Revista Venezolana de Computación
ISSN: 2244-7040
http://www.svc.net.ve/revecom
Vol. 1, No. 1, pp. 50-57, Junio 2014
Selección de los Mejores Artículos de CoNCISa 2013

Identificación de Secuencias de Scanning en Redes 802.11

Antonio Araujo[1], Andrés Arcia-Moret[2]
aaraujo@cenditel.gob.ve, aarcia_m@ictp.it

[1] Centro Nacional de Desarrollo e Investigación en Tecnologías Libres (CENDITEL), Mérida, Venezuela
[2] Marconi Laboratory, International Centre for Theoretical Physics (ICTP), Trieste, Italia

Resumen: Actualmente las redes 802.11 se han convertido en la primera opción para el acceso inalámbrico a la Internet. Para conectarse, un dispositivo cualquiera se registra a algún Punto de Acceso pasando primero por un costoso proceso de búsqueda de los APs disponibles. El estándar IEEE 802.11 deja abierta la posibilidad de implementar los criterios de búsqueda estableciendo solamente los mensajes necesarios para tal efecto. En este artículo caracterizamos distintos dispositivos a través de un sistema especialmente construido para tal fin. El sistema propuesto es capaz de escuchar todos los canales de manera simultanea para así mostrar verazmente las estrategias que siguen distintos dispositivos durante el proceso de búsqueda. El sistema propuesto permite movilidad, operación bajo estándares abiertos y uso de componentes de bajo costo.

Palabras Clave: IEEE 802.11; Scanning Activo; Caracterización.

Abstract: Nowadays 802.11 networks are the first option to last-mile wireless access to Internet. Access Points operate in a set of overlapping channels conforming a rather chaotic topology. And so, connecting to an available AP requires an expensive scanning process before deciding whic AP to connect to. As the IEEE 802.11 do not address specifics on scanning, vendors implement different (closed) strategies for scanning. In this paper we discuss the building of a simple multi-channel sniffer using open standards and low cost components in order to identify scanning sequences. Then, we characterise several scanning strategies within different chipsets.

Keywords: IEEE 802.11; Active Sanning; Characterization.

I. Introducción

Las redes inalámbricas constituyen una de las principales tecnologías de acceso local y móvil a la Internet. El uso de clientes inalámbricos, que van desde sensores hasta computadores personales, requieren que las redes 802.11 den soporte a las necesidades de usuarios que demandan movilidad y mejor calidad de servicio. Estas demandas pueden verse satisfechas a través de los mecanismos de asociación de los dispositivos móviles a las redes 802.11, cuando estos pueden orientarse a minimizar los tiempos de búsqueda de Puntos de Acceso disponibles.

El proceso de búsqueda o scanning, ha sido estudiado desde diversos enfoques para disminuir los tiempos de conexión de un dispositivo móvil. Desde la modificación de variables establecidas en el estándar IEEE 802.11 [1], hasta la incorporación de información adicional a las tramas del estándar [2], las propuestas consideran distintas maneras de mejorar el algoritmo de scanning de los dispositivos. La identificación de secuencias de scanning permite comprender el comportamiento de distintos dispositivos WiFi de uso masivo, los que hasta hoy, no muestran el código fuente de la implementación del scanning. La mejora del proceso de scanning podría, a su vez, representar directamente una mejora en el proceso de *handover* (el cual ocupa hasta un 80% del tiempo de reconexión [3]), 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 scanning 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 scanning. Algunas aplicaciones de software como Kismet [4] 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 barrido 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 éste 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 trabajo 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. Se prueba experimentalmente que es viable realizar un proceso de ingeniería inversa de los algoritmos de scanning implementados en dispositivos con distintos sistemas operativos para dispositivos móviles. También, se establecen algunas premisas de diseño para la construcción del prototipo y finalmente, se realizan experimentos para una caracterización de los algoritmos de scanning.

A. El Proceso de Scanning en Redes 802.11

El *scanning* de las redes inalámbricas es una función importante del protocolo MAC IEEE 802.11, donde los nodos intentan buscar redes inalámbricas disponibles y luego intentan asociarse a ellas [5]. El propósito del procedimiento de *scanning* es encontrar todos los BSS (Basic Service Set) a los que la estación pueda unirse. Para ser capaz de procesar todos los puntos de acceso en un área específica, por lo general se ajusta un timer llamado MaxChannelTime en áreas con un gran número de puntos de acceso [6].

En el estándar IEEE 802.11 [7] se definen dos tipos de técnicas de scanning de redes: scanning activo y scanning pasivo.

- *Scanning activo*: implica la generación de tramas Probe Request y el posterior procesamiento de las tramas Probe Response recibidas. Un resumen del procedimiento haciendo énfasis en los eventos de interés es el siguiente:
  - Enviar una trama Probe Request con destino broadcast.
  - Iniciar un temporizador (ProbeTimer) a la espera de respuestas.
  - Si el medio no está ocupado antes de que ProbeTimer alcance el valor de MinChannelTime, pasar a escanear el siguiente canal.
  - En caso contrario, activar un nuevo temporizador (MaxChannelTime) y cuando expire, procesar todas las tramas Probe Response para pasar al siguiente canal.
- *Scanning pasivo*: Si el parámetro ScanType indica un scanning pasivo, la estación escucha a cada canal una duración máxima definida por el parámetro MaxChannelTime. Los dispositivos móviles generalmente implementan el algoritmo de scanning activo para el descubrimiento de Puntos de Acceso (AP por sus siglas en inglés).

Particularmente, en este trabajo extraemos nuestro análisis y conclusiones basados en el scanning activo, pues de los dos procesos, es el que puede controlarse para obtener una eventual reducción de los tiempos de conexión a una red 802.11.

El artículo está estructurado de la siguiente manera. La Sección II discute el estado del arte sobre scanning en redes 802.11. La Sección III propone el diseño de un sniffer automático para el scanning en redes 802.11 con su respectiva arquitectura. La Sección IV discute el diseño experimental y el proceso de medición. La Sección V muestra los distintos algoritmos de scanning encontrados en una muestra de dispositivos móviles. Finalmente, la Sección VI presenta las conclusiones y el trabajo futuro.

II. Trabajos Relacionados

El estándar IEEE 802.11 describe el *scanning* para dispositivos inalámbricos (así como en la sección anterior), pero no especifica ni el orden particular ni los tiempos a esperar 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 *scanning* en sus algoritmos directamente en el hardware, así como también permitir que los sistemas operativos definan una secuencia particular para el *scanning*. La razón de la escogencia particular de una secuencia de scannning es hasta ahora una pregunta abierta.

Gupta, Beyah y Corbett [5] describen un proceso empírico para caracterizar algoritmos de scanning de diferentes tarjetas inalámbricas. Se realiza un proceso de scanning 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 en base a 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 scanning activo, tanto en software como en hardware, así como servir de base para evaluar su comportamiento en simuladores de redes.

Kim y Kim [2] estudian el desempeño del scanning 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 scanning inteligente que considera diferentes prioridades y analiza el desempeño del scanning de canales con respecto al número de estaciones activas.

Laurenson [8] 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 [9] emplean el procesamiento de señales para analizar la periodicidad del tráfico generado por el scanning 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 [10] 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.

III. Sniffer Multicanal Portatil

El comportamiento de las interfaces inalámbricas 802.11 en los diferentes dispositivos móviles sigue las especificaciones del estándar IEEE 802.11. El proceso de scanning 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 veremos más adelante en la Sección V). Sin embargo, Castignani, Arcia-Moret y Montavont [1] muestran que no solamente es de interés encontrar rápidamente los APs disponibles, sino que para llevar un proceso de descubierta de calidad, también es necesario tomar en cuenta la cantidad de APs 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 scanning de clientes 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 scanning activo. La Figura 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.

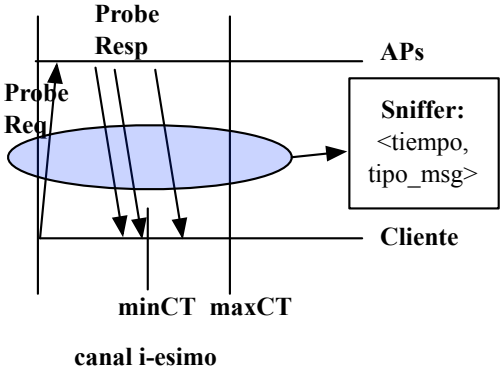

Figure 1: Bosquejo de Red IEEE 802.11 con un Sniffer

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 Figura 2 se muestra un diagrama de bloques con la arquitectura del sniffer propuesto. Cada uno de los componentes se describe a continuación.

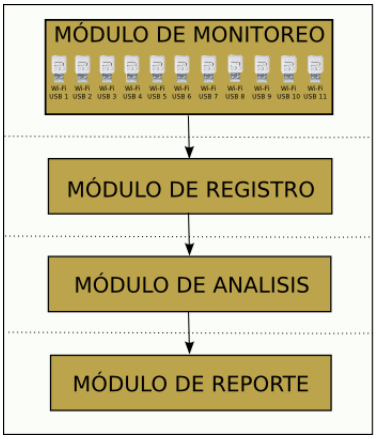

Figure 2: Diagrama de Bloques de la Arquitectura de Sniffer Propuestos

A. 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 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 scanning en los clientes.

B. 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.

1) 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 WiFi 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 en cada adaptador inalámbrico [11]. Un proceso automático habilitará o deshabilitará los adaptadores de acuerdo al entorno operativo que utilice el sniffer.

2) 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. 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 [12]. El tipo de trama se obtiene a partir de la información del encabezado radiotap [13]. 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.

3) 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 scanning de estaciones móviles. Los algoritmos de análisis pueden ser implementados por programas o scripts soportados en el entorno operativo del sniffer.

4) Módulo de Reporte: La función de éste módulo consiste en generar reportes en forma de textos, gráficos y/o animaciones que puedan representar características del comportamiento de scanning 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.

IV. Diseño Experimental

Sobre la base del diseño y premisas descritas en la sección III se construye un prototipo experimental cuyos componentes se muestran en la Figura 3.

La captura de tramas en los 11 canales de la banda 2.4GHz se realiza a través de adaptadores inalámbricos WiFi 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 [14]. 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

Figure 3: Componentes del Prototipo Experimental de Sniffer

Table I: Adaptadores Inalámbricos WiFi 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

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 [15] y 150MB/s para el disco duro [16]).

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 [17]. 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 [10], 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 scanning 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 [18] 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 I 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$.

Table II: 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	-

A. Experimentación

Para analizar y caracterizar el comportamiento de los algoritmos de scanning en redes 802.11 se realizan experimentos con el sistema descrito en la sección IV y dispositivos móviles con diferentes sistemas operativos. Como reportan Castignani, Arcia-Moret y Montavont [1], no existe una secuencia óptima predefinida ni un tiempo fijo para ejecutar la secuencia. En nuestro mejor esfuerzo de búsqueda, podemos confirmar que no está publicada una secuencia óptima de búsqueda. En la Tabla II se listan los dispositivos evaluados como clientes y un AP utilizados para pruebas del sniffer experimental.

1) Elaboración de la Traza: La función de monitoreo del sniffer experimental se realiza a través de la suite de aplicaciones *Aircrack-ng* [19], 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* [20]. *tshark* permite capturar tramas y distintos campos asociados al tráfico de redes alámbricas e inalámbricas. La salida de *tshark* se almacena en un archivo 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; ésto a través de la utilidad de línea de comando *crontab* [21]. 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 Figura 4 muestra un esquema de construcción del archivo de traza única de captura de tramas.

2) Duración del Scanning Activo: A continuación se describen los pasos realizados para la medición del proceso de scanning 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 scanning de todos los AP disponibles. Si encuentra un AP que el usuario había seleccionado como "preferido" anteriormente, automáticamente se asocia a él y no realiza el scanning completo en todos los canales [22].
- Iniciar el proceso de scanning en el cliente al activar la conexión inalámbrica.

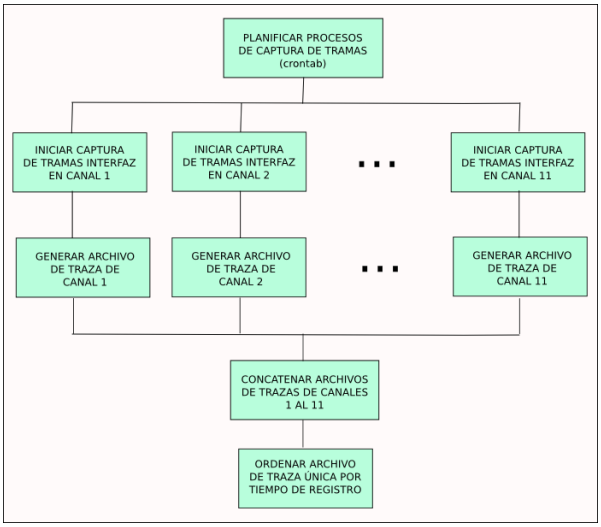

Figure 4: Esquema de Construcción de Archivo de Traza Única

Del lado del Sniffer:

- 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 III-B1.
- 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 III-B2.
- Esperar la captura (para nuestro caso particular, 4 minutos eran suficientes).
- Detener la ejecución de *tshark* en todas las interfaces y generar el archivo de tramas únicas asociado al scanning en los 11 canales. Estas actividades las realiza el módulo de Registro del sniffer como se describe en la subsección III-B2.

V. Resultados

En esta sección presentamos los dos resultados principales obtenidos. Primero, mostramos las secuencias de scanning de cada uno de los distintos clientes estudiados. Segundo, presentamos la frecuencia con la que se ejecuta un proceso de scanning en cada móvil. Notamos 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 MAC origen del dispositivo que origina la trama.
- *MAC destino*: dirección MAC destino de la trama.
- *Número de secuencia*: número de secuencia (SN) 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 ProbeRequest.
- *SSI Signal*: Potencia de la señal recibida (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.

A. Las Secuencias de Scanning

El estándar 802.11 define el algoritmo de *scanning 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 scanning de una STA 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 scanning 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 scanning o full scanning en las cuales se observan algunos patrones.

En las Figuras 5, 6, 7, 8 y 9 se muestran gráficas superpuestas de los canales en que se capturan tramas con respecto al tiempo de duración de 4 full scanning 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 APs en todos los canales. Creemos 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, observamos también que, en la Figura 9 el iPhone 4S muestra un comportamiento particular que no se había notado en ninguno de los otros dispositivos evaluados. Para éste móvil en particular, inicialmente, existe una fase

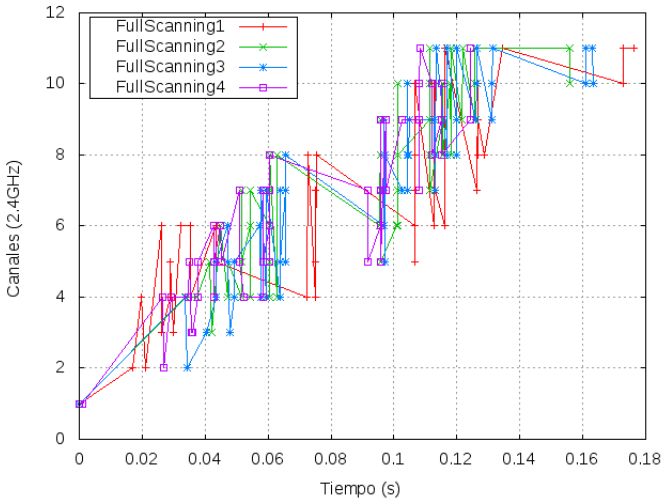

Figure 5: Secuencias de Full Scanning para Nokia N950

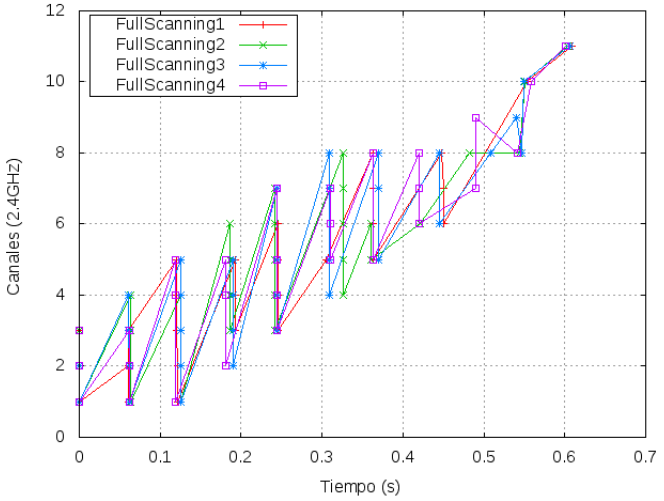

Figure 6: Secuencias de *Scanning* para Samsung Galaxy Tab

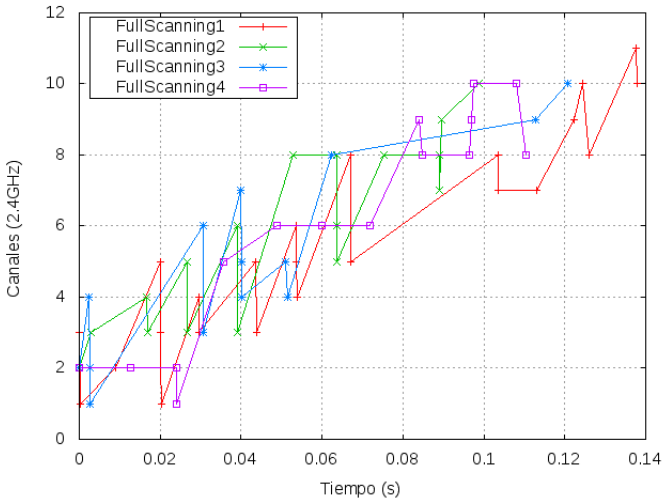

Figure 7: Secuencias de *scanning* para Laptop Dell

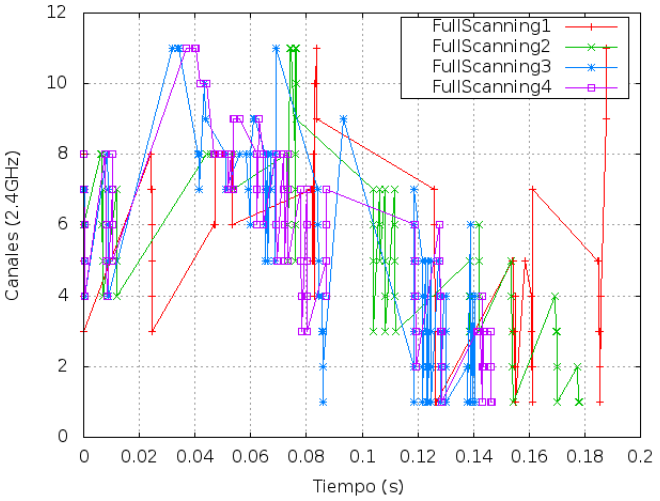

Figure 8: Secuencias de *Scanning* para Laptop HP

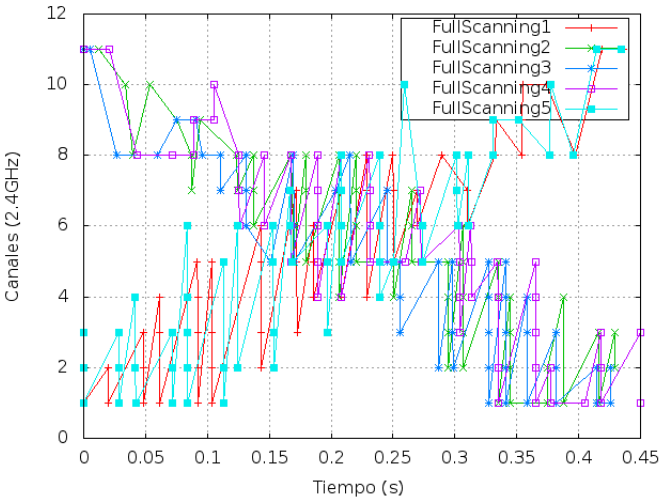

Figure 9: Secuencias de *Scanning* para iPhone 4S

de ejecución de *scanning* 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 full scanning que se realizan en secuencias ascendentes. Este comportamiento diferente resulta de la implementación particular de los algoritmos de scanning que el dispositivo Apple iPhone 4S posee y creemos que, buscaría encontrar APs más rápidamente.

B. Sobre la Frecuencia de Ejecución del Proceso Scanning

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

Table III: Frecuencia de Ejecución de Scanning 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

Table IV: Tiempos de Full Scanning para los Sistemas Operativos de los Dispositivos Evaluados

Sistema Operativo	Duración full scanning (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 $\sigma = 25,53178$

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 scanning hasta alcanzar los 60 segundos entre ejecución. El scanning 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 scanning 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 full scanning. Para el caso del sistema operativo iOS 6.1.2, se observó que la ejecución de full scanning se realiza por fases. Una fase inicial en la que el tiempo aproximado de ejecución entre full scanning no pudo ser determinado. Luego, una fase estable (o predecible) en la cual el tiempo entre ejecución de full scanning se encuentra alrededor de los 8 segundos.

C. Tiempo de Duración del Proceso de Scanning

Castignani, Arcia-Moret y Montavont [1] 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 full scanning que, corresponde al tiempo transcurrido durante el proceso de descubrimiento para escanear todos los canales disponibles en cualquier orden. En la figura 10 se muestra un bosquejo de la duración del proceso de full scanning.

En la Tabla IV se muestran los tiempos de duración promedio de full scanning 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 scanning tal como lo reportan Murray, Dixon y Koziniec [23].

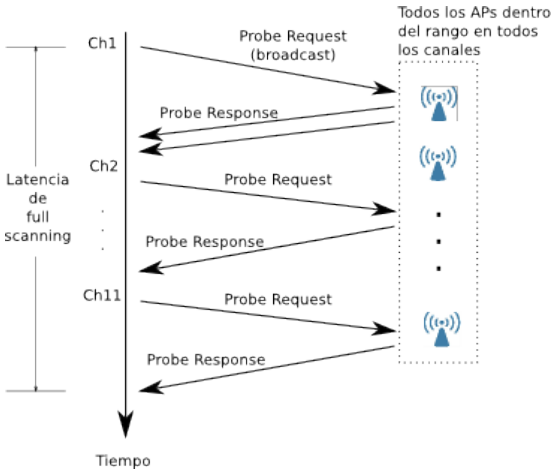

Figure 10: Bosquejo de la Duración de un Full Scanning

VI. Conclusiones y Trabajo Futuro

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

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

La frecuencia de ejecución del scanning 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 scanning real (la lista de canales propiamente dicha) para los dispositivos evaluados en estricto orden cronológico. El problema de identificación de la secuencia real de scanning puede ser abordado con estudios estadísticos y aplicación de algoritmos más elaborados para identificación de patrones.

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

Agradecimiento

Los autores agradecen al Centro de Desarrollo Científico, Humanístico y de las Artes (CDCHTA) de la Universidad de Los Andes por el financiamiento otorgado a través del proyecto I-1369-13-02-B.

References

[1] G. Castignani, A. Arcia-Moret, and N. Montavont, *A Study of Discovery Process in 802.11 Networks*, ACM SIGMOBILE Mobile Computing and Communications Review, New York, USA, 2011.

[2] Kim. I and Y. Kim, *Prediction-Based Smart Channel Scanning with Minimized Service Disruption for IEEE 802.11e*, IEEE International Conference on Consumer Electronic, Las Vegas, USA, 2011.

[3] A. Mishra, M. Shin and W.Arbaugh, *An Empirical Analysis of the IEEE 802.11 MAC Layer Handoff Process*, Process, SIGCOMM Comput. Commun, vol. 33, no. 2, pp. 93–102, New York, USA, 2003.

[4] *Sniffer de Paquetes y Sistema de Detección de Intrusos para Redes Inalámbricas 802.11*, http://www.kismetwireless.net.

[5] V. Gupta, R. Beyah, and C. Corbett, *A Characterization of Wireless NIC Active Scanning Algorithms*, Wireless Communications and Networking Conference, WCNC 2007, IEEE.

[6] R. Price and K. Heck, *Performance Comparison of Handover Mechanisms in Wireless LAN Network*, Inst. Comp. Sci., Würzburg Univ, Würzburg, Tech. Rep. 339, December, 2004.

[7] *IEEE. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE 802.11*, 2007.

[8] T. Laurenson, *Forensic Data Storage for Wireless Networks: A Compliant Architecture*, M.S. thesis, School of Computing and Mathematical Sciences, AUT Univ., Auckland, 2010.

[9] C. Corbett, R. Beyah, and J. Copeland, *Using Active Scanning to Identify Wireless NICs*, IEEE Information Assurance Workshop, 2006.

[10] P. Reddy, H. Sharme, and D. Paulraj, *Multi Channel WiFi Sniffer*. 4th International Conference onWireless Communications, Networking and Mobile Computing, 2008.

[11] *Modo Monitor de Operación en Redes Inalámbricas*, http://wireless.kernel.org/en/users/Documentation/modes.

[12] *Interfaz de Programación de Aplicaciones para Captura de Paquetes*, http://www.tcpdump.org.

[13] *Estándar de Facto para Inyección y Recepción de Tramas 802.11*, http://www.radiotap.org.

[14] *Raspberry Pi Placa Computadora Desarrollada por la Fundación Raspberry Pi*, http://www.raspberrypi.org.

[15] *SD Specifications. Part 1 Physical Layer Simplified Specification*, https://www.sdcard.org/downloads/pls/simplified_specs/part1_410.pdf.

[16] *Serial ATA. Estándar de Facto para Almacenamiento Interno en PC*, https://www.sata-io.org.

[17] *Universal Serial Bus Specification, Revision 2.0. 2000*, http://www.usb.org/developers/docs/usb_20_040413.zip.

[18] *Distribución GNU/Linux diseñada para Auditorías y Pruebas de Penetración Relacionadas con la Seguridad Informática en General*, http://www.backtrack-linux.org.

[19] *Suite Aircrack-ng para Romper Claves WEP y WPAPSK de 802.11*, http://www.aircrackng.org.

[20] *Programa Wireshark para Capturar y Analizar Trazas de Tráfico de Red en Sistemas Operativos Linux*, https://www.wireshark.org/docs/manpages/tshark.html.

[21] *Programa crontab para Administración de Procesos en Segundo Plano de UNIX/Linux*, http://pubs.opengroup.org/onlinepubs/9699919799/utilities/crontab.html.

[22] A. Nicholson, Y. Chawathe, M. Chen, B. Noble and D. Wetherall, *Improved Access Point Selection*, MobiSys'06 Proceedings of the 4th international conference on Mobile systems, applications and services, pp. 233–245, New York, USA, 2006.

[23] D. Murray. M. Dixon, T. Koziniec, *Scanning Delays in 802.11 Networks*, in proceedings of the 2007 International Conference on Next Generation Mobile Applications, Services and Technologies, pp. 255–260. Washington, DC, USA, 2007.