaneo completo.

Algoritmo 1: Algoritmo de escaneo activo

Datos: Lista de canales
Resultado: Lista de *Probe Responses*

```
1  Para Cada canal hacer
2      Esperar hasta que el tiempo ProbeDelay expire;
3      Enviar una trama Probe Request con destino broadcast;
4      Iniciar un temporizador ProbeTimer;
5      si El medio no está ocupado cuando ProbeTimer alcance MinCT
       entonces
6          Escanear el siguiente canal;
7      en otro caso
8          Cuando ProbeTimer alcance MaxCT, procesar todas las
           tramas Probe Response recibidas y escanear el siguiente canal;
9      fin
10 fin
```

El proceso de escaneo activo para encontrar puntos de acceso disponibles tiene una duración máxima que se puede denominar latencia o retardo de escaneo completo (L) y que se encuentra acotada por la expresión 2.1 como sugieren Mishra et al. [7] y Castignani et al. [5], donde N_{ch} corresponde al número de canales escaneados y MinCT $\leq$ MaxCT:

$$N_{ch} * MinCT \leq L \leq N_{ch} * MaxCT \tag{2.1}$$

2.3.1. Enfoques aplicados al descubrimiento en redes IEEE 802.11

En el proceso de descubrimiento en redes IEEE 802.11 se destacan algunas tendencias que agrupan varias perspectivas de estudio [4]:

1. Dismunición de la duración del escaneo: tendencia en la que se agrupan trabajos que proponen una optimización de la configuración del proceso de escaneo a través de las variables involucradas. Como ejemplo, Castignani et al. [5] proponen el ajuste dinámico de los valores de MinCT y MaxCT durante el proceso de escaneo para reducir el tiempo que se dedica a la revisión de cada canal. Velayos y Karlsson [11] proponen la reducción de la fase de búsqueda de puntos de acceso a través de consideraciones teóricas y simulaciones que establecen un ajuste de MinCT y MaxCT.

2. Impacto del escaneo en la asociación de redes: tendencia en la que se agrupan trabajos que optimizan el proceso de escaneo tomando en cuenta el impacto que éste tiene en la asociación de estaciones móviles durante el *handover*. Como ejemplo, Montavont et al. [12] proponen realizar el escaneo de forma periódica, esto es agrupando canales del espectro en subgrupos de pocos canales, con cada fase del *handover* revisando un canal durante exactamente MinCT. El objetivo de esta estrategia es encontrar puntos de acceso antes de iniciar el *handover* y mientras la estación móvil se mantiene conectada. Eriksson et al. [13] proponen utilizar la probabilidad de que un punto de acceso se encuentre en un canal particular para determinar la secuencia en la que se deben revisar las canales del espectro.

Otros trabajos que abordan el proceso de escaneo consideran nuevos problemas como el consumo de energía y la localización.

Para ambientes de hogar y trabajo un escaneo agresivo puede mejorar significativamente la velocidad con la que estaciones móviles se pueden unir a redes IEEE 802.11. Sin embargo, ese mismo comportamiento agresivo puede incurrir en efectos considerables a lo largo de ambientes inalámbricos densamente poblados. En este sentido, Hu et al. [14] muestran, a través de estudios

empíricos a escala amplia (estadio deportivo) y pequeña (laboratorio), cómo un escaneo agresivo tiene implicaciones significativas para la energía y el *throughput*.

Otro ámbito en el que se estudia el proceso de escaneo en redes IEEE 802.11 es la localización basada en Wi-Fi como alternativa al sistema de posicionamiento global (GPS por sus siglas en inglés) para dispositivos móviles. El escaneo consume grandes cantidades de energía en teléfonos inteligentes debido al escaneo completo que se realiza en todos los canales de la banda y que representa un comportamiento ineficiente que reduce la vida de la batería. Brouwers et al. [15] proponen un enfoque incremental de escaneo que reduce el consumo de energía de la localización Wi-Fi al escanear sólo unos pocos canales seleccionados, lo que podría reducir el consumo de energía entre 20 % y 58 %.

Recientemente, Arcia-Moret et al. [3] han observado que durante un proceso regular de escaneo, una estación móvil tiene que escanear múltiples veces para descubrir puntos de acceso en un área urbana densamente cubierta. Esto muestra la necesidad de un nuevo enfoque para la optimización del proceso de escaneo. En el capítulo 4 presentamos este análisis.

2.4. Uso de inteligencia computacional para el descubrimiento de redes IEEE 802.11

En el campo de la Inteligencia Computacional (IC), existen diversas técnicas de optimización multiobjetivo que imitan los principios evolutivos de la naturaleza y que son utilizados para ejecutar búsqueda y procesos de optimización. Algunas de estas técnicas están basadas en teorías propuestas por sociólogos para modelar la evolución cultural como los algoritmos culturales.

El proceso de escaneo definido por el estándar IEEE 802.11 establece parámetros que pueden variar de acuerdo a las distintas implementaciones de los fabricantes de dispositivos. En este sentido, realizar un proceso de optimización de esos parámetros mejora el descubrimiento de redes al obtener secuencias de escaneo completo eficientes.

2.4.1. Computación evolutiva

La computación evolutiva es considerada un paradigma orientado a la investigación de sistemas inspirados en la teoría de la evolución Darwiniana por

medio de la selección natural [16]. Se utiliza un proceso iterativo como el crecimiento o desarrollo de una población que luego se selecciona a través de una búsqueda aleatoria guiada para alcanzar un fin particular. En general el proceso está inspirado por mecanismos biológicos de evolución que representan una estrategia adaptativa aplicada típicamente en los dominios de búsqueda y optimización. Entre las técnicas populares de computación evolutiva se encuentran: algoritmos genéticos, estrategias de evolución, programación evolutiva y genética, optimización por colonias de hormigas, optimización por enjambre de partículas, algoritmos culturales, entre otros [16].

2.4.2. Algoritmos culturales

Los algoritmos culturales son una clase de modelos computacionales derivados de la observación del proceso de evolución cultural en la naturaleza [17]. La idea general es mejorar el aprendizaje o convergencia de una técnica de búsqueda a través de la evolución cultural. Para lograr esto se utilizan dos componentes principales: un espacio de población con individuos que evolucionan y un espacio de creencias que mantienen el conocimiento de la evolución cultural. Una descripción del marco de trabajo del algoritmo cultural (AC) se muestra en la Fig. 2.13.

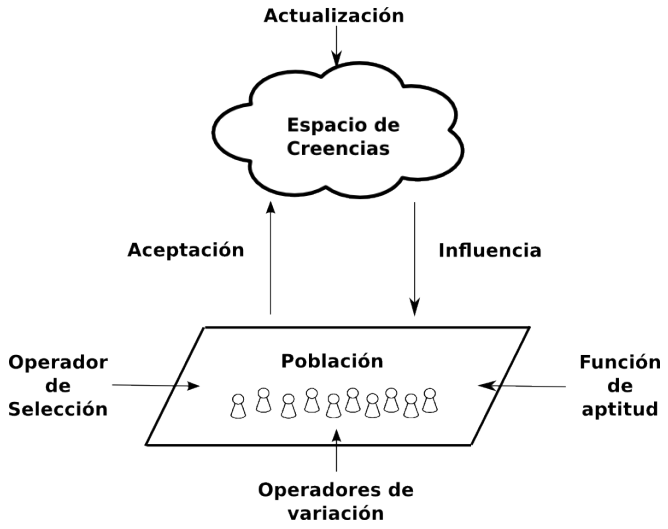


Fig. 2.13: Marco de trabajo del algoritmo cultural

El algoritmo cultural se denota como una 8-tupla tal como se muestra en la expresión 2.2.

$$AC =< P, S, V, f, B, A, U, I > \tag{2.2}$$

P es una población; S es un operador de selección, V es un operador de variación, f es la función de aptitud; B es el espacio de creencias, A es la función de aceptación, U es un operador que actualiza o ajusta el conocimiento del espacio de creencias y finalmente I es una función de influencia utilizada para influir el operador de variación V. El algoritmo cultural es un sistema de herencia dual, pues la evolución tiene lugar a dos niveles: a nivel del espacio de la población donde se mantienen los mejores individuos y a nivel del espacio de creencias donde se mantiene el conocimiento de generaciones anteriores. Estos dos componentes interactúan a través de un protocolo de comunicación representado por A e I.

Un pseudo código del algoritmo cultural se muestra en el algoritmo 2.

Algoritmo 2: Pseudocódigo básico del algoritmo cultural

Datos: Población
Resultado: Individuos optimizados

```
1 Generar población inicial;
2 Inicializar espacio de creencias;
3 Evaluar población inicial;
4 mientras no se cumpla condición de parada hacer
5 |   Actualizar el espacio de creencias;
6 |   Aplicar operadores de variación;
7 |   Evaluar cada individuo hijo;
8 |   Realizar selección;
9 fin
```

En la línea 1 se arranca el algoritmo cultural con la población inicial compuesta por individuos que representan soluciones del problema propuesto. En la línea 2 se inicializa el espacio de creencias con la representación del conocimiento y luego se evalúa el desempeño de la población inicial respecto a la función de aptitud en la línea 3. En la línea 4, el ciclo de generaciones del algoritmo cultural se repite mientras que no se cumpla una condición de parada como un número de generaciones establecido. En la línea 5 se actualiza el espacio de creencias con el conocimiento de individuos seleccionados de la

población. En la línea 6 se aplican operadores de variación a los individuos de la población actual para generar nuevos individuos influidos por el conocimiento del espacio de creencias. En la línea 7 se evalúa el desempeño de los individuos y en la línea 8 se ejecuta un proceso de selección de los mejores individuos para formar la nueva generación de la población.

Aspectos generales del diseño de algoritmos culturales

La aplicación de algoritmos culturales a la resolución de problemas exige considerar los siguientes aspectos [18]:

- Diseño del componente del conocimiento. Aquí es necesario considerar la representación del conocimiento, restricciones y la solución. Se establece cuál conocimiento se va a modificar y actualizar así como el mantenimiento del mismo.
- Diseño del componente de población. Aquí se declaran las variables que determinan el comportamiento de la solución, cómo se usan esas variables para producir un comportamiento y cómo evaluarlo.

Cada uno de los aspectos descritos tiene influencia en la aplicación adecuada de un algoritmo cultural a un problema particular.

Algunos problemas aplicables a algoritmos culturales

Reynolds [18] describe un conjunto de problemas en los cuales es posible aplicar el marco de trabajo de algoritmos culturales para su resolución. Entre estos problemas se encuentran aquellos en los que exista una cantidad significativa de conocimiento del dominio como problemas de optimización restringidos y problemas que requieren múltiples poblaciones y múltiples espacios de creencias y sus interacciones.

2.4.3. Optimización multiobjetivo

En los problemas de optimización de una sola función objetivo la tarea es encontrar una solución que optimice una función dada. A diferencia de estos, un problema de optimización multiobjetivo trata con más de una función objetivo a la vez que deben ser minimizadas o maximizadas. Así como en los problemas de un solo objetivo, en los multiobjetivo también existe un

número de restricciones que se deben satisfacer. Un problema de optimización multiobjetivo se define en su forma general como se muestra en la expresión 2.3 [19].

$$\begin{aligned} \text{Minimizar/Maximizar } & f_m(x), && m \in [1, M] \\ \text{sujeto a } & g_j(x) \geq 0, && j \in [1, J] \\ & h_k(x) = 0, && k \in [1, K] \\ & x_i^{(L)} \leq x_i \leq x_i^{(U)}, && i \in [1, n] \end{aligned} \tag{2.3}$$

Una solución x es un vector de n variables de decisión $x = (x_1, x_2, ..., x_n)^T$. El último conjunto de restricciones es llamado límite de variables y restringen a cada variable de decisión x_i a tomar un valor entre el límite menor $x_i^{(L)}$ y el límite mayor $x_i^{(U)}$. Estos límites constituyen el espacio de decisión. Para cada solución en el espacio de decisión existe un punto en un espacio denominado objetivo constituido por los valores de las funciones objetivo. Existe un mapeo que tiene lugar entre un vector de solución n-dimensional y un vector objetivo m-dimensional. La Fig. 2.14 ilustra el problema de optimización, que incluye el espacio de decisión y el espacio objetivo.

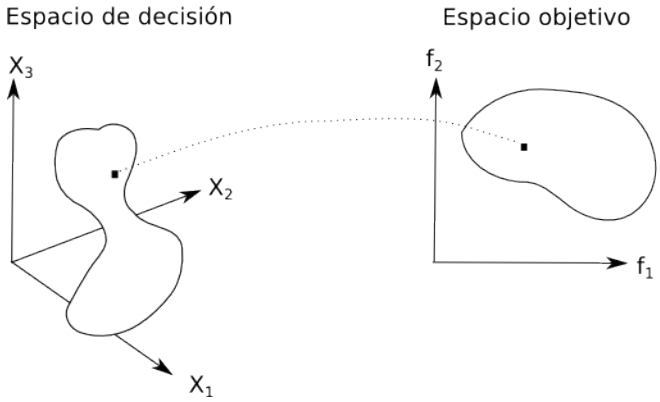


Fig. 2.14: Espacio de decisión y objetivo para un problema multiobjetivo

Una de las metas de la optimización multiobjetivo es encontrar un conjunto de soluciones tan cercano como sea posible del Frente de Pareto. Este frente comprende el conjunto de soluciones no dominadas del espacio objetivo.

2.4.4. Concepto de dominación

La mayoría de algoritmos de optimización multiobjetivo emplean el concepto de dominación en el cual dos soluciones se comparan sobre la base de si una domina a la otra o no [19]. Dadas dos soluciones x_1 y x_2 del espacio objetivo, es posible afirmar que x_1 domina a x_2 si se cumplen las siguientes condiciones:

1. x_1 es mejor o igual que x_2 en todos los objetivos.
2. x_1 es estrictamente mejor que x_2 en al menos un objetivo.

Si alguna de las condiciones descritas no se cumple, entonces la solución x_1 no domina a la solución x_2. Es posible encontrar casos en los cuales dos soluciones x_1 y x_2 son no comparables entre sí; esto es x_1 no domina a x_2 ni x_2 domina a x_1.

Como el concepto de dominación proporciona una manera de comparar soluciones con múltiples objetivos, la mayoría de los métodos de optimización multiobjetivo lo utilizan para ejecutar la búsqueda de soluciones no dominadas.

2.4.5. Optimalidad de Pareto

Dado un conjunto finito de soluciones en un problema de optimización multiobjetivo, es posible realizar todas las combinaciones posibles de comparaciones entre pares de soluciones y encontrar cuál solución domina a otra y cuáles soluciones son no dominadas con respecto a cada una de las otras [19]. El proceso de comparaciones da lugar al denominado conjunto de soluciones no dominadas ya que resultan ser mejores comparadas con el resto de soluciones. Cuando el conjunto finito de soluciones de un problema corresponde al espacio objetivo mostrado en la Fig. 2.14, el conjunto de soluciones no dominadas se denomina Pareto Óptimo. El término Pareto Óptimo tiene su origen en el ingeniero y economista italiano Vilfredo Pareto (1848–1923) quien lo utilizó en estudios de eficiencia económica y distribución del ingreso [20].

2.4.6. Principios de la optimización multiobjetivo

En la literatura se describen dos objetivos principales de la optimización multiobjetivo [21]:

1. Encontrar un conjunto de soluciones que pertenezcan al frente de Pareto.

2. Encontrar un conjunto de soluciones que sean lo suficientemente diversas para representar el rango completo del frente de Pareto.

Estos objetivos son alcanzados por las técnicas de resolución de problemas de optimización multiobjetivo a través de la implementación de distintos mecanismos y variantes particulares. En este trabajo de tesis se presentarán mecanismos orientados a satisfacer estos objetivos a través de la optimización multiobjetivo incorporada en un algoritmo cultural como se describirá en el capítulo 4.

2.4.7. Computación evolutiva en problemas multiobjetivo

En el dominio de la computación evolutiva se encuentra la optimización multiobjetivo evolutiva (EMO por sus siglas en inglés) y es reconocida como campo de estudio reciente. Los algoritmos de optimización multiobjetivo evolutiva utilizan un enfoque basado en una población en la cual más de una solución participa en una iteración y evoluciona una nueva población de soluciones en cada iteración.

Procedimiento general de la optimización multiobjetivo evolutiva

Un algoritmo de optimización multiobjetivo evolutiva sigue un procedimiento general que incluye dos pasos [21]:

1. Encontrar múltiples soluciones no dominadas tan cerca como sea posible del frente de Pareto con una amplia compensación entre los objetivos.

2. Escoger una de las soluciones obtenidas utilizando información de alto nivel.

La Fig. 2.15 muestra un esquema del procedimiento general que sigue la optimización multiobjetivo evolutiva.

En el paso 1 se encuentran múltiples soluciones no dominadas. Luego en el paso 2, un tomador de decisiones con información de alto nivel puede escoger una de las soluciones resultantes para el problema práctico que está resolviendo.

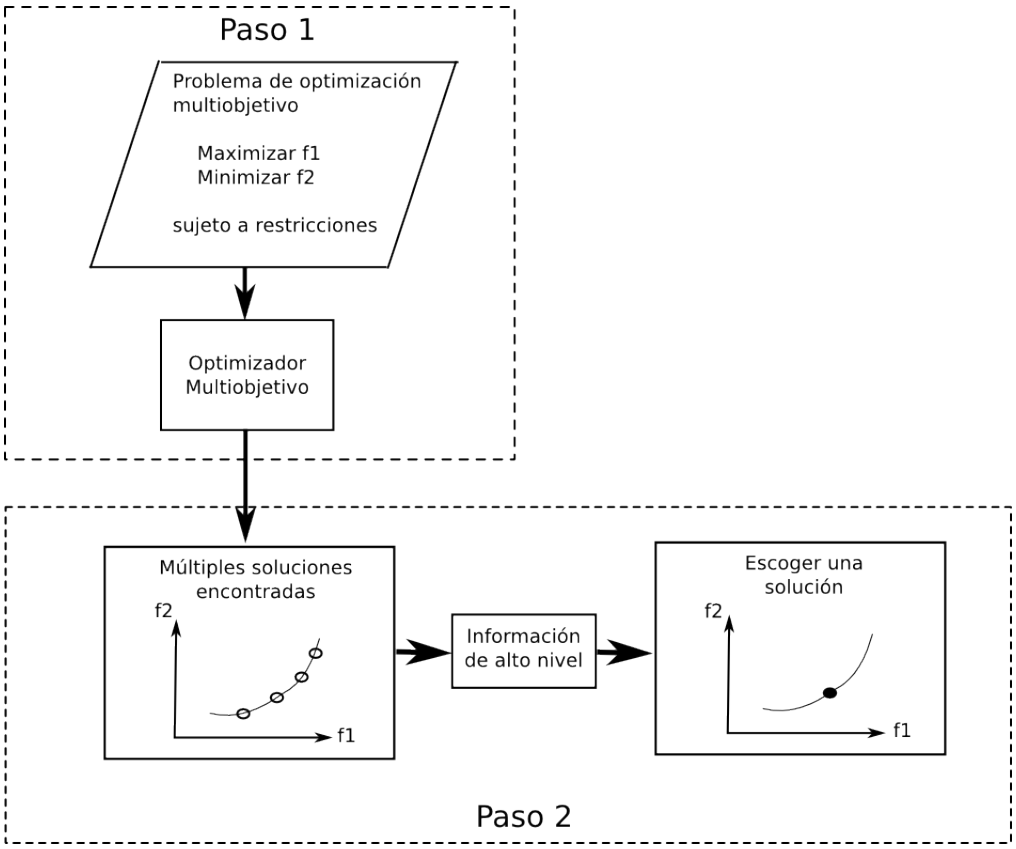


Fig. 2.15: Esquema del procedimiento de dos pasos de la optimización multiobjetivo evolutiva

2.4.8. Elitismo en optimización multiobjetivo

En los problemas de optimización multiobjetivo se utilizan técnicas en las que se intenta generar soluciones que representen los compromisos entre las funciones objetivo, para que luego un tomador de decisiones escoja la mejor solución práctica de acuerdo al problema. Algunas técnicas esperan la finalización de la generación de soluciones para establecer las preferencias sobre las funciones objetivo y otras lo hacen antes de que la técnica comience con la búsqueda.

Entre las técnicas que esperan la finalización de una generación de soluciones se utiliza el muestreo de Pareto con selección basada en elitismo para

escoger y mantener los mejores individuos [22]. En el enfoque de elitismo se almacenan los individuos no dominados para conservarlos entre generaciones mediante un archivo externo o base de datos de individuos élite.

Capítulo 3

Caracterización de la dinámica del proceso de escaneo

Las estaciones móviles y dispositivos con interfaces IEEE 802.11 se conectan a las redes luego de un proceso de descubrimiento de los puntos de acceso disponibles en su entorno. Para encontrarlos, este proceso censa las bandas del espectro ISM a través de una secuencia de canales conocida como secuencia de escaneo. La identificación de secuencias de escaneo permite comprender el comportamiento de distintos dispositivos Wi-Fi de uso masivo, los que hasta hoy, no muestran el código fuente de la implementación del escaneo. La mejora del proceso de escaneo podría, a su vez, representar directamente una mejora en el proceso de *handover* (el cual ocupa hasta un 90 % del tiempo de reconexión [7]), pues los clientes en movimiento buscan constantemente reconectarse.

Para entender el tráfico real de una red, se utiliza un analizador de paquetes o *sniffer* que captura tramas a través de la interfaz de red. En el caso de las redes 802.11, un *sniffer* común captura el tráfico de red en un canal particular de operación; esto es, en uno de los 11 canales de la banda de 2.4GHz. Debido a que los dispositivos inalámbricos ejecutan el proceso de escaneo enviando tramas para detectar puntos de acceso en todos los canales, un *sniffer* con una sola interfaz de red no es suficiente para capturar todo el tráfico generado durante el proceso de escaneo. Algunas aplicaciones de software como Kismet [23] permiten capturar tramas en redes inalámbricas 802.11 en los 11 canales. El proceso de captura se realiza en intervalos de tiempo ajustables en los que el *sniffer* permanece en un canal particular y luego salta a otro hasta recorrer todos los canales. A pesar de hacer un barri-

do por todos los canales de la banda 2.4GHz es posible que algunas tramas no sean capturadas mientras se realiza el salto entre canales. Regularmente, con este enfoque, se está capturando menos de un 10 % del tráfico presente en los 11 canales, pues se puede examinar 1 canal a la vez.

En este capítulo se presenta el diseño y prueba de un *sniffer* de redes 802.11 que pueda escuchar, simultáneamente, los 11 canales de la banda 2.4GHz reportado en [24]. Se prueba experimentalmente que es viable realizar un proceso de ingeniería inversa de los algoritmos de escaneo implementados en dispositivos móviles con distintos sistemas operativos. También, se establecen algunas premisas de diseño para la construcción del prototipo y finalmente, se realizan experimentos para una caracterización de los algoritmos de escaneo.

El capítulo está estructurado de la siguiente manera. La Sección 3.1 presenta el estado del arte y trabajos relacionados sobre escaneo en redes 802.11. La Sección 3.2 propone el diseño de un *sniffer* automático para el escaneo en redes 802.11 con su respectiva arquitectura. La Sección 3.3 discute el diseño experimental y el proceso de medición. La Sección 3.4 muestra los distintos algoritmos de escaneo encontrados en una muestra de dispositivos móviles. Finalmente, la Sección 3.5 presenta las conclusiones de este capítulo y el trabajo futuro.

3.1. Trabajos relacionados

El estándar IEEE 802.11 describe el escaneo para dispositivos inalámbricos (así como en la sección 2.3), pero no especifica ni el orden particular ni los tiempos de espera que deben seguirse para enviar las tramas de administración *Probe Request* en los canales del espectro. En consecuencia, así como lo prueban los distintos experimentos realizados, los fabricantes de tarjetas de red inalámbricas pueden implementar secuencias arbitrarias de escaneo en sus algoritmos directamente en el hardware, así como también permitir que los sistemas operativos definan una secuencia particular para el escaneo. La razón de la escogencia particular de una secuencia de escaneo es hasta ahora una pregunta abierta.

Gupta, Beyah y Corbett [25] describen un proceso empírico para caracterizar algoritmos de escaneo de diferentes tarjetas inalámbricas. Se realiza un proceso de escaneo completo para probar todos los canales del espectro 2.4GHz y para utilizar un algoritmo que genera una sola traza de captura

de tramas. La caracterización de interfaces inalámbricas se realiza sobre la base de los canales en los cuales se envía la primera trama *Probe Request*, el número de tramas *Probe Request* enviadas por cada canal, ráfagas de tramas *Probe Request* y la cantidad de tiempo de espera de la interfaz inalámbrica en cada canal del espectro 2.4GHz. Los autores proponen la caracterización para ayudar al entendimiento de las diferentes implementaciones de escaneo activo, tanto en software como en hardware, así como servir de base para evaluar su comportamiento en simuladores de redes.

Kim y Kim [6] estudian el desempeño del escaneo de canales en redes IEEE 802.11e, redes en las que se define un conjunto de mejoras de calidad de servicio para aplicaciones inalámbricas a través de modificaciones de la capa de control de acceso al medio (MAC por sus siglas en inglés). La propuesta se basa en un escaneo inteligente que considera diferentes prioridades y analiza el desempeño del escaneo de canales con respecto al número de estaciones activas.

Laurenson [26] propone el diseño de un sistema para adquisición y preservación de tráfico de red inalámbrico. El autor revisa varios enfoques para la recolección de tráfico útil en procedimientos de análisis forense digital que garanticen su confiabilidad. El sistema propuesto está compuesto por sensores inalámbricos, representados por puntos de acceso con software modificado, para recolectar y enviar tráfico de red a un servidor forense centralizado que almacena y preserva los datos adquiridos en un entorno WLAN existente.

Corbett, Beyah y Copeland [27] emplean el procesamiento de señales para analizar la periodicidad del tráfico generado por el escaneo con un análisis espectral. El mecanismo puede ser utilizado para detectar sistemas no autorizados que usan tarjetas de interfaz de red diferentes de las reconocidas como válidas. Se muestra que pueden distinguir tarjetas de diferentes fabricantes a través de un perfil espectral.

Reddy, Sharme y Paulraj [28] proponen un sistema en el que se pueden escanear los canales en la banda de radio de 2.4GHz y múltiples canales en el espectro inalámbrico de 5GHz simultáneamente. Los autores utilizan una arquitectura basada en dos subsistemas: un computador personal y un conjunto de tarjetas de computador del tipo *Single Board Computer* (SBC por sus siglas en inglés). Cada una de ellas con soporte para múltiples tarjetas inalámbricas en formato Mini PCI. Estos subsistemas se comunican a través de interfaces Ethernet. Un software específico se ejecuta en el subsistema Host y controla las tarjetas inalámbricas conectadas en cada SBC para realizar el proceso de escaneo. El uso de un computador Host (PC de escritorio) y un

Switch Ethernet para su interconexión, hacen al sistema "estacionario" o de escritorio y de costos elevados.

3.2. *Sniffer* multicanal portátil

El comportamiento de las interfaces inalámbricas 802.11 en los diferentes dispositivos móviles sigue las especificaciones del estándar IEEE 802.11. El proceso de escaneo activo está definido por ese estándar, sin embargo los fabricantes de los dispositivos pueden implementar distintas versiones del algoritmo, tanto en software, como hardware, para encontrar puntos de acceso disponibles más o menos rápido (como se muestra en la sección 3.4). Sin embargo, Castignani, Arcia-Moret y Montavont [5] muestran que no solamente es de interés encontrar rápidamente los puntos de acceso disponibles, sino que para llevar un proceso de descubierta de calidad, también es necesario tomar en cuenta la cantidad de puntos de acceso que dependera básicamente del tráfico circulante en el momento del descubrimiento.

En esta sección mostramos el diseño y construcción de un dispositivo automático para realizar ingeniería inversa del proceso de escaneo de estaciones móviles en redes 802.11. En este estudio el *sniffer* actúa como elemento adicional e inocuo para el desempeño de una red 802.11 y que permitirá recolectar datos de las estaciones móviles y analizar la manera en que realizan el proceso de escaneo activo. La Fig. 3.1 muestra un bosquejo de una red 802.11 con un *sniffer* automático. Como se puede observar, el *sniffer* debe ser capaz de recoger los *Probe Request* y *Probe Response* de diferentes canales y reportar, correctamente, la estampilla de tiempo de la trama escuchada.

En el sistema propuesto para redes 802.11, a diferencia de los *sniffer* de red comunes, se puede escuchar tráfico en los 11 canales de la banda 2.4GHz simultáneamente. En la Fig. 3.2 se muestra un diagrama de bloques con la arquitectura del *sniffer* propuesto. Cada uno de los componentes se describe a continuación.

3.2.1. Premisas de diseño

El *sniffer* se diseñó siguiendo las siguientes premisas:

- Usar componentes de fácil acceso y bajo costo. La facilidad para conseguir los elementos que forman parte del *sniffer* permitirá su rápida

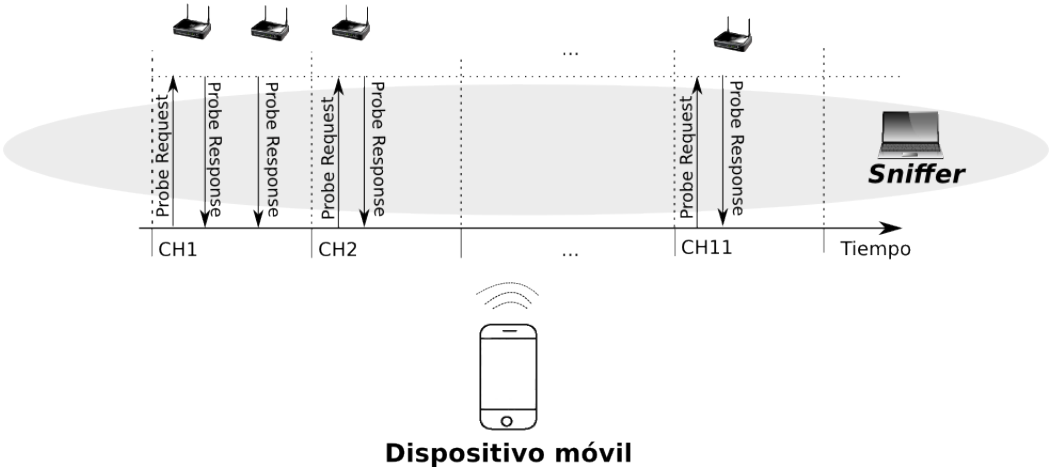


Fig. 3.1: Bosquejo de red IEEE 802.11 con un *sniffer*

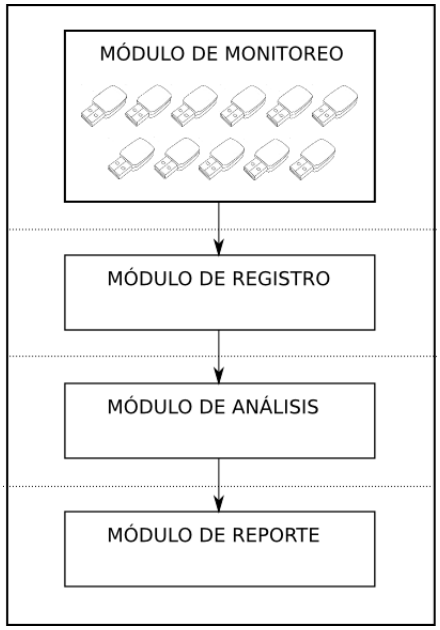


Fig. 3.2: Diagrama de bloques de la arquitectura del *sniffer* propuesto

construcción así como la reproducción de cualquier estudio que se proponga de manera sencilla.

- Construir el dispositivo que sea móvil y tenga autonomía de energía. La posibilidad de movilizar el *sniffer* con autonomía en distintos ambientes permitirá la realización de estudios de dispositivos clientes en redes con distintas configuraciones y concurridas por diferentes tipos de clientes.
- Operar bajos estándares abiertos. Se promueve el uso de estándar y tecnologías abiertas y libres. De esta manera se puede lograr un estudio preciso del funcionamiento de los algoritmos de escaneo en los clientes.

3.2.2. Módulos del sistema

A continuación se describe la función específica de cada módulo. Cada uno representa una pieza de software separable según su función, asegurando un mínimo acoplamiento entre ellos.

Módulo de monitoreo

El módulo de monitoreo se encarga de habilitar y deshabilitar adecuadamente los dispositivos inalámbricos para la captura de tramas en el *sniffer*. El dispositivo de captura de tramas que se propone es un adaptador externo, inalámbrico Wi-Fi USB. Como la propuesta del *sniffer* está orientada al estudio en el espectro de 2.4GHz de las redes 802.11, el módulo debe tener la capacidad de gestionar 11 dispositivos inalámbricos correspondientes a los 11 canales. Cada adaptador debe estar soportado por el entorno operativo del *sniffer* para habilitar su funcionamiento correctamente. La captura de tramas en cada canal ocurre luego de habilitar el modo monitor[1] en cada adaptador inalámbrico. En este modo una estación móvil puede monitorizar todo el tráfico recibido de la red inalámbrica. Un proceso automático habilitará o deshabilitará los adaptadores de acuerdo al entorno operativo que utilice el *sniffer*.

Módulo de registro

La función del módulo de registro es activar y desactivar la captura de tramas en los adaptadores inalámbricos habilitados por el módulo de monitoreo.

[1] http://wireless.kernel.org/en/users/Documentation/modes

Se generan archivos separados para cada uno de los canales en el que opera el adaptador (i.e., cada captura es atendida por un proceso por separado). Cada trama registrada es almacenada como una entrada en el archivo bajo el formato pcap [2]. El tipo de trama se obtiene a partir de la información del encabezado radiotap [3]. Los archivos generados por cada canal son integrados en un solo archivo (compactado a partir de varias fuentes) que luego pasa a ser procesado en el módulo de análisis. Para hacer la captura eficiente (por ejemplo, en medio de alto tráfico), se utilizan filtros que establecen la información asociada al estándar 802.11 que se desea almacenar en los registros. La información que se puede registrar para cada trama está limitada por la herramienta soportada en el entorno operativo del *sniffer*.

Módulo de análisis

La función del módulo de análisis consiste en aplicar algoritmos de procesamiento sobre los archivos de tramas capturadas para los 11 canales del espectro de 2.4GHz. Los algoritmos permiten realizar cálculos, generar datos y estadísticas para responder preguntas sobre el comportamiento del proceso de escaneo de estaciones móviles. Los algoritmos de análisis pueden ser implementados por programas o scripts soportados en el entorno operativo del sniffer.

Módulo de reporte

La función de este módulo consiste en generar reportes en forma de textos, gráficos y/o animaciones que puedan representar características del comportamiento de escaneo activo. Si bien se trata de un módulo que no es obligatorio para llevar a cabo la función principal del sistema, puede ser usado extemporáneamente sobre los datos obtenidos en el módulo anterior. Los reportes pueden ser implementados en programas scripts soportados en el entorno operativo del *sniffer* y que podrían apoyarse en algún tipo de interfaz de usuario.

[2]Interfaz de Programación de Aplicaciones para captura de paquetes. http://www.tcpdump.org/

[3]Estándar de facto para inyección y recepción de tramas 802.11. http://www.radiotap.org/

3.3. Diseño experimental

Sobre la base del diseño y premisas descritas en la sección 3.2 se construye un prototipo experimental cuyos componentes se muestran en la Fig. 3.3.

Fig. 3.3: Componentes del prototipo experimental de *sniffer*

La captura de tramas en los 11 canales de la banda 2.4GHz se realiza a través de adaptadores inalámbricos Wi-Fi USB. Los adaptadores están conectados a un par de concentradores de puertos USB que a su vez se conectan a dos puertos USB del computador portátil.

Originalmente, el sistema empezó a ser diseñado en base a dispositivos de bajo costo como la Raspberry Pi [4]. Sin embargo, hemos constatado que el límite en capacidad de procesamiento hace poco escalable la adquisición de tramas en distintos canales en paralelo, así como es posible en un computador portátil. La diferencia en velocidad de CPU es de 1:3 (700MHz:2.2GHz) y en memoria RAM de 1:4 (512MB:2GB). Respecto al almacenamiento, la Raspberry Pi utiliza una SD card a diferencia del computador portátil que utiliza un disco duro. Para una SD card clase 4 (class 4) el acceso de lectura y escritura mínimo comparado con un disco duro de 5400 RPM es de aproximadamente 1:37 (4MB/s para la SD card [29] y 150MB/s para el disco duro[5]).

[4]Placa computadora desarrollada por la Fundación Raspberry Pi. http://www.raspberrypi.org/

[5]Serial ATA. Estándar de facto para almacenamiento interno en PC. https://www.sata-io.org/

Tabla 3.1: Adaptadores inalámbricos Wi-Fi USB utilizados

Marca	**Modelo**	**Cantidad**	**Estándares**
TPLINK	TLW723N	8	IEEE 802.11 b/g/n
TENDA	W311M	2	IEEE 802.11 b/g/n
TENDA	W311MI	1	IEEE 802.11 b/g/n

El uso compartido de los puertos USB restringe la tasa de transferencia de datos teórica a 480 Mbit/s por cada controlador USB 2.0 en un computador [30]. En los experimentos realizados no se ha encontrado evidencia de que el uso de concentradores USB genere un cuello de botella artificial que limite el registro de tramas de administración del estándar IEEE 802.11.

A diferencia de la propuesta de Reddy, Sharme y Paulraj [28], el diseño propuesto en este documento utiliza elementos de hardware de fácil acceso, bajo costo y hace énfasis en la portabilidad del equipo para recolectar datos e identificar cómo los diferentes sistemas operativos realizan el proceso de escaneo en redes 802.11.

Para la construcción del *sniffer* multicanal se utilizaron los siguientes equipos:

- Computador portátil: Dell Inspiron 1420 con sistema operativo Backtrack Linux [6] con kernel 3.2.6, 2 GB de memoria RAM y disco duro SATA de 160 GB a 5400 RPM.
- 2 concentradores USB con 7 puertos.
- 11 adaptadores de red WiFi USB. En la Tabla 3.1 se muestran las características de los adaptadores.

Sin contar el computador portátil, el costo total para construir este prototipo (fácilmente portable) es de aproximadamente 164 US$.

[6]Distribución GNU/Linux diseñada para auditorías y pruebas de penetración relacionadas con la seguridad informática. http://www.backtrack-linux.org/

Tabla 3.2: Dispositivos clientes utilizados para las pruebas

Dispositivo	**Sistema Operativo**
Teléfono Nokia N950	Meego 1.2 Harmattan
Samsung Galaxy Tab	Android 3.2
Laptop Dell Inspiron	Debian GNU/Linux 6.0
Laptop HP	Microsoft Windows 7 Ultimate
Apple iPhone 4S	iOS 6.1.2
AP Linksys WRT54G	-

3.3.1. Experimentación

Para analizar y caracterizar el comportamiento de los algoritmos de escaneo en redes 802.11 se realizan experimentos con el sistema descrito en la sección 3.3 y dispositivos móviles con diferentes sistemas operativos. Como reportan Castignani, Arcia-Moret y Montavont [5], no existe una secuencia óptima predefinida ni un tiempo fijo para ejecutar la secuencia. En el mejor esfuerzo de búsqueda de este trabajo, se puede afirmar que no está publicada una secuencia óptima de escaneo y única. En la Tabla 3.2 se listan los dispositivos evaluados como clientes y un punto de acceso utilizados para pruebas del *sniffer* experimental.

Elaboración de la traza

La función de monitoreo del *sniffer* experimental se realiza a través de la suite de aplicaciones *Aircrack-ng*[7], específicamente con la utilidad de línea de comando *airmon-ng* que habilita el modo monitor en cada interfaz inalámbrica para escuchar todos los paquetes del estándar IEEE 802.11.

La función de registro del *sniffer* experimental se realiza con la utilidad de línea de comando *tshark* del analizador de protocolo de red *Wireshark*[8]. *tshark* permite capturar tramas y distintos campos asociados al tráfico de redes alámbricas e inalámbricas. La salida de *tshark* se almacena en un archivo

[7]Suite Aircrack-ng para romper claves WEP y WPAPSK de 802.11. http://www.aircrackng.org.

[8]Programa Wireshark para capturar y analizar trazas de tráfico de red en sistemas operativos Linux. https://www.wireshark.org/docs/manpages/tshark.html

que mantiene la traza asociada a la captura de cada uno de los 11 canales de la banda 2.4GHz.

El prototipo experimental de *sniffer* planifica la ejecución del comando *tshark* para cada una de las interfaces inalámbricas a través de la utilidad de línea de comando *crontab*[9]. Se planifica la ejecución del comando *tshark* en el mismo instante de tiempo para cada una de las once (11) interfaces inalámbricas asociadas a los canales de la banda 2.4GHz. Esto permite que la captura de tramas mantengan la misma línea de tiempo de referencia y se capture el comportamiento de clientes en todos los canales por un período determinado.

Un proceso automático (tipo script) realiza la concatenación de cada archivo de captura de canal uno tras otro, desde el canal 1 hasta el canal 11, para obtener un archivo con todas las tramas capturadas. El archivo generado de la concatenación mantiene los mismos campos de un archivo de captura de canal, se emplean utilidades de línea de comando para procesar el archivo y realizar un proceso de ordenamiento de los registros sobre la base del tiempo de captura. La Fig. 3.5 muestra un esquema de construcción del archivo de traza única de captura de tramas.

Duración del escaneo activo

A continuación se describen los pasos realizados para la medición del proceso de escaneo activo. En el cliente:

- Borrar el caché de las redes inalámbricas de la estación. Pues, cada vez que el sistema operativo se inicia o despierta luego del proceso de hibernación, realiza un escaneo de todos los puntos de acceso disponibles. Si encuentra un punto de acceso que el usuario había seleccionado como "preferido" anteriormente, automáticamente se asocia a él y no realiza el escaneo completo en todos los canales [31].

- Iniciar el proceso de escaneo en el cliente al activar la conexión inalámbrica.

Del lado del Sniffer:

[9]Programa crontab para administración de procesos en segundo plano de UNIX/Linux. http://pubs.opengroup.org/onlinepubs/9699919799/utilities/crontab.html

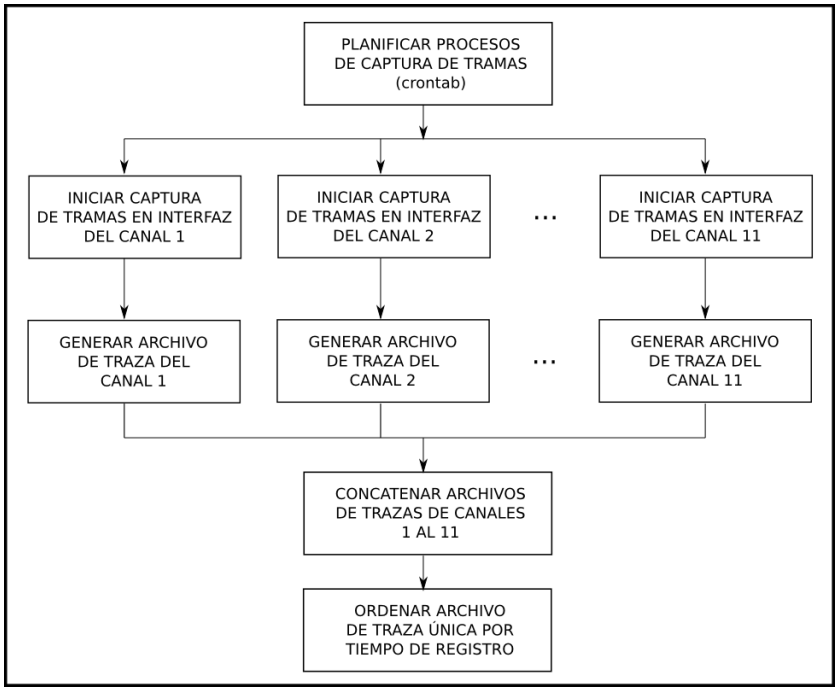


Fig. 3.4: Esquema de construcción de archivo de traza única

- Deshabilitar procesos que gestionan servicios de red como *networkmanager* y *bluetooth*.
- Configurar los adaptadores inalámbricos para ejecutar el modo monitor en cada uno de los 11 canales. Esto habilita el módulo de Monitoreo del sniffer como se describe en la subsección 3.2.2.
- Ejecutar el programa *tshark* en cada una de las interfaces en modo Monitor para iniciar la captura de tramas. Este paso habilita el módulo de registro del *sniffer* como se describe en la subsección 3.2.2.
- Esperar la captura (para este estudio particular, 4 minutos eran suficientes).
- Detener la ejecución de *tshark* en todas las interfaces y generar el archivo de tramas únicas asociado al escaneo en los 11 canales. Estas actividades las realiza el módulo de registro del *sniffer* como se describe en la subsección 3.2.2.

3.4. Resultados

En esta sección se presentan los dos resultados principales obtenidos. Primero, se muestran las secuencias de escaneo de cada uno de los distintos clientes estudiados. Segundo, se presenta la frecuencia con la que se ejecuta un proceso de escaneo en cada dispositivo móvil. Se observa que se trata en cada caso de un proceso bien particular, propio de cada fabricante.

Las tramas capturadas para todos los clientes, tienen la siguiente estructura:

- *epoch_time*: tiempo en segundos transcurridos desde el 1 de enero de 1970 hasta el momento de captura de la trama.
- *MAC origen*: dirección física o de control de acceso al medio (MAC) origen del dispositivo que origina la trama.
- *MAC destino*: dirección física o de control de acceso al medio (MAC) destino de la trama.
- *Número de secuencia*: número de secuencia (SN por sus siglas en inglés) de la trama.

- *Tipo de trama*: subtipo de la trama de acuerdo al estándar IEEE 802.11. En este caso corresponde al valor 4 para tramas *Probe Request*.
- *SSI Signal*: Potencia de la señal recibida medida en decibelio-milivatio (dBm).
- *Canal*: número del canal en el que se registra la trama.

Los archivos consolidados de captura de tramas registran los números de secuencia de cada trama escuchada y su correspondiente tiempo de captura. Con estos datos es posible establecer un orden único a los 11 archivos generados por el sniffer cuando se escucha el tráfico, simultáneamente, en todos los canales.

Una muestra de un archivo de captura de traza única se muestra en la Fig. 3.5.

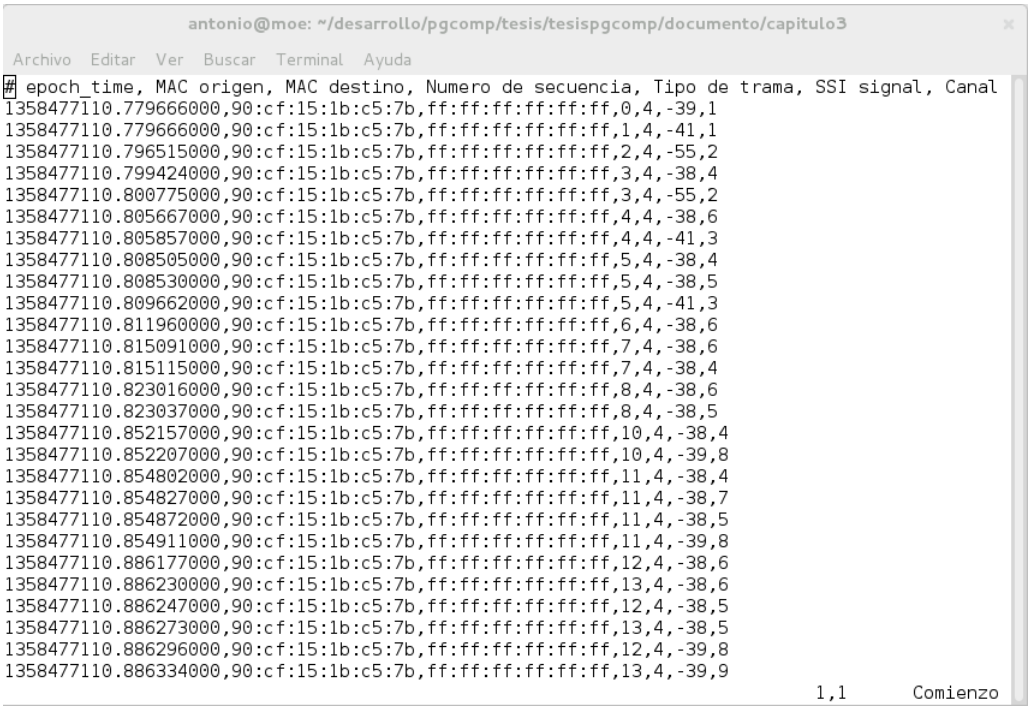


Fig. 3.5: Muestra de captura de tramas

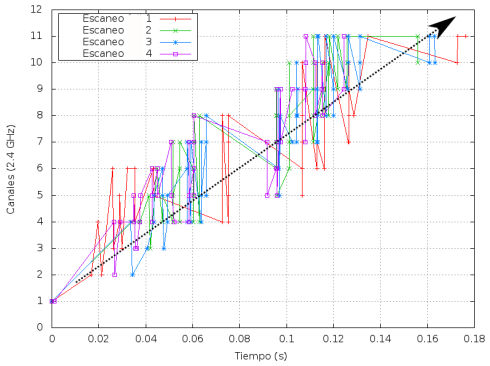
3.4.1. Las secuencias de escaneo

El estándar IEEE 802.11 [1] define el algoritmo de escaneo activo pero no especifica el orden en el cual la tarjeta inalámbrica de un dispositivo móvil debe probar cada uno de los canales. Se podría pensar que, en la banda de 2.4GHz, los canales del 1 al 11 se prueban de forma secuencial. La posibilidad de capturar tramas de los 11 canales del espectro 2.4GHz permite observar la secuencia de escaneo de una estación móvil a partir de los archivos de trama única, como los que se obtienen con el prototipo de *sniffer* construido.

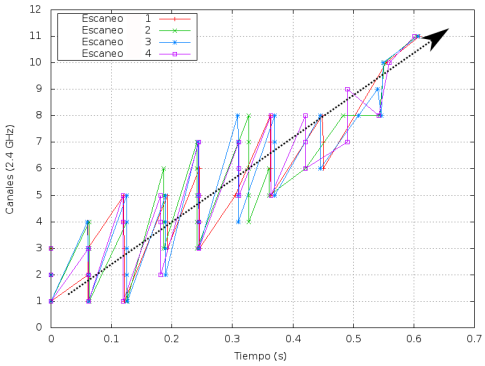
Una representación de la secuencia de escaneo se puede obtener al graficar el canal en el que se recibe la trama en función del tiempo. En esta representación es posible superponer capturas de tramas correspondientes a secuencias completas de escaneo o escaneo completo en las cuales se observan algunos patrones.

En las Fig. 3.6a, 3.6b, 3.6c, 3.6d y 3.6e se muestran gráficas superpuestas de los canales en que se capturan tramas con respecto al tiempo de duración de 4 escaneos completos para cada uno de los dispositivos evaluados. Cada punto de la gráfica representa, el tiempo en el que una trama *Probe Request* es recibida en un canal particular del espectro 2.4GHz (entre 1 y 11).

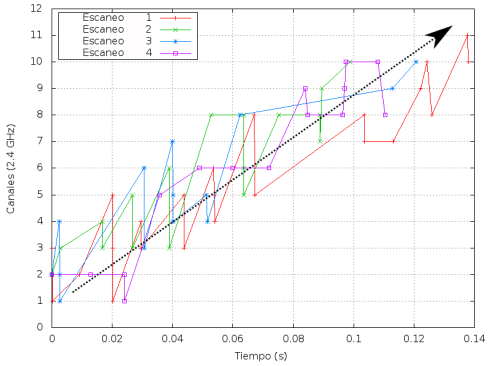
En general, en las pruebas realizadas se observó una búsqueda lineal en el sondeo de canales que van desde los números más bajos a los más altos. A medida que se sube de los números más bajos se prueban los canales vecinos próximos generando un recorrido zigzagueante. Estos recorridos hacen una búsqueda exhaustiva de puntos de acceso en todos los canales. Se considera que este patrón de búsqueda es simple y no considera la escucha de un mismo *Probe Request* en diferentes canales. Esta situación sugiere que la búsqueda podría optimizarse. Sin embargo, se observa también que, en la Fig. 3.6e el iPhone 4S muestra un comportamiento particular que no se había notado en ninguno de los otros dispositivos evaluados. Para este móvil en particular, inicialmente, existe una fase de ejecución de escaneo que hace recorrer los canales de forma ascendente (i.e., del canal 1 al 11) y otras que lo hacen en una secuencia descendente (del canal 11 al 1). Luego de un período de tiempo existe una fase de ejecuciones de escaneo completo que se realizan en secuencias ascendentes. Este comportamiento diferente resulta de la implementación particular de los algoritmos de escaneo que el dispositivo Apple iPhone 4S posee y se cree que, buscaría encontrar puntos de acceso más rápidamente.



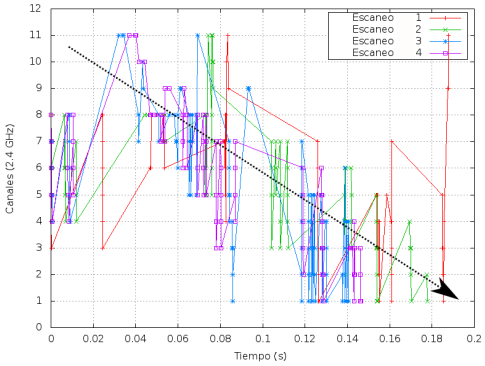
(a) Nokia N950



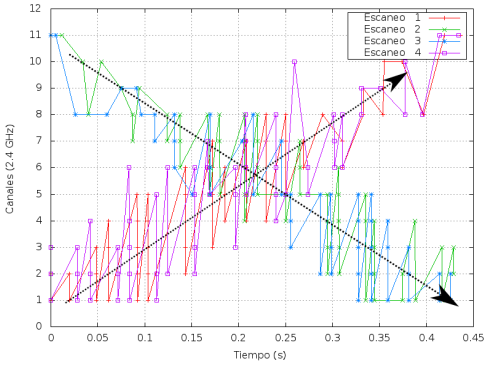
(b) Samsung Galaxy Tab



(c) Laptop Dell



(d) Laptop HP



(e) iPhone 4S

Fig. 3.6: Secuencias de escaneo para los dispositivos evaluados.

Tabla 3.3: Frecuencia de ejecución de escaneo completo en dispositivos evaluados

Sistema Operativo	**Frecuencia de ejecución (s)**
Meego©1.2 Harmattan	10
Android©3.2	10
Debian GNU/Linux 6.0	$t_0 = 0$ $t_1 = 20$ $t_2 = t_1 + 10 = 20 + 10 = 30$ $t_i = t_{i-1} + 10$ con $i = 2, 3, 4, 5$ $t_i = 60$ con $i = 6, ..., n$
Microsoft Windows 7 Ultimate©	60
iOS 6.1.2	Fase inicial: no identificable Fase estable: cada 8 s

3.4.2. Sobre la frecuencia de ejecución del proceso escaneo

Los experimentos realizados permitieron generar archivos de captura de traza única que muestran un comportamiento en el que se realiza el proceso de escaneo por intervalos de tiempo. En los distintos sistemas operativos se observa que los algoritmos de escaneo envían tramas *Probe Request* en distintos canales por un tiempo y luego esperan la recepción de una trama *Probe Response*. En la Tabla 3.3 se muestran las frecuencias de ejecución observadas para 50 repeticiones por experimento.

Para los sistemas operativos Meego©y Android©se pudo observar valores constantes y cercanos a los 10 segundos. En el caso del sistema Linux el comportamiento es incremental desde el inicio del escaneo hasta alcanzar los 60 segundos entre ejecución. El escaneo comienza en el tiempo $t_0 = 0s$, luego transcurren 20s para la segunda ejecución en t_1. A partir de este momento, la próxima ejecución de escaneo se realiza en $t_i = t_{i-1} + 10$ con $i = 2, 3, 4, 5$. Para $i = 6, \ ... \ , n$ el tiempo entre ejecución es de 60 segundos.

Para el sistema operativo Microsoft Windows 7 Ultimate©se observaron valores constantes de 60 segundos entre ejecución de escaneo completo. Para el caso del sistema operativo iOS©6.1.2, se observó que la ejecución de escaneo completo se realiza por fases. Una fase inicial en la que el tiempo aproximado de ejecución entre escaneo completo no pudo ser determinado.

Luego, una fase estable (o predecible) en la cual el tiempo entre ejecución de escaneo completo se encuentra alrededor de los 8 segundos.

3.4.3. Tiempo de duración del proceso de escaneo

Castignani, Arcia-Moret y Montavont [5] presentan un estudio del proceso de descubrimiento en redes 802.11. Una de las métricas seleccionadas para caracterizar a este proceso es el tiempo de duración de un escaneo completo que, corresponde al tiempo transcurrido durante el proceso de descubrimiento para escanear todos los canales disponibles en cualquier orden. En la Fig. 3.7 se muestra un bosquejo de la duración del proceso de escaneo completo.

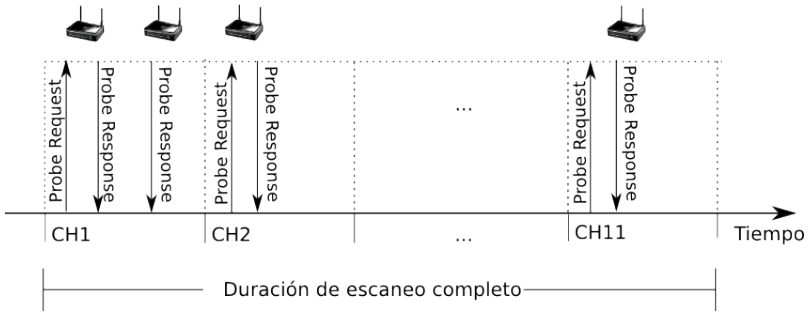


Fig. 3.7: Bosquejo de la duración de un escaneo completo

En la Tabla 3.4 se muestran los tiempos de duración promedio de escaneo completo para los sistemas operativos de los dispositivos evaluados para 50 repeticiones por experimento. Estos tiempos se encuentran entre los valores sugeridos que varían entre 70ms y 600ms para el retardo de escaneo tal como lo reportan Murray, Dixon y Koziniec [32].

3.5. Conclusiones

El prototipo experimental de *sniffer* para redes 802.11 descrito permitió capturar tramas de administración del estándar IEEE 802.11 en los 11 canales de la banda 2.4GHz. En la construcción se utilizaron componentes de bajo costo como: adaptadores inalámbricos USB, concentradores USB y un computador portátil. El proceso de captura de tramas se automatizó a

Tabla 3.4: Tiempos de escaneo completo para los sistemas operativos de los dispositivos evaluados

Sistema Operativo	**Duración de escaneo completo (ms)**
Meego©1.2 Harmattan	$\mu = 184,35719$ y $\sigma = 30,06954$
Android©3.2	$\mu = 601,59232$ y $\sigma = 11,8229$
Debian GNU/Linux 6.0	$\mu = 113,2495$ y $\sigma = 14,0071$
Microsoft Windows 7 Ultimate©	$\mu = 168,0186$ y $\sigma = 22,7389$
iOS©6.1.2	$\mu = 429,23747$ y $\sigma = 25,53178$

través de un enfoque modular, cuyo objetivo final es generar una traza única de la actividad de escaneo para un dispositivo móvil cualquiera.

Se observó que los dispositivos inalámbricos evaluados presentan distintas secuencias de ejecución de escaneo. Unos dispositivos tienen una tendencia a ejecutar el escaneo de forma ascendente desde los primeros canales a los últimos de la banda 2.4GHz. Sin embargo, se encontró que pueden haber casos donde se ejecuta el escaneo en secuencias alternadas desde los primeros canales a los últimos y viceversa.

La frecuencia de ejecución del escaneo es distinta para cada dispositivo. Se observaron valores constantes para su ejecución mientras que otros tienen un comportamiento incremental hasta un umbral como se ve en el sistema operativo Linux.

Sobre la base de los resultados presentados aún no se ha identificado la secuencia de escaneo real (la lista de canales propiamente dicha) para los dispositivos evaluados en estricto orden cronológico. El problema de identificación de la secuencia real de escaneo puede ser abordado con estudios estadísticos y aplicación de algoritmos más elaborados para identificación de patrones.

En futuras versiones de este trabajo, se propone estudiar la conducta de los algoritmos de escaneo en presencia de tráfico de fondo. También se propone realizar medidas del consumo de energía que aporta un algoritmo de escaneo.

Capítulo 4

Inteligencia computacional en el proceso de escaneo

En este trabajo se adapta e implementa un algoritmo cultural (AC) propuesto por Coello y Becerra [33] para abordar problemas de optimización multiobjetivo. El AC está basado en programación evolutiva, eficiencia de Pareto y elitismo (individuos elegidos finamente) para lograr una búsqueda y construcción más eficiente de secuencias de escaneo óptimas.

Como se explicó en § 2.3 el proceso de descubrimiento o escaneo es una función del estándar IEEE 802.11 en la que las estaciones móviles buscan puntos de acceso disponibles para luego asociarse a las redes. Durante el escaneo se utilizan parámetros como los temporizadores MinCT y MaxCT definidos por el estándar para ajustar su desempeño. Así como se explica en Castignani et al. [5], este desempeño puede ser caracterizado por un conjunto de métricas como la tasa de descubierta que representa la fracción de puntos de acceso descubiertos sobre el total de puntos de acceso desplegados durante el escaneo, la latencia de descubierta que representa el tiempo que transcurre para escanear todo el conjunto de canales (once canales en la banda de 2.4GHz) y la tasa de falla que representa la probabilidad de no encontrar ningún punto de acceso después de completar el escaneo.

Las métricas de escaneo varían de acuerdo a los parámetros que generan un compromiso entre ellas; esto es: disminuir la latencia para un menor impacto sobre el tipo de aplicación e incrementar el número de puntos de acceso que se descubren en el proceso de escaneo. Sobre la base de estas métricas es posible definir el problema de optimización de los parámetros de desempeño del escaneo en redes IEEE 802.11 (ver sección § 4.2.1).

En este capítulo se presenta un enfoque nuevo para encontrar la configuración de secuencias de escaneo óptima. El problema de escaneo puede ser declarado como un problema de optimización multiobjetivo; esto es, encontrar un compromiso entre métricas de desempeño como lo describen Montavont et al. [34]. En este sentido, se utiliza el AC para producir secuencias de escaneo adaptadas a una topología espontánea. Se realizan emulaciones con el uso de mediciones reales [3] y se utiliza un modelo de emulación. Con el uso de este modelo, el AC genera un conjunto de secuencias a lo largo del frente de Pareto (ver § 2.4.5) que pueden ser luego utilizadas de acuerdo a las necesidades de los usuarios. El enfoque propuesto en este capítulo está reportado en [35].

4.1. Trabajos relacionados

Aunque no está relacionado directamente con el problema del escaneo, Castignani et al. [36] presentan un enfoque para abordar el *multihoming* en dispositivos móviles con diferentes interfaces y seleccionar la mejor red para distintos flujos de aplicaciones. En este trabajo se aborda la selección de interfaz de red para distintos flujos de datos en estaciones móviles con múltiples interfaces. Este fenómeno se modela a través de un problema de optimización multiobjetivo en el que se aborda el compromiso entre la insatisfacción de ancho de banda y el consumo de energía a través de un algoritmo genético. En este caso se utiliza una técnica de inteligencia computacional para resolver un problema de optimización multiobjetivo en el dominio de las redes inalámbricas.

Con respecto al proceso de escaneo en redes 802.11 se han realizado trabajos que apuntan a estudiarlo desde distintas perspectivas. Una de las primeras propuestas de modificación de los temporizadores del proceso de escaneo fue abordada por Velayos y Karlsson [11]. Los autores propusieron valores fijos para los temporizadores basados en los mejores valores teóricos y estimaciones de los peores casos para el envío y recepción de tramas *Probe Request* y *Probe Response*. Castignani et al. [37] abordan un enfoque en el que adaptan los temporizadores de acuerdo a la carga de la red, adaptando eficientemente los temporizadores durante el proceso de escaneo.

Algunos enfoques agrupan canales para escanearlos cada cierto tiempo y permitir el envío alternado de tramas de prueba *Probe Request* y tráfico de datos para producir el mínimo impacto en transferencias continuas. Monta-

vont et al. [12] usan un período fijo y uno independiente. Liao y Cao [38] usan una técnica de escaneo que introduce períodos de escaneo variable. Nah et al. [39] utilizan el mismo principio de alternar ráfagas de escaneo para mejorar la experiencia del usuario.

Otros enfoques apuntan a minimizar el tiempo o latencia de escaneo al reducir las secuencias de canales escaneados. Shing et al. [40] proponen un escaneo en los canales no solapados 1, 6 y 11 de la banda 2.4 GHz, sin embargo no consideran la adaptación a la topología de la red. Eriksson et al. [13] proponen calcular progresivamente la probabilidad de obtener un punto de acceso en un canal particular y luego calcular la secuencia de escaneo.

Castignani et al. [5] realizan un estudio a partir de simulaciones y un banco de pruebas real en el que evalúan el proceso de escaneo enfocados en los valores de los temporizadores IEEE 802.11: MinCT y MaxCT. Los autores varían los valores de los temporizadores y proponen una estrategia de descubrimiento adaptativa que muestra mejoras notables comparadas con una estrategia de temporizadores fijos. Es importante notar que, a diferencia de lo propuesto por Castignani et al. [5], nosotros proponemos un modelo que no depende del conocimiento total de la topología. En consecuencia la tasa de descubierta no tiene sentido en este contexto.

Montavont et al. [34] aplican la técnica de algoritmos genéticos para encontrar secuencias de escaneo eficientes sobre la base de las métricas descritas en [5]. Las secuencias generadas son utilizadas para minimizar la latencia de escaneo completo reduciendo el impacto del escaneo y, al mismo tiempo, maximizar el número de puntos de acceso descubiertos.

4.2. Modelo de optimización

4.2.1. Caracterización del desempeño del escaneo IEEE 802.11

Castignani et al. [5] han caracterizado el desempeño del escaneo a través de la latencia de escaneo que representa el tiempo que transcurre para escanear todo el conjunto de canales (once canales en la banda de 2.4GHz), la tasa de falla que representa la probabilidad de no encontrar ningún punto de acceso después de completar el escaneo y, finalmente, la tasa de descubrimiento que representa la fracción de puntos de acceso descubiertos sobre el total de puntos de acceso disponibles.

En este trabajo la tasa de descubrimiento representa el número de puntos de acceso encontrados por unidad de tiempo al recorrer todo el conjunto de canales de la banda 2.4GHz. De esta manera es posible realizar un proceso de optimización que permita encontrar la máxima tasa de descubrimiento de puntos de acceso útiles por unidad de tiempo en cada canal al establecer un compromiso con la latencia de escaneo.

4.2.2. Modelo de optimización propuesto

En este trabajo la optimización de los parámetros de desempeño del escaneo en redes IEEE 802.11 se aborda como un problema de optimización multiobjetivo para encontrar secuencias de escaneo óptimas. Se propone un modelo de optimización de dos funciones objetivo, maximizar FO_1 representada por la ecuación 4.1 y minimizar FO_2 representada por la ecuación 4.2, basadas en las métricas de desempeño del escaneo IEEE 802.11 tasa de descubrimiento y latencia de escaneo.

$$R = \sum_{i=1}^{11} Nmin_{C_i}/MinCT_{C_i} + Nmax_{C_i}/MaxCT_{C_i} \tag{4.1}$$

$$L = \sum_{i=1}^{11} (MinCT_{C_i} + p_i \cdot MaxCT_{C_i}) \forall C_i \in [1, 11] \tag{4.2}$$

La ecuación 4.1 corresponde a la tasa de descubrimiento por unidad de tiempo (R), expresada en número de APs por ms y la ecuación 4.2 corresponde a la latencia agregada (L), expresada en milisegundos (ms). $Nmin_{C_i}$ y $Nmax_{C_i}$ corresponden al número total de APs descubiertos en el canal i, con los temporizadores MinCT y MaxCT respectivamente. Luego de realizar un análisis de varios miles de trazas, se impusieron las restricciones a los temporizadores MinCT y MaxCT descritas en las ecuaciones 4.3 y 4.4.

$$5 \leq \text{MinCT} \leq 15 \tag{4.3}$$

$$3 \leq \text{MaxCT} \leq 90. \tag{4.4}$$

Los valores para los intervalos de MinCT y MaxCT están expresados en milisegundos, y se obtuvieron considerando los percentiles 20 y 80 respectivamente reportados por Arcia-Moret et al. [3] y Montavont et al. [12].

En la ecuación 4.2, p_i corresponde a la probabilidad de tener al menos una respuesta dentro de cualquier canal. Esta probabilidad se obtiene por canal, considerando que (1) cada canal tiene un efecto diferente sobre APs que solapan y (2) el tiempo entre respuesta de un AP (*Probe Response*) varía de acuerdo al número total de APs que se encuentran en un canal. C_i representa el canal i de un total de 11 canales en la banda 2.4GHz.

4.3. Algoritmo cultural

Originalmente el algoritmo cultural fue propuesto por Coello y Becerra [33], en este trabajo se incorpora una modifición en la mutación y en el espacio de creencias del algoritmo cultural. La mutación se realiza de manera dirigida por el conocimiento que se mantiene en una nueva estructura del espacio de creencias denominada tabla de superindividuo. La mutación dirigida por este nuevo conocimiento se va a ejecutar de acuerdo a una probabilidad de ocurrencia, así, en algunos casos se ejecuta la mutación original y en otros la mutación dirigida. El objeto de la mutación dirigida es orientar a las individuos mutados a ordenar secuencias de escaneo por los canales que obtengan mejores valores de desempeño de acuerdo a las funciones seleccionadas para la optimización multiobjetivo.

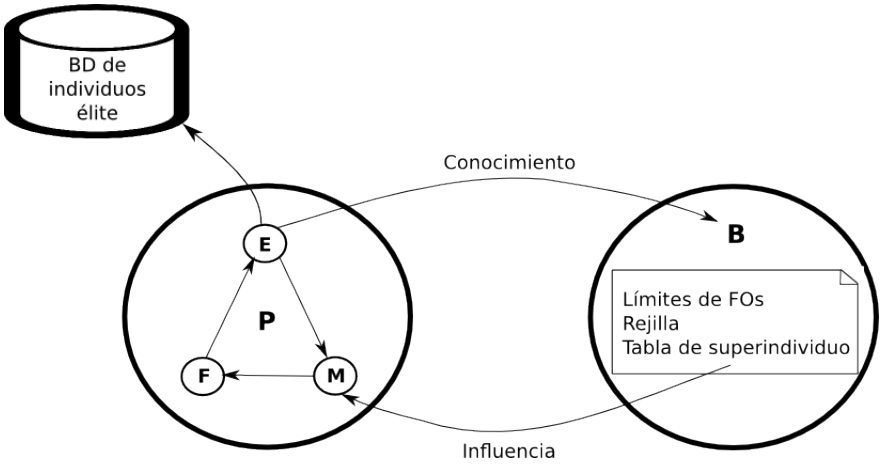


Fig. 4.1: Perspectiva general del algoritmo cultural adaptado

La Fig. 4.1 muestra una macro representación del **AC** adaptado. El en-

foque depende de dos grandes procesos, las operaciones que transforman la población **P**, y los ajustes del espacio de creencias **B**. Una población inicial **P** pasa a través de diversos ciclos de **M**utación, luego **F**iltrado de los individuos con mejor desempeño y finalmente un proceso de **E**xtracción de los individuos no dominados que se insertan en una base de datos de individuos élite (**BDe**). Estos individuos contribuyen al conocimiento dentro del espacio de creencias (**B**) sobre los límites de las variables estudiadas y la exploración de un frente de Pareto más amplio. Consecuentemente, **B** tiene influencia sobre los subsecuentes procesos de **M**utación para obtener las futuras generaciones.

Los pasos que describen el **AC** se listan en el Algoritmo 3. En la línea 1 del algoritmo se define una población de tamaño **P** que será mutada durante el proceso de evolución. El número inicial de individuos se define de acuerdo a lo propuesto por Coello y Becerra [33] y es un parámetro del algoritmo cultural. Luego en la línea 2 se evalúa el desempeño de cada individuo de la población **P** con respecto a las funciones objetivo FO_1 y FO_2. En la línea 3 se calcula el superindividuo que se utilizará para ejercer influencia sobre cada individuo durante la mutación. Luego en la línea 4 se inicializa el espacio de creencias; aquí se determinan los valores mínimo y máximo de FO_1 y FO_2, se creará una rejilla para enmarcar individuos de manera que soluciones agrupadas ejerzan influencia en el proceso de filtrado de individuos. El espacio de creencias **B** tiene un impacto sobre la población **P** a través de la adaptación de los límites de los rangos de las funciones objetivo. Adicionalmente se incorpora un nuevo operador de mutación para guiar a la población y tener la posibilidad de explorar el frente de Pareto óptimo.

Después del proceso de inicialización, comienza el refinamiento de la población en la línea 5. Los refinamientos inician con un proceso de mutación (línea 6) para obtener **2P** individuos, que luego son filtrados a **P** individuos a través de torneos (línea 7). Luego, se obtienen los individuos no dominados (línea 8) que ayudan a realizar ajustes en el espacio de creencias. En las líneas 9, 10 y 11 se actualizan la rejilla (ver Fig. 4.3), el componente normativo fenotípico y tabla de superindividuo respectivamente. Finalmente, el proceso de iteración para refinar los individuos actuales consiste en crear una nueva generación. Después de un número predefinido de generaciones[1], los individuos resultantes son la entrada para que un tomador de decisiones (que vive dentro del gestor de topología) ajuste luego las secuencias de escaneo a necesidades particulares de aplicaciones.

[1]parámetro del AC.

Algoritmo 3: El Algoritmo Cultural

Datos: modelo de topología inalámbrica
Resultado: secuencia de escaneo adaptada (individuos)

```
1  Inicializar con P individuos aleatorios;
2  Calcular FO_1 y FO_2 para cada individuo;
3  Calcular Super_Individuo;
4  Inicializar el espacio de creencias (B);
5  mientras i ≤ Total_Generaciones hacer
6      Mutar individuos para generar población 2P P_new;
7      Filtrar P_mejores individuos a través de torneos en P_new;
8      Agregar individuos no dominados (P_nd) de P_mejores a la base de
       datos de individuos élite (BDe);
9      Actualizar rejilla usando P_nd;
10     Actualizar la parte normativa fenotípica cada
       M < Total_Generaciones (ver § 4.3.2);
11     Actualizar Super_Individuo;
12 fin
```

4.3.1. Estructura de la población

La población de un algoritmo cultural se concibe como individuos que representan soluciones candidatas y sus características se traducen en una función objetivo (FO) [16]. En este trabajo un individuo es asociado a una secuencia de canales con sus respectivos temporizadores que se utiliza en el proceso de escaneo. Para realizar la búsqueda evolutiva se propone un individuo que tiene la siguiente estructura: $< X_1, ..., X_{11} >< FO_1, FO_2 >$ en la cual cada gen X_i está compuesto por $< C_i, Min_i, Max_i, AP_i >$ con $i = 1, ..., 11$, donde C_i es el canal enumerado i, Min_i es MinCT (expresado en ms) para el canal i, Max_i es MaxCT (expresado en ms) para el canal i, AP_i es el número de puntos de acceso descubiertos en el canal i, FO_1 es el valor que da cuenta de la tasa de descubrimiento de APs por unidad de tiempo y que debe ser maximizada, y FO_2 es el valor que da cuenta de la latencia de escaneo que debe ser minimizada. Existen 11 genes en un individuo debido a que en este estudio se utiliza la banda de 2.4GHz que posee 11 canales.

Un individuo tendrá un total de 44 parámetros y dos valores de funciones objetivo (tasa de descubrimiento de APs por unidad de tiempo y valor de latencia). Un ejemplo de un individuo de la población del algoritmo cultural

se muestra en la tabla 4.1. En la tabla 4.1 el gen X_1 del individuo tiene valores de C=6, Min=7, Max=48 y AP=4.

Tabla 4.1: Ejemplo de un individuo de la población del algoritmo cultural

$< X_1 >< X_2 >< X_3 >< X_4 >< X_5 >< X_6 >< X_7 >< X_8 >< X_9 >< X_{10} >< X_{11} >< FO_1, FO_2 >$
<6,7,48,4><8,5,33,0><5,6,86,0><7,6,30,3><1,6,12,1><2,5,43,0><4,7,15,0><3,7,22,0><9,10,16,0><11,6,31,5><10,2,40,0><13,443>

4.3.2. Estructura del espacio de creencias

El espacio de creencias consta de tres partes:

1. Parte normativa fenotípica.
2. Rejilla.
3. Tabla de superindividuos.

La parte normativa fenotípica del espacio de creencias **B** mantiene únicamente los límites inferior y superior de los intervalos para cada función objetivo (FO) dentro de los cuales se construirá la rejilla. La estructura se muestra en la Fig. 4.2.

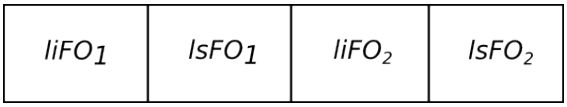


Fig. 4.2: Parte normativa fenotípica del espacio de creencias

donde los límites corresponden a los valores límites de los individuos no dominados:
$liFO_1$: corresponde al valor del límite inferior de FO_1.
$lsFO_1$: corresponde al valor del límite superior de FO_1.
$liFO_2$: corresponde al valor del límite inferior de FO_2.
$lsFO_2$: corresponde al valor del límite superior de FO_2.

La rejilla del espacio de creencias **B** se usa para enfatizar la generación de soluciones no dominadas distribuidas uniformemente a lo largo del frente de Pareto. La estructura de la rejilla se muestra en la Fig. 4.3.

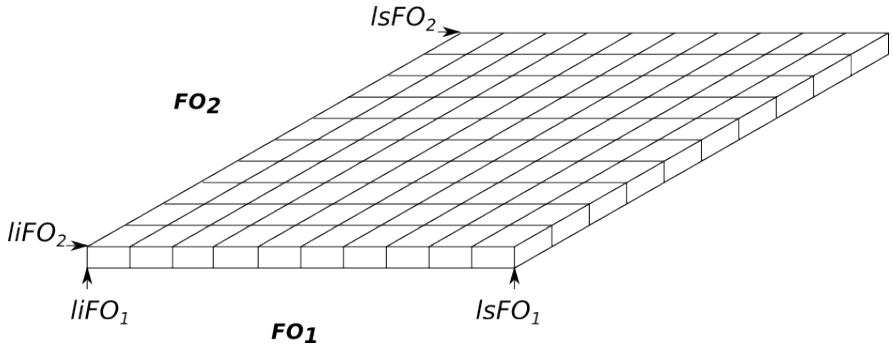


Fig. 4.3: Rejilla del espacio de creencias

La rejilla se construye utilizando los valores de la parte normativa fenotípica para ubicar cada solución sobre la base de los valores de sus funciones objetivo. Con los valores de los intervalos de las funciones objetivo, se requiere un número de subintervalos iguales[2], s_1 (número de subintervalos para FO_1) y s_2 (número de subintervalos para FO_2) en los que se dividirá cada uno, para construir la rejilla. Como resultado se tendrán $s_1 * s_2$ celdas, en cada una de las cuales se almacena la cuenta de los individuos no dominados de la base de datos de individuos élite que estén dentro de cada una. Con esto se busca distribuir adecuadamente las soluciones no dominadas entre las celdas, evitando que se agrupen todas en una zona única del frente de Pareto. En la Fig. 4.4 se muestra el estado de la rejilla al momento de ser inicializada. Con el avance de las generaciones del algoritmo cultural las celdas de la rejilla incrementan los contadores de individuos no dominados que caen dentro de ellas. En la Fig. 4.5 se muestra el estado de la rejilla luego de varias generaciones del algoritmo cultural. El proceso de actualización de la rejilla se describe en § 4.3.5.

La tabla de superindividuo del espacio de creencias **B** registra la estructura de un individuo identificado como superindividuo a lo largo de la evolución del algoritmo como nuevo conocimiento del espacio creencias. Tal como se describe en § 4.3.1, el superindividuo está compuesto por los 11 genes $< X_1, ..., X_{11} >$ en el que cada X_i incluye $< C_i, Min_i, Max_i, AP_i >$ con $i = 1, ..., 11$. Se denomina superindividuo porque éste posee el mejor gen i de todos los individuos de la población **P** en una iteración del algoritmo cul-

[2]El número de subintervalos de la rejilla es un parámetro del algoritmo cultural.

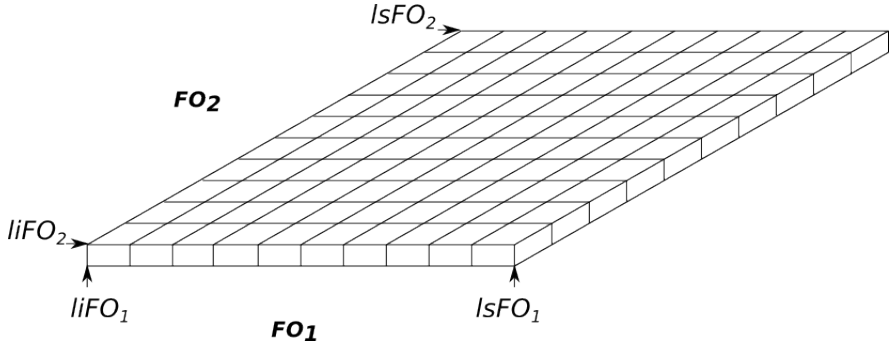


Fig. 4.4: Estado de la rejilla del espacio de creencias al ser inicializada

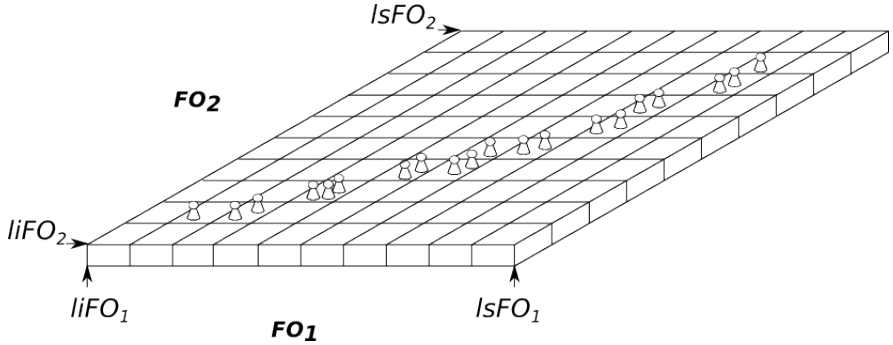


Fig. 4.5: Un estado de la rejilla del espacio de creencias luego de varias generaciones.

tural. Con esto se busca ordenar los genes en el individuo de acuerdo a los mejores valores de desempeño que corresponden a los mejores canales de la banda 2.4GHz en una secuencia de escaneo. Este superindividuo es utilizado en el proceso de mutación dirigida para seleccionar una ventana de mutación[3] (W), que incluye un número específico de genes X_i, que influirán en la generación de nuevos individuos. La estructura de la tabla de superindividuo se muestra en la Fig. 4.6. Adicionalmente, se mantiene un registro de la evolución del superindividuo a lo largo de toda la ejecución del algoritmo cultural. Es decir, por cada ciclo o generación se genera un superindividuo y se actualiza.

[3]El tamaño de la ventana de mutación es un parámetro del algoritmo cultural.

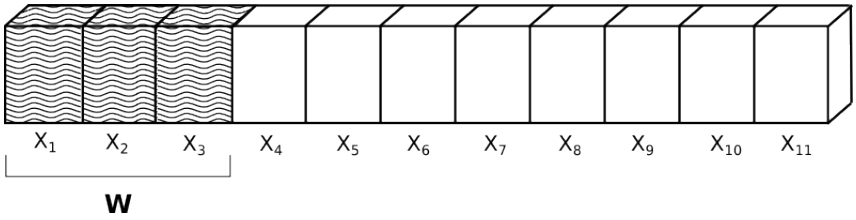


Fig. 4.6: Tabla de superindividuo del espacio de creencias

4.3.3. Inicialización del espacio de población

La inicialización de la población del AC consiste en crear **P** individuos aleatorios de acuerdo a la estructura descrita anteriormente. Una descripción detallada de la plataforma experimental utilizada para el proceso de creación de un individuo de la población **P** se presenta en § 5.1.

4.3.4. Inicialización del espacio de creencias

Para inicializar el espacio de creencias se utiliza el espacio de población creado e inicializado previamente (ver § 4.3.3). Sobre la población de tamaño **P** se ejecuta un proceso para determinar los individuos no dominados (ver § 2.4.4) que servirán para inicializar la parte normativa fenotípica y la rejilla del espacio de creencias.

Inicialización de la parte normativa fenotípica

La inicialización de la parte normativa fenotípica del espacio de creencias consiste en encontrar los valores inferior y superior de cada función objetivo que se encuentren en los individuos no dominados de la población inicial. Para esto se ejecuta un ordenamiento de los individuos no dominados con respecto a FO_1 y se asigna el de menor valor para $liFO_1$ y el de mayor valor para $lsFO_1$. Este mismo procedimiento se realiza para FO_2.

Inicialización de la rejilla

La rejilla se crea tomando como intervalos los valores almacenados en la parte normativa fenotípica y se divide utilizando los parámetros de entrada $s1$ y $s2$. Los contadores de los individuos no dominados dentro de cada celda

se inicializan en cero. El proceso consiste en: leer los límites $liFO_1$, $lsFO_1$, $liFO_2$, $lsFO_2$ y crear una matriz de dimensión $s_1 * s_2$ celdas[4] con un contador entero en cada celda. Para cada celda de la rejilla se asigna cero al valor del contador.

Inicialización de la tabla de superindividuos

La tabla de superindividuos se inicializa al recorrer todos los individuos de la población **P** y seleccionar el gen X_i con $i = 1, ..., 11$ que tenga mejor valor de desempeño de la tasa de descubrimiento de puntos de acceso por unidad de tiempo (FO_1) entre todos los genes de todos los individuos. En el algoritmo 4 se muestran los pasos de la inicialización.

Algoritmo 4: Inicialización de la tabla superindividuo

```
Datos: Población P
Resultado: Superindividuo
Crear un individuo y etiquetarlo superindividuo;
Para Todos los individuos i de la población P hacer
    Para Todos los genes X_{i,j} =< C_{i,j}, Min_{i,j}, Max_{i,j}, AP_{i,j} > de cada
    miembro i de P hacer
        Seleccionar el mejor gen < C_{1,j}, Min_{1,j}, Max_{1,j}, AP_{1,j} >de
        todos los j;
        Asignar el gen X_i seleccionado en la posición j en el
        superindividuo;
    fin
fin
```

4.3.5. Actualización del espacio de creencias

La rejilla y la tabla de superindividuos del espacio de creencias se actualizan cada generación. La actualización de la parte normativa fenotípica se realiza para garantizar que puedan incluirse nuevos límites de las funciones objetivo a partir de individuos no dominados que se han registrado en las

[4]Está reportado por Coello y Becerra [33] que 10 es un buen número para los subintervalos.

generaciones del algoritmo cultural. La parte normativa fenotípica se actualiza de acuerdo a un número de generaciones que especifica la frecuencia de actualización porque implica una reconstrucción de la rejilla. La frecuencia de actualización es un parámetro del algoritmo cultural y se denomina g_N. Todos los parámetros del algoritmo cultural se describen en § 4.3.10. En la Fig. 4.7 se ilustra la frecuencia de actualización g_N.

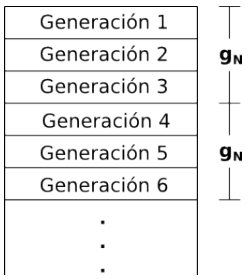


Fig. 4.7: Frecuencia de actualización de la parte normativa genotípica.

Actualización de la rejilla

La actualización de la rejilla consiste en incrementar los contadores de los individuos no dominados con todos los individuos recién agregados a la base de datos de individuos élite (BDe) durante la generación actual. Los contadores se incrementan con el objetivo de ser utilizados luego en el proceso de filtrado de individuos por torneos (ver § 4.3.8) para generar una nueva población cada generación del algoritmo cultural. Esta parte del espacio de creencias se actualiza utilizando la población de la BDe y elige únicamente a los individuos nuevos de esa población.

Actualización de la parte fenotípica normativa

Para la actualización de la parte normativa fenotípica es necesario identificar los valores inferior y superior de cada función objetivo de los individuos que se encuentran en la BDe. Con estos nuevos valores se reconstruya la rejilla del espacio de creencias. En el Algoritmo 5 se describen los pasos para la inicialización.

Algoritmo 5: Actualización de la Parte Normativa Fenotípica

Datos: Individuos de la base de datos de élite BD
Resultado: Límites inferiores y superiores de funciones objetivo

1 Ordenar los individuos de la base de datos de élite con respecto a FO_1;
2 Seleccionar el individuo de menor valor de FO_1 y asignar a $liFO_1$;
3 Seleccionar el individuo de mayor valor de FO_1 y asignar a $lsFO_1$;
4 Ordenar los individuos de la base de datos de élite con respecto a FO_2;
5 Seleccionar el individuo de menor valor de FO_2 y asignar a $liFO_2$;
6 Seleccionar el individuo de mayor valor de FO_2 y asignar a $lsFO_2$;
7 Reconstruir la rejilla con los nuevos valores de $lsFO_1$, $lsFO_1$, $liFO_2$, $lsFO_2$;
8 Reinicializar todos los contadores de la rejilla en cero;
9 Agregar todos los individuos de la base de datos de élite al contador de su celda correspondiente;

Actualización de la tabla de superindividuo

La actualización de la tabla de superindividuo consiste en recorrer el conjunto de todos los individuos recién agregados a la BDe durante la generación actual y seleccionar el gen X_i con $i = 1, ..., 11$ que tenga mejor valor de desempeño. El superindividuo servirá para ordenar los genes de acuerdo a los mejores valores de desempeño con respecto a los canales de la banda 2.4GHz en una secuencia de escaneo. En el Algoritmo 6 se describen los pasos de la actualización.

4.3.6. Mutación

La mutación se realiza sobre cada uno de los parámetros de cada gen $X_i =< C_i, Min_i, Max_i, AP_i >$ con i = 1,...,11 de cada individuo de la población de tamaño **P** para generar un nuevo individuo. Este proceso dará lugar a una población de tamaño **2P**. Inicialmente se aplica una mutación Gaussiana para obtener el nuevo valor del parámetro de cada gen de un individuo a través de la expresión 4.5:

$$x'_i = x_i + N(m, \sigma) \tag{4.5}$$

En la expresión 4.5, x'_i corresponde al nuevo valor del parámetro y x_i corresponde al valor actual. $N(m, \sigma)$ es una variable aleatoria con una dis-

Algoritmo 6: Actualización de la tabla de superindividuo

Datos: Población **P**
Resultado: *Superindividuo*
Crear un individuo y etiquetarlo *superindividuo*;
Para *Todos los individuos i del conjunto de individuos recién agregados a la base de datos de individuos élite durante la generación actual* **hacer**
  Para *Todos los genes* $X_{i,j} =< C_{i,j}, Min_{i,j}, Max_{i,j}, AP_{i,j} >$ *de cada miembro i del conjunto de individuos recién agregados a la BDe durante la generación actual* **hacer**
    Seleccionar el mejor gen $< C_{1,j}, Min_{1,j}, Max_{1,j}, AP_{i,j} >$ de todos los j;
    Asignar el gen X_i en la posición j en el *superindividuo*;
  fin
fin

tribución normal de media m y desviación estándar σ. Para la mutación de los individuos se considera $m = 0$. En el caso de que el parámetro a mutar x_i corresponda a AP_i, se realiza una consulta al emulador de la topología como se describe en § 5.1. Es importante notar que esta mutación permite cambios confinados y graduales a los individuos, es decir alrededor de los valores anteriores en los parámetros de los genes.

4.3.7. Mutación dirigida

Como se describió en § 4.3, la ejecución del proceso de mutación está basado en el conocimiento dentro del espacio de creencias (**B**) (Fig. 4.1). El objeto de la mutación dirigida es orientar a los individuos mutados a generar secuencias de escaneo ordenadas con respecto a los canales que obtengan mejores valores de desempeño en la función objetivo seleccionada para la optimización multiobjetivo. Para tratar de mejorar los resultados del algoritmo cultural, se definió una probabilidad de ocurrencia de la mutación dirigida[5]. De esta manera el proceso de mutación en la evolución del algoritmo cultural utiliza la mutación (§ 4.3.6) o la mutación dirigida de acuerdo a dicha

[5]Esta probabilidad es un parámetro de entrada del algoritmo cultural. Todos los parámetros del algoritmo están descritos en § 4.3.10.

probabilidad.

El proceso de mutación dirigida utiliza el superindividuo almacenado en la tabla de superindividuos del espacio de creencias descrita en § 4.3.2 para obtener individuos de mejor calidad. El superindividuo mantiene la ventana de mutación W que incluye un número específico de genes que influirán en la generación de nuevos individuos (ver Fig. 4.6). En el Algoritmo 7 se muestran los pasos realizados en la mutación dirigida. En la línea 1 se realiza un lazo de repetición sobre cada uno de los individuos de la población **P**. En la línea 2 se crea un nuevo individuo. En la línea 3 se inicia un lazo de repetición sobre cada uno de los genes X_i que corresponden a la ventana de mutación W, parámetro del algoritmo cultural. En las líneas 4-8 se obtiene cada gen X_i de la ventana de mutación, su correspondiente canal C_i, la posición que ocupa este canal en el individuo I_i para luego sustituir el gen X_i de I_i en esa posición y finalmente asignar el gen X_i de la ventana de mutación W. De esta manera se han utilizado los genes de la ventana de mutación para orientar los genes hacia los canales que van reportando mejores valores de desempeño.

En las líneas 10-12 se realiza un lazo de repetición para los genes X_i restantes que no pertenecen a la ventana de mutación. En estos genes los temporizadores MinCT y MaxCT se modifican de acuerdo a la expresión 4.6:

$$x_i' = x_i + N(m, \sigma) \tag{4.6}$$

En expresión 4.6, x_i' corresponde al nuevo valor del temporizador y x_i corresponde al valor actual del temporizador. $N(m, \sigma)$ es una variable aleatoria con una distribución normal de media $m = 0$ y desviación estándar σ de MinCT y MaxCT respectivamente[6]. Luego se obtiene el nuevo valor de AP_i consultando al emulador de topología como se describe en § 5.1. Finalmente, en la línea 13 se agrega el individuo mutado a la población **P** para obtener una población de tamaño **2P** que pasará luego al proceso de filtrado a través de torneos para seleccionar los mejores.

[6]Esta desviaciones estándar son parámetros de entrada del algoritmo cultural y están descritas en § 4.3.10.

Algoritmo 7: Mutación dirigida sobre individuos de población **P**

Datos: Población **P**, tabla de *superindividuo*
Resultado: Población de tamaño **2P**

Para *Cada individuo i de la población* ***P*** **hacer**
  Crear un nuevo individuo I_i idéntico a i;
  Para *Cada gen $X_{i,i}$ de la ventana de mutación* ***W*** **hacer**
    Seleccionar el gen $X_{i,j}$ de la ventana de mutación **W**;
    Leer el canal C_i del gen $X_{i,j}$ de la ventana de mutación **W**;
    Encontrar la posición pos_i del gen $X_{i,j}$ cuyo valor de canal C_i sea igual al C_i de j en el individuo I_i;
    Sustituir el gen $X_{i,j}$ en la posición pos_i;
    Asignar el gen X_i j de la ventana de mutación **W** en j;
  fin
  Para *Cada gen $X_{i,j}$ que no pertenece a la ventana de mutación* ***W*** **hacer**
    Cambiar los temporizadores MinCT y MaxCT en el individuo I_i;
    Consultar al emulador de la topología para obtener el valor de AP_i;
  fin
  Agregar el individuo I_i a la población **P**;
fin

En la Fig. 4.8 se muestra un bosquejo del proceso de mutación dirigida. Para propósitos ilustrativos sólo se muestra la mutación de canales de acuerdo a la ventana de mutación **W**. De la ventana de mutación del superindividuo se toma el gen X_1 cuyo valor de canal es 1. En el individuo i a mutar se toma el primer gen X_1 cuyo valor de canal es 7. Se ubica en el individuo i la posición en la cual se encuentra el canal 1 del gen X_1 de la ventana de mutación, en este caso corresponde al gen X_4. En la posición de X_4 se asigna el gen X_1 cuyo valor de canal es 7, y en la posición de X_1 se asigna el gen X_1 de la ventana cuyo valor de canal es 1. Este proceso de ubicación y asignación de genes se repite con los restantes genes de la ventana de mutación X_2 y X_3.

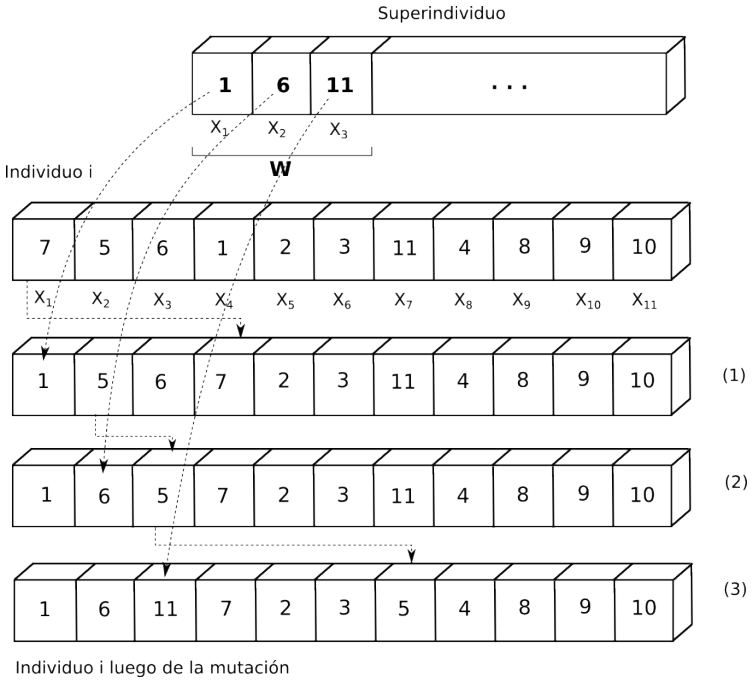


Fig. 4.8: Esquema general de la mutación dirigida

4.3.8. Filtrado de individuos por torneos

El proceso de torneos se lleva a cabo sobre la población de tamaño **2P** creada luego de la mutación. Cada individuo se enfrentará con otros C individuos[7] seleccionados aleatoriamente de la población principal. Se establecen dos reglas básicas para el torneo. En primer lugar, si un individuo domina al otro (ver § 2.4.4), entonces el individuo dominante se considera ganador. En segundo lugar, si ambos individuos son no comparables o sus valores de funciones objetivo son iguales, entonces: (a) estando ambos dentro de la rejilla del espacio de creencias, gana el torneo aquel que se encuentre en la celda menos poblada (de acuerdo al contador de las celdas), (b) de lo contrario, si un individuo cae fuera de la rejilla, entonces gana éste y se conserva ya que generará una nueva porción del frente de Pareto (para incrementar el espacio de exploración) al reconstruir la rejilla en la actualización de la parte

[7]El número de individuos a enfrentar en los torneos es un parámetro del algoritmo cultural.

normativa fenotípica del espacio de creencias. En el experimento 4 de § 5.3.1 se describen variaciones del criterio de selección del individuo ganador de un torneo entre individuos que son no comparables para observar su influencia en la obtención de secuencias de escaneo eficientes.

Una vez que se completan los torneos, los **P** individuos con el mayor número de victorias se seleccionan para convertirse en la nueva generación.

4.3.9. Inserción de individuos en la base de datos de individuos élite

La **BDe** mantiene solamente individuos no dominados. Para agregar un individuo a esta base de datos, si el individuo candidato (**IC**) es dominado por otro individuo existente en **BDe**, entonces el **IC** es descartado. Por consiguiente, todos los individuos que residen en la **BDe** corresponden a individuos no dominados entre ellos y esparcidos a lo largo del frente de Pareto.

Al final de la ejecución del algoritmo cultural la **BDe** tendrá los individuos no dominados encontrados durante todo el proceso evolutivo y presentados como resultado final a un tomador de decisiones. Estos individuos corresponden a las secuencias de escaneo optimizadas para el problema de optimización multiobjetivo.

4.3.10. Parámetros del algoritmo cultural

Los parámetros del algoritmo cultural se describen a continuación:

- Población inicial: especifica el número de individuos que conforman la población del algoritmo cultural. El número de individuos se mantiene constante durante el proceso de evolución; en la mutación se duplica la población pero en el proceso de filtrado de individuos se regresa al valor inicial.

- Máximo número de generaciones: especifica el número de generaciones durante las cuales el algoritmo se ejecuta.

- Desviación estándar para mutación Gausiana de MinCT de cada gen X_i de un individuo al momento de realizar la mutación.

- Desviación estándar para mutación Gausiana de MaxCT de cada gen X_i de un individuo al momento de realizar la mutación.

- Número de subintervalos de la rejilla: especifica el número de subintervalos en los cuales se divide la rejilla del espacio de creencias explicado en § 4.3.4.
- Frecuencia de actualización de la parte normativa fenotípica g_N: especifica el número de generaciones que transcurren para actualizar los límites de la funciones objetivo en la parte normativa fenotípica y disparar la reconstrucción de la rejilla del espacio de creencias.
- Torneos por individuo: especifica el número de torneos aleatorios que un individuo enfrentará en el proceso de filtrado para pasar a una nueva generación de individuos durante la evolución del algoritmo.
- Probabilidad de mutación dirigida: especifica la probabilidad de ocurrencia de la mutación dirigida, parte de la adaptación realizada al algoritmo cultural (ver § 4.3.7).
- Tamaño de la ventana de mutación (W): especifica el número de genes X_i de la tabla de superindividuo que se utilizan para ejercer influencia sobre un individuo en el proceso de mutación dirigida.

4.4. Conclusiones

En este capítulo se describió una técnica de inteligencia computacional para mejorar el proceso de escaneo. Se propone un modelo de optimización multiobjetivo que aborda el compromiso entre la maximización de la tasa de descubrimiento de puntos de acceso por unidad de tiempo y latencia del escaneo completo. Con este modelo se aplica un algoritmo cultural adaptado a partir del algoritmo propuesto por Coello y Becerra [33] para obtener secuencias de escaneo eficientes para el descubrimiento de redes IEEE 802.11.

Capítulo 5

Determinación de secuencias óptimas

5.1. Plataforma experimental

Para generar un individuo de la población **P** del algoritmo cultural, que representa un escaneo de un dispositivo, se consulta, desde la implementación del algoritmo cultural a un emulador de un modelo de topología inalámbrica. El emulador es resultado de una extensa campaña de medición de despliegues reales de puntos de acceso (AP) hecha en el centro de Rennes, Francia. Durante esa campaña, se almacenó información asociada a la topología del despliegue como el número total de APs descubiertos en distintos puntos a lo largo de una ruta, distribución de canales y retardos en respuestas de APs. Este emulador permite obtener el número de puntos de acceso (AP) que se encontrarían en el despliegue real descrito por Molina [41].

Para obtener el valor AP_i, que corresponde al número de puntos de acceso encontrados en el canal i, de cada gen $X_i =< C_i, Min_i, Max_i, AP_i >$ de un individuo (ver § 4.3.1), la implementación del algoritmo cultural realiza una consulta al emulador con los parámetros $< C_i, Min_i, Max_i >$ treinta (30) veces para encontrar un valor promedio de puntos de acceso. La consulta al emulador está representada por una función como la que se muestra en la expresión 5.1. Nuestra hipótesis consistió en reemplazar con una medida simple de todo un entorno la respuesta óptima dada una secuencia ineficiente.

$$AP_i = consultaTopologia(C_i, Min_i, Max_i) \tag{5.1}$$

El procedimiento general que realiza el emulador se inicia al indicar el canal a probar y el valor de los temporizadores MinCT y MaxCT. Se consulta el modelo de canales independientes formado con las funciones de distribución acumuladas producto de recolecciones reales. Se obtiene el número de APs encontrados para los temporizadores dados en el escaneo siempre que exista al menos un AP e incluyendo sólo aquellos con un retardo menor que el indicado por MaxCT. Si no hay resultados o no hay APs con un retardo menor que MinCT se declara que el canal está vacío. Con este procedimiento tanto la población **P** del algoritmo cultural como los resultados obtenidos en el algoritmo cultural están basados en despliegues reales de topologías IEEE 802.11.

5.2. Implementación del algoritmo cultural

El algoritmo cultural propuesto en § 4.3 se implementó usando el lenguaje de programación C++ y el marco de trabajo Qt[1]. Se utilizaron herramientas de software libre para el proceso de desarrollo, implementación y documentación.

5.2.1. Arquitectura del sistema

La implementación del algoritmo cultural se hizo siguiendo el paradigma de programación orientada a objetos. En la Fig. 5.1 se representa la arquitectura del sistema a través de un diagrama de componentes. Los componentes del sistema se describen a continuación:

- Interfaz web de emulador: componente de prueba del emulador de escaneo para usuarios finales (interfaz mínima para ejecutar scripts remotos y salida en formato JSON) y puede ser utilizado en otros sistemas para apoyar nuevas investigaciones sobre escaneo de redes IEEE 802.11.
- Emulador de escaneo: componente que permite emular el comportamiento del proceso de escaneo en redes IEEE 802.11 y que constituye parte importante de la plataforma experimental. El emulador permite que el componente algoritmo cultural genere secuencias de escaneo cuyos valores de puntos de acceso encontrados reflejen la topología inalámbrica descrita en [41].

[1] https://www.qt.io/

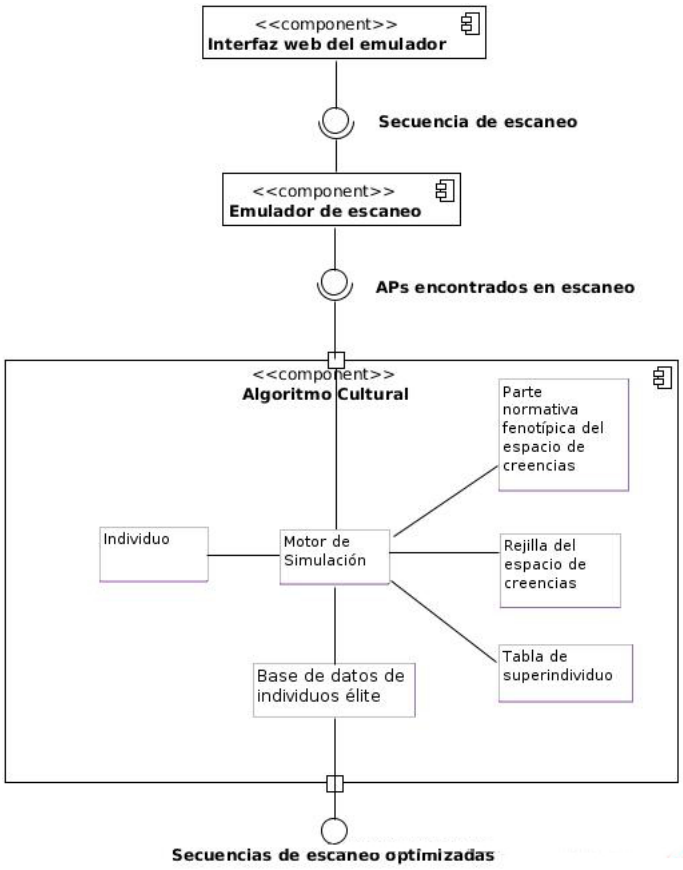


Fig. 5.1: Arquitectura del sistema que implementa el algoritmo cultural

- Algoritmo cultural: componente principal del sistema ya que en él se implementan las diferentes entidades descritas en § 4.3 que abstraen el proceso de evolución del algoritmo cultural. Se incluyen en este componente los individuos del algoritmo cultural que representan una secuencia de escaneo, la parte normativa fenotípica, la rejilla y la tabla de superindividuo del espacio de creencias, así como un motor de simulación que ejecuta la evolución del algoritmo a través de las generaciones.

5.2.2. Interfaz de emulador de escaneo

Un escaneo representa una secuencia de canales que prueba un dispositivo móvil con una interfaz IEEE 802.11 para encontrar los APs cercanos a su entorno. Se construyó entonces una aplicación web que permite interactuar con el emulador de escaneo descrito en § 5.1.

Inicialmente el emulador responde a consultas realizadas como se muestra en la expresión 5.1 para obtener el número de APs, sin embargo, la interacción de un usuario con el emulador se realiza a través de una interfaz de la línea de comandos en un sistema operativo. Para proporcionar una interacción más versátil se implementó un proceso de consulta al emulador que simula una secuencia de escaneo con sus respectivos temporizadores MinCT y MaxCT y de distintos tamaños, es decir se consultan secuencias de escaneo que van desde un solo canal hasta los once canales de la banda 2.4GHz. La interfaz web puede ser utilizada por otros trabajos de investigación que requieran obtener valores experimentales de despliegues reales levantados según la técnica descrita por Molina en [41]. Eventualmente es posible proveer las consultas al emulador a través de recursos de un servicio web.

La aplicación web desarrollada está disponible en una URL[2] accesible desde la Internet. En la Fig. 5.2 se muestra una captura de la pantalla de la interfaz gráfica desarrollada para el emulador de escaneo. Los parámetros de entrada para emular un escaneo son los siguientes:

- Tamaño de la secuencia: establece el número de canales que tendrá la secuencia que se desea emular.

- Número de repeticiones: establece el número de consultas que se realizan al emulador de la topología para obtener el valor promedio de APs descubiertos por canal.

[2] http://150.185.138.59/emulator/emulator.php

- Secuencia: establece la secuencia que se desea emular.

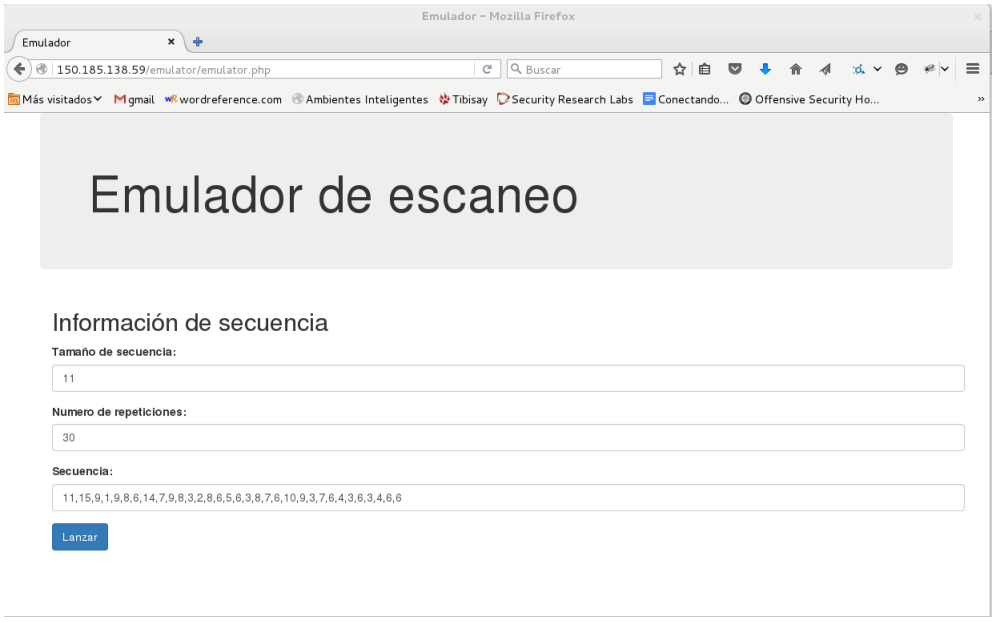


Fig. 5.2: Interfaz gráfica del emulador de escaneo

Un resultado de ejemplo de emulación de una secuencia de escaneo se muestra en el Fig. 5.3. De acuerdo al tamaño de la secuencia es posible identificar el valor de descubierta para cada canal así como los respectivos valores de función objetivos descritos en 4.2.2.

En el algoritmo 8 se muestran los pasos realizados en el emulador durante un proceso de búsqueda de puntos de acceso a partir de una secuencia de escaneo dada.

Entre las líneas 1 la 6 se inicializan contadores para registrar el tiempo entre respuestas, el número de respuesta encontrada en un mismo canal, el número de puntos de acceso encontrados, el tiempo que se acumula mientras se reciben respuestas en un canal, así como el tiempo total que transcurre mientras se escanea un canal (suma de los temporizadores MinCT y MaxCT). A partir de la línea 7 se realiza un lazo de repetición para cada uno de los canales, con sus respectivos temporizadores, que forman parte de una secuencia de escaneo dada. Entre las líneas 9 y 32 se ejecuta un lazo de repetición en el que se consulta la función de distribución acumulada (CDF por sus

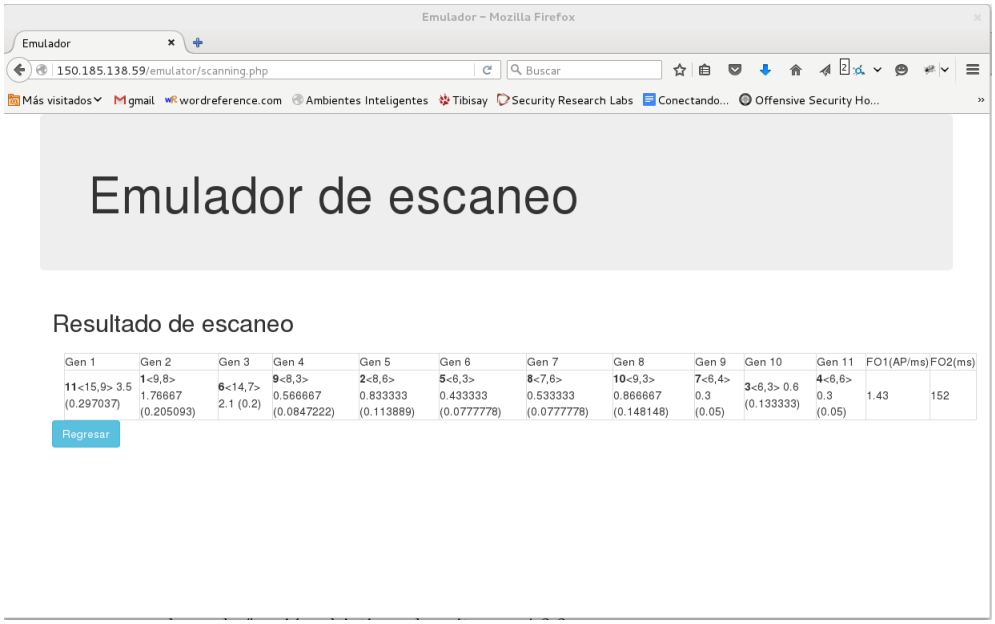


Fig. 5.3: Un resultado de consulta al emulador de escaneo

siglas en inglés) de los tiempos entre respuestas por cada canal de acuerdo al procedimiento descrito por Molina en [41]. En la línea 9 se obtiene el tiempo entre las primeras respuestas, segundas respuestas y sucesivas respuestas para un canal particular. En caso de que no se obtenga un tiempo válido se rompe el lazo de repetición de las respuestas y se pasa al siguiente canal sin contabilizar ningún punto de acceso encontrado.

A partir de la línea 14 se acumula el tiempo entre respuestas recibidas de puntos acceso en un canal dado y se verifica si éste tiempo es mayor que el temporizador MinCT del canal dado. Si se cumple esta condición se verifica que no sea la primera respuesta recibida en el canal, pues estaría violando el hecho de que la respuesta sea recibida antes de que el MinCT (tiempo mínimo de espera para recibir una respuesta de un punto de acceso antes de cambiar de canal) expire. En caso de que la condición se cumpla, se verifica que ahora el tiempo acumulado no exceda la suma de los temporizadores MinCT y MaxCT. En este caso se procede a registrar un punto de acceso encontrado en el canal dado y se pasa a la siguiente respuesta en el canal. En la línea 27 se verifica que el tiempo acumulado es menor que el MinCT por lo tanto se registra un nuevo punto de acceso encontrado en el canal y se

pasa a la siguiente respuesta. Finalmente en la línea 35 se retorna el número total de puntos de acceso registrados luego de consultar todas las respuestas del emulador en los canales de la secuencia dada.

Algoritmo 8: Algoritmo de búsqueda de puntos de acceso en el emulador

Datos: Secuencia de escaneo, función de distribución acumulada (CDF) de los tiempos entre respuestas de los 11 canales de la banda 2.4GHz.

Resultado: Número de puntos de acceso encontrados para la secuencia dada.

```
1  Inicializar contador tiempo_entre_respuestas en cero (0);
2  Inicializar contador número_de_respuesta en uno (1);
3  Inicializar contador puntos_de_acceso_encontrados en cero (0);
4  Inicializar contador tiempo_acumulado en cero (0);
5  Inicializar contador tiempo_total en cero (0);
6  Establecer bandera_de_primera_iteración en verdadero;
7  Para Cada canal de la secuencia de escaneo hacer
8      tiempo_total = MinCT + MaxCT;
9      mientras verdadero hacer
10         tiempo_entre_respuestas = CDF(canal, número_de_respuesta);
11         si tiempo_entre_respuestas == -1 entonces
12             Romper el lazo repita mientras;
13         fin
14         tiempo_acumulado = tiempo_acumulado +
           tiempo_entre_respuestas;
15         si tiempo_acumulado > MinCT entonces
16             si bandera_de_primera_iteración es verdadero entonces
17                 Romper el lazo repita mientras;
18             en otro caso
19                 si tiempo_acumulado ≤ tiempo_total entonces
20                     puntos_de_acceso_encontrados + + ;
21                     número_de_respuesta + +;
22                     Asignar bandera_de_primera_iteración en falso;
23                 en otro caso
24                     Romper el lazo repita mientras;
25                 fin
26             fin
27         en otro caso
28             puntos_de_acceso_encontrados + +;
29             número_de_respuesta + +;
30             Asignar bandera_de_primera_iteración en falso;
31         fin
32     fin
33     Asignar bandera_de_primera_iteración en verdadero;
34 fin
35 Retornar puntos_de_acceso_encontrados;
```

5.2.3. Ejecuciones del algoritmo

Para evaluar el desempeño del algoritmo cultural propuesto en 4.3, éste se ejecutó sobre un computador con las siguientes características: Intel©Core i5 @ 650 3.20GHz x 4CPU, 4 GB de memoria RAM y sistema operativo Debian GNU/Linux[3] de 64 bits.

5.2.4. Parámetros de ejecución del algoritmo

Para las ejecuciones del algoritmo se utilizaron los valores presentados en la tabla 5.1. Los valores utilizados se proponen de acuerdo a las observaciones luego de ejecuciones de prueba y sobre la base de los resultados presentados en [33].

Tabla 5.1: Valores de parámetros del algoritmo cultural

Parámetro	Valor
Población inicial (**P**)	20
Máximo número de generaciones	200
Desviación estándar para mutación Gausiana de MinCT	1
Desviación estándar para mutación Gausiana de MaxCT	3
Número de subintervalos de la rejilla	10
Frecuencia de actualización de la parte normativa fenotípica g_N	5
Torneos por individuo	10
Probabilidad de mutación dirigida	0.7
Tamaño de la ventana de mutación (W)	3

5.3. Experimentación

A partir de la implementación del algoritmo cultural descrito en la § 4.3, se diseñaron cinco experimentos sobre el algoritmo cultural que se describen a continuación.

[3]https://www.debian.org/

5.3.1. Descripción de experimentos

1. Experimento 1. Se generan 20 individuos con parámetros aleatorios para crear la población inicial del algoritmo cultural. En este caso se utilizaron individuos generados aleatoriamente dentro de los posibles valores para cada parámetro del individuo como se describe en § 4.3.1.

2. Experimento 2. Se generan 10 individuos con parámetros aleatorios y 10 individuos con parámetros predefinidos o inteligentes para crear la población inicial del algoritmo cultural. En este caso se utilizaron individuos generados aleatoriamente dentro de los posibles valores para cada parámetro del individuo como se describe en § 4.3.1 y también se utilizó un individuo predefinido o inteligente observado de los resultados experimentales mostrados en § 3.4.3.

3. Experimento 3. Se varía el número de individuos generados, aleatoriamente y a partir de individuos inteligentes, para crear la población inicial del algoritmo cultural. En este caso se generan 15 individuos aleatoriamente y 5 individuos predefinidos o inteligentes observado de los resultados experimentales mostrados en § 3.4.3.

4. Experimento 4. Se varía el criterio de selección de individuos durante la ejecución de un torneo entre individuos que son no comparables para formar las nuevas generaciones del algoritmo cultural (ver § 4.3.8). En este caso, los individuos que en el proceso de torneo son no comparables; esto es, que en un par de individuos que se enfrentan ninguno domina al otro o tienen valores de funciones objetivo iguales, utilizan como criterio de selección uno de los siguientes:

   a) Gana el torneo el individuo que posea una mayor proporción de puntos de acceso descubiertos de acuerdo a la expresión 5.2.

$$\sum_{i=1}^{11} Nmin_{C_i}/MinCT_{C_i} + Nmax_{C_i}/MaxCT_{C_i} \qquad (5.2)$$

   Donde: $Nmin_{C_i}$ y $Nmax_{C_i}$ corresponden al número total de APs descubiertos en el canal i, con los temporizadores MinCT y MaxCT respectivamente.

b) Gana el torneo el individuo que posea una mayor proporción de puntos de acceso descubiertos durante el periodo inicial (expresión 5.3).

$$\sum_{i=1}^{11} Nmin_{C_i}/MinCT_{C_i} \tag{5.3}$$

Donde: $Nmin_{C_i}$ corresponde al número total de APs descubiertos en el canal i, con el temporizador MinCT.

c) Gana el torneo el individuo que posea el mayor número de puntos de acceso descubiertos como se muestra en la expresión 5.4, siempre que la latencia del individuo sea menor que la latencia del individuo predefinido o inteligente.

$$\sum_{i=1}^{11} AP_i \tag{5.4}$$

Donde: AP_i corresponde al número de puntos de acceso encontrados por el individuo con sus parámetros.

5. Experimento 5. Se varía la influencia de la mutación dirigida a partir de la tabla de superindividuos del espacio de creencias (sección 4.3.7). En este caso, cuando un individuo se va a mutar se utiliza uno de los siguientes criterios:

   a) Mutar los primeros genes de un individuo que corresponden al tamaño de la ventana **W** de la tabla de superindividuos del espacio de creencias y dejar los restantes genes sin mutarlos.

   b) Mutar los primeros genes de un individuo que corresponden al tamaño de la ventana **W** de la tabla de superindividuos del espacio de creencias y mutar los restantes genes del individuo con una distribución normal. En este caso, la distribución normal utiliza como media el valor original del parámetro del individuo que se va a mutar, y como desviación estándar la desviación estándar para mutación gausiana de MinCT o MaxCT tal como se describe en la tabla 5.1.

5.4. Resultados de la experimentación

5.4.1. Comparación de secuencias óptimas derivadas del algoritmo cultural

El algoritmo cultural genera un conjunto de secuencias a lo largo del frente de Pareto que pueden ser utilizadas de acuerdo a las necesidades de los usuarios. Estas secuencias se comparan con otras que han sido observadas a través de un proceso de ingeniería inversa a partir de dispositivos con interfaces inalámbricas mostrando ventajas del enfoque propuesto.

En § 3.4.3 se presentaron los resultados experimentales para el tiempo de duración promedio del proceso de escaneo para sistemas operativos de distintos dispositivos móviles. Para establecer una comparación entre las secuencias óptimas derivadas del algoritmo cultural y las secuencias configuradas en los dispositivos se seleccionó como referencia el sistema operativo iOS©versión 6.1.2 [24].

En los resultados experimentales se observó que el iOS©realiza un escaneo de forma secuencial y que dura en promedio 429ms; esto representa 39ms que el dispositivo espera aproximadamente por cada uno de los 11 canales de la banda 2.4GHz. La tabla 5.2 muestra la estructura de la cadena de referencia en la que se refleja la secuencia de canales del 1 al 11, el $MinCT = 39$ y el $MaxCT = 0$ como parámetros.

Tabla 5.2: Individuo de referencia de secuencia de escaneo de iOS©

$< X_1 >< X_2 >< X_3 >< X_4 >< X_5 >< X_6 >< X_7 >< X_8 >< X_9 >< X_{10} >< X_{11} >< FO_1, FO_2 >$
<1,39,0,-><2,39,0,-><3,39,0,-><4,39,0,-><5,39,0,-><6,39,0,-><7,39,0,-><8,39,0,-><9,39,0,-><10,39,0,-><11,39,0,-><-,->

El algoritmo cultural se ejecutó con los parámetros descritos en § 5.2.3 y se observó una variedad de secuencias de escaneo cuyas tasas de descubrimiento se muestran en la Fig. 5.4. Es posible verificar que todas las secuencias tienen una tasa de descubrimiento considerablemente mayor comparadas con la secuencia de referencia que representan mejoras en el desempeño del descubrimiento entre 230 % y 600 %. Adicionalmente se observa que en los canales usualmente más poblados se registra un mejor rendimiento de descubierta que en los canales menos poblados. Con respecto a la latencia de escaneo se observaron mejoras que van entre el 30 % y 60 %.

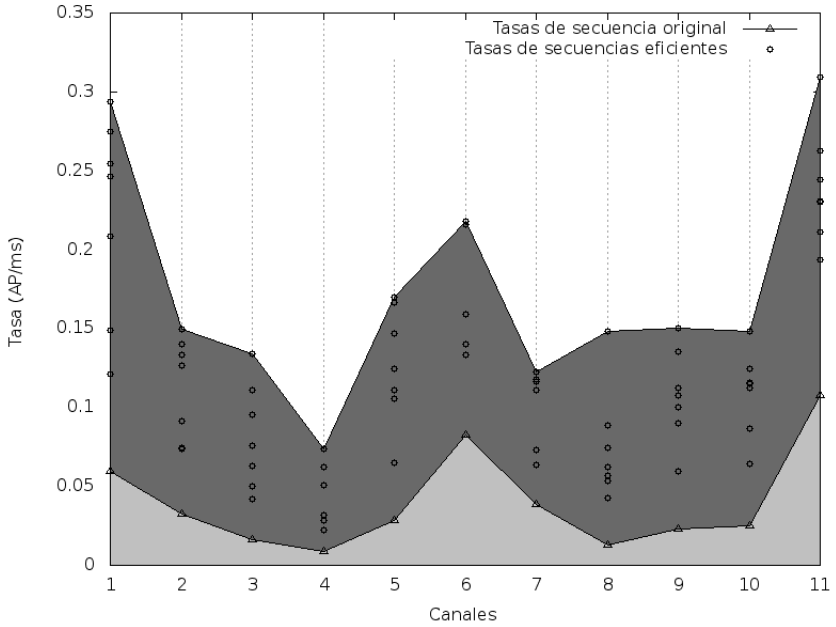


Fig. 5.4: Comparación de tasas de descubrimiento para secuencias eficientes del algoritmo cultural

5.4.2. Secuencias de escaneo obtenidas en los experimentos

Luego de la ejecución del algoritmo cultural se obtuvieron individuos no dominados que representan secuencias de escaneo con valores optimizados para los parámetros MinCT y MaxCT para cada uno de los experimentos descritos en § 5.3.1.

Para reportar los resultados de cada experimento de ejecución del algoritmo cultural se presentan dos tablas. La primera tabla muestra las secuencias de escaneo optimizadas para el experimento a través de tres columnas. En la primera columna, se muestra la secuencia de canales en la que cada canal está representado como se muestra en la expresión 5.5:

$$\underset{\substack{<m,M>AP \\ D}}{\mathbf{Ch}^{l}} \tag{5.5}$$

donde:

- Ch: es un canal de la secuencia (entre 1 y 11).
- l: es la latencia obtenida como $MinCT + MaxCT$ acumulada de un canal y los anteriores.
- m: es el parámetro MinCT.
- M: es el parámetro MaxCT.
- AP: es el número de puntos de acceso encontrados en el canal Ch.
- D: es el número de puntos de acceso encontrados en el canal Ch por unidad de tiempo.

El número de puntos de acceso encontrados corresponde al promedio de puntos de acceso encontrados luego de ejecutar 30 escaneos independientes sobre el emulador descrito en § 5.1. En la segunda y tercera columna de la tabla se presentan los valores de las funciones objetivo del modelo de optimización multiobjetivo descrito en § 4.2.2.

En la segunda tabla se reportan los intervalos de confianza del 95 % para el número de puntos de acceso (AP) encontrados por cada secuencia de escaneo generada por el algoritmo cultural. En esta tabla también se incluyen los intervalos de confianza del 95 % para el número de puntos de acceso (AP) encontrados por las secuencias de escaneo identificadas de distintos dispositivos a partir de un proceso de ingeniería inversa como el que se describe en § 3.4.1. En las tablas 5.3 a la 5.17 se muestran las secuencias de escaneo resultantes para los distintos experimentos realizados.

5.4.3. Discusión de resultados

De los resultados de los experimentos es importante notar que el orden de los canales es diferente para cada secuencia. Esto se debe a que el algoritmo cultural ha adaptado la velocidad del canal de acuerdo a la respuesta del sistema en cada canal diferente, encontrando así la máxima tasa de puntos de acceso útiles por unidad de tiempo. Una de las características relevantes de este trabajo es que se aborda la compensación entre el número de puntos de acceso encontrados por unidad de tiempo a diferencia de otros estudios reportados que utilizan el número de puntos de acceso totales encontrados como reportan Montavont et al. [34].

Tabla 5.3: Secuencias de escaneo optimizadas del experimento 1.

Canales de la secuencia											FO_1 (AP/ms)	FO_2 (ms)
$\mathbf{11}^{15}$	$\mathbf{6}^{32}$	$\mathbf{10}^{43}$	$\mathbf{1}^{52}$	$\mathbf{5}^{65}$	$\mathbf{9}^{80}$	$\mathbf{2}^{95}$	$\mathbf{7}^{106}$	$\mathbf{8}^{115}$	$\mathbf{3}^{127}$	$\mathbf{4}^{137}$	1.18429	137
<11,4>2.33333	<10,7>1.06667	<8,3>0.766667	<6,3>0.733333	<10,3>0.6	<6,9>0.533333	<11,4>0.5	<6,5>0.366667	<6,3>0.266667	<7,5>0.233333	<7,3>0.133333		
0.29697	0.11381	0.109722	0.15	0.0744444	0.135185	0.0825758	0.0611111	0.0611111	0.067619	0.031746		
$\mathbf{11}^{15}$	$\mathbf{6}^{33}$	$\mathbf{1}^{42}$	$\mathbf{10}^{59}$	$\mathbf{2}^{70}$	$\mathbf{7}^{89}$	$\mathbf{9}^{101}$	$\mathbf{5}^{112}$	$\mathbf{3}^{124}$	$\mathbf{8}^{133}$	$\mathbf{4}^{147}$	1.35072	147
<11,4>2.26667	<15,3>1.46667	<6,3>1.03333	<9,8>0.8	<8,3>0.733333	<14,5>0.7	<7,5>0.666667	<8,3>0.4	<9,3>0.4	<6,3>0.3	<9,5>0.166667		
0.253788	0.186667	0.238889	0.0888889	0.133333	0.0680952	0.104762	0.111111	0.0740741	0.0666667	0.0244444		
$\mathbf{11}^{12}$	$\mathbf{1}^{31}$	$\mathbf{10}^{47}$	$\mathbf{7}^{61}$	$\mathbf{5}^{74}$	$\mathbf{9}^{87}$	$\mathbf{8}^{101}$	$\mathbf{6}^{112}$	$\mathbf{2}^{121}$	$\mathbf{4}^{135}$	$\mathbf{3}^{152}$	1.3714	152
<8,4>2.26667	<14,5>2.03333	<10,6>1.2	<11,3>0.766667	<10,3>0.6	<10,3>0.6	<11,3>0.6	<6,5>0.4	<6,3>0.3	<11,3>0.266667	<14,3>0.266667		
0.404167	0.192381	0.157778	0.10202	0.06	0.0988889	0.0787879	0.0677778	0.0611111	0.0484848	0.1		
$\mathbf{6}^{22}$	$\mathbf{11}^{37}$	$\mathbf{1}^{53}$	$\mathbf{10}^{72}$	$\mathbf{5}^{94}$	$\mathbf{8}^{118}$	$\mathbf{9}^{133}$	$\mathbf{7}^{144}$	$\mathbf{3}^{159}$	$\mathbf{2}^{168}$	$\mathbf{4}^{182}$	1.66348	182
<15,7>2.26667	<12,3>2.2	<13,3>2.03333	<15,4>1.16667	<12,10>1.1	<12,12>0.7	<9,6>0.666667	<8,3>0.633333	<12,3>0.5	<6,3>0.266667	<11,3>0.133333		
0.224762	0.375	0.318803	0.2	0.100556	0.0583333	0.0759259	0.127778	0.0666667	0.0611111	0.0545455		

Tabla 5.4: Intervalos de confianza del 95 % de número de AP encontrados por secuencias de escaneo optimizadas del experimento 1.

Cadena	**APs**	**Min**	**Max**	**Latencia (ms)**
cadena1	7.83333	6.83531	8.83136	137
cadena2	8.7	7.73557	9.66443	147
cadena3	8.83333	7.96702	9.69964	152
cadena4	10.1667	9.42859	10.9047	182
Debian	7.633333	6.810066	8.456600	113
Windows	10.63333	9.764601	11.50206	168
Meego	10.7	9.607376	11.79262	184
iOS	16.4	14.96683	17.83317	429
Android	18.83333	17.60040	20.06627	601

Tabla 5.5: Secuencias de escaneo optimizadas del experimento 2.

Canales de la secuencia											FO_1 (AP/ms)	FO_2 (ms)
$\mathbf{11}^{39}$	$\mathbf{6}^{78}$	$\mathbf{1}^{117}$	$\mathbf{7}^{156}$	$\mathbf{10}^{195}$	$\mathbf{5}^{234}$	$\mathbf{2}^{273}$	$\mathbf{8}^{312}$	$\mathbf{3}^{351}$	$\mathbf{9}^{390}$	$\mathbf{4}^{429}$	0.484615	429
<39,0>4.46667	<39,0>3.9	<39,0>3.33333	<39,0>1.66667	<39,0>1.3	<39,0>1.16667	<39,0>0.833333	<39,0>0.7	<39,0>0.633333	<39,0>0.566667	<39,0>0.333333		
0.11453	0.1	0.0854701	0.042735	0.0333333	0.0299145	0.0213675	0.0179487	0.0162393	0.0145299	0.00854701		
$\mathbf{11}^{14}$	$\mathbf{1}^{33}$	$\mathbf{6}^{46}$	$\mathbf{9}^{64}$	$\mathbf{7}^{82}$	$\mathbf{2}^{96}$	$\mathbf{10}^{113}$	$\mathbf{5}^{128}$	$\mathbf{8}^{146}$	$\mathbf{3}^{155}$	$\mathbf{4}^{164}$	1.23989	164
<11,3>1.8	<15,4>1.6	<10,3>0.833333	<15,3>0.733333	<15,3>0.633333	<11,3>0.6	<8,9>0.566667	<12,3>0.466667	<11,7>0.4	<6,3>0.266667	<6,3>0.1		
0.248485	0.138333	0.161111	0.0755556	0.148889	0.0868687	0.0787037	0.177778	0.0519481	0.0444444	0.0277778		
$\mathbf{1}^{22}$	$\mathbf{11}^{39}$	$\mathbf{6}^{56}$	$\mathbf{7}^{72}$	$\mathbf{10}^{86}$	$\mathbf{9}^{103}$	$\mathbf{2}^{112}$	$\mathbf{3}^{124}$	$\mathbf{5}^{134}$	$\mathbf{8}^{153}$	$\mathbf{4}^{177}$	1.44941	177
<14,8>2.43333	<14,3>2.16667	<14,3>1.53333	<12,4>0.933333	<7,7>0.833333	<14,3>0.8	<6,3>0.433333	<9,3>0.433333	<7,3>0.4	<12,7>0.366667	<13,11>0.2		
0.214881	0.289683	0.188095	0.111111	0.119048	0.065873	0.105556	0.062963	0.152381	0.090873	0.048951		
$\mathbf{11}^{17}$	$\mathbf{6}^{35}$	$\mathbf{1}^{56}$	$\mathbf{9}^{70}$	$\mathbf{5}^{85}$	$\mathbf{3}^{108}$	$\mathbf{7}^{123}$	$\mathbf{10}^{134}$	$\mathbf{2}^{153}$	$\mathbf{8}^{166}$	$\mathbf{4}^{179}$	1.66155	179
<14,3>2.73333	<15,3>1.96667	<12,9>1.83333	<11,3>0.8	<12,3>0.733333	<13,10>0.666667	<12,3>0.633333	<8,3>0.566667	<15,4>0.466667	<10,3>0.366667	<10,3>0.033333		
0.247619	0.335556	0.166667	0.129293	0.130556	0.054359	0.0861111	0.223611	0.0311111	0.21	0.0466667		

Tabla 5.6: Intervalos de confianza del 95 % de número de AP encontrados por secuencias de escaneo optimizadas del experimento 2.

Cadena	**APs**	**Min**	**Max**	**Latencia (ms)**
indBase	18.6	17.5143	19.6857	429
cadena1	10.8333	9.87258	11.7941	164
cadena2	11.1333	10.0783	12.1884	177
cadena3	11.8333	10.8026	12.8641	179
Debian	7.633333	6.810066	8.456600	113
Windows	10.63333	9.764601	11.50206	168
Meego	10.7	9.607376	11.79262	184
iOS	16.4	14.96683	17.83317	429
Android	18.83333	17.60040	20.06627	601

Tabla 5.7: Secuencias de escaneo optimizadas del experimento 3.

Canales de la secuencia											FO_1 (AP/ms)	FO_2 (ms)
$\mathbf{11}^{39}$	$\mathbf{6}^{78}$	$\mathbf{1}^{117}$	$\mathbf{7}^{156}$	$\mathbf{5}^{195}$	$\mathbf{10}^{234}$	$\mathbf{2}^{273}$	$\mathbf{9}^{312}$	$\mathbf{8}^{351}$	$\mathbf{3}^{390}$	$\mathbf{4}^{429}$	0.462393	429
<39,0>4.23333	<39,0>3.86667	<39,0>2.6	<39,0>1.76667	<39,0>1.63333	<39,0>1.03333	<39,0>1.03333	<39,0>0.6	<39,0>0.566667	<39,0>0.466667	<39,0>0.233333		
0.108547	0.0991453	0.0666667	0.0452991	0.0418803	0.0264957	0.0264957	0.0153846	0.0145299	0.0119658	0.00598291		
$\mathbf{11}^{18}$	$\mathbf{1}^{33}$	$\mathbf{6}^{51}$	$\mathbf{10}^{75}$	$\mathbf{2}^{88}$	$\mathbf{5}^{99}$	$\mathbf{3}^{111}$	$\mathbf{9}^{122}$	$\mathbf{7}^{136}$	$\mathbf{8}^{153}$	$\mathbf{4}^{163}$	1.3046	159
<15,3>2.83333	<12,3>1.76667	<15,3>1.5	<15,9>0.966667	<6,7>0.633333	<8,3>0.533333	<9,3>0.533333	<8,3>0.466667	<10,4>0.466667	<14,3>0.433333	<6,4>0		
0.295556	0.280556	0.188889	0.0733333	0.0968254	0.0666667	0.111111	0.0888889	0.0583333	0.0444444	0		
$\mathbf{11}^{18}$	$\mathbf{1}^{32}$	$\mathbf{10}^{45}$	$\mathbf{6}^{57}$	$\mathbf{2}^{74}$	$\mathbf{9}^{94}$	$\mathbf{7}^{111}$	$\mathbf{5}^{124}$	$\mathbf{8}^{137}$	$\mathbf{3}^{155}$	$\mathbf{4}^{165}$	1.32265	161
<15,3>3.2	<9,5>1.33333	<10,3>1	<9,3>0.833333	<14,3>0.766667	<15,5>0.766667	<12,5>0.733333	<10,3>0.7	<6,7>0.5	<15,3>0.433333	<6,4>0		
0.311111	0.192593	0.177778	0.137037	0.0809524	0.108889	0.0766667	0.101111	0.0809524	0.0555556	0		
$\mathbf{11}^{16}$	$\mathbf{1}^{34}$	$\mathbf{6}^{59}$	$\mathbf{5}^{79}$	$\mathbf{10}^{97}$	$\mathbf{8}^{113}$	$\mathbf{9}^{126}$	$\mathbf{2}^{137}$	$\mathbf{7}^{153}$	$\mathbf{3}^{166}$	$\mathbf{4}^{186}$	1.65276	186
<12,4>2.63333	<15,3>2.26667	<12,13>2	<11,9>0.933333	<15,3>0.833333	<13,3>0.533333	<7,6>0.5	<7,4>0.466667	<13,3>0.333333	<9,4>0.3	<15,5>0.233333		
0.391667	0.284444	0.162393	0.0909091	0.162222	0.0666667	0.0753968	0.0880952	0.135043	0.153704	0.0422222		

Tabla 5.8: Intervalos de confianza del 95 % de número de AP encontrados por secuencias de escaneo optimizadas del experimento 3.

Cadena	**APs**	**Min**	**Max**	**Latencia (ms)**
indBase	17.6	16.4577	18.7423	429
cadena1	10.1333	9.2345	11.0322	159
cadena2	10.1333	9.18073	11.0859	161
cadena3	11.5333	10.3203	12.7464	186
Debian	7.633333	6.810066	8.456600	113
Windows	10.63333	9.764601	11.50206	168
Meego	10.7	9.607376	11.79262	184
iOS	16.4	14.96683	17.83317	429
Android	18.83333	17.60040	20.06627	601

Tabla 5.9: Secuencias de escaneo optimizadas del experimento 4.a.

Canales de la secuencia											FO_1 (AP/ms)	FO_2 (ms)
$\mathbf{11}^{39}$	$\mathbf{1}^{78}$	$\mathbf{6}^{117}$	$\mathbf{5}^{156}$	$\mathbf{7}^{195}$	$\mathbf{10}^{234}$	$\mathbf{2}^{273}$	$\mathbf{9}^{312}$	$\mathbf{3}^{351}$	$\mathbf{8}^{390}$	$\mathbf{4}^{429}$	0.449573	429
<39,0>4.3	<39,0>3.53333	<39,0>3.1	<39,0>1.6	<39,0>1.33333	<39,0>0.9	<39,0>0.8	<39,0>0.633333	<39,0>0.533333	<39,0>0.433333	<39,0>0.366667		
0.110256	0.0905983	0.0794872	0.0410256	0.034188	0.0230769	0.0205128	0.0162393	0.0136752	0.0111111	0.00940171		
$\mathbf{11}^{17}$	$\mathbf{1}^{28}$	$\mathbf{6}^{43}$	$\mathbf{10}^{54}$	$\mathbf{2}^{71}$	$\mathbf{3}^{87}$	$\mathbf{8}^{100}$	$\mathbf{9}^{110}$	$\mathbf{7}^{122}$	$\mathbf{5}^{135}$	$\mathbf{4}^{149}$	1.32193	149
<14,3>2.4	<8,3>1.13333	<12,3>0.966667	<8,3>0.833333	<11,6>0.766667	<7,9>0.666667	<7,6>0.533333	<7,3>0.433333	<9,3>0.433333	<8,5>0.366667	<11,3>0.3		
0.215079	0.225	0.136111	0.111111	0.079798	0.0878307	0.0833333	0.0936508	0.0925926	0.121667	0.0757576		
$\mathbf{11}^{18}$	$\mathbf{1}^{37}$	$\mathbf{6}^{55}$	$\mathbf{3}^{70}$	$\mathbf{5}^{94}$	$\mathbf{10}^{108}$	$\mathbf{2}^{121}$	$\mathbf{7}^{136}$	$\mathbf{9}^{148}$	$\mathbf{8}^{166}$	$\mathbf{4}^{176}$	1.60775	173
<15,3>2.8	<14,5>2	<15,3>1.96667	<12,3>0.9	<14,10>0.8	<11,3>0.566667	<10,3>0.533333	<6,9>0.5	<9,3>0.433333	<10,8>0.3	<7,3>0		
0.311111	0.177143	0.308889	0.225	0.0580952	0.079798	0.0766667	0.0740741	0.151852	0.0975	0.047619		
$\mathbf{11}^{18}$	$\mathbf{6}^{35}$	$\mathbf{1}^{50}$	$\mathbf{10}^{64}$	$\mathbf{9}^{79}$	$\mathbf{2}^{98}$	$\mathbf{5}^{116}$	$\mathbf{7}^{138}$	$\mathbf{8}^{156}$	$\mathbf{3}^{169}$	$\mathbf{4}^{187}$	1.66069	182
<15,3>2.83333	<14,3>1.6	<8,7>1.36667	<11,3>1.06667	<12,3>0.833333	<15,4>0.766667	<15,3>0.633333	<13,9>0.5	<15,3>0.466667	<9,4>0.233333	<13,5>0		
0.384444	0.149206	0.17381	0.137374	0.0944444	0.198889	0.131111	0.088604	0.0577778	0.134259	0.110769		

Tabla 5.10: Intervalos de confianza del 95 % de número de AP encontrados por secuencias de escaneo optimizadas del experimento 4.a.

Cadena	**APs**	**Min**	**Max**	**Latencia (ms)**
indBase	17.4	16.2433	18.5567	429
cadena1	10.4	9.33283	11.4672	149
cadena2	10.9333	9.75297	12.1137	173
cadena3	11.6333	10.5728	12.6939	182
Debian	7.633333	6.810066	8.456600	113
Windows	10.63333	9.764601	11.50206	168
Meego	10.7	9.607376	11.79262	184
iOS	16.4	14.96683	17.83317	429
Android	18.83333	17.60040	20.06627	601

Tabla 5.11: Secuencias de escaneo optimizadas del experimento 4.b.

Canales de la secuencia											FO_1 (AP/ms)	FO_2 (ms)
$\mathbf{11}^{39}$	$\mathbf{1}^{78}$	$\mathbf{6}^{117}$	$\mathbf{7}^{156}$	$\mathbf{10}^{195}$	$\mathbf{2}^{234}$	$\mathbf{5}^{273}$	$\mathbf{9}^{312}$	$\mathbf{3}^{351}$	$\mathbf{8}^{390}$	$\mathbf{4}^{429}$	0.475214	429
<39,0>4.56667	<39,0>3.66667	<39,0>2.96667	<39,0>1.93333	<39,0>1.33333	<39,0>1.03333	<39,0>1	<39,0>0.733333	<39,0>0.633333	<39,0>0.5	<39,0>0.166667		
0.117094	0.0940171	0.0760684	0.0495726	0.034188	0.0264957	0.025641	0.0188034	0.0162393	0.0128205	0.0042735		
$\mathbf{11}^{14}$	$\mathbf{2}^{28}$	$\mathbf{1}^{37}$	$\mathbf{10}^{46}$	$\mathbf{8}^{58}$	$\mathbf{6}^{67}$	$\mathbf{9}^{77}$	$\mathbf{7}^{89}$	$\mathbf{5}^{98}$	$\mathbf{3}^{109}$	$\mathbf{4}^{119}$	1.23451	119
<11,3>1.96667	<11,3>0.866667	<6,3>0.8	<6,3>0.566667	<9,3>0.533333	<6,3>0.466667	<7,3>0.466667	<9,3>0.466667	<6,3>0.333333	<8,3>0.333333	<7,3>0.1		
0.186869	0.167677	0.177778	0.105556	0.0888889	0.122222	0.130159	0.111111	0.0666667	0.0569444	0.0206349		
$\mathbf{1}^{15}$	$\mathbf{11}^{26}$	$\mathbf{6}^{38}$	$\mathbf{3}^{48}$	$\mathbf{10}^{59}$	$\mathbf{7}^{72}$	$\mathbf{9}^{81}$	$\mathbf{2}^{90}$	$\mathbf{8}^{102}$	$\mathbf{5}^{111}$	$\mathbf{4}^{120}$	1.57139	120
<10,5>1.96667	<8,3>1.76667	<9,3>0.866667	<6,4>0.7	<8,3>0.566667	<8,5>0.5	<6,3>0.4	<6,3>0.4	<9,3>0.4	<6,3>0.3	<6,3>0.1		
0.306667	0.269444	0.133333	0.147222	0.0861111	0.0675	0.116667	0.0888889	0.0888889	0.183333	0.0833333		

Tabla 5.12: Intervalos de confianza del 95 % de número de AP encontrados por secuencias de escaneo optimizadas del experimento 4.b.

Cadena	**APs**	**Min**	**Max**	**Latencia (ms)**
indBase	17.1667	15.939	18.3943	429
cadena1	6.63333	5.84567	7.421	119
cadena2	7.03333	6.0816	7.98506	120
Debian	7.633333	6.810066	8.456600	113
Windows	10.63333	9.764601	11.50206	168
Meego	10.7	9.607376	11.79262	184
iOS	16.4	14.96683	17.83317	429
Android	18.83333	17.60040	20.06627	601

Tabla 5.13: Secuencias de escaneo optimizadas del experimento 4.c.

Canales de la secuencia											FO_1 (AP/ms)	FO_2 (ms)
$\mathbf{11}^{39}$	$\mathbf{6}^{78}$	$\mathbf{1}^{117}$	$\mathbf{10}^{156}$	$\mathbf{5}^{195}$	$\mathbf{7}^{234}$	$\mathbf{2}^{273}$	$\mathbf{9}^{312}$	$\mathbf{3}^{351}$	$\mathbf{8}^{390}$	$\mathbf{4}^{429}$	0.442735	429
<39,0>4.33333	<39,0>3.33333	<39,0>2.83333	<39,0>1.53333	<39,0>1.36667	<39,0>1.06667	<39,0>0.933333	<39,0>0.733333	<39,0>0.533333	<39,0>0.533333	<39,0>0.0666667		
0.111111	0.0854701	0.0726496	0.0393162	0.0350427	0.0273504	0.0239316	0.0188034	0.0136752	0.0136752	0.0017094		
$\mathbf{11}^{9}$	$\mathbf{1}^{18}$	$\mathbf{10}^{29}$	$\mathbf{5}^{39}$	$\mathbf{8}^{52}$	$\mathbf{9}^{61}$	$\mathbf{2}^{70}$	$\mathbf{6}^{79}$	$\mathbf{7}^{89}$	$\mathbf{3}^{100}$	$\mathbf{4}^{114}$	1.32905	114
<6,3>1.13333	<6,3>0.9	<8,3>0.8	<7,3>0.733333	<10,3>0.733333	<6,3>0.566667	<6,3>0.433333	<6,3>0.433333	<7,3>0.3	<6,5>0.3	<8,6>0.0333333		
0.211111	0.161111	0.113889	0.130159	0.151111	0.133333	0.0777778	0.0888889	0.122222	0.0866667	0.0527778		
$\mathbf{11}^{10}$	$\mathbf{1}^{19}$	$\mathbf{10}^{30}$	$\mathbf{9}^{39}$	$\mathbf{7}^{51}$	$\mathbf{6}^{60}$	$\mathbf{2}^{74}$	$\mathbf{8}^{83}$	$\mathbf{3}^{94}$	$\mathbf{5}^{105}$	$\mathbf{4}^{117}$	1.49009	117
<6,4>1.83333	<6,3>1	<7,4>0.8	<6,3>0.766667	<9,3>0.666667	<6,3>0.566667	<8,6>0.466667	<6,3>0.333333	<6,5>0.233333	<7,4>0.233333	<7,5>0.166667		
0.383333	0.183333	0.114286	0.183333	0.125926	0.111111	0.0625	0.105556	0.04	0.15119	0.0295238		
$\mathbf{1}^{14}$	$\mathbf{11}^{29}$	$\mathbf{6}^{38}$	$\mathbf{10}^{47}$	$\mathbf{5}^{58}$	$\mathbf{7}^{67}$	$\mathbf{9}^{82}$	$\mathbf{3}^{91}$	$\mathbf{2}^{100}$	$\mathbf{8}^{109}$	$\mathbf{4}^{122}$	1.54874	122
<11,3>2.03333	<12,3>1.76667	<6,3>0.6	<6,3>0.6	<8,3>0.6	<6,3>0.533333	<7,8>0.533333	<6,3>0.366667	<6,3>0.366667	<6,3>0.333333	<6,7>0.133333		
0.362626	0.313889	0.105556	0.138889	0.109722	0.144444	0.0744048	0.111111	0.0944444	0.0722222	0.0214286		
$\mathbf{11}^{10}$	$\mathbf{6}^{29}$	$\mathbf{1}^{38}$	$\mathbf{5}^{49}$	$\mathbf{7}^{62}$	$\mathbf{3}^{72}$	$\mathbf{8}^{83}$	$\mathbf{9}^{93}$	$\mathbf{2}^{104}$	$\mathbf{10}^{115}$	$\mathbf{4}^{128}$	1.71921	128
<6,4>1.43333	<9,10>1.3	<6,3>1.2	<8,3>0.933333	<10,3>0.766667	<7,3>0.7	<8,3>0.666667	<7,3>0.533333	<8,3>0.466667	<8,3>0.4	<10,3>0.0666667		
0.247222	0.138889	0.255556	0.186111	0.201111	0.131746	0.145833	0.139683	0.0652778	0.172222	0.0355556		

Tabla 5.14: Intervalos de confianza del 95 % de número de AP encontrados por secuencias de escaneo optimizadas del experimento 4.c.

Cadena	**APs**	**Min**	**Max**	**Latencia (ms)**
indBase	17.9	16.3622	19.4378	429
cadena1	5.53333	4.54315	6.52351	114
cadena2	6.43333	5.47051	7.39616	117
cadena3	8.63333	7.83313	9.43354	122
cadena4	7.23333	6.17214	8.29453	128
Debian	7.633333	6.810066	8.456600	113
Windows	10.63333	9.764601	11.50206	168
Meego	10.7	9.607376	11.79262	184
iOS	16.4	14.96683	17.83317	429
Android	18.83333	17.60040	20.06627	601

Tabla 5.15: Secuencias de escaneo optimizadas del experimento 5.a.

Canales de la secuencia											FO_1 (AP/ms)	FO_2 (ms)
$\mathbf{11}^{39}$	$\mathbf{6}^{78}$	$\mathbf{1}^{117}$	$\mathbf{5}^{156}$	$\mathbf{10}^{195}$	$\mathbf{7}^{234}$	$\mathbf{9}^{273}$	$\mathbf{2}^{312}$	$\mathbf{3}^{351}$	$\mathbf{8}^{390}$	$\mathbf{4}^{429}$	0.435897	429
<39,0>4.1	<39,0>3.63333	<39,0>2.73333	<39,0>1.53333	<39,0>1.2	<39,0>1.13333	<39,0>0.733333	<39,0>0.7	<39,0>0.466667	<39,0>0.466667	<39,0>0.3		
0.105128	0.0931624	0.0700855	0.0393162	0.0307692	0.0290598	0.0188034	0.0179487	0.0119658	0.0119658	0.00769231		
$\mathbf{11}^{11}$	$\mathbf{9}^{21}$	$\mathbf{1}^{31}$	$\mathbf{2}^{40}$	$\mathbf{6}^{49}$	$\mathbf{3}^{58}$	$\mathbf{10}^{67}$	$\mathbf{8}^{78}$	$\mathbf{5}^{87}$	$\mathbf{7}^{97}$	$\mathbf{4}^{106}$	1.29702	106
<8,3>1.7	<7,3>0.733333	<7,3>0.633333	<6,3>0.466667	<6,3>0.466667	<6,3>0.333333	<6,3>0.3	<8,3>0.3	<6,3>0.266667	<7,3>0.266667	<6,3>0.0333333		
0.281944	0.11746	0.0904762	0.0888889	0.227778	0.105556	0.133333	0.0833333	0.0611111	0.101587	0.00555556		
$\mathbf{11}^{11}$	$\mathbf{1}^{21}$	$\mathbf{8}^{33}$	$\mathbf{9}^{43}$	$\mathbf{7}^{52}$	$\mathbf{2}^{61}$	$\mathbf{10}^{70}$	$\mathbf{6}^{79}$	$\mathbf{3}^{88}$	$\mathbf{5}^{97}$	$\mathbf{4}^{108}$	1.42464	108
<8,3>1.56667	<7,3>1.23333	<7,5>0.7	<7,3>0.6	<6,3>0.566667	<6,3>0.466667	<6,3>0.433333	<6,3>0.4	<6,3>0.3	<6,3>0.3	<8,3>0.166667		
0.244444	0.25873	0.122857	0.111111	0.15	0.0888889	0.0722222	0.0888889	0.2	0.0666667	0.0208333		
$\mathbf{11}^{11}$	$\mathbf{1}^{22}$	$\mathbf{6}^{34}$	$\mathbf{9}^{43}$	$\mathbf{7}^{53}$	$\mathbf{8}^{63}$	$\mathbf{3}^{75}$	$\mathbf{2}^{85}$	$\mathbf{10}^{94}$	$\mathbf{5}^{104}$	$\mathbf{4}^{118}$	1.84917	118
<8,3>2.13333	<8,3>0.966667	<9,3>0.833333	<6,3>0.766667	<7,3>0.666667	<7,3>0.6	<9,3>0.566667	<7,3>0.466667	<6,3>0.433333	<7,3>0.3	<11,3>0.266667		
0.481944	0.148611	0.122222	0.188889	0.171429	0.149206	0.0777778	0.0666667	0.255556	0.122222	0.0646465		

Tabla 5.16: Intervalos de confianza del 95 % de número de AP encontrados por secuencias de escaneo optimizadas del experimento 4.c.

Cadena	**APs**	**Min**	**Max**	**Latencia (ms)**
indBase	16.5667	15.58	17.5533	429
cadena1	5.8	4.996	6.604	106
cadena2	7	6.00225	7.99775	108
cadena3	6.3	5.57067	7.02933	118
Debian	7.633333	6.810066	8.456600	113
Windows	10.63333	9.764601	11.50206	168
Meego	10.7	9.607376	11.79262	184
iOS	16.4	14.96683	17.83317	429
Android	18.83333	17.60040	20.06627	601

Tabla 5.17: Secuencias de escaneo optimizadas del experimento 5.b.

Canales de la secuencia											FO_1 (AP/ms)	FO_2 (ms)
$\mathbf{11}^{39}$	$\mathbf{6}^{78}$	$\mathbf{1}^{117}$	$\mathbf{5}^{156}$	$\mathbf{7}^{195}$	$\mathbf{10}^{234}$	$\mathbf{9}^{273}$	$\mathbf{2}^{312}$	$\mathbf{8}^{351}$	$\mathbf{3}^{390}$	$\mathbf{4}^{429}$	0.458974	429
<39,0>4.16667	<39,0>3.43333	<39,0>3.16667	<39,0>1.6	<39,0>1.53333	<39,0>1.36667	<39,0>0.733333	<39,0>0.633333	<39,0>0.566667	<39,0>0.466667	<39,0>0.233333		
0.106838	0.0880342	0.0811966	0.0410256	0.0393162	0.0350427	0.0188034	0.0162393	0.0145299	0.0119658	0.00598291		
$\mathbf{11}^{18}$	$\mathbf{6}^{33}$	$\mathbf{1}^{43}$	$\mathbf{10}^{58}$	$\mathbf{9}^{76}$	$\mathbf{3}^{88}$	$\mathbf{2}^{106}$	$\mathbf{7}^{115}$	$\mathbf{5}^{128}$	$\mathbf{8}^{141}$	$\mathbf{4}^{159}$	1.25167	159
<15,3>2.06667	<12,3>1.26667	<6,4>1.13333	<9,6>0.766667	<14,4>0.7	<9,3>0.666667	<13,5>0.633333	<6,3>0.566667	<9,4>0.466667	<8,5>0.4	<15,3>0.166667		
0.173333	0.161111	0.227778	0.0981481	0.05	0.148148	0.0733333	0.133333	0.112037	0.05	0.0244444		
$\mathbf{11}^{14}$	$\mathbf{6}^{32}$	$\mathbf{1}^{46}$	$\mathbf{10}^{69}$	$\mathbf{7}^{86}$	$\mathbf{5}^{97}$	$\mathbf{3}^{112}$	$\mathbf{8}^{132}$	$\mathbf{9}^{144}$	$\mathbf{2}^{153}$	$\mathbf{4}^{165}$	1.36009	160
<11,3>2.43333	<11,7>1.46667	<7,7>1.43333	<14,9>0.933333	<14,3>0.9	<8,3>0.633333	<10,5>0.5	<13,7>0.5	<9,3>0.433333	<6,3>0.233333	<7,5>0		
0.350505	0.155844	0.204762	0.0772487	0.0992063	0.109722	0.0733333	0.0428571	0.062963	0.172222	0.0114286		
$\mathbf{1}^{19}$	$\mathbf{11}^{31}$	$\mathbf{6}^{50}$	$\mathbf{7}^{67}$	$\mathbf{9}^{85}$	$\mathbf{10}^{103}$	$\mathbf{5}^{115}$	$\mathbf{2}^{131}$	$\mathbf{8}^{151}$	$\mathbf{3}^{162}$	$\mathbf{4}^{178}$	1.46643	178
<11,8>2.23333	<8,4>1.8	<15,4>1.53333	<14,3>1.16667	<15,3>0.833333	<10,8>0.766667	<9,3>0.566667	<13,3>0.533333	<15,5>0.4	<8,3>0.366667	<11,5>0.3		
0.235985	0.316667	0.114444	0.179365	0.153333	0.0841667	0.062963	0.191453	0.0355556	0.0458333	0.0466667		

Tabla 5.18: Intervalos de confianza del 95 % de número de AP encontrados por secuencias de escaneo optimizadas del experimento 4.c.

Cadena	**APs**	**Min**	**Max**	**Latencia (ms)**
indBase	17.0667	16.0332	18.1001	429
cadena1	9.53333	8.44421	10.6225	159
cadena2	8.7	7.89291	9.50709	160
cadena3	11.0667	10.1933	11.94	178
Debian	7.633333	6.810066	8.456600	113
Windows	10.63333	9.764601	11.50206	168
Meego	10.7	9.607376	11.79262	184
iOS	16.4	14.96683	17.83317	429
Android	18.83333	17.60040	20.06627	601

Las secuencias de escaneo se encuentran ordenadas por el número promedio de puntos de acceso encontrados por canal y en orden descendente. En los resultados se observa que los canales 1, 6 y 11 aparecen en las primeras posiciones de las secuencias de escaneo. Esto se debe a que corresponden a los canales no solapados de la banda 2.4GHz y en los cuales es común encontrar la mayoría de puntos de accesos en despliegues urbanos como las redes comunitarias.

En los experimentos enumerados 2 al 5 (ver § 5.4.2) se utiliza una secuencia de referencia (individuo inteligente) para la creación de la población inicial, de allí que es posible observar en las tablas 5.5 a la 5.17 que se presenta este individuo en la primera fila. Se puede observar que el número de puntos de acceso encontrados por canal para la secuencia de referencia es sistemáticamente mayor que en una secuencia de escaneo resultante del algoritmo. Este comportamiento es debido a que se encuentra una mayor cantidad de puntos de acceso si se espera más tiempo en un canal. Una razón para que un punto de acceso sea encontrado más tarde es que las tramas de administración *Probe Response* de esos puntos de accceso pueden pasar por varias retransmisiones. Sin embargo, dado que se está limitando la latencia, estamos interesados notablemente en aquellos puntos de acceso que aparezcan al inicio de la ventana de tiempo de espera; esto es el temporizador MinCT.

En la Fig. 5.5 se muestra de manera gráfica un conjunto de secuencias encontradas en una ejecución del algoritmo cultural con los parámetros descritos en § 5.2.3. En esta figura es posible verificar que la forma que siguen las soluciones encontradas al problema de optimización multiobjetivo descrito en § 4.2.2, es la correspondiente a problemas de dos funciones objetivo, una de maximización y otra de minimización [21], tal como se muestra en la Fig. 5.6.

Las secuencias de escaneo derivadas del algoritmo cultural pueden ser utilizadas por aplicaciones que requieran valores específicos de latencias; por ejemplo las aplicaciones elásticas podrían considerar secuencias de escaneo con temporizadores mayores. Obtener secuencias de escaneo para este tipo de restricciones es más sencillo, ya que la aplicación se hace tolerante a retardos.

5.5. Conclusiones

En este capítulo se ha mostrado que el proceso de escaneo en despliegues Wi-Fi densos puede ser mejorado significativamente al utilizar técnicas

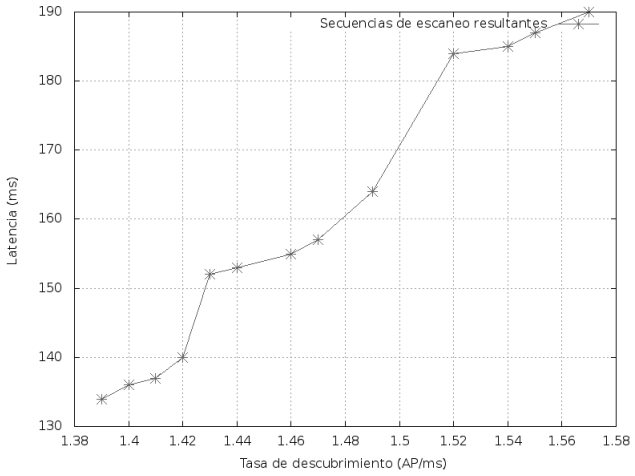


Fig. 5.5: Conjunto de soluciones del problema de optimización generado por el algoritmo cultural.

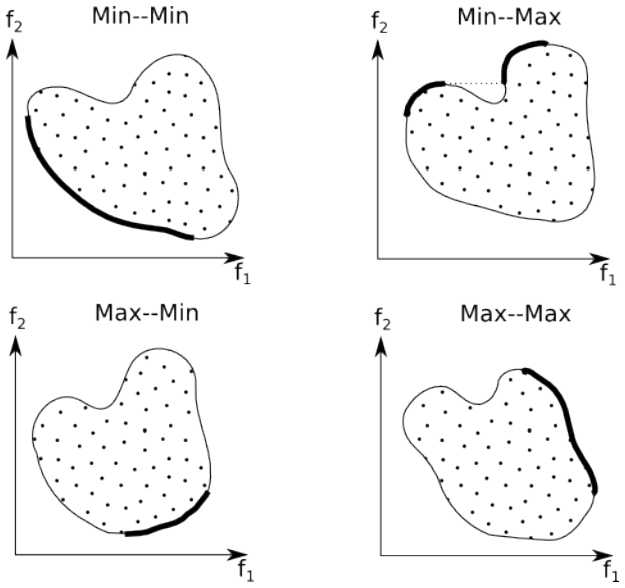


Fig. 5.6: Conjuntos de soluciones óptimas para problemas de optimización multiobjetivo de dos funciones objetivo.

de inteligencia computacional como los algoritmos culturales para encontrar valores adecuados de la secuencia de escaneo y sus temporizadores. Se ha caracterizado un compromiso entre métricas del desempeño del proceso de escaneo como la tasa de descubrimiento (APs encontrados por unidad de tiempo) versus la latencia total de un escaneo.

Las secuencias de escaneo derivadas del algoritmo cultural se compararon con secuencias de referencia identificadas a partir de un proceso de ingeniería inversa de dispositivos con interfaces Wi-Fi. Los valores de tasa de descubrimiento son superiores en las secuencias derivadas y representan mejoras entre 230 % y 600 % en el desempeño del descubrimiento. Con respecto a la latencia de escaneo las mejoras se encuentran en un rango entre 30 % y 60 %.

Las secuencias de escaneo eficientes pueden generar ahorros directos en términos de consumo de energía, debido a que los recursos de los dispositivos se dedican a una interacción con canales de la banda 2.4GHz por un tiempo de espera apropiado.

Como trabajo futuro se propone resolver el problema de optimización de parámetros de escaneo al utilizar otra técnica de inteligencia computacional como la Optimización por Enjambre de Partículas (PSO por sus siglas en inglés) para problemas de optimización multiobjetivo. Los resultados de esta nueva técnica podrían ser contrastados con los presentados en este capítulo y discutir sus ventajas. Adicionalmente, se vislumbra que el algoritmo cultural y la implementación realizada en este trabajo podría ser incorporada como un mecanismo de generación de secuencias óptimas de escaneo en entornos en los cuales una entidad centralizada soporte el proceso de descubrimiento en redes IEEE 802.11

Capítulo 6

Una arquitectura para el descubrimiento asistido en redes IEEE 802.11

En este capítulo se propone un marco de trabajo que permite asistir a las estaciones móviles en el proceso de descubrimiento en redes comunitarias. Uno de los componentes del marco de trabajo consiste en un ente centralizado que con el uso de una técnica de inteligencia computacional podría calcular secuencias de escaneo óptimas en un topología de redes IEEE 802.11 como las redes comunitarias. Una estimación correcta de los temporizadores MinCT y MaxCT depende del conocimiento de la topología de la red; de allí la necesidad de un mecanismo de escaneo asistido. El marco de trabajo propuesto en este capítulo está basado en la investigación realizada por Arcia-Moret et al. [8].

6.1. Trabajos relacionados

En la actualidad existen algunos trabajos enfocados a ofrecer servicios centralizados para controlar despliegues de redes inalámbricas. Yiakoumis et al. [42] presentan BeHop, un banco de pruebas para redes inalámbricas Wi-Fi densas que se encuentran comúnmente en zonas residenciales o en empresas. BeHop apunta a proporcionar conocimiento sobre la operación de despliegues densos y evaluar cómo distintas estrategias de administración afectan la experiencia del usuario y el comportamiento de la red. Los autores

se enfocan principalmente en estudiar los pro y los contra de nuevas maneras de controlar redes Wi-Fi, como por ejemplo el control central de una red, control de energía, asignación de canales, entre otros aspectos.

Sathiaseelan et al. [43] presentan las redes públicas virtuales (VPuN por sus siglas en inglés). Las VPuN son redes caseras creadas, desplegadas y administradas a través de una abstracción de control que le permite a usuarios y operadores de red compartir y controlar la red mientras permiten que otros participantes emerjan como operadores de red virtuales. La abstracción de control utilizada por las VPuN está basada en la noción de redes definidas por software (SDN por sus siglas en inglés) en las que se desacopla el plano de control (software) del plano de datos (hardware).

Sathiaseelan et al. [44] presentan los servicios Wi-Fi de acceso público (PAWS por sus siglas en inglés). PAWS es un nuevo paradigma de acceso a Internet basado en un conjunto de técnicas que hacen posible el uso de capacidad disponible ociosa en las redes de banda ancha de hogar, permitiendo un acceso a los recursos sin que el tráfico gratuito tenga un impacto en el desempeño del ancho de banda del usuario que comparte. Los autores discuten las restricciones y la arquitectura propuesta incluyendo mecanismos de autenticación y seguridad necesarios para este tipo de servicio.

Meraki[1] es una solución para redes inalámbricas que optimiza y garantiza un alto desempeño en ambientes inalámbricos densos y bajo intensas condiciones de interferencia. Las configuraciones de redes inalámbricas y los ajustes de estaciones móviles se adaptan automáticamente a cambios de desempeño y condiciones de interferencia sin intervención manual en los parámetros inalámbricos.

En los trabajos descritos el enfoque de servicios centralizados está orientado a perfilar tráfico y ejecutar balanceo de carga, sin embargo no ofrecen detalles sobre el proceso de descubrimiento o escaneo.

6.2. Administrador de topología para redes 802.11

En el reciente enfoque de la arquitectura de networking centrado en información descrito por Trossen y Parisis [45], un Administrador de Topología (TM por sus siglas en inglés) podría asistir oportunamente a usuarios móvi-

[1] https://meraki.cisco.com/

les para descubrir mejor y controlar una topología de red inalámbrica densa. También podría ayudar a usuarios móviles a determinar la calidad de enlace y la mejor conexión posible a las redes de acceso. Como se ha observado previamente, en un escaneo tiende a descubrirse sólo un subconjunto de los puntos de acceso disponibles, y usualmente, un cliente no tiene el tiempo para ejecutar múltiples escaneos [3]. Sin embargo, con un TM inteligente, los usuarios podrían compartir información sobre su visión de la topología, de manera que el TM pueda calcular una visión actualizada de la topología, proponiendo una vista global y más precisa del despliegue de puntos de acceso. La principal ventaja de este enfoque es generar secuencias de escaneo eficientes que permiten ahorrar tiempo a los clientes de una red inalámbrica durante el costoso proceso de descubrimiento.

Para actualizar la visión de la topología de la red en el TM, se proponen dos entidades participantes:

1. Alimentadores: que comparten la visión de la topología en forma de actualizaciones que obtienen al realizar escaneos regulares cada cierto tiempo con el TM. Estas actualizaciones se podrían realizar a través de la enmienda IEEE 802.11u conocida como Protocolo de Consulta de Red de Acceso (ANQP por sus siglas en inglés), que permite a las estaciones móviles consultar o pasar información al TM sobre un punto de acceso designado. Adicionalmente una estación móvil podría utilizar mensajes específicos para obtener información sobre un operador móvil específico cuya red sea accesible a través de un punto de acceso designado.

2. Clientes regulares: que corresponden a aquellos alimentadores que han contribuido suficiente información a la visión de la topología y que han sido promovidos por el TM como clientes regulares. De esta manera, los clientes regulares que buscan un punto de acceso candidato para conectarse podrían obtener secuencias eficientes con consultas especiales al TM, quien a su vez interactúa de forma asíncrona con un algoritmo inteligente como el que se propone en la sección 6.3.

6.2.1. Interacción con el administrador de topología comunitario

La Fig. 6.1 presenta un bosquejo de la arquitectura de la red inalámbrica de nueva generación en la que se incorpora el escaneo asistido con un TM. En esta figura se observa que con un simple intercambio de mensajes de ida y vuelta entre el cliente regular y el TM, se podrían ahorrar cientos de milisegundos mejorando la eficiencia del proceso de escaneo.

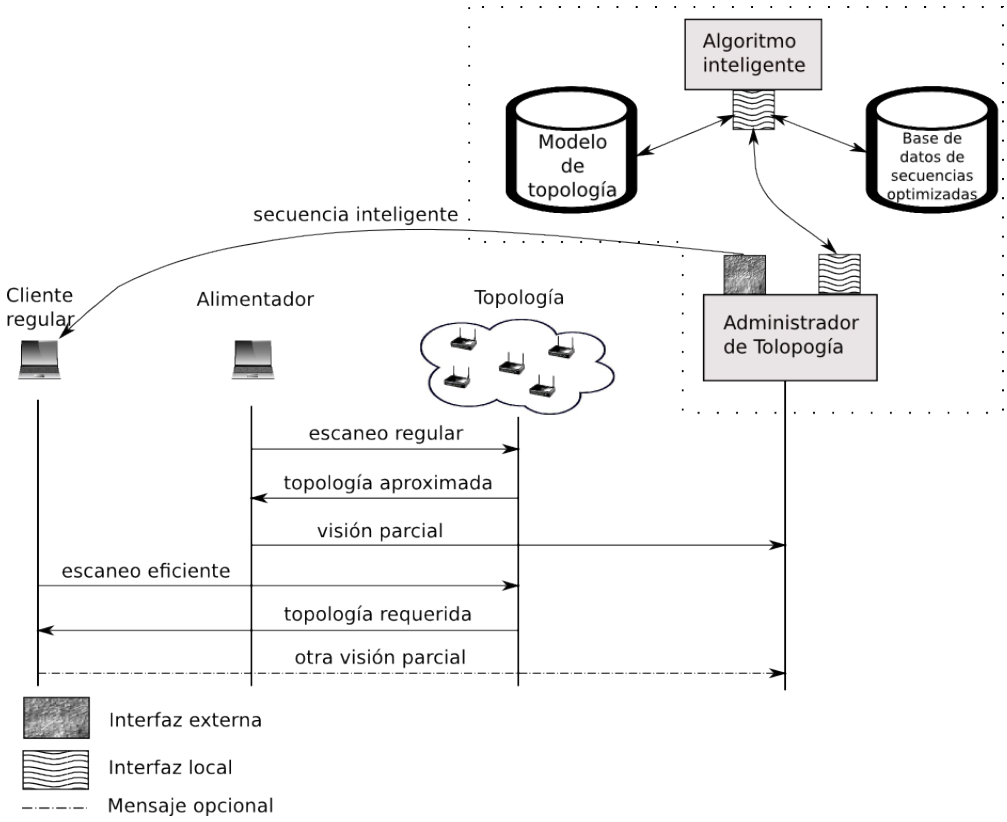


Fig. 6.1: Bosquejo de arquitectura de red inalámbrica de nueva generación

El proceso de interacción comienza como sigue. Un alimentador (A) que llega a la red contribuye con escaneos de la topología al TM, quien agrega cada nueva entrada que recibe. El TM puede almacenar información georeferenciada creada por alimentadores previos. Eventualmente el TM tendrá suficiente información sobre los tiempos entre respuestas (tramas de administra-

ción *Probe Response*) de los puntos de acceso de la topología bajo su control. Mientras tanto, esa información es almacenada en el modelo de la topología y la usa el algoritmo inteligente para calcular secuencias eficientes.

Se espera que un alimentador (A) realice escaneos regulares durante sus primeros intentos de unirse a la red. La contribución del alimentador a la visión del TM le permitirá obtener secuencias de escaneo eficientes, luego de ser promovido como cliente regular. De lo contrario, podría ser considerado como un cliente que no coopera y no se beneficiaría de las estimaciones del TM. En escaneos subsecuentes, un cliente podría saber si posee una nueva visión de la topología (ya que se encuentra en otra posición) y podría entonces enviar la nueva información al TM para mejorar el modelo de la topología.

Finalmente un cliente regular podría estar interesado en solicitarle al TM una secuencia de escaneo especial basada en requerimientos de las aplicaciones específicas que ejecuta; por ejemplo un cliente podría enfrentar restricciones de latencia debido a aplicaciones de VoIP.

6.3. Módulo inteligente para administración de topología

Como se muestra en la Fig. 6.1, utiliza un algoritmo inteligente para que los dispositivos móviles realicen el descubrimiento y asociación a una red de manera eficiente. El TM invoca, conveniente y asíncronamente, el cálculo eficiente de secuencias de escaneo. Luego estas secuencias se almacenan sistemáticamente en una base de datos de secuencias eficientes (similar a la BDe descrita en § 4.3) de manera que el TM pueda tener acceso inmediato a secuencias eficientes siempre que un cliente lo solicite.

El módulo inteligente del TM utiliza el algoritmo cultural descrito en el capítulo 4, que a su vez constituye uno de los elementos principales del sistema inteligente para el escaneo de redes 802.11 que se propone en este trabajo. El algoritmo cultural adaptado e implementado en este trabajo puede ser considerado como un motor de cálculo de secuencias de escaneo eficientes para el problema de descubrimiento de redes IEEE 802.11.

6.4. Conclusiones

En este capítulo se presentó el diseño de un administrador de topología (TM) y se discutió la interacción con clientes inalámbricos. Se han propuesto dos funcionalidades separadas para el TM: la abstracción de modelo de topología inalámbrica y el motor de cálculo de secuencias de escaneo.

Se vislumbra como trabajo futuro la implementación de un administrador de topología distribuido que permita separar la red en celdas para obtener mejores secuencias de escaneo en porciones más pequeñas de la red. También se ha observado que el modelo del TM podría ser más preciso si se incluyen áreas con las mismas características, por ejemplo, una zona comercial en un centro de ciudad.

Capítulo 7

Conclusiones y trabajo futuro

En este trabajo se ha abordado el problema del descubrimiento o escaneo en redes IEEE 802.11 desde un punto de vista experimental, de inteligencia computacional y de prospectiva.

En primer lugar, con respecto al escaneo activo de IEEE 802.11 se diseñó y construyó un prototipo de *sniffer* multicanal con componentes de bajo costo que permitió realizar un proceso de ingeniería inversa sobre el proceso de escaneo. Se capturaron tramas de administración del estándar IEEE 802.11 en los 11 canales de la banda 2.4GHz simultáneamente. Se probaron dispositivos móviles de distintos fabricantes y distintos sistemas operativos y se observó que realizan secuencias de escaneo diferentes; en algunos casos se utilizan secuencias que prueban los canales desde los más bajos a los más altos y en otros casos, se alterna con pruebas de canales desde los más altos a los más bajos. Otra observación realizada es que la frecuencia de ejecución es distinta para cada dispositivo: unos realizan el escaneo en períodos constantes, por ejemplo cada 10 segundos, otros dispositivos realizan un escaneo incremental donde, por ejemplo, después de 20 segundos incrementan en 10 segundos la espera antes de realizar otro escaneo hasta llegar a los 60 segundos. Adicionalmente se observó que la duración del escaneo es diferente para los dispositivos probados. Estos hallazgos experimentales muestran que los algoritmos de escaneo implementados en el software o hardware de los dispositivos inalámbricos son diferentes y que no hay evidencia de que exista una secuencia de escaneo única para todos los casos.

En segundo lugar, se ha caracterizado un compromiso entre métricas del desempeño del proceso de escaneo como la tasa de descubrimiento por unidad de tiempo versus la latencia total de escaneo a través de la formulación

de un problema de optimización multiobjetivo. Para resolver el problema de optimización multiobjetivo se adaptó e implementó un algoritmo cultural basado en programación evolutiva, eficiencia de Pareto y elitismo. Se incorporó un mecanismo de mutación dirigida que permite ejercer influencia en las nuevas generaciones de individuos de la población del algoritmo cultural que representan secuencias de escaneo optimizadas. Para obtener las secuencias se utilizó un emulador de un despliegue real de redes IEEE 802.11 a partir del cual se obtienen los puntos de acceso que se descubrirían con los temporizadores MinCT y MaxCT.

Se compararon secuencias derivadas del algoritmo cultural con secuencias de referencia de dispositivos identificadas con el *sniffer* multicanal. La comparación de secuencias de escaneo muestra que las secuencias derivadas del algoritmo cultural mejoran las tasas de descubrimiento entre un 230 % y 600 % y la latencia de escaneo entre un 30 % y 60 %.

La implementación del algoritmo cultural puede ser utilizada como marco de trabajo para nuevos estudios de compromiso entre otras métricas de desempeño. Asimismo, está disponible un sistema web que permite realizar consultas al emulador de topología inalámbrica para obtener el número de puntos de acceso dados los parámetros de secuencias de escaneo como los canales y los temporizadores MinCT y MaxCT.

Finalmente, y a manera de prospectiva, se propone un sistema que permitiría asistir a estaciones móviles en el proceso de descubrimiento en redes comunitarias de nueva generación. El sistema incorpora un módulo de inteligencia computacional; en este caso se propone utilizar el algoritmo cultural adaptado e implementado para optimizar las secuencias de escaneo.

Como trabajo futuro se vislumbra estudiar el comportamiento de los algoritmos de escaneo con el *sniffer* multicanal y en presencia de tráfico de fondo, así como efectuar medidas del consumo de energía. Adicionalmente, resolver el problema de optimización multiobjetivo de los parámetros de desempeño del proceso de escaneo con otra técnica de inteligencia computacional y comparar los resultados obtenidos. De esta manera se podría enriquecer el conocimiento necesario para el módulo de inteligencia computacional del sistema de descubrimiento asistido para redes comunitarias de nueva generación.

El conjunto de resultados alcanzados en este trabajo presentan un enfoque práctico para abordar el escaneo de redes IEEE 802.11 en la Internet del futuro.

Bibliografía

[1] "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," 2012.

[2] J. Saldana, A. Arcia-Moret, B. Braem, L. Navarro, E. Pietrosemoli, C. Rey-Moreno, A. Sathiaseelan, and M. Zennaro, "Alternative Network Deployments. Taxonomy, characterization, technologies and architectures." draft-irtf-gaia-alternative-network-deployments-02, November 2015.

[3] A. Arcia-Moret, L. Molina, N. Montavont, G. Castignani, and A. Blanc, "Acess point discovery in 802.11 networks," in *Wireless Days (WD), 2014 IFIP*, Noviembre 2014.

[4] L. Molina and A. Arcia-Moret, "Evaluación del proceso de escaneo en redes 802.11: una perspectiva taxonómica," in *1era Conferencia Nacional de Computación, Informática y Sistemas*, Octubre 2013.

[5] G. Castignani, A. Arcia, and N. Montavont, "A study of the discovery process in 802.11 networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 15, pp. 25–36, Marzo 2011.

[6] I. Kim and Y.-T. Kim, "Prediction-based smart channel scanning with minimized service disruption for IEEE 802.11e WLAN," in *Consumer Electronics (ICCE), 2011 IEEE International Conference on*, pp. 907–908, Jan 2011.

[7] A. Mishra, M. Shin, and W. Arbaugh, "An empirical analysis of the IEEE 802.11 MAC layer handoff process," *SIGCOMM Comput. Commun. Rev.*, vol. 33, pp. 93–102, Apr. 2003.

[8] A. Arcia-Moret, A. Sathiaseelan, A. Araujo, J. Aguilar, and L. Molina, "Assisted network discovery for next generation wireless networks," in *Presentado como poster en 6th IEEE Conference on Consumer Communications and Networking Conference (CCNC'16)*, 2016.

[9] e. J.Butler, *Wireless Networking in the Developing World.* ICTP, 2013.

[10] Cisco, "802.11 association process explained." `https://documentation.meraki.com/MR/WiFi_Basics_and_Best_Practices/802.11_Association_process_explained`. Consultado el 15 de octubre de 2015.

[11] H. Velayos and G. Karlsson, "Techniques to reduce the IEEE 802.11b handoff time," *2004 IEEE International Conference on Communications IEEE Cat No04CH37577*, vol. 00, no. c, pp. 3844–3848, 2004.

[12] N. Montavont, J. Montavont, and T. Noel, "Enhanced schemes for L2 handover in IEEE 802.11 networks and their evaluations," in *IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications PIMRC*, vol. 3, pp. 1429 – 1434, RSM - Dépt. Réseaux, Sécurité et Multimédia (Institut Mines-Télécom-Télécom Bretagne-UEB), LSIIT - Laboratoire des sciences de l'image, de l'informatique et de la télédétection (CNRS UMR 7005), 2005.

[13] J. Eriksson, H. Balakrishnan, and S. Madden, "Cabernet: vehicular content delivery using wifi," in *Proceedings of the 14th ACM international conference on Mobile computing and networking*, MobiCom '08, (New York, NY, USA), pp. 199–210, ACM, 2008.

[14] X. Hu, L. Song, D. V. Bruggen, and A. Striegel, "Is there wifi yet? how aggressive wifi probe requests deteriorate energy and throughput," *CoRR*, vol. abs/1502.01222, 2015.

[15] N. Brouwers, M. Zuniga, and K. Langendoen, "Incremental wi-fi scanning for energy-efficient localization," in *Pervasive Computing and Communications (PerCom), 2014 IEEE International Conference on*, pp. 156–162, March 2014.

[16] J. Brownlee, *Clever Algorithm: Nature-Inspired Programming Recipes.* Lulu Enterprises Incorporated, 2011.

[17] R. Reynolds, "Evolutionary multiobjective optimization using a cultural algorithm," in *Proceedings of the 3rd Annual Conference on EvolutionaryProgramming*, pp. 131–139, World Scienfific Publishing, 1994.

[18] R. G. Reynolds, "Cultural algorithms: A tutorial." `"http://groups.engin.umd.umich.edu/vi/w2_workshops/cultural_alg_reynolds_w2.pdf"`. Consultado el 13 de octubre de 2015.

[19] K. Deb, *Multi-Evolutionary Optimization Using Evolutionary Algorithms*. Wiley, 2001.

[20] V. Pareto, *Cours D'Economie Politique*. F.Rouge, 1975.

[21] K. M. a. j. Branke, K. Deb, *Multiobjective Optimization. Interactive and Evolutionary Approaches*. Springer, 2008.

[22] R. L. Becerra, "Algoritmos culturales aplicados a optimización con restricciones y optimización multiobjetivo," Master's thesis, Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, México, 2002.

[23] Kismet. `https://kismetwireless.net/`, 2011. Consultado el 19 de octubre de 2015.

[24] A. Araujo and A. Arcia-Moret, "Identificación de secuencias de scanning en redes 802.11," *Revista Venezolana de Computación*, vol. 1, no. 1, pp. 50–57, 2014.

[25] V. Gupta, R. Beyah, and C. Corbett, "A characterization of wireless NIC active scanning algorithms," in *Wireless Communications and Networking Conference, 2007.WCNC 2007. IEEE*, pp. 2385–2390, IEEE, Marzo 2007.

[26] T. Laurenson, "Forensic data storage for wireless networks: A compliant architecture," Master's thesis, School of Computing and Mathematical Sciences, Auckland University of Technology, 2010.

[27] C. Corbett, R. Beyah, and J. Copeland, "Using Active Scanning to Identify Wireless NICs.," in *Proceedings of the IEEE Information Assurance Workshop*, IAW, (West Point, New York), 2006.

[28] P. Reddy, H. Sharme, and D. Paulraj, "Multi Channel Wifi Sniffer," in *Wireless Communications, Networking and Mobile Computing. WiCOM '08. 4th International Conference on*, WiCOM, (Dalian, CHina), 2008.

[29] "SD Specifications. Part 1 Physical Layer Simplified Specification." `https://www.sdcard.org/downloads/pls/simplified\_specs/part1\_410.pdf`, 2013. Consultado el 19 de octubre de 2015.

[30] "USB Specifications." `http://www.usb.org/developers/docs/usb\_31\_072715.zip`, 2013. Consultado el 19 de octubre de 2015.

[31] A. Nicholson, Y. Chawathe, M.Chen, B. Noble, and D. Wetherall, "Improved access point selection," in *Proceedings of the 4th international conference on Mobile systems, applications and services*, pp. 233–245, 2006.

[32] D. Murray, M. Dixon, and T. Koziniec, "Scanning Delays in 802.11 Networks," in *Next Generation Mobile Applications, Services and Technologies, 2007. NGMAST '07. The 2007 International Conference on*, pp. 255–260, 2007.

[33] C. Coello and R. Becerra, "Evolutionary multiobjective optimization using a cultural algorithm," in *Proceedings of the 2003 IEEE Swarm Intelligence Symposium, SIS '03*, pp. 6–13, IEEE, April 2003.

[34] N. Montavont, A. Arcia-Moret, and G. Castignani, "On the selection of scanning parameters in IEEE 802.11 networks," in *Personal Indoor and Mobile Radio Communications (PIMRC), 2013 IEEE 24th International Symposium on*, pp. 2137–2141, 2013.

[35] A. Arcia-Moret, A. Araujo, J. Aguilar, L. Molina, and A. Sathiaseelan, "Intelligent network discovery for next generation community wireless networks," in *Proc. 12th Annual Conference on Wireless on Demand Network Systems and Services (WONS'16)*, 2016.

[36] G. Castignani, N. Montavont, and A. Arcia-Moret, "Multi-objective optimization model for network selection in multihomed devices," in *Proc. 10th Annual Conference on Wireless on Demand Network Systems and Services (WONS'13)*, pp. 113–115, 2013.

[37] G. Castignani, A. Arcia-Moret, and N. Montavont, "An evaluation of the resource discovery process in ieee 802.11 networks," in *Proceedings of the Second International Workshop on Mobile Opportunistic Networking*, MobiOpp '10, (New York, NY, USA), pp. 147–150, ACM, 2010.

[38] Y. Liao and L. Cao, "Practical schemes for smooth mac layer handoff in 802.11 wireless networks," in *Proceedings of the 2006 International Symposium on on World of Wireless, Mobile and Multimedia Networks*, WOWMOM '06, (Washington, DC, USA), pp. 181–190, IEEE Computer Society, 2006.

[39] J.-W. Nah, S.-M. Chun, S. Wang, and J.-T. Park, "Adaptive handover method with application-awareness for multimedia streaming service in wireless LAN," in *Proceedings of the 23rd international conference on Information Networking*, ICOIN09, (Piscataway, NJ, USA), pp. 1–7, IEEE Press, 2009.

[40] S. Shin, A. G. Forte, A. S. Rawat, and H. Schulzrinne, "Reducing mac layer handoff latency in ieee 802.11 wireless lans," in *Proceedings of the Second International Workshop on Mobility Management &Amp; Wireless Access Protocols*, MobiWac '04, (New York, NY, USA), pp. 19–26, ACM, 2004.

[41] L. Molina, "Estudio del descubrimiento de topologías espontáneas," Master's thesis, Universidad de Los Andes, 2014.

[42] Y. Yiakoumis, M. Bansal, A. Covington, J. van Reijendam, S. Katti, and N. McKeown, "Behop: A testbed for dense wifi networks," in *Proceedings of the 9th ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization*, WiNTECH '14, (New York, NY, USA), pp. 1–8, ACM, 2014.

[43] A. Sathiaseelan, C. Rotsos, C. S. Sriram, D. Trossen, P. Papadimitriou, and J. Crowcroft, "Virtual public networks," in *Proceedings of the 2013 Second European Workshop on Software Defined Networks*, EWSDN '13, (Washington, DC, USA), pp. 1–6, IEEE Computer Society, 2013.

[44] A. Sathiaseelan, R. Mortier, M. Goulden, C. Greiffenhagen, M. Radenkovic, J. Crowcroft, and D. McAuley, "A feasibility study of an in-the-wild experimental public access wifi network," in *Proceedings of the Fifth*

ACM Symposium on Computing for Development, ACM DEV-5 '14, (New York, NY, USA), pp. 33–42, ACM, 2014.

[45] D. Trossen and G. Parisis, "Designing and realizing an information-centric internet," *Communications Magazine, IEEE*, vol. 50, pp. 60–67, July 2012.