

## Scanning and securing wireless networks

Iván Forcada Atienza<sup>1</sup>, Juan Antonio Rodrigo Yanes<sup>2</sup>, José Antonio Gutiérrez de Mesa<sup>3</sup>

<sup>1</sup> Dep. Automática, Esc. Politécnica Superior, Univ. Alcalá,  
Alcalá de Henares, 28805 Madrid, Spain  
ivan@swscanner.org

<sup>2</sup> Dep. Automática, Esc. Politécnica Superior, Univ. Alcalá,  
Alcalá de Henares, 28805 Madrid, Spain  
jrodrigo@aut.uah.es

<sup>3</sup> Dep. Ciencias Computación, Esc. Politécnica Superior, Univ. Alcalá,  
Alcalá de Henares, 28805 Madrid, Spain  
jose.antonio@uah.es

**Abstract.** This paper presents Simple Wireless Scanner (SWScanner), an innovatory and easy-to-use application for Linux environments designed for scanning, configuring, and (in general) managing wireless networks. SWScanner has a modular design that allows novel Linux users to deal with common tasks related with wireless networks in a friendly graphical user interface, but without loosing the power of command line. SWScanner also makes use of data obtained from GPS to find the exact localization of detected networks, wich convert SWScanner in a powerfull Linux wardriving tool. Finally, SWScanner is designed to have a high level of compatibility with other well-stablished applications developed for win32 environments.

# Análisis y seguridad de las redes inalámbricas

Iván Forcada Atienza<sup>1</sup>, Juan Antonio Rodrigo Yanes<sup>2</sup> y José Antonio Gutiérrez de Mesa<sup>3</sup>

<sup>1</sup> Dep. Automática, Esc. Politécnica Superior, Univ. Alcalá  
Alcalá de Henares, 28805 Madrid, España

[ivan@swscanner.org](mailto:ivan@swscanner.org)

<sup>2</sup> Dep. Automática, Esc. Politécnica Superior, Univ. Alcalá  
Alcalá de Henares, 28805 Madrid, España

[jrodrigo@aut.uah.es](mailto:jrodrigo@aut.uah.es)

<sup>3</sup> Dep. Ciencias Computación, Esc. Politécnica Superior, Univ. Alcalá  
Alcalá de Henares, 28805 Madrid, España

[jose.antonio@uah.es](mailto:jose.antonio@uah.es)

**Abstract.** En el presente trabajo se presenta SWScanner, una sencilla e innovadora herramienta para entornos Linux diseñada para la detección y control de redes inalámbricas. El diseño modular de SWScanner permite sacarle el máximo partido tanto a usuarios noveles como a usuarios avanzados tanto de Linux como de redes inalámbricas. SWScanner utiliza los datos ofrecidos por un terminal para la localización exacta de las redes detectadas. Por último, SWScanner ha sido diseñado para mantener un elevado grado de compatibilidad con otras herramientas de similares características desarrolladas para entornos win32.

## 1 Introducción

El concepto de computación ubicua está siendo modificado por la aparición de nuevas tecnologías para las que es inherente la ubicuidad. Este es el caso de las redes inalámbricas, en las que los programas se pueden ejecutar en cualquier punto de la red. Entendiendo por punto de la red no sólo el nodo físico, si no también la posición física del nodo en el espacio.

El presente trabajo presenta una herramienta desarrollada para obtener el plano de las redes a las que se tiene acceso desde un punto dado.

## 2 Redes inalámbricas

Una red de área local inalámbrica (WLAN) puede definirse como una red de alcance local que tiene como medio de transmisión el aire. Por red de área local entendemos una red que cubre un entorno geográfico limitado, con una velocidad de transferencia de datos relativamente alta (mayor o igual a 1 Mbps tal y como especifica el IEEE), con baja tasa de errores y administrada de forma privada [4][8].

El gran atractivo de las redes inalámbricas está en la combinación de flexibilidad, ubicuidad de la red y distancia posible entre nodos, que hace que las redes inalámbricas superen a las redes cableadas en muchos aspectos[10][21]. Además, las redes inalámbricas responden a un importante e inherente rasgo que caracteriza a los seres humanos: su movilidad[14]. Por todo esto, la tecnología inalámbrica aplicada a redes de área local (WLAN), concretamente la basadas en los estándar IEEE802.11x han sufrido un tremendo auge en los últimos años y forman hoy día la alternativa más seria y económica para el consumidor [1][22].

El esquema más común de una arquitectura WLAN comprende la existencia de dos elementos: un punto de acceso, y un usuario final [3]:

- Los usuarios finales necesitan disponer de un elemento que le permita unirse a la red inalámbrica. Este elemento generalmente es una tarjeta de red inalámbrica.
- El punto de acceso es el elemento más importante del sistema, y se encarga entre otras muchas cosas del control de acceso a la red, control de flujo y difusión de su presencia (y por tanto de la existencia de la red) al medio que le rodea.

El parámetro más característico de una WLAN es el ESSID, acrónimo de *Extended Service Set Identifier*, comúnmente conocido como *nombre de red*. El ESSID es una cadena alfanumérica de 32 caracteres como máximo, que identifica un punto de acceso de manera no inequívoca. El ESSID es imprescindible en el proceso de asociación al punto de acceso (unión a la red) y, generalmente, el punto de acceso lo difunde embebido en unas tramas de control especiales.

Las bandas de frecuencia en las que operan los dispositivos con tecnología 802.11x (2.4 y 5 GHz.) [2]. En estas frecuencias son necesarios sistemas que detecten y muestren de la manera más amigable posible los puntos de acceso y, con ellos, las redes inalámbricas que nos rodean (típicamente a través del ESSID). Además, resulta conveniente dejar constancia de la existencia de las mismas fijando su posición geográfica. Este objetivo ha sido tradicionalmente conseguido combinando programas, dispositivos rastreadores de redes inalámbricas y dispositivos GPS (global positioning system). Analizando los datos obtenidos de forma conjunta, se puede situar exactamente la posición de una red inalámbrica.

Para satisfacer estos requerimientos es necesaria la existencia de:

- Una tarjeta de red inalámbrica con capacidad para rastrear el medio que la rodea.
- Un controlador capaz de interrogar adecuadamente a la tarjeta de red, indicándole el momento en que debe iniciar un rastreo. Posteriormente, el controlador debe ser capaz de recoger los resultados devueltos por la tarjeta para presentarlos a niveles superiores.
- Un programa que, utilizando las posibilidades del controlador, inicie el rastreo y le muestre al usuario los resultados obtenidos.

Al hecho de buscar, geolocalizar y analizar redes inalámbricas desplazándose en un coche se conoce con el término de *wardriving*. *Wardrive* fue el término original, del que han surgido algunos derivados, que hacen referencia a otros medios de transporte (por ejemplo: *warbiking*, *warwalking*, etc.).

Todas las tarjetas de red inalámbricas deben poseer mecanismos para rastrear el medio en busca de redes con las que asociarse. Sin embargo, no siempre existen controladores con la funcionalidad necesaria para iniciar un rastreo voluntario mediante la tarjeta de red inalámbrica que controlan. Además, en el caso de que el controlador disponga de estas funcionalidades, no siempre es fácil encontrar

programas que presenten y analicen adecuadamente los resultados que provienen de niveles inferiores [6][11].

En los entornos win32, los fabricantes han desarrollado controladores con todo tipo de capacidades diseñados para trabajar con los sistemas operativos Microsoft Windows ®. Una vez disponible el controlador, es cuestión de (generalmente poco) tiempo el que aparezcan aplicaciones que cumplan con los requisitos anteriormente expuestos. Concretamente, y para entornos win32, la aplicación más extendida (casi podría considerarse un estándar *de facto*) por su sencillez y versatilidad es NetStumbler.

En los entornos Linux existen dos importantes impedimentos, que han ocasionado un retraso respecto a la plataforma win32: En primer lugar, es poco frecuente que los fabricantes de tarjetas de red inalámbricas desarrollen controladores para entornos Linux y, se puede considerar habitual que los mismos fabricantes no hagan públicas las especificaciones técnicas de sus productos, lo que dificulta enormemente el trabajo de los desarrolladores de controladores. En segundo lugar, existe un número mucho menor de usuarios de Linux, en comparación con el número de usuarios de Windows ®, lo que inevitablemente implica un menor número de programadores capaces e interesados en desarrollar aplicaciones que resuelvan las necesidades anteriormente comentadas.

A pesar de todo, en los últimos años los avances en entornos Linux han sido tan claros, que se han reducido radicalmente las diferencias existentes frente a los entornos Windows ®.

Basándose en los trabajos realizados en los niveles inferiores por otras muchas personas, la herramienta que se presenta, SWScanner, es capaz de proporcionar unas funcionalidades muy similares a las ofertadas por NetStumbler; añadiendo, además, la posibilidad de almacenar y ejecutar configuraciones personalizadas para cada WLAN detectada, así como funciones para la conversión de los datos obtenidos a formatos estándar.

### **3 Soporte para redes inalámbricas en entornos Linux**

El soporte para los dispositivos WLAN en Linux se caracterizó por lo caótico y desorganizado hasta la serie 2.0 del núcleo. Anteriormente, no había soporte oficial para estos dispositivos en el núcleo, y había que recurrir a parches. Además, el estado de los controladores para las tarjetas de red inalámbricas era extremadamente rudimentario.

Una gran parte de los avances realizados en este campo han ocurrido gracias al trabajo de Jean Tourrilhes [24]. Él desarrolló la primera API (*Application Program Interface, Interfaz de Programación de Aplicaciones*) para manejar tarjetas inalámbricas frente al núcleo que fue incluida en éste desde la versión 2.0. A esta API se la conoce como las *wireless extensions* (WE) [35] ya que provee de funcionalidad añadida, a la ya existente para manejar tarjetas de red cableadas, que dota al usuario de la capacidad necesarias para manejar los dispositivos WLAN. Así, las WE permiten al controlador escribir en el espacio de usuario la configuración y las estadísticas específicas de una interfaz WLAN. El éxito de las WE se basa en que, con un solo grupo de herramientas o utilidades, es posible dar soporte a todas las

variaciones de WLANs, independientemente del tipo, siempre y cuando el controlador sea desarrollado con soporte para las WE.

Por otro lado, Jean Tourrilhes desarrolló una serie de utilidades para actuar sobre las WE y, así, permitir a los desarrolladores disponer de todas las funcionalidades que permiten las WE junto con el controlador subyacente. Estas utilidades reciben el nombre de *wireless tools* (WT) [35], y pueden definirse como el conjunto de utilidades que permiten manipular las WE. Las WT son herramientas en modo consola, es decir, que no poseen interfaz gráfica. Por ello, para muchas personas pueden parecer de difícil manejo, aunque soportan las WE al completo y son, por tanto, el diseño de referencia. Antes de la aparición de las WE se desarrollaban herramientas diseñadas específicamente para trabajar con un modelo o con un controlador. Es el caso de *wicontrol* para wavelan y *ancontrol* para aironet.

Por debajo de las WE debemos encontrar el controlador correspondiente al equipo de WLAN que se esté empleando. Dicho controlador deberá explotar las características del equipo subyacente, y deberá proporcionar una interfaz adecuada a las WE, es decir, que debe soportarlas. Por ejemplo, el controlador debe saber cómo preguntar a la tarjeta WLAN la calidad de la señal del enlace, para ello debe contener los mecanismos que desempeñan esa funcionalidad a través de la llamada al sistema que las WE dictaminan para obtener esa información. De poco serviría que el controlador sea capaz de sacarle todo el partido a la tarjeta WLAN, si no proporciona la interfaz adecuada para que las WE (y por encima de ellas, las WT) puedan mostrarlo en el espacio de usuario. De la misma manera, de poco sirve que las WE dispongan de la interfaz adecuada (llamadas al sistema) para obtener una información concreta, si el controlador utilizado no soporta las operaciones necesarias sobre la tarjeta WLAN para obtenerla. Por tanto, para poder sacarle el máximo partido a la tarjeta WLAN es necesario que el controlador soporte todas las funcionalidades de la tarjeta, y que las presente a los niveles superiores con la interfaz que dictaminan las WE.

A modo de resumen, podemos enumerar las partes de un sistema WLAN en Linux:

- Equipo.
- Controlador.
- El paquete *Wireless Extensions*.
- Las aplicaciones de espacio de usuario (Wireless Tools).
- Las aplicaciones de espacio de usuario de alto nivel (SWScanner).

#### **4 La aplicación SWScanner (Simple Wireless Scanner)**

SWScanner es una aplicación para entornos Linux, programada para KDE, basándose en las librerías gráficas QT, desarrolladas por Trolltech ®. Las librerías QT son la base del entorno de escritorio KDE, entre otras muchas aplicaciones gráficas multiplataforma [16].

La creación de SWScanner estuvo desde el primer momento firmemente ligada a tres premisas:

1. Diseño amigable, y de fácil manejo y configuración, especialmente pensado para usuarios recientemente llegados a entornos Linux y, por tanto, poco experimentados con la línea de comandos.
2. Estructura modular. Cada funcionalidad, aunque relacionada con el resto, debe ser capaz de operar independientemente. De esta manera, se obtienen algunas ventajas:
  - 2.1. Mejora a nivel organizativo en general.
  - 2.2. Posibilidad de desarrollar cada módulo o funcionalidad independientemente del resto.
  - 2.3. Permite el uso de los módulos o funciones que sean necesarios o, visto de otra manera, permite utilizar algunas partes de SWScanner aunque el equipo no permita el uso de otras.
3. Buscar (aparte de las mismas funcionalidades) la máxima compatibilidad posible con NetStumbler, la aplicación con mayor extensión, aceptación y calidad hasta el momento. Esta compatibilidad es imprescindible si en algún momento se pretende que SWScanner se extienda y tenga una aceptación comparable, ya que permite reutilizar el material recolectado con NetStumbler para su uso con SWScanner y viceversa.

La ventaja 2.3 implica que SWScanner será de utilidad en un elevado número de situaciones, aunque el controlador de la tarjeta de red inalámbrica que esté siendo utilizada no soporte la búsqueda de redes [13], ya que, incluso en este caso, SWScanner podría ser utilizado para convertir los datos provenientes de NetStumbler, o para gestionar las redes inalámbricas introducidas a mano en un fichero con la sintaxis adecuada, como se verá posteriormente.

#### **4.1 Funcionalidades y características de SWScanner:**

- Realización de análisis activos del entorno inalámbrico, en busca de los puntos de acceso de los alrededores. Los puntos de acceso encontrados se presentan en forma de tabla, junto con los siguientes datos: ESSID, MAC, canal, nivel de señal, nivel de ruido, relación señal/ruido (SNR), valor máximo de señal, valor máximo de relación señal/ruido y cifrado WEP activado o no [12],[23].
- Las filas de la tabla pueden ser organizadas en función de cualquiera de los campos anteriormente descritos.
- Filtrado del contenido de la tabla, en función de ESSID, canal o MAC [11].
- Datos actualizados en tiempo real del estado de la conexión en caso de estar asociados a un punto de acceso (nivel de señal, ruido, SNR, dirección IP, etc) .
- Configuración mediante la edición de un fichero de configuración, para lo que se dispone de los menús y formularios correspondientes.
- Posibilidad de almacenar el listado de puntos de acceso encontrados en un fichero de texto con formato xml para su posterior utilización.
- Permite configurar, para cada punto de acceso detectado (red inalámbrica), todos los parámetros de red necesarios para una posterior asociación a dicha red. Estos parámetros (cifrado WEP, ip estática o asignada por dhcp, essid de la red, mac del punto de acceso con el que asociarse y programa que se ejecutará tras la

asociación) son almacenados en una base de datos (sqlite) y estarán disponibles en cualquier momento posterior para permitir que el usuario se asocie a la red inalámbrica con una simple pulsación del ratón [6].

- Conexión con GPS y presentación de los datos en tiempo real (latitud, longitud, velocidad y número de puntos de acceso activos en cada instante). La configuración del puerto serie se realiza a través de un formulario de configuración en modo gráfico, al igual que el resto de los parámetros de configuración.
- Combinación en tiempo real de los datos recibidos desde el GPS con los datos recibidos de la tarjeta de red, en lo relativo a las redes inalámbricas detectadas, para generar un archivo de texto compatible con los generados por NetStumbler.
- Posibilidad de abrir tanto los ficheros xml generados por SWScanner, como los ficheros de texto generados por NetStumbler.
- Conversión de los datos de localización de las redes almacenados en ficheros de texto (tanto por SWScanner como por NetStumbler) en archivos de puntos con formato ESRI Shapefiles. Éste es un formato estándar muy utilizado por importantes aplicaciones de geolocalización.
- Paso de argumentos por la línea de comandos para abrir un fichero o pre-seleccionar el interfaz de red.

#### 4.2 Módulo de rastreo de las redes inalámbricas circundantes.

Para rastrear [13], SWScanner se basa en parte del código perteneciente a las WT, desarrolladas por Jean Tourrilhes. Una aproximación (quizás más sencilla) podría haber sido utilizar las WT tal cual, y analizar su salida para mostrar los resultados en la lista de redes de SWScanner. La otra aproximación (la finalmente tomada) consiste en extraer la parte del código relacionada con el rastreo de redes de las WT, adaptarla y mejorarla en SWScanner. Analizamos las dos aproximaciones en la tabla 1:

**Tabla 1.** Comparación de las opciones de diseño de la parte de rastrear redes.

	Primera aproximación	Segunda aproximación
Ventajas	– Sencillez de codificación.	– El desarrollo de SWScanner es autónomo y, por tanto, estático a no ser que se decida lo contrario. – Posibilidad de corregir errores o deficiencias de las WT, así como de incluir mejoras que las WT incluyan.

---

Inconvenientes	<ul style="list-style-type: none"> <li>– El desarrollo de SWScanner depende del desarrollo de las WT. Puede implicar mayor mantenimiento.</li> <li>– Los fallos en las WT implican fallos en SWScanner.</li> </ul>	– Mayor complejidad en el código
----------------	--	----------------------------------

Por otro lado, cabe destacar que la parte de SWScanner que controla el rastreo de redes, debe de ejecutarse en un hilo diferente al hilo de ejecución de la interfaz gráfica del usuario (GUI). El hacerlo así permite una mayor fluidez de la GUI, ya que en caso contrario (y aunque en la mayor parte de los casos el controlador y la tarjeta responden con los resultados casi inmediatamente) hay algunos modelos de tarjeta (especialmente las denominadas *tri-banda*, por rastrear un mayor número de canales) y de controladores que necesitan un tiempo considerable. Durante este tiempo la GUI estaría bloqueada.

Por supuesto, la utilización de hilos en la programación implica el uso de los mecanismos pertinentes para exclusión mutua (mutex) y para evitar los temidos *bloqueos de la muerte (deadlock)* entre las variables compartidas por más de un hilo.

Además, para la comunicación de los distintos hilos *satélites* con el hilo principal (GUI) se ha utilizado un sistema de eventos. Para ello, se devinieron nuevos eventos, que fueron utilizados para indicar determinadas circunstancias a la GUI como, por ejemplo, que un rastreo ha concluido y debe re-dibujar la lista de redes con los nuevos datos disponibles.

### 4.3 Módulo de captura de los datos provenientes del GPS.

Actualmente, la gran mayoría de los terminales GPS del mercado con capacidad de comunicarse con el exterior, lo hacen a través del estándar RS-232. Por ello, cualquier PC con puerto serie (u otro tipo de puertos con los conversores pertinentes) pueden recibir datos del GPS. Basta con configurar el puerto serie tal y como requiera el terminal GPS.

SWScanner facilita la configuración del puerto serie, y queda a la espera de recibir datos del terminal GPS cuando voluntariamente se le insta a iniciar la escucha. Cuando recibe datos, desecha aquellos que no llevan información *interesante* y extrae los valores de latitud, longitud y velocidad en tiempo real, tal y como los captura el GPS.

Cabe destacar que la llamada de lectura de datos provenientes del puerto serie es una llamada bloqueante, por lo que para evitar el bloqueo completo de la aplicación mientras no lleguen datos del GPS, es necesario separar la parte que recibe datos del GPS en un hilo diferente al principal (GUI). Para la comunicación entre este hilo y el principal, se definió un nuevo evento específico. Cuando el hilo correspondiente a la lectura de datos del GPS dispone de nuevos datos, le indica al hilo principal mediante este nuevo evento que dispone de datos nuevos.



#### **4.4 Módulo de almacenamiento de los datos combinados.**

Cuando el hilo principal dispone de datos provenientes del GPS y del controlador de la tarjeta de red para las redes inalámbricas (y sus características), pone en marcha los mecanismos que depositan todos los datos disponibles, en un formato compatible con el usado por NetStumbler, en un fichero previamente seleccionado.

#### **4.5 Módulo de conversión de datos.**

Los datos almacenados en un fichero de texto tienen (en bruto) poca o ninguna utilidad. Tras considerarlo, se decidió incluir la posibilidad de convertir estos datos *en bruto* a un formato más extendido y estandarizado. Concretamente, se decidió utilizar el formato de georepresentación utilizado por los principales sistemas de información geográfica (SIG ó GIS, en inglés). Se trata del estándar de representación vectorial de ESRI.

Los archivos que siguen el estándar del ESRI son archivos vectoriales, es decir, que guardan la información de puntos y líneas, a través de los cuales construyen cualquier imagen (de ahí su nombre: ESRI *Shapefiles*, archivos de formas de ESRI) conservando los atributos geográficos embebidos en ella. Realmente los datos se almacenan en archivos binarios, apoyados en una base de datos con formato dBase que guarda los atributos que se deseen añadir a cada forma (punto o línea).

En el caso de SWScanner, solamente es necesario utilizar archivos vectoriales de puntos, que referencian el punto donde una red inalámbrica fue detectada con mayor nivel de señal. Gracias a la base de datos (dBase) de apoyo que llevan estos archivos de puntos de ESRI, se puede añadir a cada punto atributos como ESSID, nivel de señal, ruido, etc. Que, posteriormente, podrán ser utilizados por un SIG para presentar una visión geográfica real, en la que se puede mostrar la información incluida en la base de datos.

SWScanner ofrece, a diferencia de NetStumbler, la posibilidad de convertir los archivos de texto, que contienen los datos combinados de redes y localizaciones, en archivos de formas de ESRI (cuyo tipo de datos son puntos), en los que se han añadido todos los atributos referentes a la red que la tarjeta es capaz de detectar (ESSID, señal, ruido, SNR, cifrado, etc.). De esta manera, SWScanner ofrece una manera sencilla de exportar los datos obtenidos al realizar un rastreo con un vehículo, a todos los sistemas de georepresentación que soporten el formato de ESRI. SWScanner es el primer programa con la posibilidad de trabajar sobre vehículo que integra esta característica.

La estructura modular de SWScanner permite utilizar esta característica de conversión de datos incluso en el caso de que el controlador de la tarjeta no soporte rastreos. Permite, incluso, convertir datos procedentes de NetStumbler.

#### **4.6 Módulo de gestión de redes.**

Una vez que SWScanner ha detectado una red inalámbrica, y ésta aparece en la tabla con la lista de redes, se puede acceder (pulsando el botón derecho del ratón) al menú de configuración, en el que podemos configurar los parámetros de red de la conexión

con ella. Aparte de los parámetros de red (ip fija o asignada por dhcp y puerta de enlace), podemos especificar de ser necesario la clave WEP con la que debemos acceder a la red.

Estos datos son almacenados en una base de datos sqlite. El empleo de esta base de datos radica en la facilidad de manejo y configuración que se consiguen por el hecho de que sqlite almacena una base de datos por archivo. Esta facilidad está de acuerdo con la primera premisa que se siguió al desarrollar SWScanner: la sencillez. Ciertamente es que sqlite no es una base de datos tan robusta y estable como otras, pero cubre sobradamente los objetivos y el volumen de datos que SWScanner requiere.

Una vez que una red inalámbrica ha sido configurada, y su configuración almacenada en la base de datos, SWScanner facilitará el proceso de asociación a dicha red, con un simple *click*, sin tener que configurar claves ni direcciones IP.

De esta manera, una vez hayamos configurado todas las redes por las que nos movemos, podremos cambiar entre ellas cómodamente, de forma análoga a como permite operar Microsoft Windows®, que era un sistema hasta ahora inexistente en los entornos Linux.

La manera de llevar a cabo estas tareas se basa en el uso de las llamadas al sistema del tipo *ioctl*. Estas llamadas se definen específicamente en función del dispositivo que se desea configurar. Concretamente, las llamadas para configurar una red TCP/IP se encuentran definidas en `linux/sockios.h`, mientras que las llamadas para configurar una red inalámbrica, pertenecientes a las Wireless Extensions, están en `linux/wireless.h`.

La estructura modular de SWScanner permite utilizar su capacidad de gestión de redes, incluso en los casos en que el controlador no soporte rastreos, ya que basta con añadir las redes a configurar manualmente o, incluso, utilizar NetStumbler para detectar las redes de interés y, luego, utilizar sus resultados con SWScanner.

#### 4.7 Módulo de análisis de redes.

El hecho de que SWScanner muestre de una manera gráfica y vistosa las redes inalámbricas que encuentra es, de por sí, un paso importante que, sin embargo, se quedaría cojo de no disponerse de un sistema de análisis. Éste es realmente necesario cuando se trabaja con varias de redes. SWScanner presenta en la parte izquierda de la pantalla un menú en forma de árbol donde se puede seleccionar el tipo de filtro que se quiere aplicar a los datos existentes en la tabla-lista de redes.

SWScanner permite un análisis basado en los siguientes filtros:

- Filtro por ESSID: Al seleccionar un ESSID concreto, se ocultarán de la lista todas las redes cuyo ESSID no se corresponda con el seleccionado.
- Filtro por canal: Es un mecanismo análogo que utiliza el dato canal.
- Filtro por dirección MAC: Éste es el filtro más restrictivo, dada la *unicidad* de las direcciones MAC. Es muy útil cuando se quiere hacer un seguimiento a un punto de acceso exclusivamente.

Cabe destacar que los filtros pueden ser ejecutados tanto si los rastreos están activados como si no lo están.

Por otro lado, existe otro nivel de filtrado que permite mostrar exclusivamente las redes *visibles* en cada instante o, por el contrario, mostrar todas las redes detectadas

(actuales y pasadas). Además, la lista de redes que SWScanner muestra, puede presentarse ordenada por todos y cada uno de los campos de los que dispone, tanto de forma ascendente como descendente.

Nuevamente gracias a la estructura modular de SWScanner es posible realizar estas tareas de análisis incluso si nuestro controlador no soporta rastreos, ya que es posible utilizar las listas de redes provenientes de otro ordenador personal o del mismo, pero utilizando Microsoft Windows<sup>®</sup> y NetStumbler.

## 5 Conclusiones y futuro trabajo

La herramienta SWScanner es una aplicación para la gestión de redes inalámbricas en entornos Linux. La aplicación está totalmente operativa y se puede obtener en:

<http://www.swscanner.org/es/index.htm>

El éxito obtenido, medido en las descargas realizadas desde que la aplicación se presentó en Internet, ha sobrepasado todas las expectativas.

En la actualidad se están realizando mejoras en base a las sugerencias de los usuarios.

Por otra parte se está estudiando la posibilidad de dispersar los procesos de la aplicación entre los nodos de la red.

### 2.3 Figures and Photographs

<http://www.swscanner.org/es/index.htm>

Please produce your figures electronically, if possible, and integrate them into your text file. For LaTeX users we recommend using the style files psfig or epsf (see Sect. 3).

Check that in line drawings, lines are not interrupted and have constant width. Grids and details within the figures must be clearly readable and may not be written one on top of the other. Line drawings should have a resolution of at least 800 dpi (preferably 1200 dpi). For digital halftones 300 dpi is usually sufficient. The lettering in figures should have a height of 2 mm (10-point type). Figures should be scaled up or down accordingly. Please do not use any absolute coordinates in figures. If possible, the files of figures (e.g. PS files) should not contain binary data, but be saved in ASCII format.

If you cannot provide your figures electronically, paste originals into the manuscript and center them between the margins. For halftone figures (photos), please forward high-contrast glossy prints and mark the space in the text as well as the back of the photos clearly, so that there can be no doubt about where or which way up they should be placed.

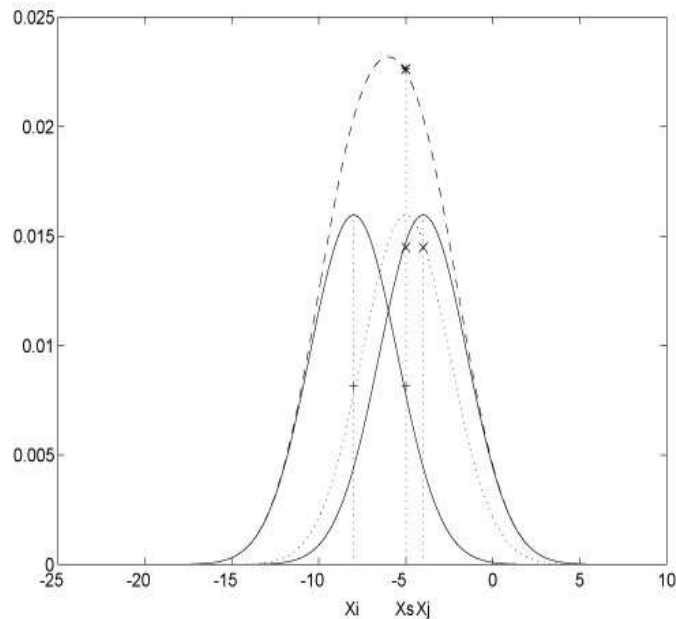
Figures should be numbered and should have a caption which should always be positioned *under* the figures, in contrast to the caption belonging to a table, which should always appear *above* the table. The final sentence of a caption, be it for a table or a figure, should end without a period. Please center the captions between the margins and set them in 9-point type (Fig. 1 shows an example). The distance

between text and figure should be about 8 mm, the distance between figure and caption about 5 mm.

If possible (e.g. if you use LaTeX) please define figures as floating objects. LaTeX users, please avoid using the location parameter h for “here”. If you have to insert a pagebreak before a figure, please ensure that the previous page is completely filled.

*Remark 1.* In the printed volumes, illustrations are generally black and white (halftones), and only in exceptional cases, and if the author is prepared to cover the extra cost for color reproduction, are color pictures accepted. If color illustrations are necessary, please send us color-separated files if possible. Color pictures are welcome in the electronic version at no additional cost.

*Remark 2.* To ensure that the reproduction of your illustrations is of reasonable quality we advise against the use of shading. The contrast should be as pronounced as possible. This particularly applies for screenshots.



**Fig. 1.** One kernel at  $x_s$  (*dotted kernel*) or two kernels at  $x_i$  and  $x_j$  (*left and right*) lead to the same summed estimate at  $x_s$ . This shows a figure consisting of different types of lines. Elements of the figure described in the caption should be set in italics, in parentheses, as shown in this sample caption. The last sentence of a figure caption should generally end without a period

## 2.4 Formulas

Displayed equations or formulas are centered and set on a separate line (with an extra line or halfline space above and below). Displayed expressions should be numbered for reference. The numbers should be consecutive within each section or within the contribution, with numbers enclosed in parentheses and set on the right margin. For example,

$$x + y = z . \quad (0)$$

Please punctuate a displayed equation in the same way as ordinary text but with a small space before the end punctuation. LaTeX users can find more examples of how to typeset equations in the file llncs.dem (see Sect. 3).

## 2.5 Program Code

Program listings or program commands in the text are normally set in typewriter font, e.g., CMTT10 or Courier.

Example of a Computer Program from Jensen K., Wirth N. (1991) Pascal user manual and report. Springer, New York

```
program Inflation (Output)
  {Assuming annual inflation rates of 7%, 8%, and
  10%,...  years};
  const  MaxYears = 10;
  var    Year: 0..MaxYears;
         Factor1, Factor2, Factor3: Real;
begin
  Year := 0;
  Factor1 := 1.0; Factor2 := 1.0; Factor3 := 1.0;
  WriteLn('Year 7% 8% 10%'); WriteLn;
  repeat
    Year := Year + 1;
    Factor1 := Factor1 * 1.07;
    Factor2 := Factor2 * 1.08;
    Factor3 := Factor3 * 1.10;
    WriteLn(Year:5,Factor1:7:3,Factor2:7:3,
            Factor3:7:3)
  until Year = MaxYears
end.
```

## 2.6 Footnotes

The superscript numeral used to refer to a footnote appears in the text either directly after the word to be discussed or – in relation to a phrase or a sentence – following the punctuation sign (comma, semicolon, or period). Footnotes should appear at the

bottom of the normal text area, with a line of about 2cm in TeX and about 5cm in Word set immediately above them.

## **2.7 Citations**

The list of references is headed “References” and is not assigned a number in the decimal system of headings. The list should be set in small print and placed at the end of your contribution, in front of the appendix, if one exists. Please do not insert a pagebreak before the list of references if the page is not completely filled. An example is given at the end of this information sheet. For citations in the text please use square brackets and consecutive numbers: [1], [2], [3], ...

## **2.8 Page Numbering and Running Heads**

Your paper should show no printed page numbers; these are allocated by the volume editor. Please indicate the ordering of your pages by numbering the sheets in pencil at the bottom of the reverse side. Do not set running heads.

## **2.9 Printing Quality**

For reproduction we need sheets which are printed on one side only. Please use a high-resolution printer, preferably a laser printer with at least 300 dpi. We prefer the text to be centered on the pages (i.e., equal margins left and right and top and bottom). The format of the paper (A4, Letter, etc.) is irrelevant.

# **3 Using LaTeX or TeX**

You will get the best results and your files will be easiest to handle if you use LaTeX2e for the preparation of your camera-ready manuscript together with the corresponding Springer class file llncs.cls. Only if you use LaTeX2e can hyperlinks be generated in the online version of your manuscript.

If you are unable to use LaTeX2e you may use one of our old macro packages llncs (for LaTeX) or plncs (for TeX).

## **3.1 How to Access the Springer LaTeX2e , LaTeX, and TeX Macro Packages**

For users of LaTeX (or TeX) Springer-Verlag provides the macro package llncs for LaTeX (or plncs for TeX). The packages can be obtained by ftp/gopher or by email as follows:

---

<sup>1</sup> The footnote numeral is set flush left and the text follows with the usual word spacing. Second and subsequent lines are indented. Footnotes should end with a period.

**Ftp:** The internet address is ftp.springer.de, the user ID is ftp or anonymous. Please enter your email address as password. The files (mentioned above) can be found in /pub/tex.

In the directory ftp://ftp.springer.de/pub/tex/latex/llncs/latex2e you will find all files belonging to the LaTeX2e package for LNCS. llncs.dem is a sample input file which you may use as a source for your own input. llncs.doc is the documentation of the class; llncs.dvi the resulting DVI file of llncs.doc.

**Gopher:** Point your client to ftp.springer.de.

**Mailserver:** Send an email message to svserv@vax.ntp.springer.de containing the line

get /tex/latex/llncs2e.zip	to get the LaTeX2e style files,
get /tex/latex/llncs.zip	to get the LaTeX style files, or
get /tex/plain/plncs.zip	to get the TeX style files.

Sending the command help to the server prompts advice on how to interact with the mail server. The style files must be unzipped and uu-decoded before use. In case of problems in getting or uu-decoding the style files please contact springer@vax.ntp.springer.de.

### 3.2 Further Instructions for LaTeX and TeX Users

Please always cancel any superfluous definitions that are not actually used in your text. If you do not, these may conflict with the definitions of the macro package, causing changes in the structure of the text and leading to numerous mistakes in the proofs.

When you use LaTeX or TeX and our macro packages, your text is typeset automatically in Computer Modern Roman (CM) fonts. Please do *not* change the preset fonts. If you have to use fonts other than the preset fonts, kindly submit these with your files.

Please use the commands \label and \ref for cross-references and the commands \bibitem and \cite for references to the bibliography, to enable us to create hyperlinks at these places.

For preparing your figures electronically and integrating them into your TEX file we recommend using the style files psfig or epsf. These can be downloaded from the DANTE ftp server at the locations

ftp://ftp.dante.de/tex-archive/graphics/psfig/psfig.sty or  
ftp://ftp.dante.de/tex-archive/systems/knuth/local/lib/epsf.tex.

These styles have always worked smoothly with our macro package. For further details about figure preparation see Sect. 2.3. In general, please refrain from using the \special command.

Remember to submit the psfig or epsf files and further style files and fonts you have used together with your source files.

## **4 Using MS Word**

We do not encourage the use of MS Word, particularly as the layout of the papers (the position of figures and paragraphs) can change from printout to printout. Having said this, we do provide the template sv-lncs.dot to help MS Word users prepare their camera-ready manuscript and to enable us to use their source files for the online version.

The template sv-lncs.dot and its documentation can be downloaded from the LNCS Web page at <http://www.springer.de/comp/lncs/authors.html>.

## **5 Supplementary Material**

If you wish to include color illustrations in the electronic version in place of or in addition to any black and white illustrations in the printed version, please provide the volume editors with the appropriate files.

If you have supplementary material, e.g., executable files, video clips, or audio recordings, on your server, simply send the volume editors a short description of the supplementary material and inform them of the URL at which it can be found. We will add the description of the supplementary material to the online version of LNCS and create a link to your server. Alternatively, if this supplementary material is not to be updated at any stage, then it can be sent directly to the volume editors, together with all the other files.

## **6 Copyright Form**

Until now, we have always had a very liberal policy regarding copyright. However, we have now had to introduce a copyright form, which we ask contributing authors to complete and sign. (It is sufficient if one author from each contribution signs the form on behalf of all the other authors.) The copyright form is located on our Web page at <http://www.springer.de/comp/lncs/copyrigh.html>. The printed form should be completed and signed and sent on to the volume editors either by normal mail or by fax, who then send it on to us, together with the printed manuscript.

## **7 Checklist**

When submitting your camera-ready manuscript to the volume editors, please make sure you include the following:

- a single-sided printout (not a photocopy) of the final version of your contribution (unless otherwise specified by the volume editor),
- your source (input) files, e.g. TEX files for the text and PS or EPS files for the figures,



- RTF files (see Sect. 1.1),
- any style files, templates, and special fonts you may have used,
- the final DVI file (for papers prepared using LaTeX or TeX),
- the final PS file (not in reverse order),
- if possible, a PDF file of the final version of your contribution,
- the completed and signed copyright form.

If supplementary material is available, please provide the volume editors with

- a short description of the supplementary material,
- the supplementary material or the URL at which it can be found,
- the files of color figures for the electronic version.

## Referencias

1. Matthew Gast: 802.11 Wireless Networks: The Definitive Guide. O'Reilly (2002)
2. Lawrence Harte: Introduction to 802.11 Wireless LAN (WLAN): Technology, Market, Operation, Profiles, & Services. ALTHOS (2004)
3. Frank Ohrtman and Konrad Roeder: Wi-Fi Handbook: Building 802.11b Wireless Networks. McGraw-Hill (2003)
4. Pejman Roshan, Jonathan Leary: 802.11 Wireless LAN Fundamentals. Cisco Press, Pearson (2003)
5. <http://home.mira.net/%7Egmb/gps/nmea.html>
6. [http://media.pearsoncmg.com/aw/aw\\_kurose\\_network\\_2/applets/csma-ca/withhidden.html](http://media.pearsoncmg.com/aw/aw_kurose_network_2/applets/csma-ca/withhidden.html)
8. <http://mia.ece.uic.edu/~papers/net/pdf00001.pdf>
9. <http://shapelib.maptools.org/>
10. <http://wi-fiplanet.com/>
11. <http://www.comlab.hut.fi/opetus/333/2004slides/topic11.pdf>
12. <http://www.comlab.hut.fi/opetus/333/2004slides/topic15.pdf>
13. <http://www.comlab.hut.fi/opetus/333/2004slides/topic26.pdf>
14. <http://www.commsdesign.com/main/2000/06/0006stand.htm>
15. [http://www.control.hut.fi/Kurssit/AS-74.340/slides\\_and\\_reports\\_adhoc\\_net/MAC%20PROTOCOLS\\_Hakan\\_report.pdf](http://www.control.hut.fi/Kurssit/AS-74.340/slides_and_reports_adhoc_net/MAC%20PROTOCOLS_Hakan_report.pdf)
16. [http://www.eetasia.com/ARTICLES/2001MAY/2001MAY25\\_NTEK\\_DSP\\_AN.PDF](http://www.eetasia.com/ARTICLES/2001MAY/2001MAY25_NTEK_DSP_AN.PDF)
17. <http://www.elgps.com/documentos.html>
18. <http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>
19. [http://www.hpl.hp.com/personal/Jean\\_Tourrilhes/Linux/Tools.html](http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.html)
20. <http://www.intel.com/netcomms/technologies/wimax/303787.pdf>
21. [http://www.intelligraphics.com/articles/80211\\_article.html](http://www.intelligraphics.com/articles/80211_article.html)
22. <http://home.jwu.edu/jwright/presentations/detecting-detectors-bu.pdf>
23. <http://www.kjhole.com/Standards/WiFi/WiFi-PDF/WLAN4alt.pdf>
24. <http://www.netstumbler.com/>
25. <http://www.nmea.org/>