CASO DE ESTUDIO DE COMUNICACIONES SEGURAS SOBRE REDES MÓVILES AD HOC

Autor: Sergio Hernán Rocabado Moreno

Director: Javier Diaz (UNLP)
Codirector: Daniel Arias Figueroa (UNSa)

Tesis presentada para obtener el grado de Magister en Redes de Datos
Facultad de Informática – Universidad Nacional de La Plata
Diciembre de 2013

Dedico este trabajo a Prudencio y Lila,
Gracias por todo abuelitos!!

INDICE DE CONTENIDO

Capítulo 1: Introducción

En este capítulo se introduce al lector en el caso de estudio de esta tesis, se realiza una breve descripción del escenario de pruebas y de los principales aspectos considerados para su implementación. También se mencionan las principales motivaciones, el objetivo general y los objetivos específicos del trabajo. Finalmente se presenta la organización general de la tesis.

1.1 Descripción

Una red móvil ad-hoc o MANET (Mobile Ad hoc NETworks en inglés) es una colección de nodos inalámbricos móviles que se comunican de manera espontánea y autoorganizada constituyendo una red temporal sin la ayuda de ninguna infraestructura preestablecida (como puntos de acceso WiFi o torres de estaciones base celulares) ni administración centralizada. Los equipos o nodos que forman parte de ella (Notebooks, PDAs, Celulares), se organizan por si mismos para ayudarse los unos a los otros en el proceso de transportar paquetes de datos entre un origen y un destino. Por tanto las MANET dan un paso mas en cuanto a movilidad (todos los nodos de la red pueden ser móviles) y flexibilidad (no se requiere inversión en infraestructura, y se minimiza la gestión de la red pues se auto-organiza ella misma) [1].

Por sus características las MANETs constituyen una tecnología ideal para facilitar servicios de comunicación a dispositivos móviles en zonas remotas donde no es posible montar y configurar redes de infraestructura debido a inconvenientes físicos y recursos limitados, como la energía y/o cobertura de red celular.

Una ventaja adicional de este tipo de redes es la posibilidad de integrarlas a redes de infraestructura con diferentes fines; entre otros podemos mencionar: el acceso a Internet y a sistemas de información de una Intranet desde los dispositivos móviles que forman parte de la MANET [2]. Las características intrínsecas de este tipo de redes (incluyendo autoconfiguración, ausencia de infraestructura, movilidad de los nodos, topología dinámica, ancho de banda limitado, falta de seguridad, conservación de energía, entre otras), plantean exigencias que deben resolverse antes de realizar la integración.

El trabajo de investigación de esta tesis se enfoca en los siguientes aspectos:

- Seguridad. Las redes móviles utilizan un medio compartido (aire) para transmitir los datos y se encuentran expuestas a "ataques" o accesos no autorizados, y por esta razón se hace necesario utilizar protocolos de seguridad que permitan una integración "segura" de los dispositivos móviles a la red de infraestructura, garantizando el cumplimiento de los siguientes aspectos de seguridad: Confidencialidad, integridad, autenticación y no repudio.
- Conservación de Energía. Los dispositivos móviles que conforman la MANET tienen capacidad limitada de energía y pocas posibilidades para recarga de baterías cuando se encuentran en zonas remotas de recursos energéticos limitados, por lo tanto se debe optimizar el consumo de energía.

- Ancho de banda limitado. La integración de una MANET en zona remota a una red de infraestructura requiere el uso de la red celular. En este tipo de zonas la cobertura de red celular es muy baja y debido a ello proporciona un ancho de banda reducido y variable.

Los tres aspectos son importantes y están directamente relacionados, se debe tener en cuenta que la implementación de un protocolo de seguridad implica un consumo de energía adicional por tres motivos: se incrementa el uso de CPU y memoria para realizar cálculos, se generan encabezados adicionales (overhead) que deben ser transmitidos y se intercambian mensajes para el establecimiento de canales de comunicación seguros.

Por otra parte, la implementación de niveles de seguridad elevados implica: un incremento en el consumo de energía en los nodos móviles que reduce drásticamente el tiempo de vida de la red y un consumo adicional de ancho de banda que puede comprometer el normal funcionamiento de las aplicaciones. Debido a estas razones se hace necesario establecer un compromiso entre seguridad y consumo de recursos.

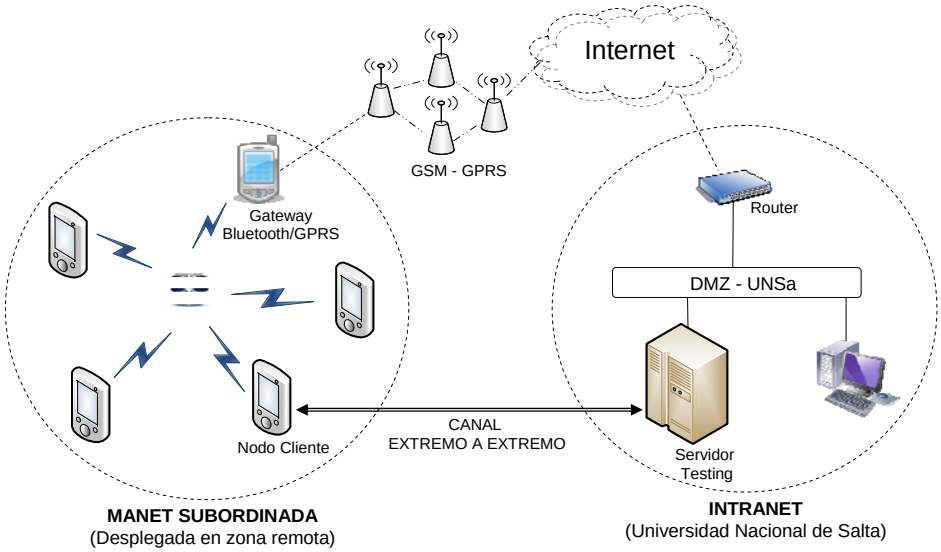

Figura 1-1: Escenario del caso de estudio

En este trabajo se presenta el estudio de un caso de integración de una MANET, desplegada en una zona remota, a una red de infraestructura. La finalidad principal es la de proporcionar, a los nodos de la red ad hoc, acceso "seguro" a un servidor de la red de infraestructura, sin comprometer recursos como ancho de banda y energía que son limitados en la zona de despliegue. Para ello, se implemento un escenario de pruebas (Figura 1-1) que comprende el despliegue de una MANET en zona remota y la integración de la misma a una red de infraestructura a través de la red celular. Sobre el escenario propuesto se establecieron canales de comunicación extremo a extremo, entre un nodo de la MANET y un servidor de infraestructura. Inicialmente, se realizaron pruebas inyectando tráfico de datos sobre un canal "no seguro" para obtener valores de referencia para latencia, throughput y consumo de energía. Luego, se efectuaron las mismas pruebas utilizando canales de comunicación "seguros" configurados sobre protocolos IPSEC y SSL/TLS. Los resultados obtenidos utilizando canales "seguros" fueron comparados con los valores de

referencia para determinar las diferencias de consumo de recursos. Las desviaciones que surgieron de estas comparaciones, permitieron:

- Establecer el consumo adicional de recursos generado por el uso de protocolos seguros.
- Realizar un estudio comparativo de rendimiento, entre diferentes configuraciones de protocolos de seguridad.
- Determinar que protocolo seguro se adapta mejor a este tipo de entornos.

1.2 Motivación

Las zonas rurales aisladas se caracterizan, entre otros aspectos, por su baja densidad demográfica, cobertura de red celular limitada y carencia de servicio de distribución de energía eléctrica. Sus habitantes utilizan energías alternativas, como paneles solares y grupos electrógenos, para cubrir necesidades energéticas elementales. En este contexto, el acceso a la información digital es prácticamente nulo debido al elevado consumo de energía que requieren los equipos computacionales.

La propuesta tecnológica de esta tesis, hace uso de redes móviles ad hoc y redes celulares para mejorar la accesibilidad a la información digital en zonas remotas de recursos limitados, y es aplicable a cualquier zona rural aislada del país que tenga características similares a las del escenario de estudio.

Desde una perspectiva social, el acceso a información digital desde zonas rurales de recursos limitados permitiría:

- Disminuir la brecha digital existente entre las poblaciones rurales aisladas y las poblaciones urbanas.
- Brindar soporte tecnológico a operaciones de emergencias en zonas aisladas que sufran desastres naturales, como ser incendios e inundaciones.
- Incrementar las posibilidades de comunicación de los pobladores rurales, permitiendo el acceso a aplicaciones de Internet tales como correo electrónico, mensajería y redes sociales.
- Implementar estrategias educativas de m-learning que mejoren el proceso de enseñanza aprendizaje en comunidades escolares rurales aisladas.
- Fomentar el turismo rural, brindando servicios de comunicación a zonas aisladas que formen parte de los circuitos turísticos.
- Mejorar el aprovechamiento de la tecnología disponible en zonas aisladas, teniendo en cuenta que por falta de recursos los equipos celulares son utilizados como reproductores de música o cámaras fotográficas y no como dispositivos de comunicación.

1.3 Objetivos

Objetivo general:

- Realizar una contribución para el despliegue seguro de redes móviles ad hoc, con integración a redes de infraestructura, en zonas de recursos limitados.

Objetivos específicos:

- Efectuar una revisión bibliográfica del estado del arte de seguridad en redes móviles ad hoc.
- Estudiar las principales métricas, técnicas y herramientas utilizadas para medir el rendimiento y el consumo de energía en dispositivos móviles.
- Implementar el escenario de pruebas para el caso de estudio, teniendo en cuenta los requerimientos de seguridad necesarios y las limitaciones de recursos propias de las zonas desfavorables.
- Realizar un estudio comparativo de protocolos de seguridad, en capa de transporte, que permitan integrar MANETs desplegadas en zonas remotas a redes de infraestructura.
- Seleccionar el protocolo de seguridad que mejor se adapte a este tipo de entornos, buscando un equilibrio entre el nivel de seguridad y el consumo de recursos como la energía y el ancho de banda que son limitados en la zona de despliegue.

1.4 Estructura de la tesis

El resto de la tesis está organizada de la siguiente manera:

El segundo capítulo pretende introducir al lector en los fundamentos de la tecnología Ad-hoc, así como su definición, funcionamiento, características y aplicaciones. En el tercer capítulo se presenta una revisión del estado del arte de la seguridad en redes móviles ad hoc con base en la literatura más relevante relacionada con la temática. En el cuarto apartado se describen en detalle las técnicas y herramientas para efectuar las mediciones de rendimiento y de consumo de energía en los dispositivos móviles que forman parte de una MANET, se fundamenta la elección de las métricas y herramientas utilizadas en el escenario de pruebas. En el capitulo quinto se detalla el trabajo de campo realizado, incluyendo la construcción del escenario de pruebas, implementación de canales seguros y mediciones efectuadas. El sexto capítulo expone los resultados mediante gráficos con sus correspondientes discusiones, además contiene las conclusiones principales del trabajo.

Capítulo 2: Redes Móviles Ad Hoc

2.1 Definición

Una red móvil ad hoc o MANET (del inglés Mobile Ad-hoc Networks) es una colección de nodos inalámbricos móviles que se comunican de manera espontánea y autoorganizada constituyendo una red temporal sin la ayuda de ninguna infraestructura preestablecida (como puntos de acceso WiFi o torres de estaciones base celulares con antenas 2G o 3G) ni administración centralizada.

La ausencia de infraestructura permite que una MANET se pueda configurar y desplegar rápidamente en cualquier lugar (interior o exterior) cuando sea necesario, pero también dificulta el establecimiento de los límites de la red [3].

Los nodos (dispositivos móviles con capacidades de comunicación inalámbrica como Smartphones, Tablets, PDAs, etc.) que forman parte de la red móvil ad hoc se auto-organizan para ayudarse los unos a los otros en el proceso de transportar paquetes de datos entre un nodo origen y un nodo destino.

Cada nodo de la MANET puede trabajar como emisor, receptor o reenviando los paquetes como si fuera un router, la comunicación en la red depende del rango de transmisión del dispositivo y de la confianza entre los nodos involucrados. La comunicación sólo se lleva a cabo si cada nodo coopera en el proceso de transmisión de los datos.

El siguiente algoritmo describe la comunicación en una MANET:

1. El nodo origen envía la señal a los nodos vecinos dentro de la zona de cobertura.
2. Los nodos vecinos se comunican con el nodo emisor.
3. Si el nodo destino es vecino del emisor entonces recibe el mensaje. Sino un nodo intermedio recibe el mensaje y reinicia el proceso a partir del paso número 1, reenviando el mensaje hasta alcanzar el nodo destino.

En la Figura 2-1 el dispositivo A requiere enviar un mensaje a C, como este se encuentra fuera de su alcance o cobertura envía el mensaje a B, este actúa como router y reenvía el mensaje a C. Para que esta comunicación sea segura se deben establecer relaciones de confianza entre los 3 nodos (A-B, A-C y B-C).

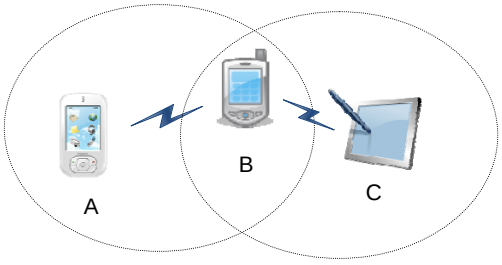

Figura 2-1: Comunicación en una MANET

Como se indica en el algoritmo de comunicación MANET, si dos nodos no vecinos (remotos) requieren comunicarse, lo hacen a través de nodos intermedios quienes se encargan de reenviar los mensajes, este tipo de comunicación se denomina: comunicación multi-salto (*multi-hop*). En la Figura 2-2 se observa una MANET formada por 5 nodos, la comunicación entre el nodo A y el nodo E se realiza a través de los nodos B, C y D que actúan como routers.

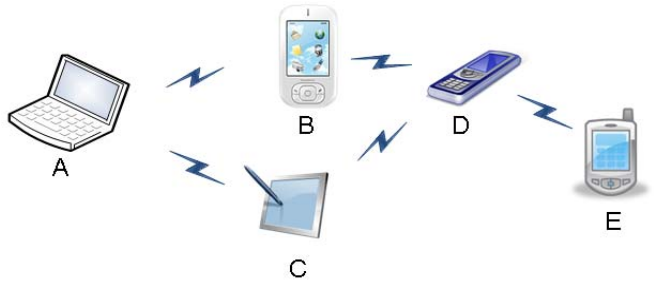

Figura 2-2: Comunicación Multi-salto en una MANET formada por 5 nodos

2.2 Clasificación

Las MANET pueden ser redes autónomas o subordinadas si se encuentran conectadas o integradas a otras redes externas con infraestructura (Internet, redes corporativas-Intranets, extranets o intranets interconectadas a través de Internet, redes celulares como GSM/GPRS/HSDPA/WCDMA, etc.).

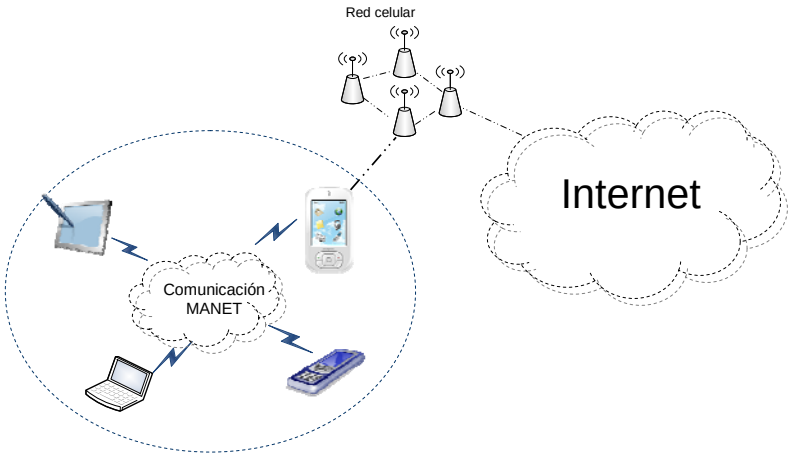

Figura 2-3: MANET subordinada a Internet

En la Figura 2-3 se ilustra un ejemplo de MANET subordinada que se conecta a Internet a utilizando los servicios de la red celular.

Otro ejemplo de MANET subordinada esta dado por aquellas que se integran a redes de infraestructura para acceder a servicios y recursos que se encuentran disponibles en la intranet de una organización. En la Figura 2-4 se puede observar una MANET que se integra a la intranet de una organización a través de la red celular, una vez establecida la comunicación los nodos de la MANET pueden acceder a los servicios facilitados por la red de la organización, por ejemplo: un Servidor de base de datos.

Es importante mencionar que los nodos móviles de una red ad hoc utilizan un medio compartido (aire) para transmitir los datos, lo cual hace que la red se encuentre expuesta a "ataques" o accesos no autorizados y por esta razón se hace necesario utilizar mecanismos de seguridad que permitan realizar una integración "segura" de los dispositivos móviles a la red de infraestructura, y de esta manera no comprometer la seguridad de la misma. La parte central de este trabajo de Tesis se centra en este tipo de cuestiones.

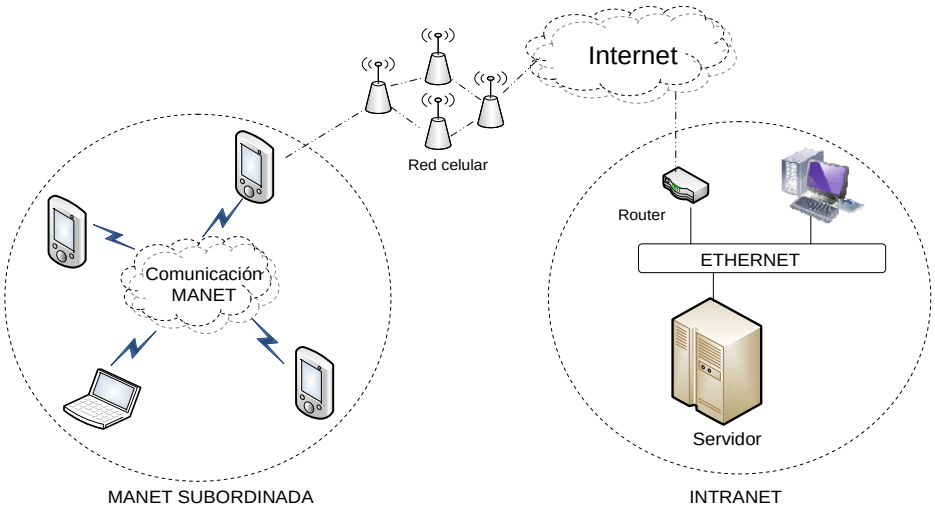

Figura 2-4: MANET subordinada a Intranet

2.3 Características

El RFC 2501 del grupo de trabajo para MANET en la IETF [4] destaca cuatro características de de las redes móviles ad hoc: Topología dinámica, ancho de banda limitado, operación con energía limitada.

Topología dinámica

La topología de la red puede cambiar rápida y aleatoriamente en tiempos impredecibles por las siguientes razones:

- Los nodos de una MANET son libres de moverse arbitrariamente, pueden hacerlo de forma independiente en diferentes direcciones y a diferentes velocidades lo que puede ocasionar que un nodo quede fuera del alcance de otros, creando particiones de la red o aislándose.
- Uno o más nodos pueden abandonar la red.
- Uno o más nodos se pueden unir a la red.

Este comportamiento dinámico de la topología dificulta el establecimiento de la conectividad en la red, la cual se debe mantener para permitir que los servicios de comunicación operen sin interrupciones.

Ancho de banda limitado

Los enlaces inalámbricos entre los nodos de una red móvil ad-hoc tienen un ancho de banda limitado y suelen ser muy inestables y propensos a errores de transmisión.

Operación con energía limitada

Los nodos en una MANET requieren de una batería para obtener la energía que asegure su funcionamiento. Por lo tanto la energía es uno de los recursos que se busca optimizar para prolongar el tiempo de vida de un nodo que no tenga posibilidades de recarga.

Seguridad física limitada

El hecho de hacer uso de un medio compartido (aire) hace que las MANETs sean vulnerables a determinadas amenazas a la seguridad, por lo que se deben implementar mecanismos de protección ante dichas amenazas.

A continuación se describen otras características importantes, no contempladas en el RFC 2501:

Auto-organización

La auto-organización engloba tres características:

- **Auto-formación**
  Los nodos que se encuentran dentro del rango de otros se vinculan entre sí para formar una red, sin necesidad de intervención humana.
- **Auto-reparación**
  La red se reorganiza automáticamente cuando los nodos se unen o abandonan la red, sin impactar en el funcionamiento de los demás nodos participantes.
- **Auto-protección**
  Los nodos resguardan la información que fluye a través de la red, defendiéndose contra amenazas que comprometan la seguridad de la red.

Existencia temporal

Las MANETs se forman generalmente con un propósito determinado, luego de cumplir su propósito cesan de existir. Esto significa que la red existe solo por un periodo de tiempo limitado.

Operación distribuida

Los nodos tienen un conocimiento local sobre su entorno. Es decir, cada nodo sólo conoce la información sobre los nodos vecinos que se encuentran dentro de su rango de transmisión y no tiene un conocimiento global de la red. Las funciones de ruteo y seguridad deben estar diseñadas de manera que puedan operar eficientemente bajo estas condiciones.

Rango de conectividad limitado

Los nodos de una MANET utilizan radio frecuencia (RF) para comunicarse, en consecuencia el rango de transmisión es muy limitado. Este rango puede ser incrementado si un nodo origen envía paquetes a los nodos vecinos que se encuentran en su rango de frecuencia, y estos a su vez operan como routers y reenvían los paquetes hasta que llegan a su destino (multi-salto) [3].

Recursos limitados

Además de las limitaciones de ancho de banda y energía mencionadas en [4], los dispositivos de una MANET (PDAs, Tablets, Smartphones, etc) presentan otras limitaciones tales como potencia de transmisión, capacidad de carga de la batería, disponibilidad de memoria y capacidad de procesamiento. Los mecanismos de seguridad se deben diseñar e implementar considerando estas limitaciones.

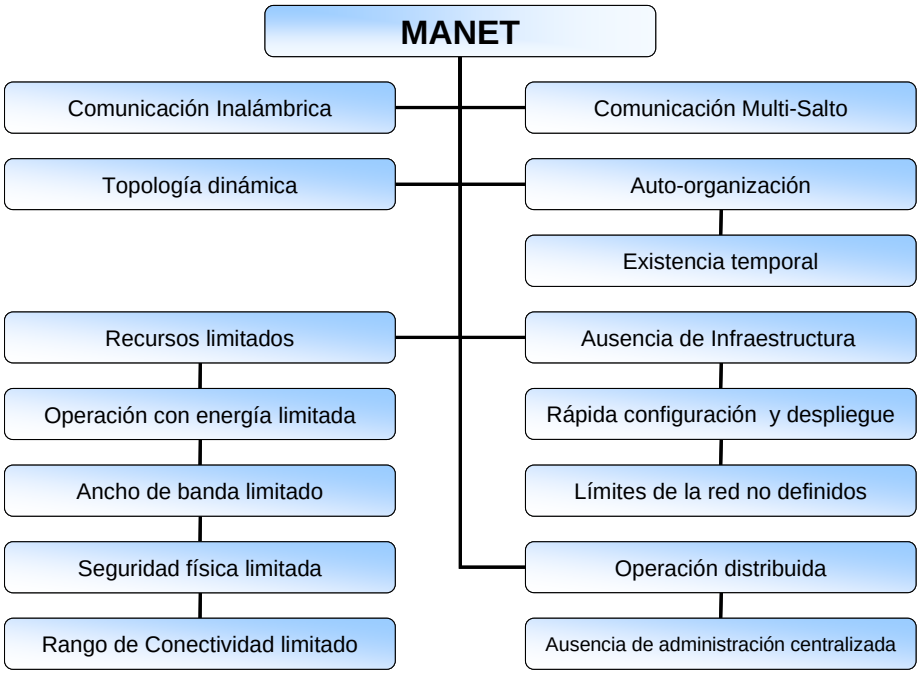

Figura 2-5: Características de una MANET

2.4 Debilidades

Ausencia de infraestructura

La ausencia de infraestructura preestablecida dificulta la implementación de soluciones clásicas de seguridad basadas en autoridades de certificación (CA) centralizadas. Existen 3 alternativas para la construcción de un entorno seguro, ambas permiten resolver cuestiones relacionadas con la confianza entre los nodos:

- Integrar la MANET a una red de infraestructura que disponga de una CA centralizada montada sobre servidores de alta disponibilidad.
- Descentralizar la autoridad de certificación configurando una CA distribuida de forma parcial o total.
- Implementar un modelo de confianza entre nodos, similar a PGP.

Tampoco es posible implementar un servidor de monitoreo centralizado, esto dificulta la detección de ataques porque es muy complicado monitorear el tráfico en MANETs de gran escala que presenten topologías dinámicas.

Nodos vulnerables

Las características de un nodo ad hoc, como movilidad y tamaño reducido, hacen que presente vulnerabilidades que lo convierten en un objetivo atractivo para potenciales atacantes. Algunas de estas vulnerabilidades son:

- Se utilizan en entornos no confiables.
  Los nodos de una MANET son utilizados en entornos con complejas relaciones de confianza, si una de las partes requiere poner un dispositivo en manos de otra, debe hacerlo teniendo la seguridad que la segunda parte no pueda modificar las características de seguridad del dispositivo. Además, debido a su tamaño reducido, los nodos pueden ser extraviados o robados y caer en manos de usuarios maliciosos. Se espera que un nodo ad hoc garantice las operaciones seguras aunque se encuentre en posesión de usuarios "no confiables".
- Vulnerabilidades inducidas por la conexión a redes inalámbricas.
  Los nodos ad hoc tienen instaladas interfaces de red inalámbricas, esto los expone a diferentes ataques (a los mecanismos de seguridad) que pueden ser realizados en forma remota.
- Ejecución de software descargado.
  La mayoría del software de los nodos ad hoc esta desarrollado por terceros y generalmente se descarga desde redes públicas como Internet. Esta circunstancia puede ser aprovechada por el atacante para introducir software malicioso como virus, gusanos y troyanos en el nodo.
- Complejidad en el proceso de diseño.
  El hardware de un nodo as hoc es normalmente desarrollado utilizando componentes de diferentes fabricantes. Si bien la seguridad de cada componente está certificada por el fabricante del mismo, se hace muy difícil verificar que la composición de las partes no exponga nuevas vulnerabilidades.

Vulnerabilidad del canal de comunicación

Los mensajes pueden ser escuchados y/o modificados por cualquiera que tenga acceso al medio compartido (aire).

Topología dinámica

Los nodos de pertenencia cambiante, que constantemente abandonan o se unen a la red, pueden perturbar las relaciones de confianza entre los nodos de la MANET, lo que implica un problema de identificación y control de accesos. La confianza también puede verse alterada si algunos nodos son detectados como comprometidos.

Si los cambios en la topología de la red ocurren con mucha frecuencia, los protocolos de enrutamiento serán complejos y la seguridad de los mismos un desafío adicional.

Capacidad de procesamiento limitada

Las operaciones criptográficas que garantizan niveles elevados de seguridad requieren gran capacidad de cómputos, que en general los procesadores de los dispositivos de una MANET no son capaces de realizar ya que sus capacidades de procesamiento son limitadas.

Energía limitada

Los dispositivos móviles que conforman la MANET tienen capacidad limitada de energía y pocas posibilidades para recarga de baterías cuando se encuentran en itinerancia, por lo tanto se debe optimizar el consumo de energía.

La implementación de niveles de seguridad elevados implica un aumento en el consumo de energía en los nodos móviles y reduce drásticamente el tiempo de vida de la red, por esta razón se hace necesario establecer un compromiso entre seguridad y consumo de energía.

Ancho de banda limitado

Los enlaces inalámbricos de las redes ad hoc son de capacidad reducida y muy susceptibles a los efectos del ruido, interferencia y atenuación de señal.

El uso de protocolos de seguridad requiere la transmisión de encabezados y tráfico adicional (*overhead*), estos protocolos no se pueden implementar si no se dispone de un ancho de banda adecuado.

Ausencia de colaboración entre los nodos

Al tratarse de redes que basan su funcionamiento en la colaboración entre sus nodos, cualquier comportamiento que pueda considerarse no colaborativo puede poner en riesgo la existencia misma de la MANET. Ej: Un nodo con baja carga de batería puede comportarse de manera egoísta, dejando de colaborar con la red y utilizando el remanente de energía para realizar tareas en beneficio propio.

2.5 Perfiles de los nodos

En un escenario ideal todos los nodos de una MANET deberían ser honestos, altruistas y colaborativos. Honestos porque actúan conforme a las reglas definidas para la red, altruistas porque facilitan sus recursos en beneficio de la red y colaborativos porque ejecutan su mejor esfuerzo para brindar soporte a los requerimientos de la red.

Esto no siempre es así, debido a que los nodos que forman parte de una MANET generalmente se encuentran en itinerancia y en riesgo permanente de caer en manos de terceros no autorizados, ya sea por robo o extravío. Los nodos que son robados o extraviados se denominan “nodos comprometidos”, un nodo comprometido en manos de un atacante puede adoptar comportamientos que pongan en riesgo el normal funcionamiento de la red, por lo que debe ser detectado rápidamente.

2.5.1 Comportamientos no deseados

En una MANET se pueden detectar tres tipos comportamientos no deseados que puede adoptar un nodo [5]: Malicioso, Egoísta y Averiado.

1. Nodo malicioso: Es aquel que persigue interceptar comunicaciones y falsear identidades para acceder a la información transmitida o también dañar de forma activa a la red para degradar su rendimiento o causar su interrupción.

2. Nodo egoísta: Es aquel que utiliza los servicios de la MANET pero no encamina adecuadamente los paquetes con destino a otros nodos que pasan por él, y así no consumir sus recursos (Batería, CPU y Memoria) en beneficio propio.
3. Nodo averiado: Es aquel cuyo comportamiento es inestable e inseguro debido a algún fallo de un componente de hardware y/o en el software que ejecuta, ya sea en el Sistema Operativo o en alguna aplicación.

2.6 Encaminamiento en redes ad hoc

El encaminamiento (ruteo o enrutamiento) es la actividad más importante que cumplen los nodos de una MANET para garantizar el normal funcionamiento de la red. Si uno de los nodos deja de encaminar correctamente los paquetes, puede comprometer el normal funcionamiento de toda la red ad hoc.

Los protocolos de encaminamiento diseñados para las redes cableadas o de infraestructura no son eficientes para su aplicación en redes móviles debido a la movilidad de sus nodos, por lo tanto han sido modificados con la finalidad de que ofrezcan mejores soluciones de enrutamiento en movilidad y se adapten al entorno dinámico de las redes móviles, permitiendo que superen problemas tales como topología dinámica, recursos de ancho de banda limitada y seguridad reducida.

Las tareas principales de los protocolos de encaminamiento para MANET son el descubrimiento y mantenimiento de las rutas entre los nodos que forman parte de la red.

2.6.1 Protocolos de encaminamiento reactivos y proactivos

Los protocolos de encaminamiento pueden ser reactivos (*On-Demand,* bajo demanda) o proactivos (*Table-Driven,* manejado por tablas) [6].

En los protocolos reactivos solo se realiza el descubrimiento de rutas cuando se requiere establecer una conexión. Estos protocolos presentan tiempos de descubrimiento de ruta bastante cortos ya que trabajan bajo demanda, de esta manera hacen que el tiempo de entrega de paquetes sea menor, utilizan mecanismos de descubrimiento y mantenimiento de ruta.

En los protocolos proactivos cada nodo mantiene una tabla con información de las rutas, periódicamente los nodos intercambian mensajes a fin de mantener actualizadas sus tablas. Utilizando la tabla de encaminamiento un nodo puede elegir la mejor ruta en cada instante y ofrecer una rápida respuesta ante solicitudes de ruta.

En la Tabla 2-1 se presenta una comparación entre protocolos de encaminamiento proactivos y reactivos.

Protocolos de ruteo	Proactivos	Reactivos
Descripción	- Manejado por tablas (Table-Driven). - Buscan rutas periódicamente, suponiendo que serán útiles.	- Bajo demanda (On-demand). - Buscan una ruta sólo cuando se necesita.
Ventajas	- Una ruta puede ser utilizada de forma inmediata, sin ningún retardo.	- Requiere un ancho de banda reducido para mantener las tablas de ruteo.

		- Hace un uso eficiente de la energía.
Desventajas	- Introduce mayor tráfico de control. - Utiliza una cantidad elevada de ancho de banda. - Produce congestión en la red. - Reacción lenta en reestructuración y fallas.	- Tiempo de latencia alto en la búsqueda de rutas. - Una inundación excesiva puede llevar al congestionamiento de la red

Tabla 2-1: Comparación entre protocolos de encaminamiento proactivos y reactivos.
Tomada y modificada de [7]

2.6.2 Encaminamiento salto a salto y en origen

En el encaminamiento salto a salto (hop by hop routing) cada router decide solo el siguiente salto y la información de enrutado se guardan en los routers. En el encaminamiento de origen (Source Routing) la ruta se establece antes de realizar el envío y la información de enrutado se guarda en el paquete.

En la Tabla 2-2 se clasifican los principales protocolos de encaminamiento.

Protocolos	Encaminamiento salto a salto	Encaminamiento en origen
Proactivos	DSDV, OLSR, CGSR, WRP, TBRPF	
Reactivos	AODV, LMR, TORA	DSR, LQSR

Tabla 2-2: Protocolos de encaminamiento para redes Ad-Hoc

Por las características propias de las MANETs, los protocolos reactivos (bajo demanda) son los que mejor se adaptan a este tipo de entornos. A continuación se describen dos protocolos reactivos que están siendo considerados como estándar por la IETF: AODV [8] y DSR [9].

2.6.3 AODV - Ad hoc On Demand Distance Vector

Es un protocolo de encaminamiento reactivo (bajo demanda) que encamina salto a salto, busca una ruta sólo cuando se necesita y la información de ruteo se almacena en los routers. Ningún nodo tiene el grafo completo de la red, solo conoce el primer salto por donde debe encaminarse y la distancia a la que se encuentra.

Las principales características de este protocolo se listan a continuación:

- Bajo demanda, el proceso de descubrimiento de ruta solo se ejecuta cuando existe la necesidad de comunicarse.
- Encamina salto a salto, mantiene una tabla de encaminamiento local para los destinos ya conocidos.
- Los mensajes que intercambian los nodos solo contienen información del origen y el destino.
- Mayor retardo en la entrega de paquetes.
- Baja utilización de recursos del nodo (procesador, memoria) y del ancho de banda.
- Introduce poca sobrecarga en la red debido a que no hace una actualización constante de rutas.

Descubrimiento de ruta

Cuando se necesita una ruta desde un nodo origen "O" hasta un nodo destino "D", se inunda la red con peticiones RREQ (Route REQuest). Los nodos intermedios graban en sus tablas de enrutamiento la dirección del nodo vecino desde el cual recibieron el paquete broadcast, para luego establecer una ruta inversa, si reciben copias adicionales del mismo RREQ estos paquetes son descartados. Cuando un RREQ llega al destino buscado, o a algún nodo que conoce una ruta para el destino, se genera una respuesta RREP (Route RREP). El RREP sabe volver al origen porque la inundación de RREQ fue creando el camino de vuelta. Cuando el RREP va volviendo al origen, va creando el camino de ida, una vez que el origen ha recibido el RREP, ya puede enviar paquetes, que seguirán el camino de ida.

La Figura 2-6 ilustra el procedimiento de descubrimiento de ruta entre un nodo origen y un nodo destino, en la parte izquierda se observa el envío de mensajes RREQ en modo *broadcast* desde el nodo "O" a través de los nodos vecinos hasta llegar al destino D, en la parte derecha se muestra como el nodo D envía en sentido contrario un mensaje RREP al nodo "O" en modo *unicast*.

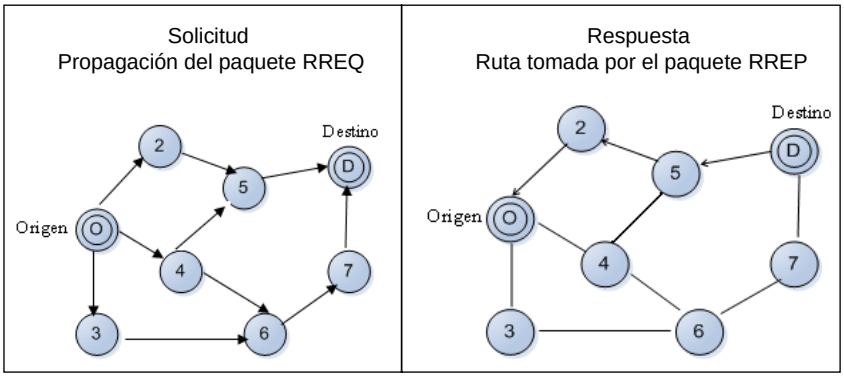

Figura 2-6: Descubrimiento de ruta en AODV

Mantenimiento de Ruta

Cada nodo mantiene información sobre sus enlaces vecinos utilizando mensajes periódicos denominados HELLO (de TTL=1), las caídas de enlaces son detectadas por la ausencia de mensajes HELLO. Una vez detectada la caída de algún enlace que afecta a una ruta activa, la información es propagada a todos los nodos y el nodo origen puede reiniciar el proceso de descubrimiento de ruta si es necesario. La desaparición de un enlace que no participa en ninguna ruta activa no provoca ninguna acción.

2.6.4 DSR - Dynamic Source Routing for Protocol Mobile Ad hoc Networks

Es un protocolo de encaminamiento reactivo (bajo demanda) que encamina en origen, busca una ruta sólo cuando se necesita y la información de ruteo se almacena en el paquete. Los paquetes de datos incluyen una cabecera con información sobre los nodos exactos que deben recorrer.

DSR permite que la red esté completamente autoorganizada y autoconfigurada sin necesidad de administración en la infraestructura de red. Es un protocolo de encaminamiento simple y eficiente diseñado para uso exclusivo de redes ad hoc inalámbricas multi-salto.

El protocolo está compuesto de dos mecanismos: Descubrimiento de Ruta y Mantenimiento de Ruta, permitiendo a los nodos descubrir y mantener rutas del origen a destinos arbitrarios.

Descubrimiento de ruta

El descubrimiento de ruta consiste en que un nodo origen envía un paquete a un nodo destino obteniendo la ruta de origen al destino. El descubrimiento de ruta es usado solamente cuando el origen intenta envía un paquete al destino y no conoce todavía una ruta. La petición de ruta se hace por inundación (Flooding), cada petición lleva un identificador para no propagar por duplicado. La petición va registrando su ruta, si llega a su destino se contesta al origen.

Mantenimiento de Ruta

El mantenimiento de ruta es el mecanismo por el cual el nodo origen es capaz de detectar un cambio en una ruta que lo comunica con el destino y actuar en consecuencia. Cada nodo es responsable del siguiente salto, si pierde un enlace con algún vecino se lo comunica al origen y no intenta recuperar el error (*best effort*). Cuando el origen se entera de que algún enlace dejo de funcionar (cambio de topología) puede intentar usar otra ruta que conozca al destino o invocar al mecanismo de descubrimiento de ruta para encontrar una nueva ruta. El mantenimiento de ruta es usado solamente cuando el nodo origen está actualmente enviando paquetes al destino.

La Figura 2-7 ilustra el descubrimiento y la construcción de la ruta entre el origen y destino (O, D) a través de mensajes RREQ y RREP respectivamente.

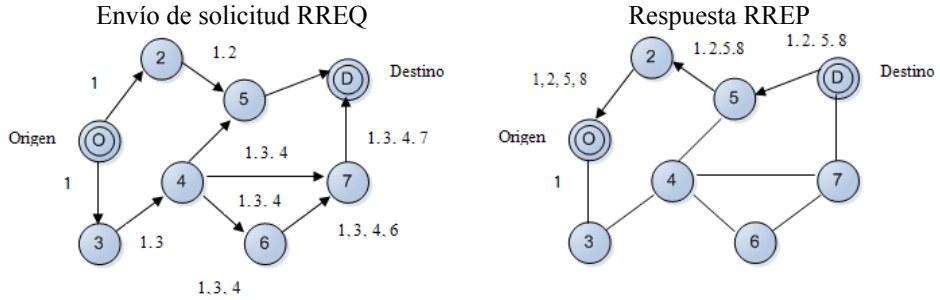

Figura 2-7: Descubrimiento y construcción de ruta en DSR

2.7 Aspectos de seguridad en redes móviles ad hoc

Las características especiales de las MANETs, hacen de este tipo de redes un entorno muy vulnerable frente a diversos ataques y amenazas de seguridad. Resulta, por tanto, imperativa la provisión de un entorno de comunicaciones seguras y libres de riesgos en este tipo de redes, lo que redundará en el aprovechamiento efectivo de sus atractivas cualidades, en ámbitos tan variados como el medioambiental, emergencias, educacional o militar.

2.7.1 Definición de Seguridad

La seguridad es una combinación de procesos, procedimientos y sistemas utilizados para garantizar los siguientes atributos o requerimientos: confidencialidad, autenticación, integridad, no repudio. En el caso de las redes móviles ad hoc se deben agregar los

siguientes atributos: disponibilidad, supervivencia, anonimato y privacidad, control de acceso y autorización.

2.7.2 Atributos de la seguridad en redes móviles ad hoc

2.7.2.1 Confidencialidad

La confidencialidad se encarga de mantener el secreto de los datos intercambiados y de garantizar que la información enviada no sea revelada a usuarios no autorizados (intrusos). Además, los usuarios no autorizados no deben ser conscientes de la existencia de datos protegidos ni de su naturaleza [10].

Las MANETs utilizan el aire como medio de transmisión y esto hace que todos los nodos que se encuentran dentro del rango de transmisión puedan obtener los datos transmitidos, por ello se hace necesario prevenir que nodos intermedios y no confiables tengan acceso o consigan interpretar el contenido de los paquetes que están siendo transmitidos. Cada nodo debe asegurar la información que intercambia con cualquier otro nodo de la MANET utilizando mecanismos criptográficos.

2.7.2.2 Autenticación

La autenticación es una confirmación de que las partes, en comunicación entre sí, son genuinas y no imitaciones, para ello es necesario que los nodos de alguna manera demuestren su identidad. Sin autenticación, un adversario podría enmascarar un nodo y tener acceso a información sensible y clasificada o bien podría interferir con el funcionamiento normal de la red.

En redes de infraestructura o redes inalámbricas con componentes de infraestructura o MANETs subordinadas se puede implementar una CA que autentique a los usuarios en un componente de infraestructura. En el caso de las MANETs autónomas o independientes al no existir infraestructura es mucho más difícil autenticar una entidad y se hace necesario utilizar arquitecturas descentralizadas, como ser: CA distribuida total o parcialmente, autenticación basada en ID, Web of Trust, entre otras. Más adelante se describen estas arquitecturas en detalle.

Existen cuatro tipos de procedimientos de autenticación [11]:

1. Autenticación de la entidad: Garantizar que las entidades (servidores, clientes, personas, etc.) que desean comunicarse con otras sean las que dicen ser.
2. Autenticación de la geo-localización (*Geo-authentication*): La geo-localización de los nodos o cualquier información relacionada con la ubicación debe ser verificada y autenticada.
3. Autenticación de los atributos: Se deben utilizar mecanismos para establecer confianza en los atributos de una entidad o dispositivo.
4. Autenticación de los datos: Es la capacidad que tienen los nodos de comprobar la autenticidad de los datos recibidos.

2.7.2.3 Integridad

La integridad garantiza que los datos transmitidos entre nodos de la MANET sean recibidos por las entidades involucradas sin sufrir modificaciones por parte de terceros y lo que se ha recibido sea lo que originalmente se ha enviado.

En las MANETs los datos son enviados a través de medios inalámbricos, por lo tanto un mensaje se podría corromper debido a razones no maliciosas tales como ruido, interferencias o atenuación de la señal, pero siempre existe la posibilidad que un atacante haya modificado maliciosamente el contenido del mensaje.

2.7.2.4 No repudio

El no repudio permite a cada lado de la comunicación probar fehacientemente que el otro lado ha participado en la comunicación, está relacionado con el hecho de que si una entidad envía un mensaje, esta entidad no puede negar que el mensaje fue enviado por ella.

En el caso de no repudio de origen, el remitente del mensaje no puede negar haberlo enviado. En el caso de no repudio de destino, el destinatario del mensaje no puede negar haberlo recibido.

2.7.2.5 Disponibilidad y Supervivencia

La disponibilidad significa mantener accesibles los servicios de la MANET aun en presencia de fallos o incidentes maliciosos. La supervivencia es la capacidad de la MANET para normalizar los servicios luego de producirse un fallo o incidente malicioso. [3]

En [12] se destacan las siguientes propiedades para la supervivencia de una red:

- **Resistencia.** Es la capacidad de repeler ataques que tiene una red, algunos mecanismos utilizados son: Autenticación de usuario, *firewalls* y criptografía.
- **Reconocimiento.** Es la capacidad de detectar ataques y evaluar la complejidad del daño que tiene una red, los sistemas de detección de intrusos (IDS) son un ejemplo de mecanismo de reconocimiento.
- **Recuperación.** Es la capacidad de restaurar la funcionalidad de una red y la disponibilidad de la información dentro de limites de tiempo tolerables, acotando el daño y manteniendo activos los servicios esenciales.

2.7.2.6 Anonimato y Privacidad

Anonimato significa que toda información que pueda ser utilizada para identificar el origen, el destino, la ruta de transmisión y el contenido de la información deben mantenerse privados y fuera del alcance de entidades no autorizadas.

En [13] se describen tres tipos de protección de la privacidad que provee el anonimato: Privacidad de la identidad, Privacidad de la ruta de transmisión, privacidad de la localización.

- **Privacidad de la identidad.** Ningún nodo puede obtener información acerca del origen y el destino, solo los nodos origen y destino pueden identificarse entre sí.

Además, los nodos origen y destino no deben recibir información acerca de las identidades "reales" de los nodos intermedios que participaron en el reenvío de datos entre origen y destino.
En caso de que se utilicen identificadores de dispositivo (ID) o de red (Dirección MAC o IP), no debe ser posible establecer ninguna asociación entre el identificador y la identidad.

- **Privacidad de la ruta de transmisión.** Ningún nodo puede predecir información acerca del camino (de origen a destino) del paquete. Además, un adversario móvil no puede obtener ninguna pista para rastrear al nodo origen a partir del contenido de un paquete capturado.
- **Privacidad de la localización.** Ningún nodo puede obtener ninguna información acerca de la localización del nodo remitente ya sea en términos de distancia física o números de saltos. Esta información solo puede ser conocida por el nodo origen, sus vecinos inmediatos y el nodo destino.

Se puede también considerar el anonimato desde la perspectiva del usuario o dueño del nodo, en [14] se define anonimato como "toda información que pueda ser utilizada para identificar al dueño o usuario actual del nodo o su geo-localización debe mantenerse privada y no puede ser distribuida ni por el nodo móvil ni por su software de sistema".

2.7.2.7 Control de acceso y Autorización

El control de acceso permite restringir el acceso a los recursos de la MANET a entidades debidamente autorizadas. La autorización establece reglas que definen lo que cada nodo de la MANET tiene o no permitido hacer.

2.8 Mecanismos de seguridad

Los mecanismos de seguridad para garantizar los atributos de seguridad en un entorno ad-hoc dependerán, en gran medida, de la aplicación y del escenario para los que se realiza el despliegue de la red, por tanto las propuestas de seguridad se centran en aspectos concretos del problema.

Generalizando se podrían identificar los siguientes aspectos clave que deben ser cubiertos por cualquier política de seguridad para MANETs:

- Criptografía (simétrica, asimétrica, de umbral).
- Autenticación de los nodos.
- Mecanismos de cooperación para detectar y reducir comportamientos egoístas en los nodos.
- Sistemas de gestión de claves.
- Seguridad física de los nodos.
- Seguridad de los protocolos de encaminamiento.
- Sistemas de detección de intrusiones (IDS).
- Sistemas de respuesta a intrusiones (IRS).
- Sistemas de tolerancia a intrusiones (ITS).
- Anonimato y privacidad.

Capítulo 3: Seguridad en Redes Móviles Ad Hoc

En este capítulo se presenta el estado del arte de la seguridad en redes móviles ad hoc, la información obtenida durante la revisión bibliográfica realizada se organizo de la siguiente manera: En primer lugar, se describen las principales formas de ataques diseñadas para redes móviles ad hoc, se clasifica a los atacantes, se clasifica a los ataques dirigidos a una MANET desde una visión general y también por capa del modelo OSI. A continuación, se clasifica a los ataques dirigidos a un nodo de la red ad hoc. Finalmente, se detallan las principales medidas de seguridad o líneas de defensa diseñadas para proteger una MANET de los atacantes y ataques.

3.1 Ataques

3.1.1 Clasificación de los atacantes

La principal clasificación los divide en atacantes externos y atacantes internos [15].

Los atacantes externos son adversarios que no pertenecen a la red o al grupo de nodos que se comunica de forma autorizada. Dichos adversarios no poseen las claves criptográficas utilizadas para proteger la red y son más fáciles de detectar. Este tipo de atacantes generalmente intenta causar congestión del tráfico o modificar el comportamiento de los nodos de la MANET, utilizando técnicas como:

- Inundación (Flooding), para producir la denegación de servicios por parte del nodo.
- Spoofing, para suplantar un nodo.
- Bloqueo de la propagación de paquetes con información de encaminamiento, para modificar y perturbar rutas operativas.

Los atacantes internos son adversarios que obtienen el control de un nodo que forma parte de la MANET por apropiación ilícita (robo), esto le permite acceder a claves criptográficas utilizadas para proteger el funcionamiento de la red. Estos adversarios son muy difíciles de detectar para poder defenderse contra ellos, y por esta razón sus resultados suelen ser muy dañinos. En los ataques procedentes de atacantes internos, el adversario utiliza el nodo comprometido como base para realizar comportamientos maliciosos, participando de las actividades de la red con la intención de obtener acceso a información confidencial.

En la Tabla 3-1 se mencionan otras clasificaciones para los atacantes de una MANET.

Tipos de atacantes	Descripción
Externos e internos	Según formen parte o no de la MANET
Fijos o móviles	Según se encuentren estacionarios o en movimiento
Sigilosos y no sigilosos	Según se mantenga oculto o se pueda percibir algo su existencia
Activos y pasivos	Según el tipo de ataque que realicen.
En solitario o en grupo	Según realicen el ataque de forma individual o de forma cooperativa entre varios nodos (ataque distribuido).

Tabla 3-1:Clasificación de los atacantes de una MANET

3.1.2 Clasificación general de los ataques

Los ataques a una MANET son generalmente clasificados en dos grandes categorías: Ataques activos y Ataques pasivos [16] [17].

3.1.2.1 Ataques pasivos

Un ataque pasivo permite obtener información que viaja por la red sin interrumpir las comunicaciones, no causan daño directo a un sistema sino que se dedican a capturar información, utilizando las siguientes técnicas:

- **Escuchas clandestinas de las comunicaciones inalámbricas.**

  La RF (Radio Frecuencia) es un medio compartido y de libre acceso, potencialmente toda entidad puede escuchar transmisiones no protegidas y protegidas si se conoce el protocolo, incluso desde mucha distancia. Los atacantes realizan escuchas no autorizadas (*Eavesdropping*) en el rango de cobertura inalámbrica de los nodos de la MANET, con la intención de obtener información confidencial como claves públicas, claves privadas o contraseñas.

- **Análisis y monitoreo de tráfico.**

  El tráfico de paquetes entre los nodos de la MANET puede ser monitoreado y posteriormente analizado por un atacante con la finalidad de determinar:

  - Quien es el emisor y/o quien es el receptor de los mensajes.
  - El rol del nodo emisor y del nodo receptor.
  - Cantidad y longitud de los mensajes intercambiados.
  - La hora de envío de los mensajes.
  - La ubicación desde donde se envían los mensajes (geo-localización).
  - La topología de la red

Los ataques pasivos normalmente se utilizan como puntos de partida para luego realizar ataques dirigidos a los nodos más importantes de la MANET.

3.1.2.2 Ataques activos

Un ataque activo involucra una interrupción, modificación o fabricación de información, alterando el normal funcionamiento de la MANET [18].

La Tabla 3-2 muestra la clasificación general de los ataques contra una MANET y algunos ejemplos.

Ataques	Ejemplos
Pasivos	- Escuchas no autorizadas (*Eavesdropping*). - *Snooping* - Análisis y monitoreo de tráfico
Activos	- Interferencia (*Jamming*) - Suplantación de identidad (*Spoofing*) - Cambio de campos del encabezado de protocolo (*Masquerade*) - Modificación de mensajes (*Modification*) - Fabricación de mensajes falsos (*Fabrication*) - Repetición de mensajes (*Message Replay*)

	- Denegación de servicio (*DoS*)

Tabla 3-2: Clasificación general de los ataques a una MANET

3.1.3 Clasificación de los ataques a MANET por capa del modelo OSI

Otra clasificación utilizada por los autores está basada en el modelo de referencia OSI aplicado a redes MANET (Figura 3-1). En esta clasificación se mencionan los principales ataques en cada capa y también ataques que pueden actuar en diferentes capas (Multi-capa).

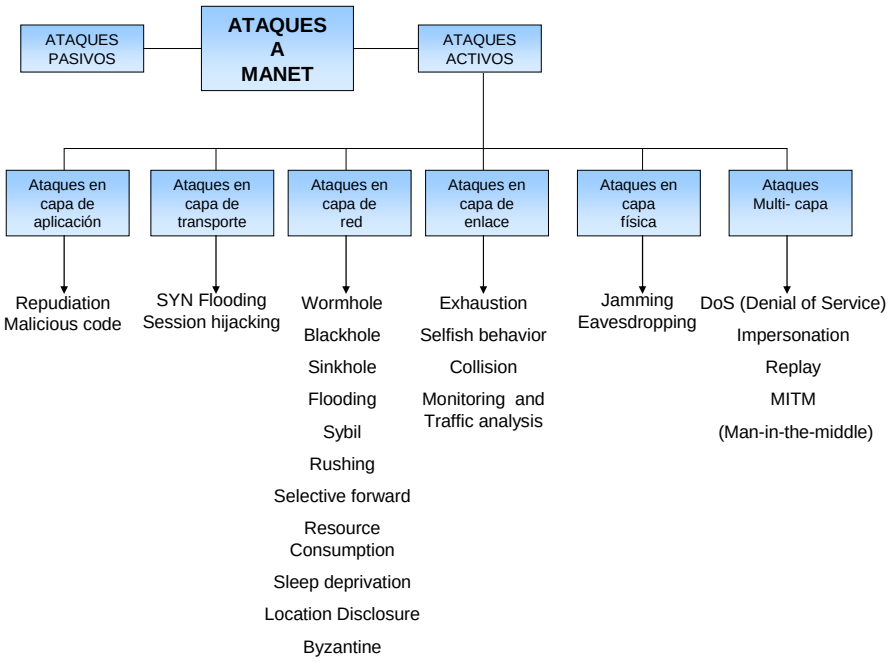

Figura 3-1: Clasificación de los ataques según la capa del modelo OSI
Tomada y ampliada de [16] y [19]

A continuación se describen en forma resumida los ataques mencionados en la Figura 3-1.

3.1.3.1 Ataques en capa física

3.1.3.1.1 Jamming

Interferencias radioeléctricas deliberadas que intentan denegar el uso del canal de comunicación a los nodos de la MANET [20]. La señal de radio frecuencia (RF) puede ser perturbada utilizando señales que se solapan y reducen la relación señal ruido (S/N) por debajo de los valores aceptables. El jamming puede realizarse de forma remota y selectivamente.

3.1.3.1.2 Eavesdropping

Se trata de un ataque pasivo en el cual un atacante realiza escuchas no autorizadas sobre el medio compartido con la finalidad de obtener información confidencial (Localización, claves publicas y privadas, contraseñas, etcétera) que puede ser utilizada más adelante para lanzar un ataque activo. [21]

3.1.3.2 Ataques en capa de enlace

3.1.3.2.1 Exhaustion

El atacante induce intentos repetidos de retransmisión hacia un nodo objetivo, con la finalidad de agotar los recursos del nodo, que por lo general tiene una cantidad limitada de energía en su batería. [19]

3.1.3.2.2 Selfish behavior

El atacante trata de modificar el comportamiento de un nodo de la red con el fin de obtener beneficios frente al resto de los nodos de la red. Por ejemplo, si un nodo comprometido deja de encaminar los paquetes que pasan por él, las retransmisiones de estos paquetes serán enviadas por rutas alternativas consumiendo recursos de otros nodos. [19]

3.1.3.2.3 Collision

Se inyecta gran cantidad de tráfico en el canal de comunicaciones para producir un número excesivo de colisiones, con la finalidad de se deniegue el uso del enlace a los nodos de la red ad hoc. [18]

3.1.3.2.4 Monitoring and Traffic análisis

El tráfico de paquetes entre los nodos de la MANET puede ser monitoreado y posteriormente analizado. [18]

3.1.3.3 Ataques en capa de red

3.1.3.3.1 Spoofing

En estos ataques un nodo malicioso suplanta a un nodo autentico, para ello cambia su dirección IP o MAC (en los paquetes que envía) por la dirección del nodo legítimo. Una vez realizada la suplantación, el nodo comprometido puede monitorear el tráfico de la red, cambiar la topología de la red y aislar a otros nodos de la red. Además puede obtener acceso a información confidencial y/o enviar información de enrutamiento falsa.

3.1.3.3.2 Wormhole

Ataque cooperativo que establece un canal entre nodos de la red maliciosos haciendo pensar a los demás que la ruta establecida por los nodos atacantes es la más rápida [20]. En la Figura 3-2 se ilustra un ejemplo de ataque wormhole, donde dos nodos maliciosos se confabulan para reenviar mensajes entre ellos uno a otro utilizando un canal de baja latencia y alto ancho de banda OOB (Out-Of-Bound). Con esto consiguen atraer el flujo de datos de nodos legítimos de las redes A y B.[16]

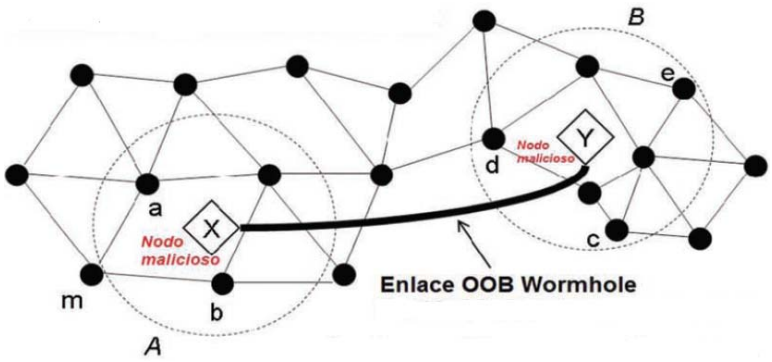

Figura 3-2: Ejemplo de ataque Wormhole

3.1.3.3.3 Blackhole

Un nodo malicioso comienza a descartar los paquetes que recibe. Si el nodo malicioso forma parte de una ruta seleccionada, impedirá que se establezca la comunicación entre los nodos origen y destino [16].

3.1.3.3.4 Sinkhole

El nodo malicioso se presenta como una buena elección de encaminamiento e intenta posicionarse en tantos flujos de red como sea posible (Figura 3-3) buscando que los mensajes dirigidos a la estación base, pasen por él. [16]

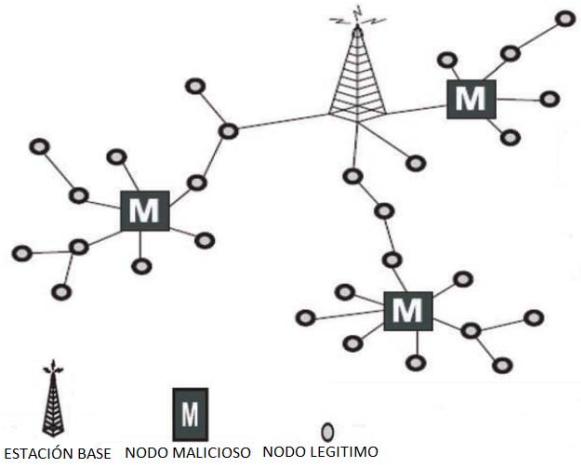

Figura 3-3: Ejemplo de ataque Sinkhole

3.1.3.3.5 Flooding

Uno o varios nodos comprometidos envían un número excesivo de solicitudes a un nodo victima (Ej: solicitud de conexión), con la finalidad de agotar los recursos de la red (ancho de banda) y consumir los recursos del nodo víctima (CPU, batería). Si el ataque es exitoso se interrumpen las operaciones de enrutamiento o se degrada la performance de la red. [15]

3.1.3.3.6 Sybil

En un ataque *sybil* un nodo se relaciona con el resto de la red utilizando múltiples identidades diferentes. El nodo atacante dice tener muchas identidades y localizaciones, dando la sensación de que puede estar en muchos lugares al mismo tiempo. Si algún nodos

objetivo toma como valida la identidad y localización del nodo atacante, puede crear rutas "erróneas" que pasen por este nodo. [15]

3.1.3.3.7 Rushing

Explota las técnicas de descubrimiento de ruta de los protocolos de encaminamiento bajo demanda. Un nodo malicioso reenvía, con mayor rapidez que otros nodos, los paquetes de petición de ruta (RREQ) hacia el destino aumentando la probabilidad del nodo malicioso de estar en la ruta seleccionada. [19]

3.1.3.3.8 Modification

Para lograr la mejor ruta al nodo destino, los nodos siempre dependen de valores de métricas como: numero de secuencia, cantidad de saltos, retardo etc. Mientras menor sea el valor, mejor será la ruta. En este tipo de ataques un nodo comprometido busca redirigir el tráfico hacia él modificando el valor de la métrica utilizada por el protocolo de encaminamiento [22]. Los ataques por modificación mas conocidos son:

- **Redirección de tráfico por modificación del número de secuencia**
  Los protocolos de encaminamiento mantienen rutas incrementando los números de secuencia para cada destino. Un nodo malicioso utiliza el protocolo de enrutamiento para anunciar que tiene el camino más corto a un nodo destino cuyos paquetes quiere interceptar, para lograr su objetivo publica una ruta hacia el nodo destino con un número de secuencia mayor o igual al verdadero valor. Una vez enterados del número de secuencia publicado, los nodos vecinos redirigen el tráfico hacia el nodo malicioso.
- **Redirección de tráfico por modificación del numero de saltos**
  En este ataque el tráfico de paquetes puede ser desviado para tomar una ruta que incluya a un nodo comprometido cambiando los parámetros de número de saltos a un valor menor.
- ***Tunneling***
  Este tipo de ataques ocurre cuando dos o más nodos maliciosos cooperan entre si para establecer un canal de comunicación independiente (túnel) por el cual intercambian mensajes de encaminamiento encapsulados. Una vez establecido el canal, los nodos maliciosos informan a los demás nodos de la red que la ruta es la más corta y consiguen que el tráfico pase por ellos.

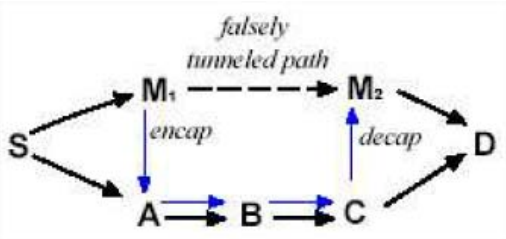

Figura 3-4: Ataque Tunneling
Fuente: [20]

En la Figura 3-4 se observan dos nodos maliciosos NO vecinos (M1 y M2) que pueden utilizar la ruta (M1;A;B;C;M2) como túnel. Cuando M1 recibe una solicitud de ruta (RREQ) del nodo origen (S), la encapsula y la envía por el túnel a M2, el cual reenvía el

paquete hacia el destino (D). Cuando M2 recibe la respuesta (RREP) desde D, la reenvía hacia M1 por el túnel, y este a su vez hacia el nodo origen (S). El resultado es la construcción de una ruta errónea mas corta (M1;M2), la cual seguramente será elegida como la optima por el origen.

- **Denegación de servicio por modificación de ruta**

  Los ataques de denegación de servicio apuntan a la a la destrucción completa de las funciones de ruteo. Un atacante utiliza modificación con la finalidad de que el tráfico de red sea descartado, redirigido a un destino diferente o por una ruta mas larga hacia el destino introduciendo un retardo innecesario.

3.1.3.3.9 Fabrication

Los ataques por fabricación tienen que ver con la generación de mensajes de encaminamiento falsos, los más utilizados son:

- **Falsificación de mensajes de error de ruta**
  Este tipo de ataque es utilizado en el protocolo de enrutamiento bajo demanda, que periódicamente realiza un mantenimiento ruta para recuperar links caídos. Cuando algún nodo cambia su localización, el nodo mas cercano envía un mensaje de error a los demás nodos informando que una ruta dejo de existir. Falsificando y enviando este tipo de mensajes de error cualquier nodo puede ser fácilmente aislado.
- **Difusión de rutas falsas**
  En este tipo de ataques, el atacante cambia la información de ruta del encabezado del paquete y la reemplaza por información de encaminamiento que incluya uno o más nodos maliciosos en la ruta. Luego reenvía el paquete a los nodos vecinos para que actualicen el cache de rutas con información falsa.
- **Desbordamiento de tablas de enrutamiento**
  En este tipo de ataques, el atacante publica una cantidad elevada de rutas inexistentes para saturar la capacidad de memoria de los demás nodos, con esto consigue que los nodos no puedan añadir rutas nuevas (validas) a sus tablas de ruteo.

3.1.3.3.10 Selective forward

Un nodo malicioso se comporta normalmente, aunque de vez en cuando descarta paquetes que son necesarios para el normal funcionamiento de la red. Este descarte selectivo es muy difícil de detectar. [23]

3.1.3.3.11 Resource Consumption

La idea de este ataque es consumir la batería del nodo victima, haciendo que realice tareas que no son necesarias para el funcionamiento de la MANET. Ej: solicitando un número elevado de solicitudes de ruta (RREQ) o reenviando paquetes innecesarios a la victima (Flooding). [24]

3.1.3.3.12 Sleep deprivation

La idea de este ataque es solicitar los servicios de un nodo, una y otra vez con la finalidad de impedir que un nodo cambie a un estado de inactividad o ahorro de energía. Esto es muy

efectivo en MANETS que tienen nodos con baterías de capacidad reducida y pocas posibilidades de recarga. [20]

3.1.3.3.13 Location Disclosure

Un atacante descubre información sobre la localización de los nodos y la estructura de la red utilizando herramientas de análisis de tráfico o monitoreando la red. Reúne información de la localización del nodo en diferentes instantes de tiempo, lo que le permite construir mapas de rutas y en función de estos planificar futuros escenarios de ataque. [19]

3.1.3.3.14 Byzantine

Nodos intermedios comprometidos operan solos o en grupo y ejecutan ataques con el propósito de interrumpir o degradar los servicios de encaminamiento. Estos ataques pueden ser: creación de bucles de enrutamiento (routing loops), reenvío de paquetes a través de rutas no óptimas o descarte selectivo de paquetes. [16]

3.1.3.4 Ataques en capa de transporte

3.1.3.4.1 SYN Flooding

El atacante envía un número grande de peticiones de establecimiento de conexión TCP half-open al nodo victima y nunca completa el handshake TCP de tres vías. Las conexiones semi-abiertas producen agotamiento de recursos en el nodo victima. [21]

3.1.3.4.2 Session hijacking

Este ataque consiste en robar una conexión o sesión después de que el nodo ha superado con éxito el proceso de autenticación ante el nodo objetivo del ataque. El atacante configura su dispositivo con la dirección IP del nodo victima y se adelanta una respuesta en la conexión TCP con el número de secuencia correcto (que obtiene por sniffing o spoofing). A partir de allí toma el control de la sesión y puede lanzar un ataque de denegación de servicios contra el nodo objetivo del ataque. [20]

3.1.3.5 Ataques en capa de aplicación

3.1.3.5.1 Repudiation

Un nodo comprometido que forme parte de una MANET puede adoptar un comportamiento egoísta o malicioso y negar su participación en toda la comunicación o en parte de la misma [24].

3.1.3.5.2 Malicious code

Aplicaciones maliciosas como virus, gusanos, spywares y troyanos pueden atacar al Sistema Operativo y/o a las aplicaciones que se ejecutan en el dispositivo móvil, degradando el rendimiento del mismo y en algunos casos dejando no operativa a la MANET. [15]

3.1.3.6 Ataques multi-capa

Los ataques multi-capa o multinivel pueden aparecer en diferentes capas del modelo OSI. A continuación se describen los cuatro más trascendentes.

3.1.3.6.1 DoS (*Denial of Service*)

El objetivo de este ataque es afectar la disponibilidad de un nodo o de toda la MANET. Si el ataque es exitoso los servicios de la red quedaran no disponibles. El atacante generalmente utiliza interferencias radioeléctricas (jamming) o inundación (flooding) para producir el agotamiento de recursos (ancho de banda, batería). [20]

3.1.3.6.2 Impersonation

Si el mecanismo de autenticación no se implementa correctamente, un nodo comprometido puede actuar como un nodo autentico y monitorear el tráfico de la red. Además puede obtener acceso a información confidencial y/o enviar información de encaminamiento falsa.[19]

3.1.3.6.3 Replay

Un nodo comprometido registra los mensajes de control de un nodo de la MANET, cuando el nodo victima se encuentra fuera de servicio o cobertura, el nodo comprometido continúa enviando información de encaminamiento relacionada con el nodo victima. Esto ocasiona que los demás nodos actualicen sus tablas con rutas obsoletas. El ataque de repetición se utiliza para suplantar un nodo específico o simplemente para perturbar el funcionamiento del encaminamiento en una MANET. [24]

3.1.3.6.4 Man-in-the-middle

Un atacante se posiciona entre el nodo emisor y el receptor y escucha la información intercambiada entre ambos nodos. En algunos casos el atacante puede suplantar al emisor para comunicarse con el receptor y en otros suplantar al receptor para comunicarse con el emisor. [24]

Contramedidas para los ataques por capa del modelo OSI

En la Tabla 3-3 se mencionan algunas de las contramedidas, organizadas por cada capa del modelo OSI, que pueden aplicarse para proteger a la MANET, las mas relevantes se explican en detalle en el apartado 3.2.

Capa	Ataques	Contramedidas
Aplicación	Código malicioso (virus, gusanos, malware) y abusos de aplicación.	Detección y prevención utilizando IDS.
Transporte	SYN Flooding, Session hijacking	Autenticación y seguridad de las comunicaciones entre nodos (extremo a extremo) utilizando sistemas de gestión de claves y criptografía pública (SSL, TLS).
Red	Ataques a los protocolos de encaminamiento: Wormhole, Sinkhole, Blackhole, Sybil, etc.	Extensiones de seguridad a los protocolos de encaminamiento.
Enlace	Exhaustion, Selfish behavior, Collision, Monitoring and Traffic analysis.	Protección del protocolo MAC y proporcionar seguridad en capa de

		enlace.
Físico	Jamming, Eavesdropping, Interceptions	Prevenir ataques utilizando técnicas de espectro expandido (Spread spectrum: direct-sequence or frequency-hopping spread spectrum) [25]

Tabla 3-3: Contramedidas para los ataques a MANET

3.1.4 Clasificación de los ataques a un nodo ad hoc

En la Figura 3-5 se ilustra mediante un gráfico la clasificación de los ataques que se pueden ejecutar sobre un nodo de la MANET, desde un punto de vista funcional y considerando los medios que un atacante utiliza para iniciar un ataque.

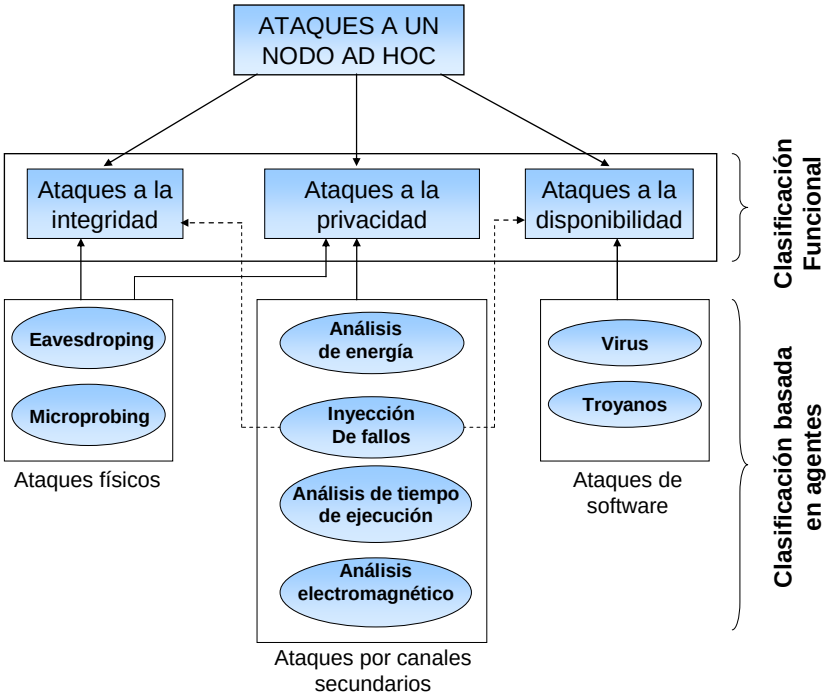

Figura 3-5: Clasificación de los ataques a un nodo ad hoc
Fuente: [26]

La clasificación funcional de los ataques a un nodo ad hoc, considera:

- Ataques a la integridad.
  El propósito detrás de estos ataques es tratar de cambiar los datos o el código en el nodo ad hoc.
- Ataques a la privacidad.
  La idea detrás de estos ataques es acceder a la información confidencial almacenada o manipulada en un nodo ad hoc.
- Ataques a la disponibilidad.
  Estos ataques perturban el normal funcionamiento de un nodo ad hoc intentando agotar los recursos necesarios para el funcionamiento del sistema (energía, ancho de banda, etc).

La clasificación basada en agentes considera los medios que un atacante utiliza para iniciar el ataque, estos se pueden agrupar en tres categorías principales:

- Ataques invasivos.
  Se produce intrusión física a componentes de hardware del nodo (Ej: Circuitos de la placa principal "*mainboard*").
- Ataques de software.
  Ejecutados a través de aplicaciones de software como ser virus, gusanos, troyanos, etc.
- Ataques por canales secundario.
  Monitoreo de propiedades de un nodo mientras realiza operaciones de criptografía con la intención de obtener información para ser utilizada en futuros ataques. Algunas de las propiedades que pueden ser monitoreadas son: tiempo de ejecución, consumo de energía y comportamiento del nodo en presencia de fallos [27].

Los agentes utilizados para lanzar ataques pueden ser: pasivos cuando no interfieren de ninguna manera con la ejecución del sistema (solo observan ciertas propiedades), o activos cuando interfieren con las operaciones del nodo objetivo del ataque. Por ejemplo, los ataques a la disponibilidad e integridad requieren que el atacante interfiera al nodo de alguna manera y esto solo se puede conseguir utilizando agentes activos.

3.1.4.1 Ataques invasivos

También llamados ataques físicos, consisten en disolver la resina que cubre el silicio del chip para luego realizar una reconstrucción del diseño del chip utilizando una combinación sistemática de microscopio y extracción invasiva de capas de cobertura [26].

Los ataques físicos a nivel chip son muy dificultosos porque requieren que el atacante tenga acceso a la misma tecnología utilizada por los fabricantes, la cual generalmente tiene un costo elevado. No obstante, si el ataque físico es exitoso el atacante dispondrá de información suficiente como para lanzar un ataque no invasivo en contra de un nodo ad hoc.

En la Figura 3-5 se destacan dos tipos de ataques invasivos: *Microprobing* y *Eavesdropping*.

Microprobing

Abarca técnicas que le permiten al atacante acceder directamente a la superficie del chip, con la finalidad de observar, manipular e interferir con el circuito integrado. Una vez que el atacante logra acceder al circuito, puede aplicar ingeniería inversa para entender la estructura interna del chip y aprender o emular a su funcionalidad.

Eavesdropping

Comprende técnicas que le permiten al atacante monitorear las características relacionadas con el microprocesador del nodo durante su operación: consumo de energía, radiaciones electromagnéticas producidas, y otras.

3.1.4.2 Ataques de software

Los ataques de software son una de las principales amenazas para los nodos ad hoc, se ejecutan utilizando agentes maliciosos (virus, gusanos) que pueden afectar atributos de seguridad como la integridad, privacidad y disponibilidad. Este tipo de ataques son los más frecuentes ya que requieren de infraestructura simple y de bajo costo.

Los agentes de software maliciosos buscan debilidades en la arquitectura del sistema, las que surgen debido a deficiencias o vulnerabilidades en el software, estas debilidades le permiten al atacante obtener acceso directo al sistema o bien proveen un punto de acceso (*entry point*) que puede ser explotado para acceder a información confidencial.

3.1.4.3 Ataques por canales secundarios (*side-channels*)

Análisis de consumo de energía

El consumo de energía de cualquier circuito de hardware está en función de la actividad que tenga el mismo internamente, la idea del atacante es intentar inferir una clave utilizada por un algoritmo criptográfico a partir de las estadísticas de consumo de energía obtenidas desde una amplia gama de datos de entrada.

Los ataques al consumo de energía se pueden dividir en dos clases: Simple Power análisis (SPA) y Differential Power Analisys (DPA) [28].

En SPA un atacante directamente observa el consumo de energía del sistema, el consumo de energía varía dependiendo de las instrucciones ejecutadas por el microprocesador. Muchas características de algoritmos de encriptación como DES, RSA pueden ser identificadas ya que las operaciones ejecutadas por el procesador hacen variar significativamente el consumo de energía del microprocesador.

DSA es un ataque mucho más potente que SPA y es muy difícil de prevenir. Mientras que SPA utiliza la inspección visual para detectar las fluctuaciones relevantes de energía, DPA utiliza técnicas de análisis estadístico para extraer información relacionada con las claves secretas.

Análisis de tiempo de ejecución

Este tipo de ataques están basados en el hecho de que los tiempos de ejecución de los cálculos para operaciones criptográficas son dependientes de los datos, por lo tanto pueden ser utilizados para inferir claves criptográficas [29].

El tiempo de ejecución puede variar, dependiendo de la implementación del algoritmo de encriptación como de la arquitectura del hardware. Sin embargo, recolectando estadísticas de tiempos de ejecución obtenidas a partir de una amplia gama de datos es posible lidiar con las diferencias en la implementación del algoritmo y obtener la clave.

Inyección de fallos

Se basan en la variación de parámetros externos y condiciones ambientales para inducir fallas en los componentes de un dispositivo[29]. Algunos parámetros que un atacante puede variar son la tensión de alimentación, la temperatura y la radiación. Estas variaciones comprometen la seguridad de un nodo y pueden ser transitorias o permanentes.

Utilizando inyección de fallos se pueden generar los siguientes ataques:

- Ataques a la disponibilidad
  Produciendo fallos en el hardware del dispositivo. Por ejemplo, un cambio de voltaje de entrada puede interrumpir el normal funcionamiento del dispositivo (denegación de servicio).
- Ataques a la integridad y privacidad
  Modificando el código y/o datos almacenados en memoria. Por ejemplo, un fallo de memoria generado por emisión de calor puede ser explotado por un programa no autorizado y no confiable, que se encuentre en ejecución en otro procesador, para asumir el control completo del entorno de ejecución.

Ataques por análisis del espectro electromagnético

Estos ataques están basados en que las radiaciones electromagnéticas que emite una pantalla de video puede ser utilizada para reconstruir su contenido, la idea de este ataque es intentar medir la radiación electromagnética que emite la pantalla de un dispositivo para revelar información sensitiva.

Contramedidas para los ataques a un nodo ad hoc

En la Tabla 3-4 se resumen las contramedidas que se pueden implementar para proteger a los nodos de la MANET de posibles ataques.

Ataques		Contramedidas
Físicos	Eavesdropping Microprobing	- Diseño resistente a la manipulación (*tamper resistance* [26]). - Verificar el cumplimiento de estándares (CC [30] o FIPS [31]).
Por canales secundarios	Análisis de energía	- Enmascaramiento de datos para ocultar información sensible. - Modificación de la amplitud de la señal.
	Inyección de fallos	- Uso de sensores para monitorear cambios ambientales
	Análisis de tiempo de ejecución	- Uso de señales de reloj aleatorias para introducir no determinismo.
	Análisis electromagnético	- Uso de técnicas avanzadas de blindaje. - Uso de técnicas de diseño para distribuir los componentes por toda la superficie del chip.
Software	Virus, gusanos y troyanos	- Instalación de antivirus - Autenticación y validación del software a instalar. - Descargas de software desde sitios de distribución certificados (Ej.: Google Play)

Tabla 3-4: Contramedidas para los ataques a un nodo ad hoc

3.2 Medidas de seguridad

Las características y debilidades de una MANET descritas en el capitulo anterior hacen que el cumplimiento de los atributos de seguridad (confidencialidad, autenticación, integridad,

no repudio, y otros) sea un problema muy complejo de abordar, mostrando la dificultad de diseñar una solución general en términos de seguridad sobre un escenario móvil ad-hoc.

Son numerosas las soluciones propuestas a los problemas de seguridad de las redes ad-hoc [32]. En general, estas soluciones plantean el uso de mecanismos preventivos para proteger protocolos y/o aplicaciones, los cuales se centran en las siguientes cuestiones:

- La seguridad física de los nodos.
  Gran parte de la seguridad de una MANET depende de que la seguridad física de los nodos, debido a que un nodo comprometido es capaz de introducir fallos de seguridad en toda la red. Es indispensable implementar mecanismos de seguridad en los dispositivos móviles antes de incorporarlos a la red ad hoc [20].
- Soluciones extremo a extremo (End to End) que requieren la integración la MANET a una red de infraestructura [2] y [33].
- El desarrollo de mecanismos de criptografía para proporcionar servicios de seguridad como confidencialidad y autenticación, esto requiere la gestión de las claves criptográficas en entornos móviles ad hoc [34].
- La implementación de extensiones de seguridad para los protocolos de encaminamiento disponibles [35].

Sin embargo, y a pesar de la amplia variedad de técnicas existentes, la mayor parte de las mismas son muy específicas y no resultan efectivas para ciertos tipos de ataques o incidentes. Una solución integral de seguridad requiere el uso combinado de estas técnicas.

Como complemento de los esquemas preventivos, existen iniciativas donde se plantea y evalúa la eficacia de técnicas en las que se combina la prevención con una segunda línea de defensa: detección. Aunque el uso conjunto de ambos tipos de mecanismos mejora cada uno de ellos por separado, sigue siendo ineficiente para abarcar una diversidad de ataques. Dentro de las técnicas de detección se destaca el uso generalizado de los denominados sistemas de detección de intrusiones, o IDS (del inglés Intrusion Detection System) [36] . Complementando a los IDS, los sistemas de respuesta a intrusiones o IRS (Intrusion Response System) han sido propuestos en la bibliografía para abordar la reacción del sistema ante posibles eventualidades [37].

Además de las líneas de defensa mencionadas (prevención y detección), surge la idea de una tercera línea de seguridad, denominada tolerancia (o Intrusion Tolerance, IT)[38]. Ésta complementa a las anteriores, dotando al sistema de mecanismos de supervivencia y capacidad de reacción para garantizar el funcionamiento de los servicios mínimos esenciales frente a los ataques, intrusos, accidentes o fallas.

La Figura 3-6 ilustra las medidas de seguridad, o líneas de defensa, que se pueden implementar en las redes móviles ad hoc y que son abordadas en este apartado.

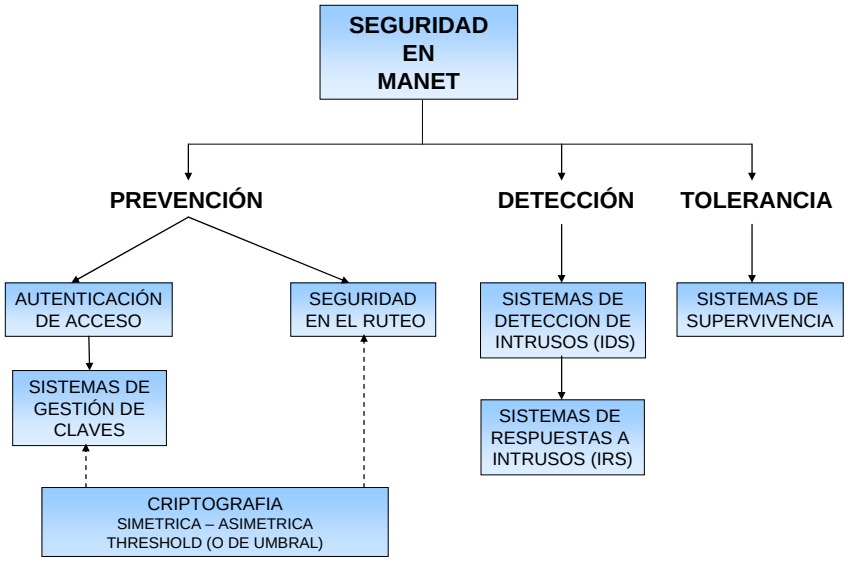

Figura 3-6: Medidas de Seguridad en MANET

3.2.1 Sistemas de gestión de claves

Un sistema o esquema de gestión de claves (*Key management protocol*) es un conjunto de técnicas y procedimientos que brindan soporte para el establecimiento y mantenimiento de claves entre entidades autorizadas. Es el encargado de generar, distribuir, instalar, actualizar, revocar y almacenar las claves asociadas a las entidades.

Para poder brindar servicios avanzados de seguridad en MANETs, es necesario el uso de un sistema de gestión de clave que se adapte a un entorno distribuido y a las características de este tipo de redes.

No se debe confundir un protocolo de gestión de claves con un protocolo de acuerdo de claves (*Key agreement protocol*). Un protocolo de acuerdo de claves es un mecanismo que permite establecer claves entre entidades participantes, el secreto compartido por estas entidades se calcula en función de información aportada por cada una de ellas, de tal manera que ninguna entidad pueda predeterminar el valor resultante por si misma.

La entidad en la cual confían las partes para proveer el servicio de gestión de claves se denomina: *Trusted Third Party* (TTP).

Las TTPs se pueden dividir en 3 categorías principales [39]:

1. ***In-Line***
   La TTP es conocida como in-line si todas las transmisiones del receptor son recibidas primero por la TTP para recién ser enviadas al receptor por la TTP.
2. ***On-Line***
   La TTP en línea recibe una copia de los mensajes transmitidos o solamente una señal de control mencionando algo acerca de la comunicación que se lleva a cabo entre las partes, la comunicación efectiva se realiza directamente entre el emisor y el receptor.

3. ***Off-Line***
   La TTP fuera de línea ayuda a inicializar y configurar la comunicación entre el emisor y el receptor y no se encuentra disponible en la red cuando se realiza la comunicación entre las partes.

En la Figura 3-7 se ilustran gráficamente las tres categorías de TTP.

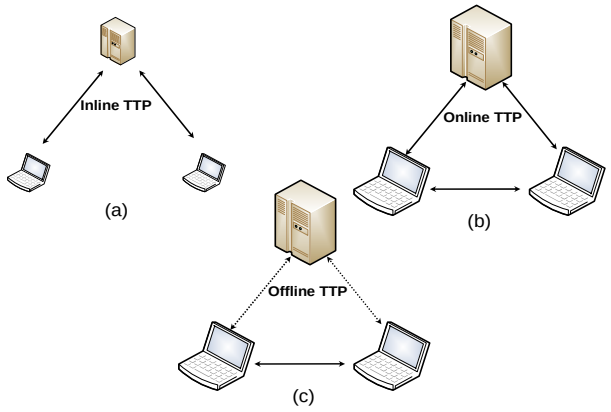

Figura 3-7: Categorías de TTPs

En [40] se considera otro enfoque referido a la disponibilidad de la TTP, también denominada CA (Certificate Authority), se distinguen 4 casos:

1. **CA siempre disponible.**
   Esta opción se puede implementar cuando la MANET se integra a una red de infraestructura que dispone de una CA configurada sobre algún servidor de alta disponibilidad, con mecanismos de tolerancia a fallos y seguridad física elevada. Si la MANET es autónoma esta solución no puede ser considerada ya que se tendría que montar la CA sobre algún nodo de la red y seria imposible garantizar niveles razonables de disponibilidad, debido a que como todo nodo de una red ad hoc se encuentra en itinerancia y puede quedar fuera del rango de transmisión de los demás nodos o bien sufrir algún ataque físico que lo deje no operativo.

2. **CA disponible en la fase de inicialización de la red y cada vez que un nodo se une.**
   La segunda opción es aplicable en escenarios donde una CA se encuentra disponible para emitir certificados y para generar y distribuir claves y parámetros del sistema en la etapa inicial de la red. La CA debe también estar accesible cuando nuevos nodos se sumen a la MANET, así estos pueden obtener las claves y parámetros del sistema.

3. **CA disponible en la fase de inicialización de la red**
   Esta opción es similar a la anterior, con la diferencia que después de la fase de inicialización la CA no puede ser accedida por ninguno de los nodos, ni los nodos en la red ni los nodos nuevos. Los nodos presentes se hacen responsables de las tareas de la CA: emitir, renovar y revocar certificados.

4. **CA no disponible**
   Si no se dispone de una CA ni de posibilidades de integrar la MANET a una red de infraestructura, los nodos deben emitir sus propios certificados implementando para

ello modelos de CA distribuida en forma parcial o total o bien un modelo auto-organizado similar a PGP.

3.2.1.1 Clasificación de los sistemas de gestión de claves

En [41] se presenta una clasificación de los sistemas de gestión (o administración) de claves para MANET (Figura 3-8).

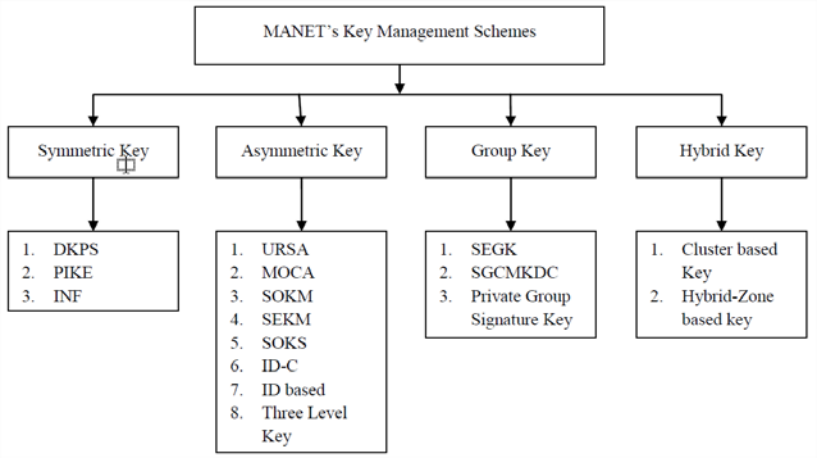

Figura 3-8: Clasificación de los esquemas de gestión de claves para MANET
Fuente: [41]

En esta clasificación se consideran cuatro categorías:

1. **Sistemas de gestión de claves simétricas.**
   En los sistemas de clave simétrica el nodo emisor y el nodo receptor utilizan la misma clave. Esta clave se utiliza para el cifrado de los datos, así como para el descifrado de los mismos. Si n nodos quieren comunicarse, k números de claves son necesarias, donde k = n (n-1) / 2.

2. **Sistemas de gestión de claves asimétricas.**
   En la criptografía asimétrica, se utilizan dos claves, una privada y otra pública. Cada receptor tiene una clave privada que se mantiene en secreto y una clave pública que se publica para todos. El emisor utiliza la clave pública del destinatario para cifrar el mensaje. El destinatario utiliza su clave privada para descifrar el mensaje y nunca publica o transmite su clave privada a nadie. Por lo tanto, la clave privada nunca está en tránsito y permanece invulnerable. Esto reduce el riesgo de pérdida de datos y la confianza se incrementa cuando las claves privadas se gestionan adecuadamente. La criptografía de clave asimétrica requiere menor número de claves, en comparación con la criptografía de clave simétrica.

3. **Sistemas de gestión de claves por grupo.**
   En la criptografía de clave por grupo, una clave única se asigna a un grupo de nodos móviles en MANET. Para establecer una clave de grupo, se crea una clave y se distribuye en secreto a los miembros del grupo. Existen tres categorías de protocolos de gestión de clave por grupo:

   - Centralizado: La generación, control y cambio de claves de grupo es realizado por una sola entidad.

- Descentralizado: Dos o más entidades son responsables de generar, distribuir y re generar la clave de grupo.
- Distribuido: Los miembros del grupo son responsables de generar, distribuir y regenerar la clave de grupo.

4. **Sistemas de gestión de claves híbridos.**
   Las claves híbridas o compuestas son aquellas que se surgen de la combinación de dos o más tipos de claves, que pueden ser simétricas, asimétricas o una combinación de claves simétricas y asimétricas.

Este trabajo de tesis se enfoca en los sistemas de gestión de claves asimétricas debido a que son los más utilizados para entornos MANET. A continuación se presentan las principales propuestas para este tipo de enfoque.

3.2.1.2 Sistemas de gestión de claves asimétricas

Las propuestas que se han hecho para la gestión de claves asimétricas en MANETs abarcan dos tipos de arquitecturas:

- Arquitecturas centralizadas, en las que la CA o TTP se encuentra en un solo lugar (nodo ad hoc o servidor de la red de infraestructura).
- Arquitecturas distribuidas, en las que la CA o TTP se distribuye entre varios nodos de la red ad hoc.

En la Figura 3-9 se clasifican de los sistemas de gestión de claves asimétricas. Se destaca el uso de una TTP (*Trusted Third Party*) en ambas arquitecturas y de la criptografía de umbral (*threshold cryptography*) en tres de las propuestas.

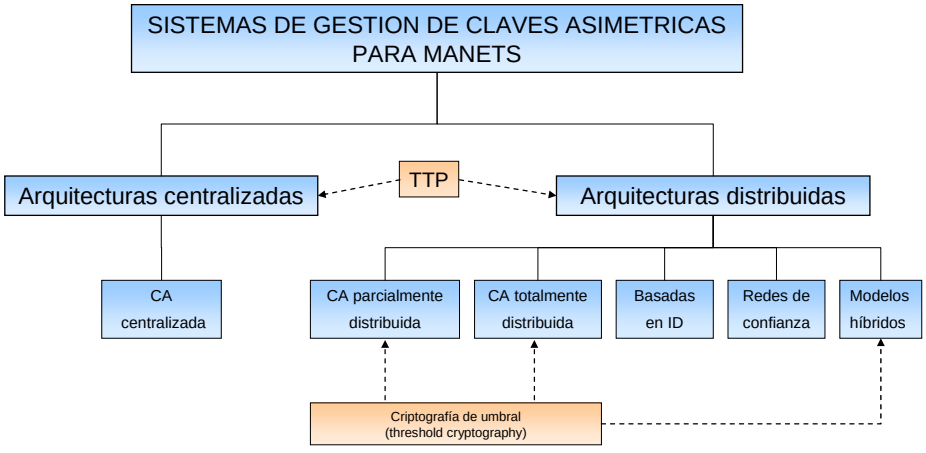

Figura 3-9: Sistemas de gestión de claves para MANETs

Las soluciones consideradas en esta clasificación son:

- **Sistemas de gestión de claves basados en CA centralizada.**
  Requieren la integración de la MANET a una red de infraestructura que tenga implementada una infraestructura de clave pública (PKI). El caso de estudio de este trabajo de tesis se implementa sobre este tipo de escenario.
- **Sistemas de gestión de claves basados en CA parcialmente distribuida.**

Un conjunto de nodos ejerce las funciones de la autoridad de certificación.

- **Sistemas de gestión de claves basados en CA totalmente distribuida.**
  Todos los nodos constituyen la CA.
- **Sistemas de gestión de claves basados en ID.**
  Los certificados se reemplazan por las propias identidades de los nodos para la autenticación.
- **Redes de confianza *(Web Of Trust)*.**
  Se emplea una aproximación similar a la de PGP consistente en crear cadenas de confianza entre los nodos en base a las relaciones de confianza que tienen estos con el resto.
- **Modelos híbridos.**
  Combinan una CA distribuida con el modelo de redes de confianza.

En los siguientes apartados se presentan las principales características de los sistemas de gestión de claves ilustrados en la Figura 3-9. Previamente, y por tratarse de un mecanismos utilizados en tres de las soluciones propuestas, se describen las características y el funcionamiento de la criptografía de umbral.

3.2.1.2.1 Criptografía de umbral

La criptografía de umbral (*threshold cryptography*) tiene como objetivo distribuir alguna funcionalidad criptográfica entre muchos participantes de tal forma que:

- Cualquier conjunto con (t+1) participantes pueda colectivamente calcular la funcionalidad.
- Ningún conjunto con sólo t participantes pueda calcular la funcionalidad.

Esto significa que la acción se lleva a cabo solo si al menos un cierto umbral de participantes (t+1) ejecuta el protocolo, se suele usar la notación "*t- threshold"* para indicar cual es el valor umbral del sistema.

Muchas de las soluciones propuestas para la gestión de claves descentralizada proponen la distribución de las funciones de la CA entre un conjunto de nodos y para llevar a cabo este objetivo utilizan criptografía de umbral.

En el esquema de secreto compartido Shamir (k,n) [42], se propone un mecanismo para distribuir un secreto entre un conjunto de k nodos sin que nadie lo posea por completo, el secreto (la clave privada) se divide en n partes utilizando un polinomio aleatorio, el mismo puede ser recuperado combinando k partes y utilizando interpolación de Lagrange.

La solución consiste en construir un polinomio de grado *k*-1 tal como el siguiente:

$$f(x) = d + a_1 x + a_2 x^2 + ... + a_{k-1} x^{k-1}$$

En el polinomio anterior el término independiente es el secreto compartido, la clave privada, con lo que $f(0) = d$. La parte del secreto que corresponde a cada uno de los n nodos que constituyen la CA es el resultada de evaluar la función en un punto:

$$S(i) = f(i) \bmod N$$

Si tenemos k secretos y una función que es un polinomio de grado (*k*-1), podemos interpolarlo mediante coeficientes de Lagrange, por lo que la función se puede escribir:

$$f(x) = \sum_{l=1}^{k} S_I - l_i(x) \bmod N$$

$$l_i(x) = \prod_{j=1, j\neq i}^{k} \frac{x-j}{i-j}$$

3.2.1.2.2 CA Centralizada

En este tipo de arquitecturas los nodos de MANET hacen uso de los servicios que brinda la PKI (*Public Key Infraestructure*) de una red de infraestructura. Los nodos ad hoc se comunican con la CA de la red de infraestructura para realizar operaciones de autenticación, firma o verificación de firma y/o operaciones de cifrado o descifrado.

La implementación de esta arquitectura requiere la integración de la MANET a una red de infraestructura y el establecimiento de canales de comunicación extremo a extremo entre los nodos ad hoc y la CA (instalada en un servidor de infraestructura).

En la Figura 3-1 se ilustra una MANET subordinada que utiliza servicios PKI de una intranet. Los nodos de la red ad hoc (Origen y Destino) se comunican con la CA centralizada utilizando canales extremo a extremo implementados sobre diferentes tecnologías y protocolos de comunicación: MANET (Bluetooth, WiFi), red celular (2G, 3G) e Internet. La CA de la intranet es la encargada de autenticar los nodos y gestionar las claves para que estos puedan intercambiar mensajes firmados y/o cifrados.

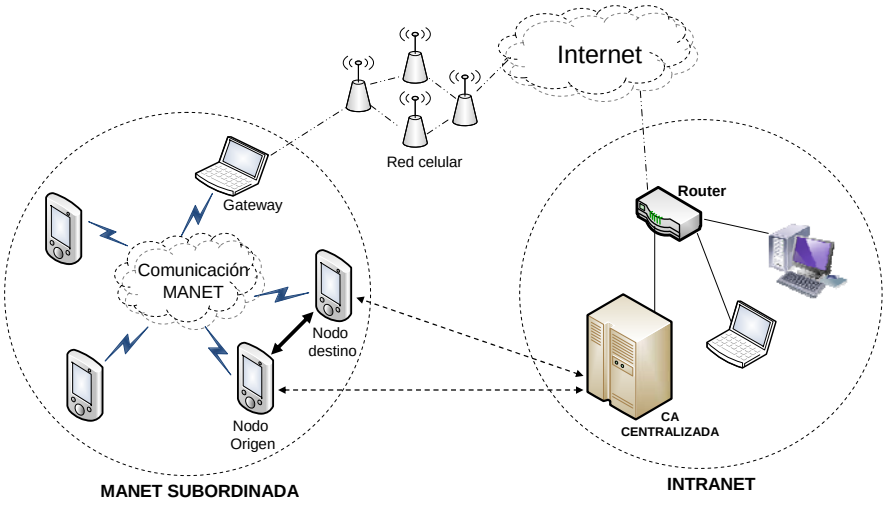

Figura 3-10: Gestión de claves en MANET utilizando una CA centralizada

Funcionamiento

En la fase de inicialización se generan las claves privada (sk_{CA}) y pública para la CA (pk_{CA}). La CA genera las claves (pública y privada) y certificados para todos los nodos de

la MANET, los certificados incluyen la identidad y clave pública del nodo y se firman con la clave privada de la CA (sk_{CA}). En cada nodo de la red ad hoc se instalan la clave pública de la CA (pk_{CA}), los certificados de todos los nodos (que incluyen la clave pública) y la clave privada del nodo en cuestión.

A continuación se enumeran los pasos para enviar un mensaje firmado de un nodo origen O con claves (sk_O, pk_O) a un nodo destino D con claves (sk_D, pk_D):

1. El nodo origen (O) utiliza su clave privada sk_O, para generar una firma s del mensaje M.
2. Transmite el mensaje firmado (M,s) al nodo destino (D).
3. Luego de recibir (M, s), el nodo D se comunica con la CA para verificar la validez del certificado de O.
4. Si el certificado de O es valido el nodo D procede a verificar la firma de M usando la clave pública del nodo O (pk_O). Si la firma es genuina se acepta el mensaje, en otro caso no.

Si se requiere autenticar la identidad de los nodos origen (O) y destino (D), la lista de pasos será la siguiente:

1. El nodo origen (O) se comunica con la CA para comprobar la validez del certificado del nodo (D). Si el certificado no es valido, suspende el envío del mensaje.
2. Una vez verificada la autenticidad del destino (D), O utiliza su clave privada sk_O para generar una firma s del mensaje M.
3. O transmite el mensaje firmado (M,s) al nodo destino (D).
4. Luego de recibir (M, s), el nodo D se comunica con la CA para comprobar la validez del certificado del nodo origen (O). Si el certificado no es valido, rechaza el mensaje.
5. Si el certificado del nodo origen es valido, el nodo D comprueba si la firma del mensaje M es auténtica utilizando la clave publica del nodo O (pk_O). Si la firma es genuina se acepta el mensaje, en otro caso no.

Cuestiones a considerar

El uso de este tipo de soluciones introduce cierta problemática que debe ser considerada antes de realizar la implementación. A continuación se describen las cuestiones más relevantes:

- La comunicación extremo a extremo entre un nodo ad hoc y la CA consume recursos del nodo (extremo) y de aquellos nodos que realicen operaciones de encaminamiento. Algunos de los recursos que se comprometen son: Energía, CPU, ancho de banda.
- En entornos MANET de alta movilidad y topología dinámica la revocación de certificados es una tarea difícil de implementar. Es recomendable que cada nodo (cuando lo requiera) valide los certificados en línea con la CA, en lugar de distribuir listas de revocación de certificados (CRLs) a todos los nodos de la red.

- La incorporación de un nuevo nodo a la red ad hoc requiere: la generación de claves (publica y privada) y del certificado, la instalación de la clave privada en el nodo y la distribución del certificado a todos los nodos.
- Se requiere comunicación permanente con la CA (on line TTP), si por algún motivo un nodo no puede establecer comunicación con la CA no podrá validar los certificados de los demás nodos, salvo que disponga de la CRL.
- Se deben implementar mecanismos de seguridad para proteger la clave privada, en cada nodo. Si un atacante consigue acceso físico a un nodo, puede obtener una copia de la clave privada del nodo y utilizar esta para realizar ataques de suplantación, con la finalidad de obtener información confidencial o comprometer el normal funcionamiento de la MANET.

3.2.1.2.3 CA parcialmente distribuida

En [43] Hou propone el uso de un esquema (n, t+1) de criptografía de umbral (para repartir la autoridad de certificación entre n nodos especiales de la red, llamados servidores, los cuales pueden emitir certificados para el resto de los nodos a partir de la combinación de por lo menos t+1 firmas parciales. En este artículo también se indica cómo llevar a cabo la actualización de los secretos compartidos.

En la Figura 3-11 se ilustra la configuración del servicio. El servicio de gestión de claves tiene asignadas un par de claves publica/privada (K/k), la clave publica K es conocida por todos los nodos de la red y la clave privada k se divide en n porciones: s_1, s_2, . . . , s_n, una porción para cada servidor. A su vez cada servidor i tiene sus claves pública y privada (K_i, k_i) y conoce las claves públicas de todos los nodos miembros de la red, en particular las de los demás servidores, lo cual permite a los nodos servidores establecer vínculos seguros entre ellos.

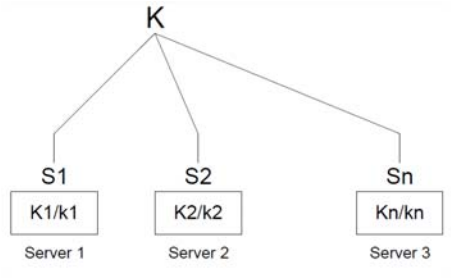

Figura 3-11: Configuración de un servicio de administración de claves parcialmente distribuido
Fuente: [43]

La Figura 3-12 ilustra la operación de construcción de firma, teniendo en cuenta un servicio compuesto por tres servidores con clave pública y privada (K/k). Se utiliza criptografía de umbral con un esquema de configuración (3,2) donde cada uno de los 3 servidores obtiene una porción si de la clave privada k.

Para autenticar al nodo m cada servidor i genera una firma parcial $P_S(m, s_i)$ utilizando para ello la parte del secreto que conoce s_i. Aunque el servidor 2 se encuentra comprometido, el

combiner todavía puede generar la firma para el certificado del nodo m ($Cert_m$) utilizando para ello las firmas parciales de los servidores 1 y 3.

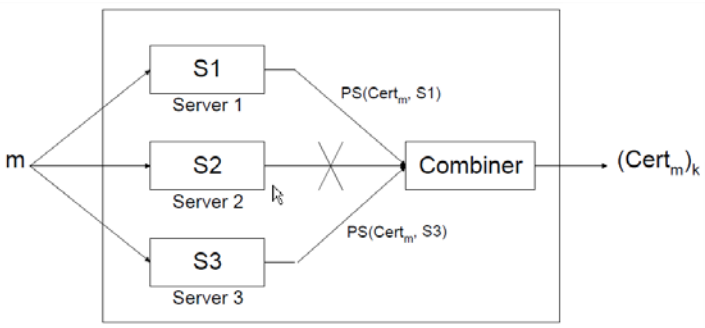

Figura 3-12: Criptografía de umbral para una configuración (3,2)
Fuente: [43]

En [44] Yi y Kravets presentan un protocolo de certificación llamado MP (MOCA Certification Protocol). MP es un ejemplo de CA parcialmente distribuida o esquema de gestión de claves descentralizado, se centra en hacer eficientes los protocolos de comunicación para la gestión de la CA y propone distribuir el secreto entre los nodos que tengan una mayor capacidad de cómputo y seguridad física, en especial en aquellos entornos que estén compuestos por nodos con diferentes características. Estos nodos son llamados MOCAs (Mobile Certificate Authorities).

A continuación se enumeran los pasos involucrados en una certificación:

- Los nodos clientes envían por broadcast mensajes de solicitud (SREQ, Send REQest).
- Cada nodo MOCA que recibe este mensaje envía una respuesta (SREP, Certif Response) con una firma parcial.
- Cuando el nodo cliente recolecta t CREPs validos, procede a calcular su firma.

MP ofrece un nivel de seguridad superior ya que no utiliza un combinador (*combiner*). Además, incluye un mecanismo para generar listas de revocación de certificados (CRLs) de manera distribuida.

En [45] Wu y Otros presentan SEKM (Secure and efficient key management in mobile ad hoc networks). Se trata de otra solución basada en los mismos principios pero que mejora las anteriores centrándose en los mecanismos para la formación y administración del grupo de servidores que realizan las funciones de CA. Estos mecanismos incluyen mecanismos para incorporar nuevos nodos al grupo de servidores y para actualizar los secretos compartidos. En SEKM el secreto (K_{ca}) se distribuye entre m servidores (*shareholders*) y se requiere un quórum de k ($1<k\leq m$) servidores (grupo de servidores) para producir un certificado valido.

Las propuestas de CA parcialmente distribuida comparten un mismo compromiso entre seguridad y disponibilidad derivado de la elección del valor de k para el esquema de criptografía de umbral [46]. Si el valor de k es elevado, la seguridad aumenta al necesitarse un mayor número de servidores para generar una firma, y por lo tanto también será

necesario capturar más nodos para comprometer el sistema. Por otro lado, la disponibilidad de la CA disminuye, ya que será más difícil encontrar y contactar a k nodos para que emitan un certificado.

En este tipo de soluciones aparecen las siguientes limitaciones:

- Suponen la existencia de una TTP off-line o CA disponible en la fase de inicialización de la red que hace la inicialización del sistema y que escoge los n nodos que contendrán la CA.
- La tarea de elegir los nodos que contendrán la CA no es sencilla, deberían ser los más seguros y los que tengan una mejor capacidad de procesamiento, ya que seguramente tendrán un mayor consumo de recursos.
- No todos los nodos tendrán el mismo rol en la red, se puede comprometer el funcionamiento de la red seleccionando para atacar a los nodos que formen parte de la CA.

3.2.1.2.4 CA totalmente distribuida

En este tipo de soluciones se distribuye la función de la CA entre todos los nodos de la red ad hoc.

En [47] Kong, Luo y Zerfos presentan URSA (Ubiquitous and Robust Access Control for Ad hoc Network). La idea básica de esta propuesta, además de distribuir la CA completamente en la red, es ofrecer un servicio de certificación localizado en el cual los nodos más próximos son los encargados de emitir certificados (o tickets) para un nodo vecino. Estos nodos a la vez tienen también la misión de monitorear la actividad de sus nodos vecinos para la toma posterior de decisiones de revocación. En URSA todos los nodos tienen un ticket o certificado, que necesitan obligatoriamente para interactuar con el resto de los nodos, y que es válido solo durante un periodo de tiempo, por lo que antes de que caduque debe ser renovado periódicamente. Los tickets siempre son emitidos por un conjunto de k nodos vecinos al nodo que los solicita. A la vez, estos nodos, como se encuentran monitoreando el comportamiento del nodo solicitante, podrán emitir mensajes de acusación si detectan un comportamiento malicioso del nodo. Si un nodo recibe por parte de sus vecinos muchas acusaciones pasa al estado de convicto y no se le volverá a renovar el certificado o ticket cuando éste caduque, quedando así fuera de la red.

En [48] Zhu, Bao y Otros presentan AKM (Autonomous Key Management), una propuesta que utiliza criptografía de umbral para proporcionar una CA totalmente distribuida. El par de claves (pública y privada) de la CA se distribuye entre un grupo de nodos vecinos de la siguiente manera:

- Cada uno de los n vecinos elige un valor secreto S_i y distribuye el secreto a los demás nodos del grupo utilizando un esquema de secreto compartido de Shamir (k, n).
- La clave privada de la CA esta representada por la suma de los secretos individuales: $S = (S_1 + S_2 + S_3 + \ldots + S_n)$.
- La clave pública de la CA es igual a g^S (g mod S). Si los nodos publican sus valores públicos individuales g^S_i (g mod S_i), la clave publica se puede obtener sin necesidad

de revelar la clave privada, multiplicando los valores individuales: $(g^S) = (g^{S_1})x(g^{S_2})x\ldots x(g^{S_n})$.

Inicialmente cada nodo recibe una parte de la clave privada de la CA, a medida que el número de nodos aumenta, se introduce una jerarquía de claves compartidas. Los nuevos nodos reciben un fragmento de la parte de la clave privada de la CA. Este procedimiento se ilustra en la Figura 3-13, en tres fases:

- Fase 1 - Inizialitation
  Se comparte el secreto S= S1+S2+S3 entre los nodos N1, N 2 y N3, cada nodo tiene parte del secreto (N1–>S1, N2–>S2 y N3->S3).

- Fase 2 - New nodes added
  A continuación se agregan nuevos nodos (N4,N5 y N6), ahora el secreto se comparte entre 6 (n=6) nodos y se requieren contactar a 3 (k=3) para mantener la CA operativa.

- Fase 3 - Split
  Se procede a la dividir el secreto, los nodos N1, N2 y N3 deciden formar un nuevo grupo y distribuyen una parte de su cuota de secreto f (N1), f (N2) y f(N3) respectivamente. El nuevo secreto regional de N1, N2 y N3 es igual a la suma de sus cuotas S' = f (N1) + f (N2) + f (N3), representados por el nodo virtual "G". Del mismo modo, el nuevo secreto regional de N4, N5 y N6 es igual a la suma de sus cuotas S''= f (N4)+f(N5)+ f(N6), representados por el nodo virtual "H".

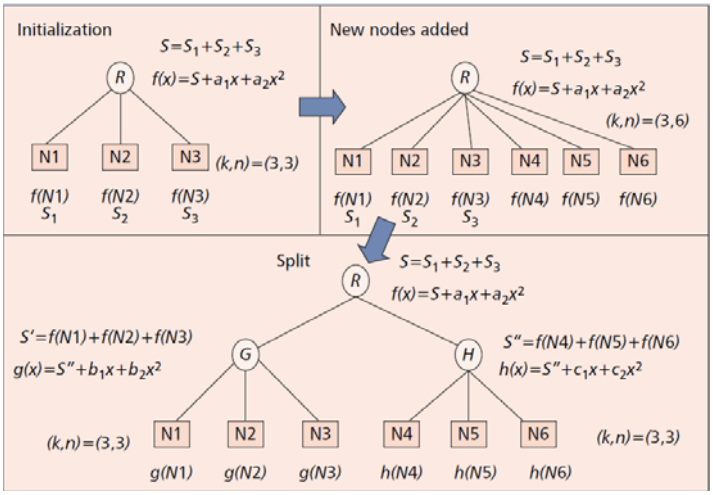

Figura 3-13: Funcionamiento de AKM
Fuente: [46]

Las soluciones propuestas en este apartado, al distribuir el secreto entre todos los nodos de la red, tienen resuelto el tema de la auto-inicialización, la admisión de nuevos nodos y la actualización de secretos, siguiendo los principios de la criptografía de umbral (*threshold*). La CA totalmente distribuida tiene ventajas con respecto a la CA parcialmente distribuida ya que preserva las relaciones simétricas entre los nodos, donde todos juegan el mismo rol, y proporcionan mejor disponibilidad y mecanismo de revocación. Por el contrario, mantienen el problema de la elección de k. Si el valor de k es alto será difícil encontrar k nodos vecinos que puedan emitir o renovar un certificado. Si el valor de k es bajo será más

sencillo para un atacante comprometer la red. En este sentido la movilidad de los nodos mejora la disponibilidad ya que favorece encontrar nodos vecinos, pero perjudica a la seguridad, incrementando las posibilidades de que nodos maliciosos no sean detectados.

3.2.1.2.5 CA Basadas en ID

La criptografía basada en identidad o IBC (*Identity Based Cryptography*) se caracteriza por el uso de atributos de identidad de los nodos (cadenas de caracteres identificativos) como ser: dirección IP, dirección MAC, etc. A partir de estas cadenas los nodos pueden cifrar y verificar firmas, sin necesidad de certificados digitales de una PKI.

En estos esquemas se usa una correspondencia entra la identidad del nodo y una clave secreta privada y única. Esta correspondencia la determina una CA, denominada PKG (Private Key Generator), que cifra la clave para cada identidad y se la entrega a cada nodo durante la fase de inicialización. Si bien no se utilizan certificados, se observa una certificación implícita entre la clave simétrica generada por el PKG y la identidad.

Antes de que las operaciones puedan comenzar el PKG genera un par de claves pública/privada, estas claves son llamadas clave pública maestra (pk_{PKG}) y clave privada maestra (sk_{PKG}) respectivamente. Para operar, el PKG primero publica la clave pública maestra (pk_{PKG}) al resto de los nodos. Si un nodo origen (O) dispone de clave pública maestra del PKG, puede calcular la clave pública de un nodo destino (D) combinando la clave pública maestra con la cadena de identidad del nodo destino (D). Para obtener su clave privada, el nodo D la solicita al PKG, el PKG utiliza su clave privada maestra (sk_{PKG}) para generar la clave privada asociada a la cadena de identidad de D, y se la envía al nodo D por canal seguro.

A continuación se describen los pasos para firmar un mensaje de un nodo origen O a un nodo destino D, usando criptografía basada en identificación:

1. El nodo origen (O) se autentica con el PKG y recibe su clave privada sk_O
2. Usando su clave privada sk_O, el nodo O genera una firma s del mensaje M y transmite el mensaje firmado (M,s) al nodo destino D.
3. Después de recibir (M, s), el nodo D verifica si la firma de M es auténtica usando la identidad del nodo O (ID_nodoO) y la clave pública maestra pk_{PKG}. Si la firma es genuina se acepta el mensaje, en otro caso no.

En la secuencia de pasos se observa que el nodo destino D no necesita en ningún momento tener ningún tipo de certificado del nodo origen O.

En [49] Kapil y Rana presentan un sistema de gestión de claves para MANETs basado en identidad criptografía asimétrica. Esta propuesta posibilita a los nodos generar sus propias claves públicas a partir de sus identidades y alguna otra información común a todos los nodos, y a continuación, distribuir estas claves en toda la red. A diferencia de otros esquemas que hacen uso intensivo de certificados de clave pública, esta solución no necesita de una entidad emisora de certificados en línea (TTP) para compartir la clave secreta.

En [50] Li, Dawei y Otros proponen una solución para crear un threshold PKG entre varios nodos, de modo similar a lo que hemos visto anteriormente para la CA parcialmente distribuida. En este caso se sustituye el problema de autenticar una clave pública por el problema de autenticar una identidad.

El uso de la criptografía basada en ID reduce la memoria necesaria para almacenar las claves públicas y la necesidad de utilizar éstas para la autenticación. Por otro parte, se debe tener total confianza en el PKG, ya que esta conoce las claves de todos los nodos.

3.2.1.2.6 Redes de confianza

En [51] Capcun y Hubaux presentan una solución basada en la construcción de cadenas de certificación, similar al mecanismo que utiliza PGP (*Pretty Good Privacy*).

En el esquema de redes de confianza, también llamadas *web of trust* o web de confianza, todos los nodos poseen un par de claves pública y privada. Cada nodo emite certificados, utilizando su clave privada, a todos los nodos en los que confía y a su vez todos los nodos que confíen en un nodo en cuestión le emitirán certificados para su clave pública. Además de emitir certificados, los nodos reciben y comparan los certificados que otros nodos les emiten a ellos y los certificados que otros nodos han emitido a otros nodos. Cada nodo posee dos listas de certificados recogidos: la *out-bound* y la *in-bound*, en la *out-bound* almacena los certificados de los nodos en que él confía y en la *in-bound* los certificados de los nodos que confían en él. Los certificados tienen un tiempo de vida limitado y deben ser renovados.

Para verificar un certificado los nodos tratan de construir un camino de confianza entre ellos utilizando los certificados que tienen en sus listas *in-bound* y *out-bound*, siempre teniendo en cuenta la confianza que les ofrece cada punto que emplean para construir la cadena. En caso de que les falte algún certificado para acabar de construir la cadena, los nodos se pueden ayudar de los certificados que tienen sus nodos vecinos (a un salto), siempre y cuando confíen en ellos. Además, los nodos pueden construir diferentes rutas de certificación y escoger la ruta que mayor confianza le brinde, un nodo podría llegar a recoger los certificados de todos contra todos.

En la Tabla 3-5 se resumen las ventajas y desventajas de esta propuesta.

Ventajas	Desventajas
- No necesita de ninguna infraestructura para su inicialización, ya que es autogestionado - Todos los nodos juegan el mismo rol dentro de la red de confianza. - La movilidad de los nodos favorece al funcionamiento del sistema.	- Ofrece una autenticación débil, ya que las relaciones del tipo A confía en B, B confía en C, por lo tanto A confía en C, no siempre son ciertas. - No existe una TTP que sirva de anclaje, por lo que en algunos casos construir un camino de certificación puede resultar imposible. - El sistema necesita de una fase de inicialización y calentamiento, hasta que los nodos hayan emitido y recogido suficientes certificados y puedan empezar a construir las rutas de confianza.

Tabla 3-5: Ventajas y desventajas de un sistema de autenticación basado en redes de confianza
Fuente: [51]

3.2.1.2.7 Modelos híbridos

Los modelos híbridos combinan funcionalidades de diferentes sistemas de gestión de claves, buscando generar nuevas soluciones que se adapten mejor a las características de los entornos ad hoc.

En [52] Luo y Hubaux presentan DICTATE, en este trabajo se propone la combinación de una CA centralizada y una CA distribuida. La CA centralizada denominada mCA escoge N nodos llamados servidores sobre los que monta la una CA distribuida (dCA). La mCA permanece fuera de línea (*off-line*) y solo se activa para: 1. Emitir certificados a los nuevos nodos que se quieren incorporar a la red y 2. Mantener y controlar la dCA, con la que se conecta periódicamente para recoger peticiones y sincronizar información. La dCA siempre está en la red y on-line, ejerciendo las funciones de una autoridad de registro (RA) distribuida, recogiendo las peticiones de los nodos dirigidas a la mCA, renovando certificados y actuando como una entidad de revocación distribuida. La dCA utiliza el esquema de criptografía de umbral. Para que esta solución sea factible es necesario que periódicamente se pueda realizar el proceso de sincronización entre dCA y mCA.

En [53] Yi y Kravets presentan *Composite key Management,* un modelo híbrido basado en los esquemas de CA distribuida y redes de confianza (*web of trust)*. En esta propuesta se implementa una CA distribuida y se definen métricas para asignar niveles de confianza a los caminos de confianza generados por las redes *web of trust*. En *Composite key Management* se definen tres tipos de nodos: nodos que forman parte de la CA, nodos que participan en la cadena de certificación y nodos clientes que utilizan los servicios de gestión de claves. El nivel de confianza se obtiene en función de si la cadena utiliza (o no) algún nodo que forma parte de la CA y del resto de nodos intermedios usados. Sigue siendo necesaria una CA *off-line* y una fase de inicialización para crear la CA distribuida.

El artículo considera las siguientes configuraciones:

- **Virtual CA with 1-hop certificate chaining.**
  Se implementa una CA virtual distribuida formada por varios nodos de la red y una serie de nodos que están certificados por esta CA. Los nodos que están certificados por la CA virtual están habilitados a certificar a otros nodos utilizando su propia clave privada. En esta configuración si un nodo desea adquirir un certificado pero no encuentra la CA virtual, el nodo puede buscar en su vecindario hasta encontrar algún nodo que haya sido certificado por la CA virtual. En la Figura 3-14 los nodos K1, K2, K3, K4, K5, K6, K7 se encuentran certificados por la CA virtual y los nodos K12, K13, K14, K15, K17 están certificados por nodos certificados por la CA virtual. Tanto la CA virtual como los nodos autorizados a certificar asignan una métrica de confianza o a los nodos que certifican, K2=1.0, K3=0.25, K13=0.9, etc.

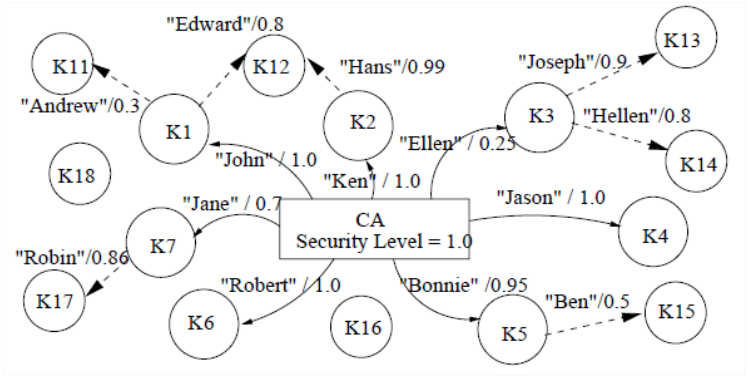

Figura 3-14: Virtual CA with 1-hop certificate chaining
Fuente: [53]

- **Certificate chaining with CA certified nodes.**
  Dentro del esquema Web of trust en el que todos se certifican contra todos, se utiliza una CA distribuida para certificar una serie de nodos para que tengan un mayor nivel de confianza. Estos nodos pueden ser utilizados para crear cadenas con niveles de confianza mas elevados. En la Figura 3-15 los nodos K5 y K8 están certificados por la CA y por lo tanto son confiables. K7 no puede autenticar K porque no dispone de una cadena de certificación desde K7 a K3, sin embargo K7 puede autenticar K3 siguiendo la cadena (K5→K4→K3).

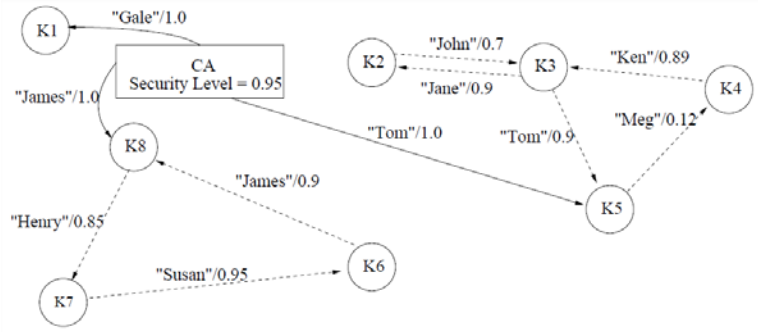

Figura 3-15: Certificate Chaining with CA certified Nodes
Fuente: [53]

3.2.2 Seguridad en el encaminamiento

El encaminamiento (ruteo o enrutamiento) es la actividad más importante que cumplen los nodos de una MANET para garantizar el normal funcionamiento de la red. Si uno de los nodos deja de encaminar correctamente los paquetes, puede comprometer el normal funcionamiento de toda la red ad hoc. El requerimiento más importante para un protocolo de ruteo es la implementación de mecanismos de seguridad que garanticen que la comunicación entre los nodos que componen la red sea segura, confiable y eficaz.

3.2.2.1 Protocolos de encaminamiento seguros

En [22] se clasifican los protocolos de encaminamiento seguros en dos clases principales: NO colaborativos y colaborativos (Figura 3-16). En los protocolos no colaborativos se utilizan mecanismos de prevención, mientras que en los protocolos colaborativos se implementan mecanismos de detección y reacción.

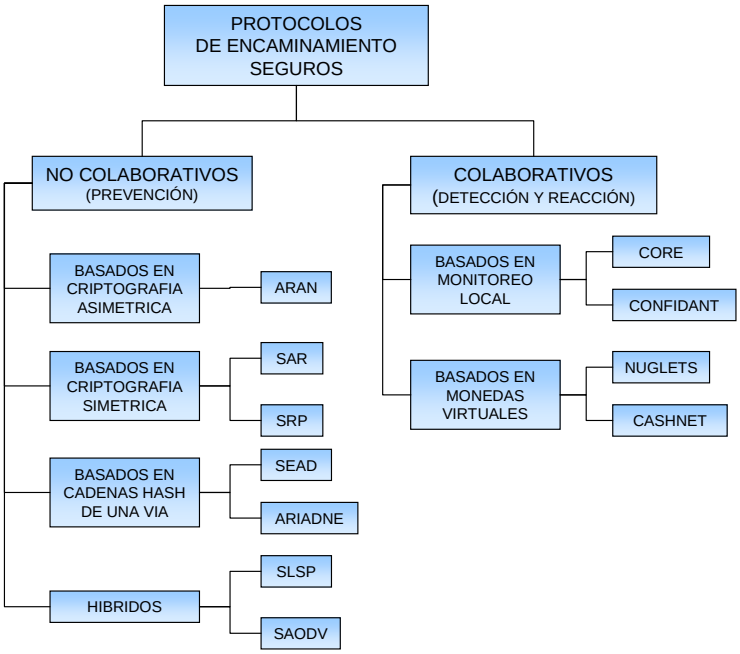

Figura 3-16:Clasificación de los protocolos de encaminamiento seguros
Tomado y ampliado de [22]

A continuación, y siguiendo el orden de aparición en la Figura 3-16, se realiza una descripción del funcionamiento de los principales protocolos de encaminamiento seguros para redes móviles ad hoc.

3.2.2.1.1 NO Colaborativos

En los protocolos de encaminamiento NO colaborativos se confía en TODOS los nodos, pero pueden existir nodos comprometidos de dos tipos, nodos maliciosos, que llevan a cabo ataques activos contra la integridad de la información transmitida, nodos egoístas, que degradan el funcionamiento de la red y pueden llegar a producir su partición efectiva.

Dentro de los protocolos de encaminamiento que utilizan mecanismos de prevención se encuentran los que utilizan criptografía asimétrica (ARAN [39]), los que utilizan criptografía simétrica (SAR [40] y SRP[41]), los que utilizan cadenas hash de una vía (SEAD [42] y ARIADNE [43]) y finalmente los que combinan criptografía y cadenas hash de una vía (SAODV [44], SLSP[45]).

3.2.2.1.1.1 ARAN

En [54] Sanzgiri, Dahill y Otros presentan ARAN, un protocolo de encaminamiento seguro que protege a la red Ad Hoc contra acciones maliciosas llevadas a cabo por terceras partes. Fue concebido como un protocolo de ruteo bajo demanda o reactivo que introduce el uso de criptografía asimétrica y firmas digitales para garantizar la autenticación, integridad y no repudio.

ARAN presenta algunas desventajas, que deben ser consideradas antes de su implementación:

- Requiere que los nodos mantengan una entrada en la tabla de encaminamiento por cada enlace activo entre nodos (origen-destino). Esto requiere una mayor utilización de recursos comparada con otros protocolos de encaminamiento que mantienen una entrada en la tabla de ruteo por destino.
- El uso de criptografía asimétrica lo vuelve muy costoso en términos de utilización CPU y consumo de energía.

ARAN esta compuesto por dos etapas. La primera etapa es obligatoria y consiste en un fase de certificación preliminar y otra de autenticación extremo a extremo (End to End) entre los nodos origen y destino. La segunda etapa es opcional y posibilita la obtención de rutas seguras más cortas.

Primera etapa

Fase de certificación primaria

ARAN requiere el uso de un servidor de certificados confiable llamado "T". Antes de entrar a la red, cada nodo debe solicitar un certificado firmado por el nodo "T". El certificado contiene la dirección IP del nodo, su clave pública, un sello de tiempo (*timestamp*) que indica cuando fue creado el certificado y el tiempo en que este expira junto con la firma del nodo T. Todos los nodos deben mantener certificados actualizados por el servidor de certificados y conocer la clave pública de este.

Cuando se necesita revocar un certificado, el servidor de certificados "T" envía un mensaje de broadcast que anuncia la revocación al grupo Ad Hoc. Cualquier nodo que recibe este mensaje lo retransmite hacia sus vecinos. Las notificaciones de revocación deben ser almacenadas hasta que el certificado revocado expire normalmente. Los vecinos del nodo con el certificado revocado modifican la información de encaminamiento (que tienen almacenada) para evitar la transmisión a través del nodo no confiable.

Fase de autenticación extremo a extremo

En esta fase el nodo origen verifica que el nodo destino esperado fue alcanzado. Para ello, el nodo origen "O" inicia el proceso de descubrimiento de ruta difundiendo un paquete RDP (Route Discovery Process) hacia sus vecinos. Este paquete incluye la dirección IP del destino y su certificado ($Cert_O$), todo esto firmado con la clave privada de "O". Cada nodo intermedio registra al vecino desde donde recibe el mensaje RDP y reenvía el mensaje firmado a sus vecinos, esta firma previene ataques de suplantación que pueden alterar la ruta o producir ciclos.

Debido a que el mensaje RDP se propaga por difusión, pueden llegar a destino varias copias del mensaje (por diferentes rutas). El nodo destino "D" responde solamente al primer RDP que recibe desde el origen, no existe ninguna garantía de que el primer RDP corresponda a la ruta mas corta desde el origen. La respuesta del nodo destino se envía utilizando un paquete unicast de respuesta (REP) que viaja hasta el origen por la ruta que siguió el paquete RDP pero en sentido inverso. Los nodos intermedios que reciben la respuesta REP, verifican la firma del salto previo, firman el paquete REP, y lo reenvían

hacia el nodo desde donde recibieron el RDP original. Esta firma del paquete REP protege contra ataques de suplantación o reproducción de mensajes de encaminamiento.

Cuando el nodo origen "O" recibe el paquete REP, verifica que el mismo tenga la firma del nodo destino "D". Solo el nodo destino responde un paquete RDP, los nodos intermedios que tengan caminos hacia el destino no pueden responder por el nodo destino.

Segunda etapa

En esta fase se realiza el descubrimiento de la ruta mas corta de forma segura.

El nodo origen "O" inicia el proceso de descubrimiento de ruta difundiendo un paquete SPC (Shortest Path Confirmation) hacia sus vecinos. Los nodos intermedios que reciben el mensaje, firman el mensaje y lo reenvían a sus vecinos. Cada vez que llega un paquete SPC, los nodos intermedios crean una entrada en sus tablas de ruteo que les sirve para no reenviar paquetes duplicados y para encaminar los paquetes de respuesta hacia el origen "O" en sentido inverso.

Cuando el nodo destino "D" recibe el SPC, verifica que todas las firmas sean validas y responde al primer SPC recibido y a cualquier SPC que tenga registrado un camino mas corto. Para la respuesta del SPC se utiliza un paquete RSP (Recorded Shortest Path) que se envía hacia el nodo origen "O" a través del nodo desde donde llego el SPC.

Los nodos intermedios que reciban los mensajes RSP, previa verificación de la autenticidad, actualizan sus tablas y reenvían el mensaje hacia el nodo origen "O" a través del nodo desde donde llego el mensaje SPC. Como el mensaje RSP solamente se envía cuando se detecta una ruta mas corta, las tablas de los nodos que intervienen quedarán actualizadas con esta ruta.

En la Figura 3-17 se observa el descubrimiento de ruta en ARAN, en el gráfico de la izquierda los paquetes RSP regresan al nodo origen A por la única ruta (E-C-B-A), en el gráfico de la derecha se incorpora el nodo D a la red y esta vez los paquetes RSP regresan al nodo origen por la ruta mas corta (E-D-A).

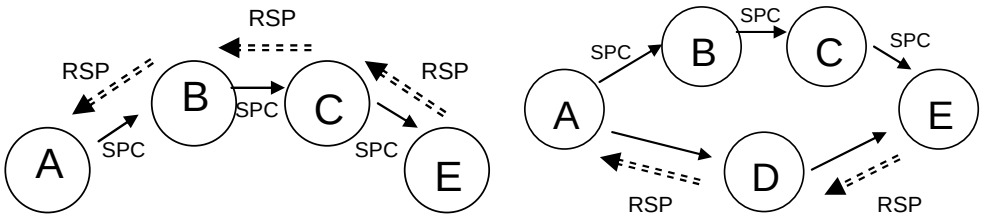

Figura 3-17: Descubrimiento de ruta en ARAN

Si no se detecta tráfico en una ruta existente, la misma se desactiva luego de transcurrido el tiempo de vida definido para esa ruta. Cuando se reciben datos que correspondan a una ruta desactivada, los nodos generan un mensaje de error (ERR) que viaja firmado en sentido inverso hacia el nodo origen "O". Los mensajes de error también se utilizan para informar de enlaces caídos en la red que pueden ser ocasionados por movilidad o fallo en los nodos.

3.2.2.1.1.2 SAR

SAR [55] hace uso de atributos de seguridad asignados a los nodos para tomar decisiones de encaminamiento. Los nodos seguros de la red ad hoc disponen de las claves necesarias para realizar las operaciones criptográficas. Un nodo confiable es capaz de descifrar los datos recibidos, tomar decisiones de encaminamiento, y si es necesario cifrar y reenviar los datos.

SAR permite descubrir una ruta entre dos nodos pasando por nodos intermedios que tengan atributos de seguridad deseados, por ejemplo: rutas formadas por nodos que tengan una clave compartida (Figura 3-18). Si una ruta con los atributos de seguridad requeridos es encontrada, un mensaje RREP es enviado desde el nodo intermedio hasta el nodo origen. En caso de que exista más de una ruta segura, se elige la ruta mas corta.

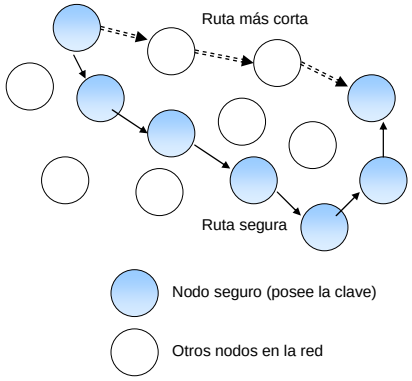

Figura 3-18: Descubrimiento de ruta segura en SAR

La distribución de claves se puede implementar utilizando alguno de los mecanismos descritos en el apartado 3.2.1

La principal desventaja de SAR es que los nodos intermedios deben realizar una gran cantidad de operaciones de cifrado y descifrado, lo cual impacta de forma directa en el consumo de recursos de cada nodo (CPU, energía).

3.2.2.1.1.3 SRP

En [56] Panagiotis Papadimitratos y Zygmunt Haas presentan SAR. Este protocolo fue diseñado como una extensión compatible con una variedad de protocolos de enrutamiento reactivos existentes (DSR, AODV y otros).

SRP confía en la disponibilidad de una asociación segura SA (Security Association) entre el nodo origen “O” y el nodo destino “D”. La SA puede establecerse mediante un esquema de distribución de claves basado en claves públicas de las partes en comunicación. “O” y “D” pueden intercambiar una clave secreta simétrica ($K_{O,D}$) utilizando las claves públicas de cualquiera de ellos para establecer un canal seguro. Utilizando la clave secreta simétrica ($K_{O,D}$), “O” y “D” pueden autenticarse mutuamente y autenticar los mensajes de encaminamiento.

SRP hace frente a nodos maliciosos no cooperativos capaces de modificar, reenviar y fabricar paquetes de encaminamiento. Además, posibilita al nodo iniciador del descubrimiento de ruta detectar y eliminar réplicas de información falsa.

3.2.2.1.1.4 SEAD

En [57] Hu y Johnson presentan SEAD, se trata de un protocolo de enrutamiento seguro proactivo basado en el Destination-Sequenced Distance Vector protocol (DSDV). En los protocolos proactivos de este tipo, los nodos periódicamente intercambian información de enrutamiento entre sí con la intención de que cada uno de los nodos conozca siempre la ruta actual hacia todos los destinos. El principal objetivo de SEAD es evitar que cualquier nodo malicioso publique rutas falsas o manipule el número de secuencia recibido por el origen. Implementa mecanismos para proteger modificaciones en la información de enrutamiento, tales como: cadenas hash de una vía, número de secuencia y ruta de origen. La naturaleza de las cadenas de hash de una vía previene que cualquier nodo anuncie una ruta con un número de secuencia mayor que el número de secuencia del origen.

Para apoyar el uso de SEAD en nodos de CPU limitada, y defenderse en contra de ataques del tipo DoS (denegación de servicio) en el cual un atacante intenta causar que otros nodos consuman excesivo ancho de banda o tiempo de procesamiento, el protocolo usa funciones hash en lugar de las operaciones de criptografía asimétrica que son altamente costosas en términos de CPU.

Este protocolo requiere de claves de secreto compartido para autenticar a los vecinos, estas claves se pueden intercambiar utilizando un protocolo de establecimiento de clave por pares (*pairwise shared secret keys*) [58] o bien protocolos de difusión como TESLA [59] o TESLA con *Instant Key Disclosure* (TIK) [60] .

3.2.2.1.1.5 ARIADNE

En [61] Allen, Gaughan y Otros proponen ARIADNE, un protocolo de ruteo bajo demanda basado en DSR. Según los autores este protocolo cuenta con un esquema altamente eficiente de criptografía simétrica, resiste la presencia de nodos comprometidos en la red y no requiere de hardware de alta confiabilidad ni de procesadores de alta performance.

ARIADNE implementa autenticación punto a punto entre los nodos utilizando códigos de autenticación de mensajes (MAC, Message Authentication Code) y una clave compartida entre los nodos participantes. Los esquemas de autenticación empleados por ARIADNE son:

- Claves secretas compartidas entre todos los pares de nodos
- Claves secretas compartidas entre los nodos en comunicación combinada con autenticación por difusión (broadcast).
- Firmas digitales.

El primero de los esquemas es el más eficiente sin embargo requiere de secretos compartidos entre todos los pares de nodos (*pairwise shared secret keys*) [58] , lo cual no siempre es posible de establecer. La segunda opción es el esquema de autenticación de TESLA [59], el cual está basado en encriptación asimétrica, lo que requiere de claves previamente desplegadas. El esquema que utiliza firmas digitales requiere de una autoridad de certificación y de la distribución de certificados.

Cuando un nodo origen “O” quiere comunicarse con el nodo destino “D” envía una solicitud de ruta RREQ, el mensaje RREQ lleva un código de autenticación de mensaje (MAC) que se calcula aplicando una función hash con la clave compartida K_{OD} sobre datos únicos (por ejemplo, un sello tiempo). El destino puede verificar fácilmente la autenticidad y que tan reciente es una petición de ruta, para ello calcula nuevamente el MAC usando la clave compartida K_{OD} y lo compara con el MAC recibido.

Durante el descubrimiento de ruta el nodo destino debe autenticar cada nodo de la lista de nodos del REQ (nodos intermedios), así él regresará un REP sólo a través de caminos que contienen nodos legítimos. Esto lo consigue aplicando la función hash en cada salto.

Los nodos intermedios, luego de verificar que el paquete sea valido, agregan su dirección a la lista de nodos, reemplazan la cadena hash por una nueva q incluya su dirección y agregan un MAC de todo el paquete a la lista de MAC. El nodo destino “D” verifica cada salto de la ruta comparando los MAC recibidos con los que calcula.

ARIADNE se utiliza para prevenir ataques realizados por nodos maliciosos que modifican y fabrican información de enrutamiento, es también efectivo frente a ataques de suplantación.

3.2.2.1.1.6 SAODV

En [62] Guerrero y Asokan presentan SAODV, una extensión de seguridad para proporcionar integridad y autenticación al protocolo de encaminamiento AODV. SAODV utiliza firmas digitales para autenticar los campos de los mensajes de Route Request (RREQ) y Route Reply (RREP) y cadenas hash para autenticar el contador de saltos (*hop count*). El encabezado AODV fue modificado para soportar estas extensiones de seguridad (Figura 3-19).

TYPE	LENGTH	HASH FUNCTION	MAX HOP COUNT
TOP HASH			
SIGNATURE			
HASH			

Figura 3-19: Encabezado del protocolo SAODV

Las firmas digitales se utilizan para proteger la integridad de los datos no modificables de los mensajes RREQ y RREP, se firma todo el mensaje excepto el contador de saltos de AODV y el hash de la extensión SAODV (se modifican en cada salto). Cuando un nodo requiere enviar un mensaje de encaminamiento (RREQ o RREP), firma digitalmente el mensaje y lo envía a los nodos vecinos. Los nodos intermedios que reciben un mensaje verifican la autenticidad del mensaje a través de la firma digital antes de ejecutar cualquier otra acción. Las cadenas hash son utilizadas para autenticar el contador de saltos de los mensajes RREQ y RREP, aseguran que esta métrica no sea alterada por algún atacante.

Antes de enviar el mensaje el nodo origen realiza el cálculo del TOP HASH de la siguiente manera:

- Genera un número aleatorio denominado "semilla", que almacena en el campo HASH.
- Ajusta el valor del campo Max_Hop_Count en el mensaje con el valor del TTL del paquete.
- Utiliza la función Hash para calcular el campo TOP HASH. Este valor se obtiene aplicando la función hash a la semilla la cantidad de veces especificada por Max_Hop_Count.

  $$\text{TOP HASH} = H^{\text{Max Hop Count}}(\text{seed}),\ \text{donde H es la función hash}$$

- Toda esta información es almacenada en el mensaje a enviar

Cuando un nodo intermedio recibe un mensaje RREQ o RREP, la verificación de la cantidad de saltos (hop count) se efectúa de la siguiente forma:

- Aplica la función hash al valor almacenado en el campo HASH (seed) la cantidad de veces especificada por la diferencia entre MAX HOP COUNT Y HOP COUNT.

  $$\text{Top\_Hash\_verificador} = ^{\text{Max Hop Count} - \text{Hop\_Count}}(\text{campo HASH})$$

- Verifica que el resultado obtenido en "Top_Hash_verificador" sea igual al que esta almacenado en el campo TOP HASH. Si son iguales el mensaje será considerado valido y reenviado, si no son iguales el mensaje se descarta.
- Si el mensaje es considerado valido el nodo intermedio aplica la función hash al valor del campo HASH y lo reenvía a los nodos vecinos.

3.2.2.1.2 Colaborativos

En los protocolos de encaminamiento colaborativos se implementan los siguientes esquemas de seguridad para lidiar con nodos maliciosos o egoístas:

- **Basados en monitoreo local**
  Donde cada nodo monitorea a sus vecinos evaluando para cada uno una métrica que refleja su comportamiento (reputación) hasta ese momento, lo que permite aislar gradualmente a los nodos egoístas. (CORE[63], CONFIDANT[64]).
- **Basados en monedas virtuales**
  En los que los usuarios reciben un incentivo para cooperar en las operaciones de la red. (NUGLETS [65], CASHNET[66]).
- **Híbridos**
  Combinan ambos conceptos, cada nodo debe poseer un ticket para participar en la red, y son sus vecinos, que monitorean su comportamiento, quienes deciden al vencimiento de dicho ticket, si puede renovarlo o no.

3.2.2.1.2.1 CORE

En [63] Molva y Michiardi presentan CORE (A COllaborative REputation mechanism to enforce node cooperation in Mobile Ad Hoc Networks), se trata de un protocolo que

estimula la cooperación entre los nodos utilizando una técnica de monitoreo colaborativo y un mecanismo basado en la reputación de los nodos.

Cada nodo de la red monitorea el comportamiento de sus vecinos con respecto a una función y reúne observaciones acerca de la ejecución de esa función. En base a las observaciones recolectadas, cada nodo calcula un valor de reputación de sus vecinos utilizando un mecanismo que toma en cuenta:

- Las observaciones realizadas (reputación subjetiva).
- Los reportes positivos de otros nodos (reputación indirecta).
- El comportamiento del nodo frente a una tarea específica (reputación funcional).

La fórmula utilizada para calcular la reputación evita falsas detecciones a través del uso de un factor de envejecimiento el cual da más relevancia a observaciones pasadas, las variaciones del comportamiento de un nodo son filtradas. Además, si la función que está siendo monitoreada provee un mensaje de asentimiento, la información de reputación puede también obtenerse de los nodos que no se encuentran en el rango de radio del nodo monitoreado. En este caso, solamente las calificaciones positivas son asignadas a los nodos que participaron en la ejecución de la función en su totalidad.

El mecanismo de reputación de CORE permite a los nodos de la MANET aislar gradualmente a los nodos egoístas, cuando la reputación asignada a un nodo vecino se reduce por debajo al umbral predefinido, la provisión del servicio hacia el nodo con mal comportamiento se interrumpe. Ninguna calificación negativa es desplegada entre los nodos, por lo tanto es imposible que un nodo reduzca de manera maliciosa la reputación de otro nodo.

3.2.2.1.2.2 CONFIDANT

En [64] Buchegger y Le Boudec proponen CONFIDANT (Cooperation of nodes, Fairness in dynamic ad-hoc networks). En primer lugar, utilizan un *watchdog*[1] para detectar qué nodos no participan en las rutas de reenvío de paquetes, y en segundo lugar definen un mecanismo de reputación en sustitución para proteger el protocolo de encaminamiento DSR. El watchdog, que se encuentra en ejecución en cada nodo consigue realizar su tarea escuchando la red en modo promiscuo. Si el siguiente nodo de la ruta encamina correctamente el paquete se lo considera colaborativo, en caso contrario, se lo considera egoísta.

CONFIDANT detecta nodos atacantes mediante la combinación de monitoreo y establecimiento de rutas que eviten nodos egoístas. Los paquetes provenientes de nodos egoístas no son enviados por los nodos honestos. Si un nodo es acusado erróneamente, o se arrepiente y se comporta de forma honesta durante cierto tiempo, se lleva a cabo su redención (re-socialización) y reintegración en la red.

3.2.2.1.2.3 NUGLETS

[1] Servicio activo en cada nodo que monitorea las comunicaciones de su vecindario para, mediante el uso de herramientas estadísticas, identificar a aquellos nodos cuyo comportamiento no se ajuste a parámetros de normalidad.

En [65] Buttyán y Hubaux presentan una propuesta basada en moneda virtual. En este trabajo definen una moneda de pago a la que llaman nuglet, que es utilizada para construir un mecanismo (gestionado por hardware) que incentiva el reenvío de paquetes.

Las precondiciones del esquema propuesto son dos:

- Cada nodo está equipado con un contador de nuglets, que debe estar necesariamente dentro de un módulo de seguridad resistente a manipulaciones (tamper resistant, en inglés).
- Cada nodo tiene instalado un certificado en el módulo de seguridad, que esta perfectamente identificado por un mecanismo de Infraestructura de Clave Pública (o PKI, de sus siglas en inglés).

El mecanismo propuesto funciona de la siguiente manera:

- Cuando un nodo desea enviar un paquete, consulta su contador de nuglets.
- Si tiene suficientes nuglets como para conseguir retribuir a todos los nodos intermedios hasta el destino, lo envía al primer nodo intermedio de la ruta, y ejecuta el protocolo de sincronización de nuglets controlado por el módulo de seguridad.
- Cada nodo intermedio irá recibiendo su nuglet por el reenvío del paquete, hasta que éste llegue a su destino. Los nuglets que acumule por reenvío de paquetes, son los que luego cada nodo usa para enviar sus propios paquetes siguiendo este mismo esquema.

A continuación se describen los problemas que surgen en la implementación del mecanismo de pago virtual propuesto:

- Es complicado construir hardware que sea resistente a manipulaciones. Si un usuario malicioso logra manipular el hardware de un nodo y modificar el contador de nuglets, podría disponer de crédito ilimitado para el envío de sus paquetes y seguir negándose a reenviar los de otros sin verse penalizado por ello.
- Se requiere montar una PKI sobre la MANET, esta es una tarea de cierta complejidad como se explico en el apartado 3.2.1.
- El uso de las técnicas de criptografía supone una sobrecarga importante en nodos con capacidad de procesamiento reducida.
- No se tiene en cuenta el tamaño de los paquetes que se transmiten para valorar el costo del servicio. Se cobra siempre un nuglet por paquete reenviado.
- Se pueden producir situaciones de inanición en los nodos ubicados en los bordes de la MANET, que no son requeridos por ningún otro para reenvío de paquetes, pero que sí consumen su crédito de nuglets para enviar los suyos. Por ello, periódicamente, el sistema deba recargar el contador de nuglets de cada nodo con un determinado valor, lo que desvirtúa la motivación de obtener nuglets a cambio de reenviar paquetes de otros.

3.2.2.1.2.4 CASHnet

En [66] Wetland and Braun presentan CASHnet (Cooperation and Accounting Strategy for Hybrid Networks). Los autores proponen un esquema que soporta un sistema de pago virtual basado en el rol del nodo (remitente o receptor), esto conduce a una mejor equidad

ya que no tenemos que preocuparnos por el hecho de que el remitente y el receptor cambien sus roles de acuerdo con la dirección de la transmisión. Por ejemplo, en una descarga de Internet, el receptor es el que tiene todo el beneficio, y debe pagar por ello.

En el esquema CashNet, existen dos tipos de monedas: créditos de tráfico y créditos auxiliares. Cada vez que un nodo desea enviar un mensaje sólo tiene que pagar créditos de tráfico de acuerdo con el número de saltos en su red. Si un nodo reenvía un mensaje, recibe un crédito auxiliar. El esquema introduce estaciones de servicio que se encuentran dentro de las redes ad-hoc. Un usuario que quiera cambiar sus créditos auxiliares por créditos de tráfico tiene que ir a una estación de servicio. También es posible comprar créditos adicionales de tráfico en estas estaciones de servicio. Este mecanismo les deja a los nodos dos opciones: cooperar o pagar. Como un nodo no gana créditos auxiliares si no envía nada, la cooperación sería una alternativa lucrativa para el egoísmo.

En la Figura 3-20 se ilustra la topología de CASHnet. Existe por lo menos una estación de servicio de créditos en cada red y un Gateway de comunicaciones que conecta la MANET a Internet.

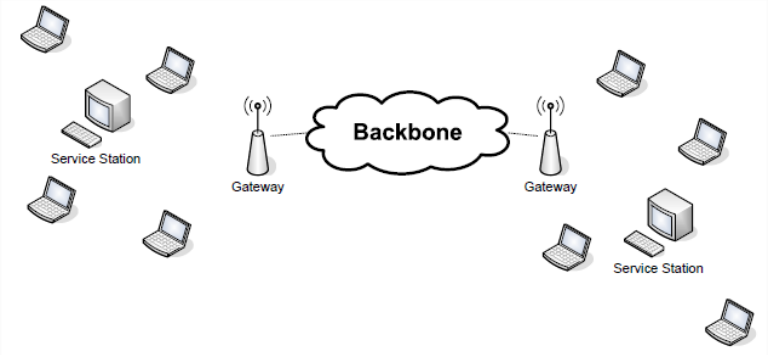

Figura 3-20: Topología de CASHnet
Fuente: [66]

De manera similar a NUGLETS [65], para proteger la información crítica, CASHnet requiere que los nodos participantes sean dispositivos resistentes a la manipulación (*tamper resistant*). Cada nodo debe garantizar la integridad de los datos y autenticación del origen de datos, para ello debe tener las claves necesarias y capacidad de procesamiento suficiente para realizar las operaciones de firma y verificación.

3.2.3 Sistemas de detección de intrusiones (IDS)

El IDS es un programa encargado de controlar la actividad de la red con el fin de detectar acciones maliciosas y, en última instancia, un intruso. Tras la detección de un intruso, el IDS toma una acción apropiada que puede ir desde una simple notificación de usuario a una más amplia como eliminar la comunicación del nodo intruso dentro de la red y excluirlo por completo de la red para evitar una nueva intrusión en futuras ocasiones.

Las funciones principales de un IDS son:

- **Monitorear**
  Examinar y procesar información acerca de las actividades dentro de la MANET.

- **Reportar**
  Generar información acerca del sistema monitoreado dentro de una infraestructura de seguridad del sistema. Por ejemplo, generando logs de eventos de seguridad en la red.
- **Responder**
  Reaccionar y responder a la intrusión detectada. Por ejemplo, generando alarmas para informar al administrador(es) de seguridad de la red.

Un IDS se puede dividir en dos partes principales:

- La arquitectura, que ejemplifica la estructura operativa de la IDS.
- El motor de detección, que es el mecanismo utilizado para detectar intrusos o comportamientos maliciosos.

Los motores de detección de intrusos se clasifican en tres categorías principales (Figura 3-21):

1. Motores basados en firmas: se basan en un conjunto predefinido de patrones para identificar ataques.
2. Motores basados en anomalías: se basan en modelos particulares de comportamiento, los nodos que se desvían de estos modelos de comportamiento se marcan como maliciosos.
3. Motores basados en especificaciones: se basan en un conjunto de especificaciones de operación correcta de programas y protocolos. El motor supervisa la ejecución de programas y protocolos con respecto a estas especificaciones.

A su vez, las arquitecturas de IDS para Manet se dividen en tres categorías: Autónomas, cooperativas y jerárquicas (Figura 3-21).

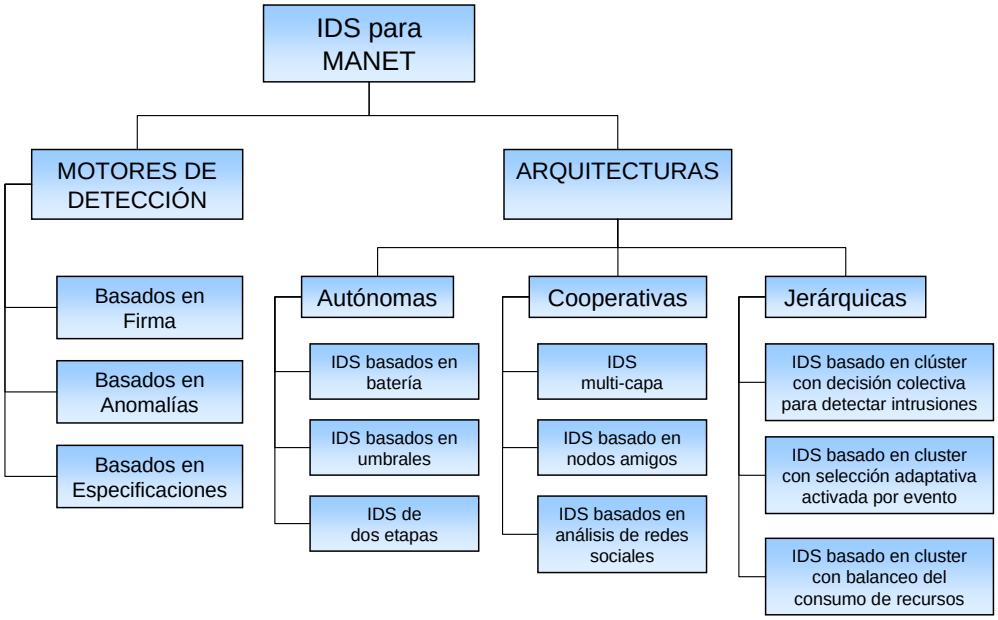

Figura 3-21: Clasificación de las arquitecturas IDS para MANET
Fuente: [67]

A continuación se describen las principales características de las soluciones propuestas para arquitecturas IDS en cada una de las categorías.

3.2.3.1 Arquitecturas Autónomas

Las arquitecturas autónomas se basan en un enfoque auto-contenido (self-contained) para la detección de acciones maliciosas en cada nodo de la red. Instalan un motor de detección de intrusos (en cada nodo) que utiliza solamente los datos de auditoría locales. El hecho de que estas soluciones se basan sólo en datos de auditoría locales para resolver comportamientos maliciosos, los limita en términos de precisión en la detección y el tipo de ataques que detectan.

3.2.3.1.1 IDS basados en batería

En [68] Jacoby y Davis presentan una arquitectura autónoma para detectar acciones maliciosas en MANETs, su trabajo esta basado en el control del consumo de energía (batería) de cada nodo. La detección se logra mediante la comparación de consumo de energía de un nodo con un conjunto de patrones de consumo de energía inducidos por ataques conocidos, esto se consigue utilizando tecnología inteligente para la gestión de la batería.

Debido a que se basa en componentes de hardware, esta arquitectura es muy fiable en comparación con otras arquitecturas IDS que se basan en datos de auditoría y detección de anomalías, ya que estos datos pueden llegar a ser manipulados por intrusos. Por otro lado, sólo detecta ataques que causan irregularidades de consumo de energía en los casos en que los nodos están inactivos, algo que rara vez ocurre en los sistemas reales.

3.2.3.1.2 IDS basados en umbrales

En [69] Nadkarni y Mishra proponen una arquitectura de IDS independiente que utiliza umbrales de ajuste para determinar comportamientos maliciosos, con el objetivo de reducir la cantidad de alertas falso-positivo que aparecen en la detección de anomalías. Durante la fase de inicialización, el motor de detección de intrusiones instalado en cada nodo crea el perfil de normalidad del tráfico de la red. Basado en este perfil, se calculan los valores de umbral, más allá de los cuales hay una indicación de posibles ataques. Cada vez que se detecta un síntoma de un ataque conocido, un contador llamado “mis-incident” se incrementa y el nodo responsable por el síntoma se marca como sospechoso. Si se repite el incidente y el contador “mis-incident” excede el valor de umbral para el ataque específico, el nodo desde donde se origina el incidente se etiqueta como malicioso. Después de un período de tiempo predeterminado en el que no hay comportamientos maliciosos detectados, el umbral se eleva, de lo contrario se baja.

Debido a la utilización de los umbrales variables esta arquitectura es adaptable a los cambios en la red. Por ejemplo, los síntomas periódicos de comportamientos sospechosos, causados por cambios en la topología de la red, se mantendrán por debajo de los umbrales de detección, mientras que los comportamientos maliciosos que son constantes superarán los umbrales que indiquen la ocurrencia de ataques o intrusiones.

Por otro lado, el uso de umbrales de ajuste introduce nuevos fallos de seguridad que pueden ser explotados por nodos maliciosos. Por ejemplo, un nodo malicioso puede generar el aumento de los valores de umbral mediante la ejecución legítima (comportamiento NO malicioso) durante un cierto período de tiempo. Cuando los valores de umbral sean lo suficientemente elevados, puede modificar su buen comportamiento para realizar un ataque, teniendo la precaución de no exceder los valores límite de umbral para no generar alarmas.

3.2.3.1.3 IDS de dos etapas

En [70] Lauf y Robinson han propuesto una arquitectura de dos etapas diseñada para operar en entornos MANET con recursos limitados. Se instalan dos motores de detección diferentes en cada nodo, el primero llamado sistema de detección de máximos (MDS) se utiliza para identificar rápidamente una amenaza potencial y calibrar el segundo motor, denominado sistema de detección cruzada correlativa (CCDS).

MDS es un motor de detección de anomalías que identifica diferencias estadísticas en las interacciones observadas en la capa de aplicación. Para conseguir esto, MDS mantiene un archivo histórico de las interacciones de capa de aplicación, las interacciones almacenadas en este archivo se comparan con un perfil de normalidad creado fuera de línea. Si se identifica un posible ataque, MDS activa CCDS que calibra un valor de umbral teniendo en cuenta el ataque. A continuación, calcula los valores medios del comportamiento de la aplicación en cada nodo y los compara con el valor de umbral. Los comportamientos que exceden el umbral serán marcados como maliciosos.

Mediante el empleo de dos motores de detección en cada nodo, la propuesta IDS aumenta la precisión de detección, en comparación con arquitecturas IDS de un solo motor.

3.2.3.2 Arquitecturas Cooperativas

Las arquitecturas cooperativas utilizan técnicas de colaboración entre nodos para detectar con mayor precisión un conjunto más amplio de intrusiones y ataques.

Los IDS cooperativos instalan un motor de detección de intrusos en cada nodo, este motor es el encargado de monitorear los datos de auditoría locales y de garantizar la disponibilidad del mecanismo de detección de intrusos. Ante la presencia de un ataque y/o intrusión se consultan los datos de auditoria locales para identificarlos, si los datos de auditoria locales no son suficientes, el motor de detección del nodo intercambia datos de auditoría y/o resultados de detección con los motores de nodos vecinos. En base a la información local (en el nodo) y global (facilitada por los nodos vecinos) el nodo puede resolver intrusiones inconclusas y detectar con precisión ataques avanzados.

3.2.3.2.1 IDS multi-capa

En [71] Bose y Kannan proponen una arquitectura IDS cooperativa que utiliza tres motores paralelos de detección de anomalías instalados en cada nodo:

- Motor de detección de capa MAC, monitorea el control de acceso y direccionamiento en capa de enlace de datos .

- Motor de detección de encaminamiento, supervisa la capa de red y realiza un seguimiento de la entrega de paquetes y la información de ruteo.
- Motor de detección de nivel de aplicación, monitorea las funciones de la capa de aplicación.

Cada motor recoge los datos de auditoría apropiados, los procesa y busca comportamientos maliciosos dentro de ellos. En cada nodo, un módulo de integración local combina los resultados de los tres motores de detección diferentes, mientras que un módulo de integración global combina los resultados recibidos de los nodos vecinos.

El uso de la detección en múltiples capas tiene como objetivo aumentar la precisión de detección, ya que los ataques a los protocolos de capas superiores pueden ser vistos como eventos legítimos en-capas inferiores, y viceversa. Los motores de detección de cada capa se complementan mutuamente, esto hace que la capacidad de detección sea superior en comparación con soluciones de detección individuales. Por otro lado, su funcionamiento aumenta la sobrecarga de procesamiento en cada nodo, en comparación con otras soluciones de un solo motor, ya que el IDS despliega tres motores de detección en lugar de uno.

3.2.3.2.2 IDS basado en nodos amigos

En [72] Razak y Clarke proponen una arquitectura cooperativa de dos niveles, uno para la detección local y otro para la global. Además, cada nivel implementa dos motores de detección.

El primer nivel utiliza un mecanismo de detección que recoge los datos de auditoría locales (en el nodo) y los procesa utilizando un motor de detección basado en firmas. Si se detecta una actividad sospechosa y no se puede determinar con precisión un ataque específico, se activa un segundo motor (que también se encuentra en el primer nivel) que lleva a cabo la detección de anomalías. Si los dos motores en el primer nivel no pueden determinar si la actividad sospechosa es maliciosa, se dispara el segundo nivel de la arquitectura. El segundo nivel utiliza un mecanismo de detección global que recoge datos de auditoría de los nodos vecinos, primero realiza una detección basada en firmas y a continuación una detección basada en anomalías, de manera similar a la de primer nivel. Cada nodo construye y mantiene una lista de nodos confiables (amigos), que se utiliza para garantizar que los datos de auditoría proporcionados por los nodos amigos sean confiables.

Esta arquitectura proporciona una alta precisión de detección ya que cada nodo contiene un módulo de detección de dos niveles, y cada nivel incluye dos motores de detección diferentes que se complementan, uno que utiliza detección basada en firmas y otro que utiliza detección basada en anomalías. Por otra parte, el uso de múltiples detecciones (dos niveles, cada uno con dos motores de detección) y la gestión de la confianza añaden una complejidad considerable y carga de procesamiento adicional.

3.2.3.2.3 IDS basados en análisis de redes sociales

En [73] Wang, Man y Liu proponen una arquitectura de IDS cooperativa que utiliza métodos de análisis de redes sociales. Según esta propuesta, cada nodo despliega un motor de detección de intrusos para realizar detecciones utilizando los datos de auditoría recibidos

de su red "ego", los motores desplegados utilizan las relaciones sociales como métricas de interés. Una red de "ego" se compone de un nodo de alojamiento ("yo") y los nodos ("alters") que están directamente conectados a él. Los motores de detección desplegados por los nodos son capaces de monitorear la capa de enlace (MAC) y la capa de red.

El IDS propuesto se compone de tres módulos:

1. El módulo de pre-procesamiento de datos, que recolecta y pre procesa los datos de auditoría.
2. El módulo de análisis social que realiza la detección de intrusiones.
3. El módulo de respuesta que integra alertas de intrusión local y global (obtenidas de los nodos vecinos).

Durante la operación de IDS, el módulo de pre-procesamiento de datos periódicamente recolecta los datos de auditoría de sus nodos vecinos. A continuación, el módulo de análisis social, procesa los datos recogidos con el fin de darse cuenta de las relaciones sociales entre los nodos de la red "ego", que representan el comportamiento de estos nodos en un momento determinado. Posteriormente, las relaciones sociales se comparan con el perfil normal de los comportamientos esperados, cualquier variación de estos constituye una intrusión. Si se detecta una intrusión, el módulo de respuesta notifica a los nodos vecinos.

La principal ventaja de esta arquitectura es que los motores de detección empleados demandan una complejidad de cálculo menor, en comparación con los motores convencionales de detección de anomalías. Por otro lado, el intercambio de datos de auditoría puede aumentar la carga de comunicación entre los nodos y producir una baja en el rendimiento de la red.

3.2.3.3 Arquitecturas Jerárquicas

Las arquitecturas jerárquicas implementan un enfoque de múltiples capas, mediante la división de la MANET en agrupaciones de nodos denominadas clusters. En la Figura 3-22 se ilustra la división de una MANET en tres clusters, los nodos pueden adoptar tres roles diferenciados:

- Coordinador de cluster (*Cluster-Header*): Nodo que gestiona la formación y el mantenimiento del cluster.
- Miembro de cluster: Nodo normal.
- Gateway de cluster: Nodo con enlaces inter-cluster. Los nodos Gateway son los encargados de reenviar información entre clusters.

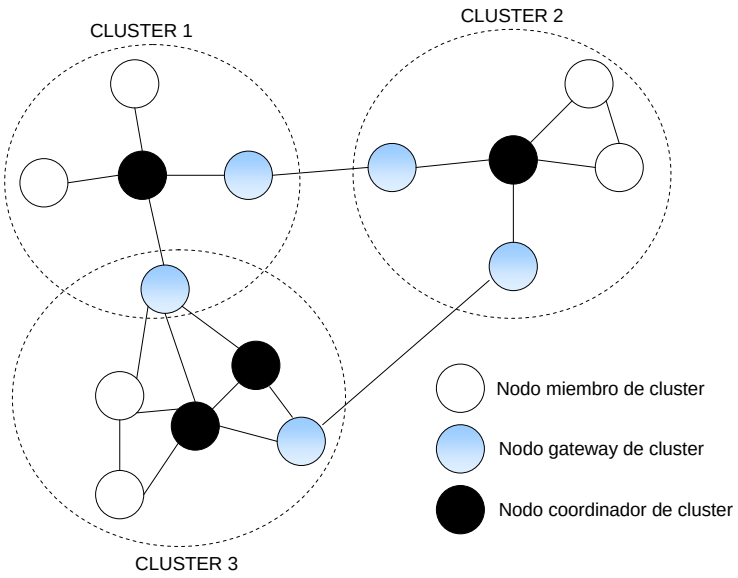

Figura 3-22: MANET dividida en clusters

En las arquitecturas jerárquicas de IDS, una cantidad de nodos son seleccionados (sobre la base de criterios específicos) para actuar como coordinadores de cluster y llevar a cabo responsabilidades y funciones de detección de intrusos que suelen ser diferentes de las de los miembros normales del clúster. Los nodos normales del cluster ejecutan un motor de detección ligero que realiza la detección utilizando datos de auditoría locales, mientras que los coordinadores de cluster ejecutan un motor integral que actúa como una segunda capa de detección sobre la base de datos de auditoría de todos los miembros del clúster.

3.2.3.3.1 IDS basado en cluster y detección de anomalías

En [74] Zhang y Otros proponen una arquitectura de IDS basada en clúster en la que sólo los nodos coordinadores llevan a cabo la detección de intrusiones. En esta arquitectura, cada nodo vota como coordinador al nodo vecino que tenga el mayor número de conexiones y el remanente de energía más alto, los nodos con mayor cantidad votos se convierten en coordinadores.

Los nodos coordinadores despliegan un motor de detección de anomalías que monitorea:

- La propagación de paquetes de protocolo de enrutamiento específicas (Hello, Error, Solicitud de ruta (RREQ), Respuesta (RREP), Etcétera)
- Los cambios en las tablas de ruteo.
- La transmisión de paquetes de datos.

Estas características son monitoreadas aleatoriamente, o mediante la selección de un coordinador al cual los miembros del clúster le transmitan su conjunto propio de características, o activamente mediante la configuración de un coordinador de cluster para escuchar el tráfico generado en el clúster.

Los nodos coordinadores se reeligen periódicamente, después de un intervalo de tiempo predefinido, y teniendo en cuenta la conectividad del coordinador con otros nodos de la red. Esto asegura que los coordinadores cluster elegidos controlen gran parte de las actividades de la red, facilitando IDS para llegar a decisiones más precisas. Además, como los coordinadores rotan después de un cierto período de tiempo, la carga de de procesamiento se distribuye equitativamente entre los nodos. Por otro lado, la gran desventaja de este tipo de IDS es que el motor de detección empleado es sólo capaz de detectar ataques de encaminamiento.

3.2.3.3.2 IDS basado en clúster con elección de coordinador activada por evento

En [75] Ma Chuan y Fang Ze Ming proponen una arquitectura modular basada en clusters, donde los nodos coordinadores se eligen en función de la cantidad de energía disponible en su batería.

Durante la inicialización de la red, cada nodo informa el nivel de carga de energía (en su batería) a sus vecinos. En función de esta información, los nodos eligen al nodo coordinador de cluster, será aquel que tenga el mayor nivel de carga de batería.

El procedimiento de reelección de coordinador se activa si se produce alguno de los siguientes eventos:

- Un nuevo nodo se une a la red.
  Se notifica primero a todos los nodos vecinos para que inicien el procedimiento de reelección.
- El nodo coordinador electo abandona la red.
  Se envía un paquete de notificación (por difusión) a todos los nodos miembros del cluster para que inicien el procedimiento de reelección del nodo coordinador de cluster.
- La energía de la batería del nodo coordinador es menor que un umbral predefinido.
  Se procede ídem al evento anterior.

En esta arquitectura IDS, los nodos de la MANET contiene cuatro módulos, que se describen a continuación:

- El módulo de detección local, que vigila al nodo y genera alertas locales si se detectan actividades maliciosas. Este módulo está siempre activo en cada nodo.
- El módulo de detección de red, que proporciona un control de paquetes de red dentro de un cluster. Se activa sólo si el nodo es elegido como coordinador.
- El módulo de gestión de recursos, que supervisa los recursos energéticos de un nodo que actúa como coordinador. Cuando la energía de la batería es inferior a un umbral predefinido, el módulo inicia el procedimiento de reelección de nodo coordinador.
- El modulo de monitoreo de estado, que gestiona si el módulo de detección de red está activo. Por ejemplo, cuando un nodo es elegido coordinador.

La arquitectura de IDS presenta las siguientes ventajas:

- Los nodos con mejor carga de batería (remanente) son elegidos para actuar como coordinadores.

- Implementa dos capas de detección (local y de red), lo cual proporciona un mayor grado de precisión en la detección.

Por otro lado, surgen las siguientes desventajas:

- La creación y mantenimiento de clusters y la elección de nodos coordinadores añaden procesamiento y comunicación adicional (*overhead*).
- Los nodos del clúster elegidos como coordinadores son injustamente sobrecargados, ya que son responsables del funcionamiento de los módulos de detección, tanto locales como de red.
- En entornos MANET de alta movilidad se reduce la precisión en la detección y aumenta la proporción de falsos positivos. Esto se debe a que algunos nodos miembro pueden salir de la rango de cobertura de un coordinador, lo que limita la información que el módulo de detección de red puede utilizar para efectuar la detección.

3.2.3.3.3 IDS basado en cluster con equilibrio del consumo de recursos

En [76] Otrok, Mohammed, Wang y Otros proponen un enfoque jerárquico que intenta balancear el consumo de recursos, utilizados para las tareas de detección de intrusos, entre los nodos que forman parte de un clúster. Incentiva a los nodos de la red para participar en la elección de nodos coordinadores e intenta evitar que nodos con mal comportamiento sean elegidos coordinadores.

En la arquitectura propuesta, los nodos pueden funcionar como:

- Coordinadores de cluster, que son responsables de la detección de intrusos dentro de un cluster. Para detectar intrusiones, el nodo coordinador despliega un motor de detección basado en un mecanismo de juego no-cooperativo de suma cero[2], donde los jugadores son el coordinador y un posible intruso.
- Inspectores de cluster, que son miembros del clúster seleccionados al azar para monitorear y detectar comportamientos egoístas o maliciosos en el nodo coordinador. Si un inspector tiene indicaciones de mala conducta de parte de un coordinador, intercambia información con otros nodos inspectores para tomar una decisión cooperativa, que puede derivar en la elección de un nuevo coordinador.
- Miembros de cluster, que no tienen responsabilidades de detección de intrusiones.

Durante la fase de inicialización, los nodos de la red informan de la energía disponible en sus baterías a sus nodos vecinos, en base a esta información cada nodo crea una lista con la capacidad energética de sus vecinos. A partir de esta lista, cada nodo vota al nodo con la mayor capacidad energética para ser elegido como coordinador, el nodo elegido solamente monitorea a los nodos participantes del proceso electoral. Cuando el remanente de energía de la batería del nodo coordinador se encuentra por debajo de un valor de referencia, se repite el proceso electoral (después de un tiempo de periodo transcurrido) y se elige un nuevo coordinador de cluster.

[2] En teoría de juegos no-cooperativos, un juego de suma cero describe una situación en la que la ganancia o pérdida de un participante se equilibra con exactitud con las pérdidas o ganancias de los otros participantes.

Las principales ventajas de esta arquitectura son:

- Se eligen como coordinadores a aquellos nodos que tengan un nivel alto de carga en su batería.
- El mal comportamiento de los nodos coordinadores puede ser detectado por los nodos inspectores.

Por otro lado, esta arquitectura presenta las siguientes desventajas:

- La formación y mantenimiento de clusters y el funcionamiento de los nodos inspectores, requiere procesamiento adicional e introduce sobrecarga en las comunicaciones (*overhead*).
- Los nodos coordinadores e inspectores son injustamente sobrecargados con responsabilidades de monitoreo y detección

3.2.3.3.4 Debilidades de las arquitecturas jerárquicas basadas en cluster

A continuación se enumeran las principales debilidades que aparecen en las arquitecturas jerárquicas basadas en cluster que fueron descriptas con anterioridad:

- Los nodos coordinadores pueden convertirse en puntos de falla.
- Los nodos maliciosos o egoístas que no cooperan pueden obstaculizar la detección de intrusos o inducir errores.
- Los nodos maliciosos pueden acusar falsamente (como maliciosos) a nodos legítimos.
- No se tienen en cuenta la capacidad de procesamiento de los nodos en la elección de los nodos coordinadores.

3.2.4 Sistemas de tolerancia a intrusiones

La tolerancia a intrusiones se centra en la capacidad de "supervivencia" de una red cuando está bajo el ataque de un adversario. La supervivencia es la capacidad de la MANET para normalizar los servicios esenciales luego de producirse un fallo o incidente malicioso y la rápida recuperación de los servicios completos cuando las condiciones mejoren. A diferencia de las medidas de seguridad tradicionales que requieren administración centralizada, la supervivencia abarca entornos de redes distribuidas que carecen de control centralizado y políticas de seguridad unificadas.

En [77] se destacan las siguientes propiedades de las que depende la supervivencia de una MANET: Resistencia, Reconocimiento, Recuperación y Adaptabilidad. La Figura 3-23 ilustra la interacción entre estas propiedades.

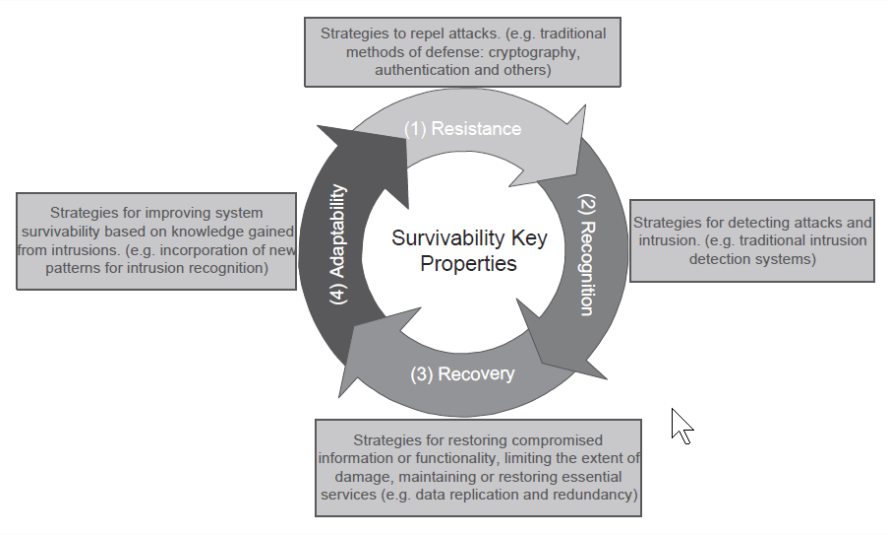

Figura 3-23: Propiedades de la supervivencia en MANET
Fuente: [77]

- La resistencia es la capacidad de un sistema para resistir los ataques. La autenticación y la criptografía son ejemplos de mecanismos de resistencia.
- El reconocimiento es la capacidad del sistema para detectar ataques y evaluar la magnitud de los daños. los sistemas de detección de intrusos (IDS) son un ejemplo de mecanismo de reconocimiento.
- La recuperación es la capacidad de restaurar la funcionalidad interrumpida dentro de límites de tiempo tolerables, acotando el daño y mantenimiento activos los servicios esenciales. En general, las estrategias convencionales utilizadas para lograr la recuperación son la replicación y la redundancia.
- La adaptabilidad es la capacidad del sistema para incorporar rápidamente las lecciones aprendidas de los fracasos y la adaptación a nuevas amenazas.

3.2.4.1 Supervivencia de una MANET

Se debe garantizar la supervivencia de los servicios esenciales de una MANET: Conectividad, encaminamiento y comunicación [12].

Conectividad

La red se debe mantener conectada, esto implica garantizar la comunicación extremo a extremo y un enrutamiento eficaz. Para lograr esto, los sistemas deben:

- Trabajar en redes heterogéneas.
- Ser auto configurables, principalmente para la asignación de direcciones a los nodos y el descubrimiento de servicios.
- Ajustar potencias de transmisión de los nodos de forma adaptativa en respuesta a la movilidad, entornos y ataques.
- Administrar el uso de energía del nodo y otros recursos de manera eficiente cuando se sospecha que el sistema está siendo atacado.

Encaminamiento

El encaminamiento es otro servicio esencial, cuyo trabajo cooperativo (involucra a muchos nodos) introduce debilidades de seguridad. Por esta razón, los sistemas de supervivencia deben:

- Controlar el acceso de los nodos a la red.
- Proteger la comunicación inalámbrica en las capas física y de enlace de datos.
- Trabajar con enfoques redundantes como ser múltiples rutas y protocolos de encaminamiento dobles, para garantizar un enrutamiento sólido y eficiente.

Comunicación

La comunicación en las MANET es un desafío, debido a los problemas de movilidad y topología dinámica que aparecen en este tipo de redes. Para garantizar la supervivencia de la comunicación se debe considerar:

- El uso de múltiples canales de comunicación. Ante la caída de un canal se puede establecer la comunicación por un canal redundante, que puede llegar a tener un ancha de banda significativamente menor.
- El diseño de protocolos que funcionan normalmente en diferentes condiciones. Por ejemplo, canales de comunicación con retardo variable.
- La comunicación extremo a extremo debe funcionar sin necesidad de un canal de retorno para acuses de recibo (Best-effort delivery).

3.2.4.2 Medidas de seguridad para la supervivencia de una MANET

Las medidas de seguridad destinadas a mejorar la supervivencia de una MANET se clasifican en tres grandes grupos: descubrimiento de ruta, reenvío de datos y gestión de claves (Figura 3-24).

- El descubrimiento de ruta, que consta de enfoques que introducen niveles mas elevados de resistencia y tolerancia, a diferentes tipos de ataques e intrusiones, en la fase de descubrimiento de ruta de los protocolos de encaminamiento.
- El reenvío de datos, que se compone de iniciativas especializadas en el reenvío de datos utilizando mecanismos de seguridad preventivos o reactivos y algunas técnicas de tolerancia (Ej: redundancia).
- La gestión de claves, que abarca soluciones criptográficas y enfoques de control de acceso diseñados para ser tolerantes a los ataques.

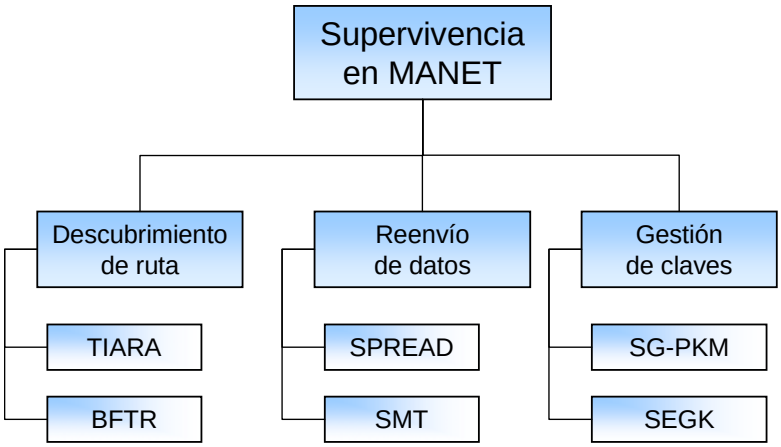

Figura 3-24: Medidas de seguridad para la supervivencia en MANET

3.2.4.2.1 Descubrimiento de ruta

En el apartado 3.2.2.1 se han descrito protocolos de encaminamiento seguros tales como SAR [55], SRP [56] y SAODV [62], estos protocolos se basan principalmente en algoritmos de autenticación y cifrado, si bien cada uno implementa características de prevención y reacción, ninguno considera mecanismos de tolerancia. Por esta razón, en la bibliografía surgen enfoques de encaminamiento tolerantes a intrusiones como TIARA [78] (Techniques for Intrusion-resistant Ad Hoc Routing Algorithms) o BFTR [79] (Best-Effort Fault Tolerant Routing).

3.2.4.2.1.1 TIARA

En [78] Ramanujan y Nguyen presentan un conjunto de técnicas diseñadas para ser aplicadas en protocolos de encaminamiento bajo demanda (como DSR y AODV). La implementación de estas técnicas posibilita un funcionamiento aceptable de la red en presencia de ataques.

Las técnicas más importantes propuestas en TIARA son:

- Control de acceso a ruta basado en flujo (FLAC, FLow-based route Access Control). Cada nodo que participa en la red ad hoc contiene una lista de control de acceso, donde se definen los flujos[3] autorizados. En base a esta lista el nodo descarta los paquetes que pertenecen a flujos no autorizados o envía los paquetes de flujos autorizados.
- Encaminamiento de múltiples rutas. Se efectúa el descubrimiento y mantenimiento de diferentes rutas de encaminamiento para un flujo específico, sólo una ruta se elige inicialmente para el reenvío de los datos.
- Monitoreo de flujo. Los fallos en la red se comprueban enviando periódicamente mensajes de control, llamados paquetes de estado de flujo. Si se identifica un fallo de ruta, se puede seleccionar un camino alternativo que se encuentra en la fase de descubrimiento.

[3] Un flujo es una secuencia de paquetes que viaja desde el nodo de origen a un nodo de destino.

- Autenticación ligera. Las etiquetas de ruta y números de secuencia se colocan en una posición secreta del paquete, cada nodo puede utilizar una posición diferente para las etiquetas dentro del paquete. Esta información se utiliza para la autenticación del nodo y proporciona una contramedida para los ataques de repetición.
- Reenvío de paquetes basado en la asignación de recursos. Se utiliza una asignación limitada de recursos con el fin de evitar que los flujos de tráfico autorizados agoten los recursos de red. Los nodos que participan en tareas de ruteo, definen un umbral para la cantidad máxima de recursos de red que pueden ser asignados a un flujo determinado.

3.2.4.2.1.2 BFTR

En [79] Xue y Nahrstedt presentan BFTR (Best-Effort Fault Tolerant Routing), en este trabajo proponen un algoritmo de encaminamiento en origen que introduce tolerancia a fallos mediante la redundancia de rutas. Su objetivo es mantener el servicio de enrutamiento con alta relación de entrega y bajo overhead en presencia de nodos con mal comportamiento. Para elegir la ruta más factible tiene en cuenta estadísticas existentes del comportamiento de los nodos que forman parte de la ruta (Ej: Porcentaje de entrega correcta de paquetes) y la retroalimentación del receptor.

BFTR utiliza inundaciones DSR para generar un conjunto de rutas entre los nodos origen y destino, seguidamente elige la ruta más corta y comienza a enviar paquetes de datos. El algoritmo considera que el comportamiento de un nodo es bueno si tiene una elevada proporción de entrega correcta de paquetes. De esta manera, una buena ruta consta de nodos con un buen patrón de comportamiento (extremo a extremo). Cualquier camino que se desvía de dicho modelo, supone un mal camino y debe ser descartado y sustituido por el siguiente camino más corto.

BFTR no requiere soporte de seguridad en los nodos intermedios, esto hace que la solución sea más genérica y eficiente. Antes de establecer un canal de comunicación, se asume un buen comportamiento de los nodos origen y destino y se crea una relación de confianza para posibilitar la autenticación mutua entre ambos nodos. Para evaluar si un paquete se entrega correctamente, el nodo de origen observa si el paquete llega correctamente al nodo destino (observación del funcionamiento extremo a extremo).

3.2.4.2.2 Reenvío de datos

Los protocolos de encaminamiento seguros, descritos en el apartado 3.2.2.1, se enfocan en mecanismos de descubrimiento y mantenimiento de de ruta y no contemplan la entrega segura y continúa (sin interrupciones) de los datos. Un atacante puede situarse en una ruta, siguiendo las reglas del descubrimiento de ruta y luego redireccionar, eliminar, inyectar o modificar tráfico. En otras palabras, un adversario puede ocultar su comportamiento malicioso por un período de tiempo y lanzar un ataque en el momento menos esperado, lo que complica su detección. Por estas razones, es necesario utilizar mecanismos que proporcionen confidencialidad, disponibilidad e integridad para los datos que son enviados por la red.

Las soluciones propuestas para proporcionar confidencialidad y disponibilidad de los datos aplican técnicas como la redundancia y protección de mensajes para ser más resistentes a

los ataques. En SPREAD [80] y SMT [81], el mensaje se divide en múltiples partes mediante un algoritmo de división de mensaje. Estas partes se envían simultáneamente desde el origen hasta el destino a través de múltiples rutas.

3.2.4.2.2.1 SPREAD

En [80] Lou, Liu y Fang presentan SPREAD (Secure Protocol for Reliable Data Delivery), Este protocolo propone el uso de algunas técnicas para mejorar la disponibilidad de los datos sin comprometer la confidencialidad. El mensaje a enviar es dividido por el nodo origen en N partes utilizando un esquema de secreto compartido de umbral (K,N), cada parte se cifra y se envía a través de un camino independiente. Para la reconstrucción del mensaje en destino solo se necesitan K de las N partes, esto introduce cierto nivel de tolerancia a fallos e incrementa la disponibilidad.

SPREAD implementa cifrado de enlace entre nodos vecinos, con una clave diferente para cada enlace. El esquema supone la existencia de un sistema de gestión de claves y se centra en tres actividades principales:

- **Dividir el mensaje en piezas.**
  Utiliza un esquema de secreto compartido de umbral (K, N) [42] para distribuir un secreto de N piezas entre un conjunto de K nodos. El secreto de N partes puede ser recuperado combinando K piezas. Por lo tanto, el enemigo tiene que comprometer al menos K nodos para recuperar el mensaje original.
- **Seleccionar múltiples rutas.**
  Selecciona varias rutas independientes, teniendo en cuenta factores de seguridad como la probabilidad de que la ruta sea comprometida.
- **Asignar las piezas de los mensajes a las rutas.**
  Utiliza un esquema de asignación para distribuir las N partes en M rutas seguras disponibles. Plantea dos objetivos para la asignación de las partes: 1. Reducir al mínimo la probabilidad de daño, evitando el uso de rutas con gran probabilidad de ser comprometidas y 2. Asignar las N partes garantizando un nivel de seguridad y proporcionando redundancia.

La confidencialidad de datos óptima se logra cuando K = N, y entre 1 y (K-1) partes se asignan a cada ruta utilizada. Sin embargo, para mejorar la disponibilidad de datos, la redundancia se debe introducir con la elección de (K <N). Esta elección de K asegura que el mensaje original se pueda reconstruir en presencia de errores en los nodos, cambios topológicos o ataques activos, siempre y cuando no más de (N – K) partes se pierdan.

Se puede demostrar que la asignación de entre (N - K + 1) y (K – 1) partes a cada ruta óptima proporciona confidencialidad de datos cuando se introduce redundancia. Por lo tanto, incluso si están en peligro un número reducido de partes, la confidencialidad del mensaje original permanece intacta. Estas restricciones obligan al atacante a poner en peligro varias rutas para dañar el mensaje original.

3.2.4.2.2.2 SMT

En [81] Papadimitratos y Hass presentan SMT (Secure Message Transmission), esta propuesta apunta a mejorar la confidencialidad, integridad y disponibilidad de datos en

entornos móviles. El esquema de SMT opera suponiendo una Asociación de Seguridad (SA) entre los nodos origen y destino (extremo a extremo), por lo que no se necesita ningún cifrado de enlace. Esta SA entre nodos finales se utiliza para proporcionar integridad de datos y autenticación de origen, pero también podría ser utilizada para facilitar el cifrado de mensajes de extremo a extremo.

Al igual que el esquema SPREAD, SMT utiliza encaminamiento de rutas múltiples (*multi-path*) para mejorar estadísticamente la confidencialidad y disponibilidad de los mensajes intercambiados entre los nodos de origen y destino. Mientras que SPREAD fue diseñado considerando principalmente la confidencialidad de la transmisión de datos, los diseñadores de SMT se centraron principalmente en la confiabilidad de la transmisión de datos. SMT proporciona un mecanismo seguro y robusto de retroalimentación extremo a extremo que permite la reconfiguración rápida de la trayectoria en caso de que un nodo falle o sea comprometido. Cada ruta tiene asociado un índice de fiabilidad basado en el número de transmisiones exitosas y no exitosas. SMT utiliza estos índices en conjunción con un algoritmo de encaminamiento de rutas múltiples para determinar y mantener un conjunto de rutas (*path-set*) seguras. Periódicamente estos parámetros se ajustan para maximizar la eficiencia y la eficacia del protocolo.

El esquema SMT propone el uso de un algoritmo de dispersión de la Información (IDA, Information Dispersal Algorithm) [82] para dividir los mensajes en varias partes. Cada parte se transmite por un camino de nodos disjuntos diferente. Un código de autenticación de mensajes (MAC) se transmite con cada parte para proporcionar integridad de datos y autenticación del origen. El factor de redundancia de la información es la relación N / M, donde M partes (de N transmitidas) son necesarios para reconstruir el mensaje original. Tenga en cuenta que, a diferencia del algoritmo de secreto compartido de umbral, no se garantiza que menos de M piezas no revelarán el mensaje original. La redundancia de datos junto con el encaminamiento de múltiples rutas asegura de que el destino puede reconstruir el mensaje original, incluso si algunas de las piezas se pierden en la red. Por lo tanto, la retransmisión de paquetes perdidos es a menudo eliminada, lo que potencialmente permite a SMT soportar tráfico de tiempo real.

En la Tabla 3-6 se resumen las principales características de los esquemas SMT y SPREAD.

Esquema	SPREAD [80]	SMT [81]
Confidencialidad	SI	SI
Integridad	NO	SI
Disponibilidad	SI	SI
Algoritmo de división de Mensajes	Secreto compartido de umbral	Dispersión de la información
Tipo de cifrado	Enlace	Extremo a extremo
Múltiples caminos	SI	SI
Criterio para la elección de rutas	Probabilidad de que la ruta sea comprometida.	Índice de confiabilidad de la ruta.
Conjunto de caminos optimizado	SI	SI

Tabla 3-6: Comparación entre SPREAD y SMT

3.2.4.2.3 Gestión de claves

Los sistemas de gestión de claves, presentados en el apartado 3.2.1, se enfocan en la distribución y auto organización de una CA (o TTP) y no contemplan mecanismos de tolerancia a fallos que aseguren la disponibilidad de la CA. Si un atacante logra comprometer uno o mas nodos que formen parte de la CA, puede interrumpir el normal funcionamiento de protocolos seguros que hagan uso de los servicios proporcionados por la CA. Por esta razón, es necesario utilizar mecanismos de tolerancia a fallos para optimizar la disponibilidad de la CA ante fallos o ataques.

En este apartado se describen las principales características de dos soluciones diseñados para proporcionar tolerancia a fallos en la gestión de claves en MANET: SEGK [71] y SG-PKM [83] .

3.2.4.2.3.1 SEGK

En [84] Wu Bing, Wu Jie y Dong presentan SEGK (Simple and Efficient Group Key Management). La idea básica de SEGK es la generación de grupos de multidifusión para la distribución de claves entre los nodos, para mejorar la eficiencia los grupos se forman utilizando una topología de árbol. Los miembros del grupo se turnan para actuar como coordinador de grupo, el nodo coordinador es el responsable de: 1. Calcular y distribuir las claves a los demás miembros a través de los enlaces de árbol activos y 2. Mantener la conexión del grupo de multidifusión. Para generar la clave de grupo, cada miembro del grupo (nodo móvil) participa con una parte de la clave, la misma se actualiza periódicamente.

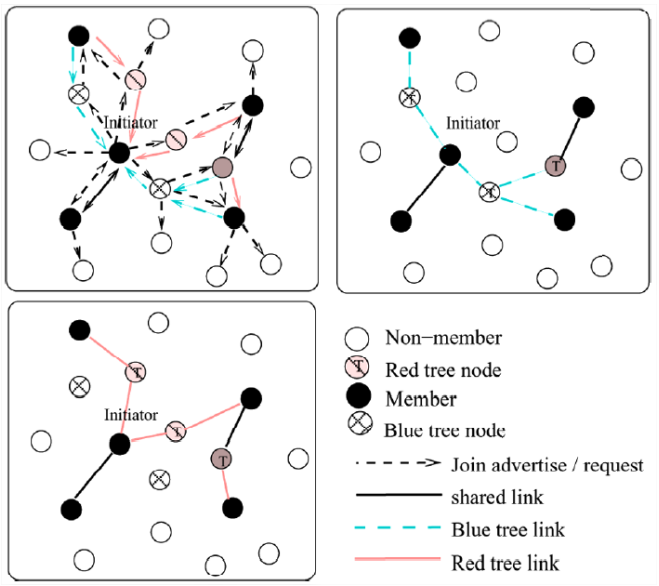

Figura 3-25: Arquitectura de doble árbol multicast en SEGK
Fuente: [84]

SEGK incorpora protocolos para la formación y mantenimiento de árboles de multidifusión. Dos árboles de multidifusión, uno azul y otro rojo, se construyen y se mantienen en paralelo para conseguir la tolerancia a fallos (Figura 3-25). Cuando un enlace de un árbol se rompe, se sustituye por el otro árbol.

En el modelo SEGK, cualquier nodo o miembro de grupo puede abandonar el grupo de manera física o lógica. La salida física se produce cuando un nodo se mueve fuera del rango de cobertura de la red o su transmisor. La salida lógica se produce cuando un nodo se mantiene dentro de la red, pero no participa en las actividades del grupo. Se presentan dos métodos para detectar la salida de un miembro:

1. Periódicamente el nodo coordinador envía mensajes de verificación a través de los enlaces de árbol, todos los miembros del grupo deben contestar con un mensaje de confirmación (ack) para confirmar su afiliación al grupo. Este método se utiliza cuando la movilidad de los nodos no es significativa
2. Periódicamente el nodo coordinador envía mensajes de control utilizando en un esquema de inundación controlada, todos los miembros activos del grupo responden al mensaje de control. Este método se utiliza para entornos de alta movilidad.

3.2.4.2.3.2 SG-PKM

En [83] Da silva, Lima y Otros presentan SG-PKM (Secure Group-Based Public Key Management), se trata de una PKI robusta, totalmente distribuida y autogestionada. Para aumentar la resistencia y la supervivencia frente a ataques, SG-PKM incorpora redundancia y tolerancia a fallos en las operaciones que efectúa la PKI.

SG-PKM se basa en formación segura de grupos a través de las relaciones sociales de sus usuarios. Los grupos, denominados iniciadores, promueven la redundancia, ayudan a la distribución de las claves públicas, y garantizan la asociación entre las claves públicas y las identidades de nodo de una manera descentralizada.

Inicialmente, cada nodo genera su par de claves (pública y privada). A continuación, los nodos deben construir grupos de iniciadores con el fin de participar en el sistema. Un grupo iniciador se compone de M nodos confiables lo que significa que todos ellos mutuamente han intercambiado sus claves públicas siguiendo relaciones de amistad (relaciones sociales) existentes entre sus propietarios. Cada grupo iniciador tiene su propio par de claves, llamada clave de grupo. Este par de claves se puede construir utilizando el esquema de umbral de shamir (k, n) [42]. La clave privada de grupo se utiliza para firmar los certificados digitales emitidos por sus miembros.

Los certificados de clave pública se utilizan para vincular una clave pública a una identidad. Por lo tanto, el SG-PKM tiene dos tipos de certificados de clave pública: certificado de nodo y certificado de grupo. Los certificados de nodos vinculan las claves públicas de los nodos con sus respectivas identidades, mientras que los certificados de grupo asocian las claves públicas de grupo con la identificación del grupo. Los certificados de nodo son firmados por el grupo en el que participa el nodo. Los certificados de grupo son firmados por otro grupo iniciador.

Los nodos pueden unirse o abandonar la red en cualquier momento. Para unirse a la red, el nuevo nodo debe crear un nuevo grupo iniciador. Tenga en cuenta que los nodos pueden ser parte de varios grupos. Por lo tanto, el nuevo nodo debe encontrar (m-1) amigos para formar un grupo, y conectar el grupo a la red. Por otro lado, la salida de un nodo de la red no compromete su funcionalidad. De hecho, SGPKM puede tolerar la salida de hasta (m-t)

nodos sin interferir en sus operaciones. Si más de (m-t) nodos dejan un grupo, el grupo se vuelve incapaz de realizar cualquiera de sus operaciones y se considera inválido. En este caso, los nodos restantes pueden formar un nuevo grupo iniciador con m nodos y volver a la red. Todos los certificados, de nodos y de grupo emitidos por el grupo siniestrado son válidos hasta su fecha de caducidad, después de expirados estos certificados no pueden ser revalidados.

Los autores destacan las siguientes características de su propuesta:

- A diferencia de los clusters, los nodos participantes se organizan en grupos iniciadores y no requieren de nodos coordinadores.
- Los nodos pueden formar parte de diferentes grupos iniciadores.
- Formación de grupos iniciadores basados en relaciones sociales entre nodos.
- Cooperación entre los nodos para probar la vinculación entre la identidad de los usuarios y sus claves públicas.
- Mejoras en la generación de certificados y en la distribución de claves.
- Utilización de mecanismos para mejorar la supervivencia de la PKI.

"Esta página se dejó en blanco intencionalmente".

Capítulo 4: Medición del rendimiento y el consumo de energía

4.1 Medición del rendimiento

Las mediciones de rendimiento (o performance) pueden ser activas o pasivas. Las mediciones activas inyectan tráfico a través de la red y observan el efecto del mismo, mientras que las mediciones pasivas observan el tráfico de red existente.

En la Tabla 4-1 se presenta un resumen de las principales métricas utilizadas para evaluar el rendimiento de una MANETs. Todas las métricas de la tabla son utilizadas para efectuar mediciones activas. Además, se incluye una columna con las herramientas (aplicaciones Android) utilizadas para efectuar la medición.

Métrica	Descripción	Herramientas
Latencia	El tiempo que tarda (retardo) en llegar un paquete enviado desde un emisor a un receptor. Puede ser de ida y vuelta (RTT) o solo ida (One way).	- Ping, Thrulay (RTT) - OWAMP (One Way)
Jitter	Es la fluctuación del retardo, se utiliza para detectar congestión en una red.	- Thrulay
Path	Trayectoria o ruta que sigue un paquete desde un emisor hasta llegar al receptor (Numero de saltos, retardo entre salto, etc.).	- Traceroute. - MTR. Herramienta hibrida que busca la trayectoria y calcula el retardo en cada salto
Bandwidth / *Throughput*	Bandwidth. Cantidad máxima de datos que un dispositivo puede enviar datos a través de un canal de comunicación en un periodo de tiempo. Throughput. Cantidad de datos que un dispositivo está en realidad enviando a través de un canal de comunicación	- Iperf (Throughput TCP) - Wget (Throughput HTTP)
Paquetes perdidos (*Loss*)	Porcentaje de paquetes enviados que no llegan a destino.	- Ping - Traceroute.

Tabla 4-1**:** Métricas de rendimiento
Fuente: [10]

A continuación se describen las principales características de las métricas utilizadas en esta tesis: Latencia y Throughput.

4.1.1 Latencia

Se denomina latencia a la suma de retardos temporales dentro de una red, los factores que influyen en la latencia de una red son:

- El retardo de propagación, es el tiempo que demora un bit que se envía desde el emisor hasta llegar al receptor. Depende de la distancia entre el emisor y el receptor y la velocidad a la que viajan los datos por el medio físico de transmisión, generalmente la velocidad de la luz.
- El retardo de transmisión, es el que tiempo tomo inyectar una cantidad determinada de bits (trama). Depende de la capacidad del canal y del tamaño de trama.

- Los retardos introducidos por los dispositivos intermedios (repetidores, amplificadores, *switchs*, routers, *gateways*, etc.).
- El tamaño de los paquetes transmitidos.
- El tamaño de los *buffers* en los equipos de conectividad intermedios. Si un paquete transmitido queda encolado, se introduce un retardo adicional.

La latencia se puede calcular utilizando el tiempo que tarda un paquete enviado desde un emisor en volver a este mismo habiendo pasado por el receptor de destino (RTT, Tiempo de ida y vuelta), o bien utilizando el tiempo que tarda un paquete enviado desde un emisor a un receptor (One Way). One Way no necesariamente es RTT/2.

4.1.2 Throughput

En transmisión de datos, el Throughput es la cantidad de datos "útiles" transferidos con éxito de un extremo a otro a través de un canal en un período de tiempo dado. Generalmente se expresa en bits por segundo (bps).

No se debe confundir Throughput con Bandwidth (Ancho de banda). El ancho de banda representa la capacidad teórica de un canal y el Throughput la capacidad real alcanzada en ese canal, en la Tabla 4-2 se señalan algunas diferencias.

Bandwidth	Throughput
Capacidad teórica del canal	Utilización que se puede lograr del canal (Ancho de banda alcanzable)
Cantidad máxima de datos que un dispositivo puede enviar datos a través de un canal de comunicación.	Cantidad de datos que un dispositivo está en realidad enviando a través de un canal de comunicación
Bandwidth $\geq$ Throughput	Throughput $\leq$ Bandwidth
También conocido como "Throughput Máximo"	También conocido como "Ancho de banda consumido"

Tabla 4-2: Diferencias entre ancho de banda y throughput

Si un canal esta formado por enlaces de diferentes tamaños (Figura 4-1):

- La capacidad del canal (extremo a extremo) esta dada por la capacidad del enlace con la menor capacidad (*Narrow Link*) a lo largo de una ruta.
- El ancho de banda disponible esta dado por la capacidad del canal (extremo a extremo) menos el ancho de banda utilizado (carga de tráfico actual).
- El enlace más ajustado (*Tight Link*) es aquel que tenga disponible el ancho de banda más pequeño a lo largo de la ruta.

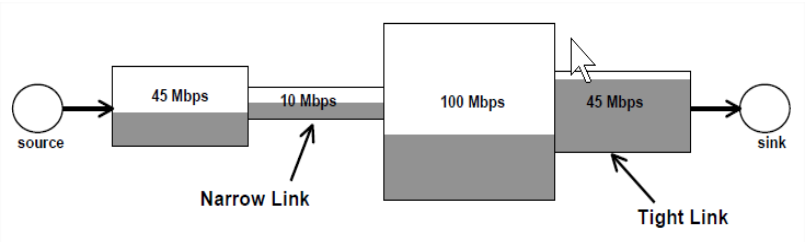

Figura 4-1: Canal extremo a extremo con enlaces de diferentes tamaños
Fuente: [10]

También se debe tener en cuenta que los protocolos de comunicaciones introducen encabezados (*overhead*), que son transmitidos junto con la carga útil de datos (*payload*), e impactan directamente en el Throughput. A continuación se presentan dos ejemplos que muestran la diferencia de Throughput cuando se utilizan encabezados de diferentes tamaños:

- Ejemplo1:
  Si se transmiten tramas de 550 bytes con un payload de 512bytes y un overhead de 38 bytes por un canal con una capacidad de 10 Mbps, el throughput o ancho de banda alcanzable será:
  Frame rate = capacidad del canal/tamaño total de la trama
  Frame rate = 10.000.000/((512+38)x8)= 2272,72
  Throughput =(Frame rate x payload)=2272,72x(512x8)= 9309061,12 = 9,3 Mbps
  Throughput = 9,3 Mbps
- Ejemplo2:
  Ídem al ejemplo 1, utilizando 60 bytes de overhead en lugar de 38:
  Frame rate = capacidad del canal/tamaño total de la trama
  Frame rate = 10.000.000/((490+60)x8)= 2272,72
  Throughput = (Frame rate x payload)=2272,72x(490x8)= 8909062,4= 8,9 Mbps
  Throughput = 8,9 Mbps

Se observa que el throughput baja de 9,3 Mbps a 8,9 Mbps cuando el overhead crece.

4.1.3 Herramientas para medir el rendimiento

En la Tabla 4-3 se presentan las aplicaciones Android utilizadas para obtener las métricas de rendimiento. Todas se distribuyen en forma gratuita a través de la tienda de aplicaciones Google (*Google Play*) [85].

Métrica	Protocolo	Aplicación Android
Latencia	ICMP	Ping [86]
	HTTP	HTTPing [87]
Throughput	TCP	Iperf [88]
	HTTP	Wget [86]
	FTP	ANDftp [89]

Tabla 4-3: Aplicaciones utilizadas para obtener las métricas de rendimiento

4.1.3.1 Ping

Ping es una herramienta diseñada para medir la latencia de ida y vuelta entre un dispositivo móvil y un servidor. La aplicación forma parte del paquete de aplicaciones BusyBox [86]. Por lo tanto, para poder utilizarla, primero se debe instalar Busybox en el dispositivo.

Figura 4-2: Ping desde un emulador de terminal para Android

Figura 4-3: Funcionamiento de HTTPing

En la Figura 4-2 se muestra el funcionamiento de la aplicación Ping de Busybox, la misma se encuentra instalada en la carpeta /system/xbin y se ejecuta desde línea de comandos utilizando el Emulador de Terminal para Android de Jackpal [90].

Entre los resultados que presenta la herramienta se destacan: La cantidad de paquetes transmitidos, el porcentaje de paquetes perdidos y tiempo de ida y vuelta o RTT (mínimo, máximo y promedio). Estos resultados pueden ser re dirigidos a un archivo de texto, lo que permite realizar un gran número de pruebas secuenciales y almacenar los resultados de cada prueba en un mismo archivo. Al finalizar las pruebas, el archivo de resultados puede ser transferido a una PC para ser procesado y analizado.

4.1.3.2 HTTPing

HTTPing [87] es una aplicación Android diseñada para medir la latencia y el throughput de servidores HTTP y HTTPS. HTTPing envía peticiones GET o HEAD a un servidor HTTP y registra las respuestas y el tiempo de ida y vuelta (RTT). Implementa una interfaz gráfica de usuario (GUI) que facilita el ingreso de los parámetros y opciones de configuración.

En la Figura 4-3 se ilustra un ejemplo del funcionamiento de HTTPing, se le pasa como parámetro la URL (http://e-cidia.unsa.edu.ar). En este caso, HTTPing enviara 4 peticiones HEAD (una por segundo) al servidor especificado, para cada petición muestra el tiempo de ida y vuelta y al finalizar muestra el tiempo de ida y vuelta mínimo, máximo y promedio.

HTTPing también permite configurar una opción para medir los tiempos de acceso a un servidor HTTPS. Además, los resultados pueden ser exportados a un archivo de texto o enviados por mail para ser procesados y analizados.

4.1.3.3 Iperf

Iperf [91] es una herramienta desarrollada por NLANR/DAST como una alternativa para medir el rendimiento extremo a extremo en una red. Permite al usuario configurar varios parámetros que pueden ser utilizados para determinar el rendimiento de una red, o alternativamente para la optimización y puesta a punto de la misma.

Iperf inyecta tráfico TCP (o UDP) en un canal que comunica a un cliente y un servidor y mide los tiempos de transferencia, la cantidad de datos transferidos y el throughput o ancho de banda alcanzable. El tráfico entre los dos extremos puede ser unidireccional o bi-direccional.

Los parámetros que se pueden utilizar en el extremo servidor son:

iperf –s –o <logfile>

-s indica que es un servidor
-o almacena los resultados en un archivo de texto: <logfile>

Los parámetros que se pueden utilizar del extremo cliente son:

iperf –c <IP server> -w <tamaño ventana> -l <tamaño buffer> -n <payload>

-c indica que es un cliente
<IP server> IP del servidor, es necesario conocer la dirección IP del servidor donde se ejecuta Iperf en modo servidor.
-w tamaño ventana TCP en kbytes.
-l tamaño del buffer de lectura escritura.
-n total de Kbytes de carga útil (*payload*) a transmitir, sin importar el tiempo.

Para mostrar el funcionamiento de esta herramienta, a continuación se presenta un ejemplo de transferencia entre un cliente Android y un servidor Windows.

Ejemplo: Transferencia TCP entre un cliente Android y un servidor Windows, con los siguientes parámetros: Payload de 1024 KBytes, tamaño de ventana TCP de 64Kbytes, tamaño de los buffers de lectura escritura de 8kbytes y dirección IP del servidor 190.221.183.220. Archivo de resultados de salida log001.txt, almacenado en el servidor.

SERVIDOR WINDOWS	**CLIENTE ANDROID**
D:\iperf>iperf -s -o log001.txt -- Server listening on TCP port 5001 TCP window size: 64.0 KByte (default) -- [4] local 192.168.1.147 port 5001 connected with 181.87.91.41 port 4147 [ID] Interval Transfer Bandwidth [4] 0.0-85.9 sec 1.00 MBytes 97,65 Kbits/sec	iperf -c 190.221.183.220 -n 1024k –l 8k –w64k -- Client connecting to 190.221.183.220, TCP port 5001 TCP window size: 64.0 KByte -- [3] local 192.168.44.27 port 56955 connected with 190.221.183.220 port 5001 [ID] Interval Transfer Bandwidth [3] 0.0-85.5 sec 1.00 MBytes 98,1 Kbits/sec

En los resultados del cliente observamos un tiempo de 85,5 segundos para transferir un payload o carga útil de datos de 1 Mbyte (1024 Kbytes), a partir de estos valores Iperf calcula el throughput de la siguiente forma:

Throughput = 1024Kbytes/85,5seg = (1024x1024x8)bits/85,5seg = 98112,37 bps
Throughput = 98112,37 /1000 = 98,1 kbps

El throughput o ancho de banda alcanzado para esta prueba es de 98,1 Kbps.

Aclaración: Iperf presenta el throughput como ancho de banda alcanzable, por esta razón en el encabezado de los resultados aparece la leyenda bandwidth en lugar de throughput.

La aplicación Iperf fue elegida para efectuar las mediciones de esta tesis, por tratarse de una herramienta de código abierto que se puede ejecutar en diferentes plataformas, incluyendo Linux, Windows y Android.

Iperf para Windows

El código fuente de Iperf fue re compilado utilizando Cygwin para producir un ejecutable Windows. Cygwin es una colección de herramientas que proporciona a Windows funcionalidades propias del entorno Linux mediante una DLL (cygwin1.dll) que actúa como si fuera un API de Linux. Cygwin No posibilita ejecutar aplicaciones nativas de Linux en Windows, para poder hacerlo el código fuente de la aplicación se debe recompilar para crear un ejecutable Windows.

Iperf para Windows se encuentra disponible en dos versiones: La versión 2.0.5 [92] con interfaz de línea de comandos y la versión 3.0 [93] que implementa una GUI avanzada.

Iperf para Android

Iperf para Android [88] es una aplicación que permite medir el rendimiento de los protocolos TCP y/o UDP entre dos dispositivos Android, uno de los dispositivos actúa como servidor y el otro como cliente. Para obtener mejores resultados y debido a que el rendimiento siempre esta limitado por la capacidad del dispositivo más lento, es recomendable realizar las pruebas entre un cliente Android y un servidor Windows o Linux. La aplicación es de libre distribución y se puede descargar desde la tienda de aplicaciones de Google [85].

La interfaz gráfica de la aplicación es básica y requiere el uso de la línea de comandos para introducir los parámetros. En la Figura 4-4 se observa el funcionamiento de un cliente iperf con los siguientes parámetros: modo cliente (-c), dirección IP del servidor iperf (170.210.201.211), tamaño de ventana TCP de 64 kbytes (-w 64k) y transferencia de 1024 kbytes de carga útil (-n 1024k).

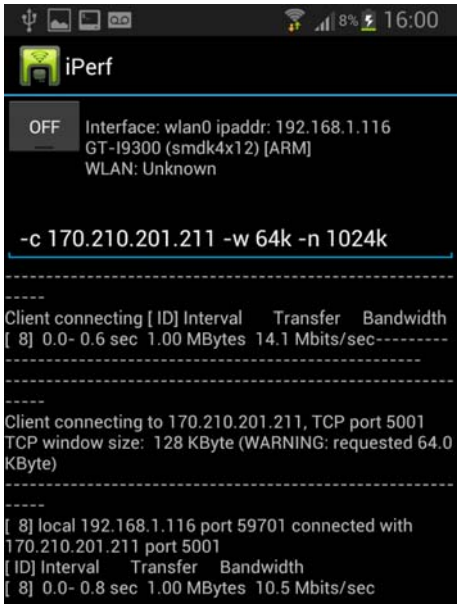

Figura 4-4: Iperf para Android

4.1.3.4 Wget

Wget es una aplicación que permite realizar descargas de archivos desde servidores HTTP y FTP, y medir el throughput de cada descarga. La herramienta forma parte del paquete de aplicaciones Busybox [86]. Por lo tanto, para poder utilizarla, primero se debe instalar Busybox en el dispositivo.

Al igual que la aplicación Ping, Wget se encuentra en la carpeta /system/xbin y se ejecuta desde línea de comandos utilizando un emulador de terminal para Android [90]. A continuación se presentan dos ejemplos de uso de la herramienta:

- Ejemplo1: Descarga de archivo de 1 Mbyte, utilizando HTTP.

  cd /system/xbin
  wget -o salida.txt http://190.221.183.220/archivo1024.txt

  2012-12-21 14:25:58-- http://190.221.183.220/archivo1024.txt
  Connecting to 190.221.183.220:80... connected.
  HTTP request sent, awaiting response... 200 OK
  Length: 1048576 (1,0M) [text/plain]
  Saving to: `archivo1024.txt'
  ……
  2012-12-21 14:28:45 (6,27 KB/s) - `archivo1024.txt' saved [1048576/1048576]

- Ejemplo2: Descarga de archivo de 1 Mbyte, utilizando HTTPs.

  cd /system/xbin
  wget -o salida.txt https://testing.unsa.edu.ar:453/archivo1024.txt

  2012-12-21 14:28:45-- https://testing.unsa.edu.ar:453/archivo1024.txt

```
Connecting to 190.221.183.220:453... connected.
Verifying testing.unsa.edu.ar certificate, issued by `/CN=TESTING': OK
HTTP request sent, awaiting response... 200 OK
Length: 1048576 (1,0M) [text/plain]
Saving to: `archivo1024.txt.3'
……
2012-12-21 14:33:19 (5,73 KB/s) - `archivo1024.txt' saved [1048576/1048576]
```

Se observa una diferencia de throughput de casi un 10% (6,27 KB/s a 5,73 KB/s), generada por el overhead que introduce el protocolo seguro SSL/TLS.

4.1.3.5 ANDftp

AndFTP [89] es un cliente FTP para dispositivos Android. Permite conectarse a servidores FTP desde dispositivos móviles, para realizar transferencia de archivos desde y hacia el servidor. Soporta los siguientes tipos de servidores:

- FTP (File Transfer Protocol).
- SFTP (SSH Secure FTP).
- FTPS (Implicit FTP over SSL/TLS).
- FTPS (Explicit FTP over SSL/TLS).

Genera un archivo de logs de transferencia, donde registra la información de las transferencias realizadas como ser Nombre archivo, Tamaño, Hora inicio y Hora finalización. En función de esta información el throughput FTP se calcula de la siguiente manera:

$$\text{Throughput FTP} = \frac{\text{Tamaño}}{\text{Hora finalización - Hora inicio}}$$

4.2 Medición del consumo de energía

4.2.1 Introducción

En este apartado se efectúa un breve repaso de las unidades de medida que son utilizadas por las aplicaciones para mostrar los resultados.

Para el Sistema Internacional de Magnitudes (ISO/IEC 80000):

- La energía se mide en "Joules" (J)
- La potencia se mide en "Watt" (W).
- El tiempo se mide en "segundos" (s).
- La carga eléctrica de una batería se mide en culombios (C) o amperios hora (Ah), donde 1 Ah = 3600 C.

El Joule es watt por unidad de tiempo. NO es lo mismo decir Joule que Watt, SI es lo mismo Joule que "Watt por segundo". Algunos ejemplos de esta relación:

1 joule [J] = 1 Watt por segundo [Ws]

1 milijoule [mJ] = 1x 10^{-3} [J] = 1x 10^{-3} Watt por segundo [Ws] = 1 miliwatt por segundo (mWs)
1 microjoule [µJ] = 1x 10^{-6} [J] = 1x 10^{-6} Watt por segundo [Ws] = 1 microwatt por segundo (µWs)

El amperio-hora representa la cantidad de electricidad que, en una hora, atraviesa un conductor por el que circula una corriente continua de 1 A (1 Ah = 3600 Culombios). Se emplea para evaluar la capacidad de una batería, es decir la cantidad de electricidad que puede almacenar durante la carga y devolver durante la descarga. Si una batería tiene, por ejemplo, una capacidad de 100 Ah, significa que puede dar una corriente de 10 A durante 10 h, o de 1 A durante 100 h, etc. [94]

4.2.2 Potencia vs Energía

Durante el desarrollo de esta tesis se consulto bibliografía para obtener información relacionada con el consumo de energía en tecnologías como Bluetooth, GPRS, HSPA y otras. Algunos autores presentan los resultados en función de la energía consumida (Joule) y otros en función de la potencia utilizada (Watt). Para una correcta interpretación de los datos es muy importante entender la diferencia entre los conceptos de Potencia y Energía.

En [95] La potencia y la energía se definen en función del trabajo que el dispositivo móvil realiza:

- Potencia = Carga de trabajo/Tiempo (Watts)
- Energía = Potencia * Tiempo (Joules) => Potencia= Energía/Tiempo (Joules/seg)

La potencia (P) utilizada por un dispositivo móvil es la energía (E) consumida por unidad de tiempo (t). P = E/t, en unidades Watt (W)=Joule (J)/s o J=Ws.

Mientras que la energía es la integral de la potencia en función del tiempo.

$$\text{Energía} = \int \text{Potencia}\, dt$$

Por ende, la energía es E=P.t en unidades J=W.s, por esta razón muchas veces el consumo de energía se mide en kWh (kilo Watt hora) o mWs (mili Watt segundo).

En algunos casos minimizar la potencia también minimiza la energía. Sin embargo, esto no siempre es así, algunas tareas requieren menos energía para completarse cuando se ejecutan a alta velocidad y elevada potencia, debido a que finalizan su ejecución en un periodo de tiempo corto. Si ejecutamos estas tareas a baja velocidad y utilizando menor potencia, finalizaran en un período de tiempo mas largo, lo que implica un mayor consumo de energía.

Por lo tanto, cuando se habla de la conservación de la energía, es necesario distinguir entre la reducción de potencia y la reducción de la energía. El hecho de que reducir la potencia no significa que la energía se reduce:

"Reducción de la potencia" ≠ "Reducción de la Energía"

En la Figura 4-5 se presentan dos sencillos ejemplos para demostrar este hecho:

- **Ejemplo1:**
  Potencia estática despreciable (*Negligible*) y voltaje constante (E1=E2)
  Si ejecutamos una tarea a una frecuencia F1 y a una potencia P1, el consumo de energía E1 será de 2 unidades.
  Si la frecuencia F2 se reduce a la mitad, la potencia P2 se reducirá a la mitad y se necesitará el doble de tiempo para ejecutar la tarea. El consumo de energía E2 será de dos unidades.

- **Ejemplo2:**
  Potencia estática significativa (*Significant*) y voltaje constante (E1<E2)
  Si ejecutamos una tarea a una frecuencia F1 y a una potencia P1 (activa + estática), el consumo de energía E1 será de 3 unidades.
  Si la frecuencia F2 se reduce a la mitad, la potencia P2 también se reducirá, pero no a la mitad. El consumo de energía E2 para completar la tarea será de 4 unidades.

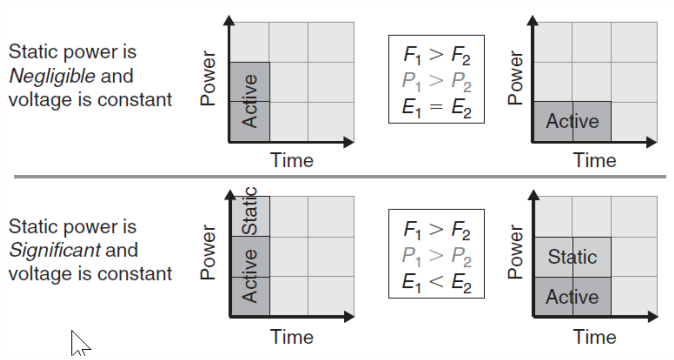

Figura 4-5: Potencia vs Energía
Fuente: Power Management in Mobile Devices [95]

4.2.3 Energía acumulada en las baterías

La unidad para medir la energía acumulada en una batería es el joule; sin embargo, para fines prácticos, y dado que el voltaje de una batería es fijo, se utiliza el Ah como unidad de carga, haciendo referencia al tiempo de carga y descarga de la batería. La equivalencia de energía máxima que se acumula en una batería viene dada por la cantidad de Ah de la batería, multiplicada por 3600 y por el voltaje.

En las baterías de dispositivos móviles es normal el uso del miliamperio hora (mAh), que representa la milésima parte del Ah, o lo que es lo mismo 3,6 C. Esto indica la máxima carga eléctrica que es capaz de almacenar la batería. A mayor carga eléctrica almacenada, mayor tiempo tardará en descargarse.

Ejemplo:

Si una batería de un teléfono móvil tiene 1390 mAh (1,390 Ah) de capacidad de carga, y proporciona un voltaje de 3,7 V, eso quiere decir que puede acumular 18514,8 joules:

1,390 Ah x 3600 x 3,7v = 18514,8 J

Con esta energía acumulada, un celular que consuma 500mJ (o 500mWs) podrá permanecer funcionando durante 10,28 horas:

500 mJ = 500 x 10^{-3} J
18514,8J/0,5J = 37029,6 seg / 3600 = 10,28 horas

Debido a que una batería almacena una cantidad determinada de energía, el objetivo principal de la gestión de la energía es reducir al mínimo la cantidad de energía necesaria para la ejecución satisfactoria de las aplicaciones en el dispositivo móvil.

4.2.4 Tiempo de descarga

El tiempo de descarga se calcula utilizando la siguiente expresión:

Tiempo de descarga = carga eléctrica de la batería/consumo eléctrico

Ejemplo:

Si una batería de 3,7 voltios puede cargar 1000 mAh, durará 50 horas en un dispositivo que consuma 200 mA por hora:

Tiempo de descarga = carga eléctrica/consumo eléctrico
Tiempo de descarga = 1000/200 = 50 horas

Y si el dispositivo móvil consume 10 mA por hora, el tiempo de vida de vida de la batería será de 10 horas.

4.2.5 Consumo eléctrico

El consumo eléctrico de un dispositivo se puede hallar aplicando la siguiente formula:

Consumo eléctrico = carga eléctrica de la batería/tiempo de descarga

Ejemplo:

Si un dispositivo móvil con una batería de 800 mAh tarda 400 horas en descargarse, el consumo del dispositivo es de 2 mA:

Consumo = carga eléctrica/tiempo de descarga = 800 / 400 = 2 mA

Y si 1 A = 1 C/s, entonces 2 mA = 0,002 A = 0,002 C/s.

4.2.6 Herramientas para medir el consumo de energía

Imaginemos el siguiente escenario: Un usuario de dispositivo móvil, utilizando el nivel de carga de la batería como herramienta de medición, intenta determinar el consumo de energía de una aplicación "A": El consumo de un equipo es de 100 mW cuando se ejecuta la aplicación "A", y es de 20 mW cuando no, se infiere entonces un consumo de 80mW para la aplicación "A". Si ejecuta la aplicación "A" y una aplicación "B" al mismo tiempo el consumo pasa a ser de 200mW, y es de 160 mW cuando solo se ejecuta la aplicación "B", se infiere entonces que la aplicación "A" consume 40 mW. La pregunta que se hace este usuario es ¿Cuál es el verdadero consumo de la aplicación "A" 40mW o 80mW?

Para evitar estas inconsistencias, se deben utilizar herramientas de software que brinden información detallada (*Fine-grained*) acerca del consumo de energía en el dispositivo móvil. En base a esta información se puede estimar correctamente el consumo de energía por aplicación y por componente de hardware.

Estas herramientas permiten a los usuarios de dispositivos móviles, desarrolladores de aplicaciones y administradores de red conocer el consumo detallado de energía de un

dispositivo, esta información puede ser utilizada con fines específicos dependiendo del tipo de usuario:

- Los usuarios finales, para determinar cual es el consumo de energía que tiene cada aplicación, esto les ayuda a tomar decisiones sobre la instalación y/o ejecución de una aplicación en su dispositivo móvil.
- Los desarrolladores, para dimensionar el impacto que tienen sus aplicaciones en el consumo de energía de un dispositivo móvil, esto conduce al uso de buenas prácticas de programación que reflejen un menor consumo de energía y un mayor tiempo de vida de las baterías (*power aware* y *network aware programming*).
- Los administradores de red, para diseñar e implementar redes móviles optimizando el consumo de energía. En la Tabla 4-4 se enumeran algunos factores que un administrador de red debe considerar.

Categoría	Factores a considerar
Diseño físico de la red	- Cantidad máxima de nodos. - Distancia entre nodos para minimizar la potencia de transmisión.
Configuración de hardware del dispositivo	- Capacidad de las baterías. - Potencia de transmisión de las interfaces. - Tipo de pantalla (LCD/OLED).
Configuración de software del dispositivo	- Sistema Operativo. - Procesos del SO a habilitar/deshabilitar. - Aplicaciones que pueden ser instaladas en los dispositivos móviles.
Tecnología WAN a utilizar	- 3G vs 2G. - HSPA vs GPRS.
Tecnología WLAN a utilizar	- Wi-Fi vs Bluetooth.
Protocolos de comunicación	- TCP vs UDP - IP over Bluetooth (BNEP) vs IP over Ethernet
Nivel de seguridad en las comunicaciones	- Canal seguro vs canal no seguro - (IP vs IPSec, HTTP vs HTTPs, FTP vs FTPs).

Tabla 4-4: Cuestiones a considerar para diseñar una red, optimizando el consumo de energía

Las siguientes características son deseables para este tipo de aplicaciones o herramientas de software:

- No utilizar dispositivos externos para las mediciones, deben utilizar sensores incorporados en el dispositivo móvil (internos) y/o llamadas a funciones del SO.
- No requerir modificaciones en el software del sistema (BIOS, Kernel del SO, etc.)
- Estimar el consumo de energía en tiempo real y de manera autónoma (sin depender de otros procesos o aplicaciones).
- Brindar información detallada y de bajo-nivel (*Fine-grained*) acerca del consumo de energía. Por ejemplo: La aplicación "A" consume 40 mW, de los cuales 10 mW corresponden a CPU, 20 mW a LCD/AMOLED, 3 mW a sistema 2G/3G, 4mW a sistema WiFi Y 3mW a sistema Bluetooth.
- Portabilidad del sistema, esto significa que no debe ser dependiente de la configuración del dispositivo móvil.

- El análisis gráfico de las mediciones de potencia y energía debe ser fácil de usar, para ello la interfaz de usuario (UI) debe estar diseñada para dispositivos con pantalla reducida y capacidades de entrada restringidas.
- Los resultados deben presentarse de dos maneras:
    1. Utilizando gráficos que posibiliten el análisis de los resultados de forma inmediata y sin necesidad de transferir archivos entre el dispositivo móvil y una PC.
    2. Guardando las mediciones de consumo en archivos de texto (logs), los cuales se pueden transferir a una PC para realizar un análisis detallado de los resultados.

Existen aplicaciones desarrolladas para distintos plataformas de Hardware y Software (Sistema Operativo), en la Tabla 4-5 se resumen las principales herramientas utilizadas para medir el consumo de energía.

Aplicación	Sistema Operativo	Inconvenientes
PowerTop [96]	Linux	Disponible para procesadores X86, no existe una versión para smartphones
Joulemeter [97]	Windows	No disponible para smartphones.
Información de batería incorporada en Android	Android	No proporciona información detallada ni de bajo nivel (fine-grained) acerca del consumo de batería.
Android Powermanager [98]	Android	Código abierto, que forma parte del API de programación de Android, requiere programación para medir el consumo de energía, es útil para medir el consumo en aplicaciones en etapa de desarrollo, no sirve para aplicaciones ejecutables.
Eprof [99]	Windows Mobile	La herramienta se encuentra en etapa de prueba y todavía no se encuentra disponible para descarga.
Powertutor [100]	Android	La estimación del consumo se efectúa en base predicciones realizadas mediante un modelo de consumo de energía (menor precisión).
Energy Profiler [101] [Nokia]	Symbian	Symbian fue discontinuado y reemplazado por Windows Mobile.
Trepn Profiler [102] [Qualcomm]	Android	Se puede utilizar sólo en un número limitado de teléfonos, aquellos que incorporan procesadores Snapdragon.

Tabla 4-5: Herramientas utilizadas para medir el consumo de energía

A continuación se describen las principales características de las herramientas utilizadas para medir el consumo de energía en dispositivos móviles. Posteriormente se justifica la elección de la herramienta utilizada para efectuar las mediciones en este trabajo de tesis.

4.2.6.1 Android Powermanager

Powermanager [98] esta formado por un grupo de funciones de código abierto, que forman parte del API de programación de Android. Estas funciones están diseñadas para ser utilizadas por programadores de aplicaciones que requieren optimizar el consumo de energía de aplicaciones en desarrollo, entre otras cosas, permiten detectar secciones de código que comprometen los recursos de energía de un dispositivo móvil.

4.2.6.2 Eprof

Eprof [103] es una herramienta que permite medir el consumo detallado de energía (fine-grained) en dispositivos móviles con SO Windows Mobile, se centra en políticas para contabilizar el consumo de energía y en cómo asignar ese consumo a las aplicaciones, para ello:

- Divide una aplicación en entidades de consumo de energía.
- Realiza el seguimiento del consumo de energía y de las actividades energéticas de cada componente de hardware.
- Asigna las actividades energéticas a las entidades responsables de las mismas.

El diseño de esta herramienta se apoya en una técnica para modelar el consumo detallado de energía (fine-grained) en tiempo real [4]. Esta técnica implementa un manejador de llamadas al sistema basado en una máquina de estados finitos[4], el cual se encarga de capturar con precisión el comportamiento de consumo de energía que tienen los componentes de hardware del dispositivo móvil (*Smartphone*).

Se tiene prevista una futura implementación de Eprof para sistemas operativos Android.

4.2.6.3 Nokia Energy Profiler

Nokia Energy Profiler (NEP) [101] es una aplicación desarrollada por Nokia para medir el consumo de energía en dispositivos móviles con Sistema Operativo Symbian S60. Ofrece detalles de consumo de energía en los siguientes ítems: CPU, red celular (2G, 3G) y tráfico IP. Proporciona APIs externas que pueden ser utilizadas por los desarrolladores para optimizar el consumo de energía en sus aplicaciones.

NEP almacena muestras o registros cada 250 milisegundos, cada muestra incluye una marca de tiempo (*timestamp*) y el consumo de energía en ese instante. Por ejemplo, si deseamos conocer cuanta energía consume una aplicación para realizar una transferencia de datos, se analizan las muestras almacenadas entre el inicio y el final de la transferencia. Los registros se pueden exportar a un archivo de texto (*log*), esto posibilita realizar un análisis detallado y determinar el impacto que tiene una aplicación en el consumo de energía del dispositivo móvil.

Figura 4-6: Mediciones de consumo de energía

[4] Se denomina máquina de estados a un modelo de comportamiento de un sistema con entradas y salidas, en donde las salidas dependen no sólo de las señales de entradas actuales sino también de las anteriores. Una máquina de estados se denomina máquina de estados finitos si el conjunto de estados de la máquina es finito[94].

En la Figura 4-6 se observan mediciones de consumo de energía realizadas utilizando NEP para una transferencia de datos utilizando 3G, se destaca la transición en el tiempo entre un estado de alto consumo y bajo consumo.

En [104] se presenta un estudio del consumo de energía en dispositivos móviles, las mediciones se realizaron utilizando el Nokia Energy Profiler, en el desarrollo de la publicación se explican detalles del funcionamiento de esta herramienta.

4.2.6.4 Trepn Profiler

Trepn profiler [102] es una herramienta que permite realizar mediciones de rendimiento y de consumo de energía de aplicaciones Android que se ejecutan en dispositivos móviles con procesadores Snapdragon [105]. Trepn Profiler utiliza una plataforma de desarrollo denominada Mobile Development Platform (MDP) [106] diseñada para facilitar el acceso a la siguiente información:

- Porcentaje y frecuencia de uso de la CPU
- Consumo de energía
- Estadísticas de consumo de memoria física y virtual
- Utilización de la red celular y Wi-Fi

El consumo de energía se mide a través de sensores (*sensor-based*) que incorporan los procesadores SnapDragon, esto permite una mayor precisión en las mediciones. Los sensores internos miden el consumo de energía en cada núcleo del procesador y a partir de estos valores obtienen el consumo total de energía (Figura 4-7).

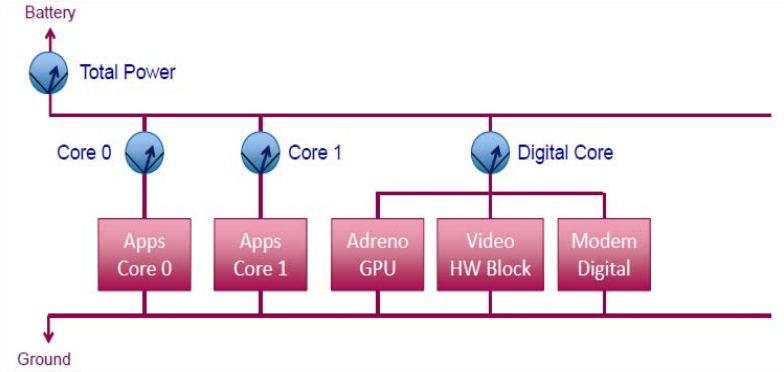

Figura 4-7: Sensores incorporados en los procesadores Snapdragon
Fuente: [103]

En [107] se explica paso a paso el funcionamiento de el funcionamiento de esta herramienta, se presentan ejemplos de aplicaciones de prueba con alta y baja utilización de recursos, en cada ejemplo se determina el consumo de energía de la aplicación, se interpretan los resultados (Figura 4-8) y finalmente se presentan los cambios necesarios para optimizar el código de la aplicación para reducir el consumo de energía.

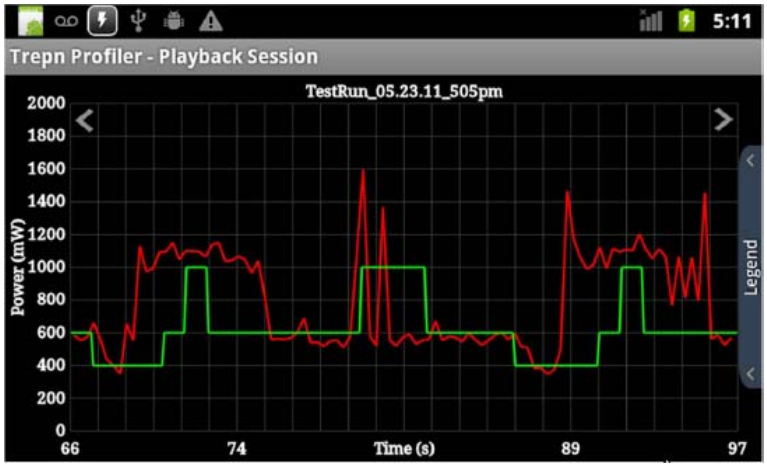

Figura 4-8: Trepn Profiler. Consumo de energía en mW de una aplicación en función del tiempo.
Fuente: [107]

4.2.6.5 Powertutor

Powertutor [100] es una aplicación Android de libre distribución. Informa a desarrolladores y/o usuarios de sistemas Android el consumo de energía detallado en tiempo real y por aplicación de los siguientes componentes: CPU, LCD, WiFi, *Bluetooth*, red celular (2G y 3G).

La Figura 4-9 muestra las estadísticas de consumo de energía reportadas por Powertutor, se observa el consumo en Joules de las aplicaciones que se encuentran en ejecución. Por ejemplo: *iperf* (270 mJ) y *Bluetooth Subsystem* (28mJ).

Figura 4-9: Powertutor. Estadísticas de consumo de energía "por aplicación"

El modelo de energía utilizado por Powertutor para estimar el consumo de energía de las aplicaciones y de los diferentes componentes de hardware del dispositivo móvil, se

denomina "*PowerBooter*" y se describe con detalle en [108]. Este modelo se divide en dos partes: Construcción del modelo de energía y Asignación energía a las aplicaciones.

4.2.6.5.1 Construcción del modelo de energía

Cada componente de hardware en un dispositivo móvil (*smartphone*) tiene estados de energía que influyen en el consumo de energía del teléfono. Por ejemplo: la utilización de la CPU, el nivel de brillo de la pantalla, la potencia de transmisión, entre otros. El modelo de energía en PowerTutor se construye mediante la correlación del consumo de energía medido y los estados de energía de hardware [108]. En la Tabla 4-6 se detallan los estados de energía considerados en este modelo.

Hardware	Estados de energía
CPU	Utilización de la CPU y el nivel de frecuencia.
OLED / LCD	Para el hardware con pantalla LCD se tiene en cuenta el nivel de brillo, para hardware con pantalla y/o la pantalla OLED se considera el nivel de brillo y la información de píxeles de la pantalla.
Wi-Fi	Tasa de datos o velocidad de datos de enlace ascendente y paquetes transmitidos por segundo.
2G/3G	Paquetes transmitidos por segundo y potencia de transmisión.
GPS	Número de satélites detectados y estados de potencia del dispositivo GPS (activo, dormido, apagado).

Tabla 4-6: Estados de energía que influyen en el consumo de energía de un dispositivo móvil

4.2.6.5.2 Asignación de consumo (potencia/energía) a las aplicaciones

El uso de energía se asigna a una aplicación como si esta fuera la única en ejecución. La razón de esto es que a veces, cuando dos aplicaciones se están ejecutando al mismo tiempo pueden causar que algunos de los componentes de hardware tengan una transición de estado, que no tendría lugar si solo una de las aplicaciones se estaría ejecutando. En este tipo de casos no está claro cómo asignar la utilización de energía a cada aplicación.

Por ejemplo, consideremos el caso en el que dos aplicaciones "A" y "B" requieren transmitir datos por la interfaz WiFi. Si la aplicación "A" se ejecuta sola, el dispositivo inalámbrico transmite en un estado de potencia baja (low power) y utiliza una cantidad baja de mW. Sin embargo, si "A" y "B" se ejecutan juntas el dispositivo inalámbrico pasa a un estado transmisión de potencia alta (high power) y utiliza una cantidad alta de mW, se plantea el problema de cómo dividir la potencia utilizada para transmitir entre las aplicaciones "A" y "B" de una manera razonable.

Para resolver este problema Powertutor predice y simula estados de hardware para cada aplicación, como si esta se ejecutará sola, con los estados simulados es posible calcular la cantidad de energía que cada aplicación utilizaría. En el ejemplo anterior a cada aplicación se le asignara un consumo por usar el dispositivo móvil en el estado de bajo consumo, salvo que la aplicación hubiera sido la causante de la transición al estado de transmisión de potencia alta.

4.2.6.6 Elección de la herramienta para medir el consumo de energía

Para las mediciones de este trabajo de tesis se utilizó la aplicación PowerTutor, la elección de esta herramienta se fundamenta en los siguientes motivos:

- La aplicación esta desarrollada para Sistema Operativo Android, según datos de Our Mobile Planet de Google [109] Android es el SO más utilizado en dispositivos móviles en la Argentina (33%).
- La herramienta es de libre distribución y se encuentra publicada en la tienda de aplicaciones Google (*Google Play)* [85].
- Powertutor esta basado en *Powermeter*, un modelo de consumo de energía detallado (*fine grained*) que permite estimar el consumo en tiempo real, por aplicación y por componente de hardware.
- Proporciona una salida de texto basada en archivos que contiene los resultados detallados (*logs*). Esto permite ejecutar una secuencia de pruebas en el dispositivo mientras Powertutor se ejecuta en background, una vez finalizadas las pruebas se analizan los resultados almacenados en los archivos de salida.

Capítulo 5: Caso de estudio

En este capítulo se presenta la metodología de trabajo utilizada en el trabajo experimental de esta tesis. En primer lugar se brinda una explicación acerca de la implementación del escenario de pruebas, abarcando el despliegue de la MANET y su integración a la red de infraestructura. A continuación, se describe el funcionamiento de las aplicaciones y protocolos utilizados para establecer canales de comunicación "no seguros" y "seguros" entre el nodo de pruebas de la MANET y el servidor de infraestructura. Finalmente, se detallan las pruebas y mediciones realizadas sobre cada configuración de canal.

5.1 Metodología de trabajo

Para cumplir con los objetivos propuestos, se realizaron las siguientes actividades:

- Construcción del escenario de pruebas.
- Selección de métricas para las mediciones.
- Selección e instalación de aplicaciones para efectuar las mediciones en el nodo cliente.
- Establecimiento de canal extremo a extremo "no seguro", entre el nodo cliente y el servidor de infraestructura.
- Medición del rendimiento y consumo de energía para el canal "no seguro", inyectando tráfico aleatorio entre el cliente y el servidor.
- Establecimiento de canal extremo a extremo "seguro", utilizando diferentes configuraciones de protocolos de seguridad. (IPSec [110], SSL/TLS [111]).
- Medición del rendimiento y consumo de energía para las distintas configuraciones de canal "seguro", inyectando tráfico aleatorio entre el cliente y el servidor.
- Evaluación y comparación de los resultados para determinar las diferencias de consumo de recursos entre las diferentes configuraciones de canal implementadas.

5.2 Construcción del escenario de pruebas

En la Figura 5-1 se observa la representación gráfica del escenario implementado para realizar las pruebas y mediciones. En el mismo se conecta una MANET, desplegada en zona remota, a la Intranet del campus universitario de la UNSa, a través de la red celular (GSM/GPRS [112]). Los dispositivos móviles (nodos) de la MANET se conectan al servidor "testing" utilizando un canal de comunicación TCP/IP extremo a extremo (*end to end*). El tráfico entre el nodo móvil y el servidor se gestiona a través de uno de los nodos que actúa como Gateway entre la MANET y la red celular. Este nodo es el encargado de enviar los paquetes de datos hacia los routers de la red celular; desde donde y a través de Internet son encaminados a la intranet para ser entregados al servidor.

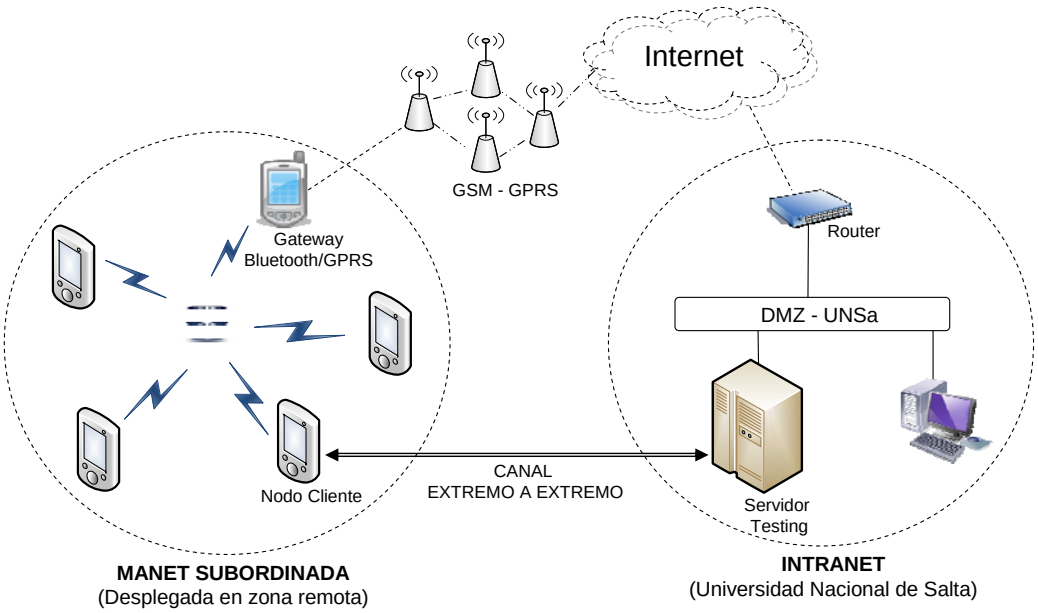

Figura 5-1: Escenario de pruebas

El despliegue y la integración de la MANET se realizaron considerando los siguientes inconvenientes y limitaciones:

- La energía en la zona de despliegue es escasa, lo que dificulta la capacidad de recarga de los dispositivos que forman parte de la MANET.
- Las redes celulares en zonas remotas no brindan servicios de tercera (3G) o cuarta generación (4G), solo se dispone de tecnología 2G (GSM/GPRS) que proporciona un ancho de banda limitado y variable.
- La mayor parte de los dispositivos móviles utilizados en zonas remotas son equipos económicos y de características básicas, que incorporan tecnologías como Bluetooth y 2G en lugar de Wi-Fi y 3G.
- Las MANETs y las redes celulares utilizan un medio compartido (aire) para transmitir los datos y se encuentran expuestas a "ataques" o accesos no autorizados. Se requiere entonces la implementación de canales de comunicación "seguros" entre los nodos de la red ad hoc y los equipos de la red de infraestructura.
- La implementación de niveles de seguridad elevados implica un incremento del consumo del ancho de banda y de la energía en los nodos móviles [33]. Ambos recursos son limitados en zonas remotas, por lo que se hace necesario elegir un nivel de seguridad que no comprometa los recursos disponibles para el normal funcionamiento de la MANET.

5.2.1 Despliegue de la MANET

Existen 4 estándares que permiten realizar comunicaciones inalámbricas de corto alcance que se pueden utilizar para la formación de redes móviles ad hoc: Bluetooth (IEEE 802.15.1), Ultra-WideBand (UWB, IEEE 802.15.3), ZigBee (IEEE 802.15.4) y WiFi (IEEE 802.11).

Para desplegar la MANET del escenario de prueba, en zona remota de recursos limitados, se eligió la tecnología Bluetooth por las siguientes razones:

- Utiliza un radio de corto alcance que ha sido optimizado para el ahorro de energía y operación adecuada de la batería [113].
- El consumo de energía de Bluetooth es inferior al de UWB y WiFi. En [114] se presenta un estudio comparativo entre diferentes tecnologías inalámbricas de corto alcance, entre los resultados de este estudio se observa que el consumo de energía de UWB y WiFi es hasta 4 veces superior al consumo de Bluetooth.
- Su bajo precio y reducido tamaño [1], posibilitan que la mayor parte de los dispositivos móviles que se consiguen en el mercado tengan incorporada la interfaz Bluetooth (en lugar de WiFi o UWB).
- Facilidad y rapidez de despliegue. Bluetooth no utiliza componentes de infraestructura como es el caso de las redes Wi-Fi, que requieren la instalación y configuración de componentes adicionales (Puntos de acceso inalámbricos) [1].
- El perfil PAN (Personal Area Networking) de Bluetooth, proporciona el transporte de datagramas IPv4 mediante el protocolo BNEP (Bluetooth Network Encapsulation Protocol) [115].

En nuestro escenario, la conexión del dispositivo móvil cliente al punto de acceso a la red (NAP - Network Access Point) se realizó utilizando el perfil PAN (Personal Área Network) del estándar Bluetooth. El punto de acceso a la red se configuro sobre el nodo Gateway utilizando la funcionalidad "Bluetooth Tethering" de Android, que utiliza el Framework netfilter e iptables [116] para implementar un puente entre la PAN bluetooth y la red GSM/GPRS.

5.2.2 Integración de la MANET a la red de infraestructura

Existen diferentes tecnologías de telefonía celular que brindan soporte para la integración de MANETs, desplegadas en zonas remotas, a redes de infraestructura. Estas tecnologías fueron evolucionando en el tiempo y esas evoluciones son las denominadas generaciones: Segunda generación o 2G (GSM, GPRS, y EDGE), Tercera generación o 3G (UMTS, HSDPA, HSUPA y HSPA+) y Cuarta generación o 4G (LTE y LTE Advanced). En el Apéndice 2 se describen características y detalles técnicos de cada tecnología.

Para integrar la MANET, desplegada en zona remota, se eligió la tecnología GPRS por sobre tecnologías como 3G o 4G, fundamentando esta elección en las siguientes razones:

- Cobertura en la zona de despliegue de la MANET.
  En zonas remotas, por lo general, solo se dispone de tecnología 2G (GSM/GPRS). Las tecnologías 3G (UMTS, HSPA) y 4G (LTE) se encuentran en zonas con gran concentración de usuarios y su implementación en zonas alejadas implica un importante recambio tecnológico por parte de las compañías de celulares.
- El consumo de energía es menor en los dispositivos móviles que utilizan GPRS, en comparación con los que utilizan UMTS o HSPA.
  En [117] y [118] se presentan estudios relacionados con el consumo de energía en diferentes tecnologías celulares, los resultados muestran que GPRS consume entre un

40% y 70% menos energía comparado con UMTS. Esta diferencia de consumo se debe a dos razones:

1. El numero de estaciones base compatibles con los estándares UMTS/HDPSA es limitado en zonas alejadas, por este motivo los dispositivos móviles 3G deben conectarse a antenas situadas a grandes distancias, lo que implica utilizar mayor potencia para transmitir los datos.
2. Las velocidades de transferencia alcanzables por los estándares 3G y 4G requieren de modulaciones más complejas, las cuales necesitan de muchos cálculos adicionales y obligan a un mayor uso de CPU a los dispositivos y, por lo tanto, a un mayor consumo de energía.

- Velocidad de transferencia.
  En la Tabla 5-1 se muestran las velocidades de transferencia de las diferentes tecnologías, desde GPRS hasta LTE-Advanced. Note que a partir de la tecnología WCDMA se utilizan diferentes velocidades para el enlace descendente (DL - DownLink) y para el ascendente (UL – UpLink). Si bien la velocidad máxima de transferencia que soporta la red GPRS (171,2 Kbit/s) es pequeña en comparación a UMTS, alcanza para establecer una conexión segura con la red de infraestructura.

Generación	**2G**		**3G**				**4G**	
Tecnología	GPRS	EDGE	WCDMA (UMTS)	HSDPA	HSUPA	HSPA+	LTE	LTE-Advanced
Tasa de datos máx. teórica (**DL - Downlink**)	171,2 Kbps	473,6 Kbps	2,0 Mbps	7,2 Mbps	14,4 Mbps	21/42 Mbps	100 Mbps	1,0 Gbps
Tasa de datos máx. teórica (**UL - Uplink**)	171,2 Kbps	473,6 Kbps	474 Kbps	384 Kbps	5,76 Mbps	7,2/11,5 Mbps	50 Mbps	0,5 Gbps

Tabla 5-1: Velocidad de transferencia para las tecnologías de celular 2G, 3G y 4G.

- Proporciona transporte de datagramas IPv4 mediante el protocolo GTP (GPRS Tunneling Protocol) [119].
- La mayor parte de los dispositivos móviles, utilizados en zonas remotas, solo soportan GSM/GPRS.
  Esto se debe a que el costo de un dispositivo 2G/3G es muy superior al de un dispositivo 2G, y su adquisición no se justifica en zonas alejadas donde solo se dispone de redes 2G y la tecnología 3G es limitada o directamente no existe.

5.2.3 Comunicación entre el cliente y el servidor

El canal de comunicación provee comunicación TCP/IP, extremo a extremo, entre el nodo cliente y el servidor "Testing". En el trayecto los datagramas IP son encapsulados en BNEP [115] por la red Bluetooth y GTP [119] por la red GPRS.

La Figura 5-2 ilustra el envío de un datagrama IP desde el nodo móvil hasta el servidor de la intranet siguiendo los siguientes pasos:

1. El nodo móvil envía el datagrama IP, encapsulado en BNEP, al punto de acceso a la red (NAP).
2. El NAP transmite el datagrama al SGSN de la red GPRS, desde donde viaja al GGSN encapsulado en GTP.

3. El GGSN re-envía el datagrama a Internet, por donde viaja hasta llegar al router frontera de la red destino.
4. El router frontera de la red destino encamina el datagrama hacia el servidor, encapsulado en una trama Ethernet.

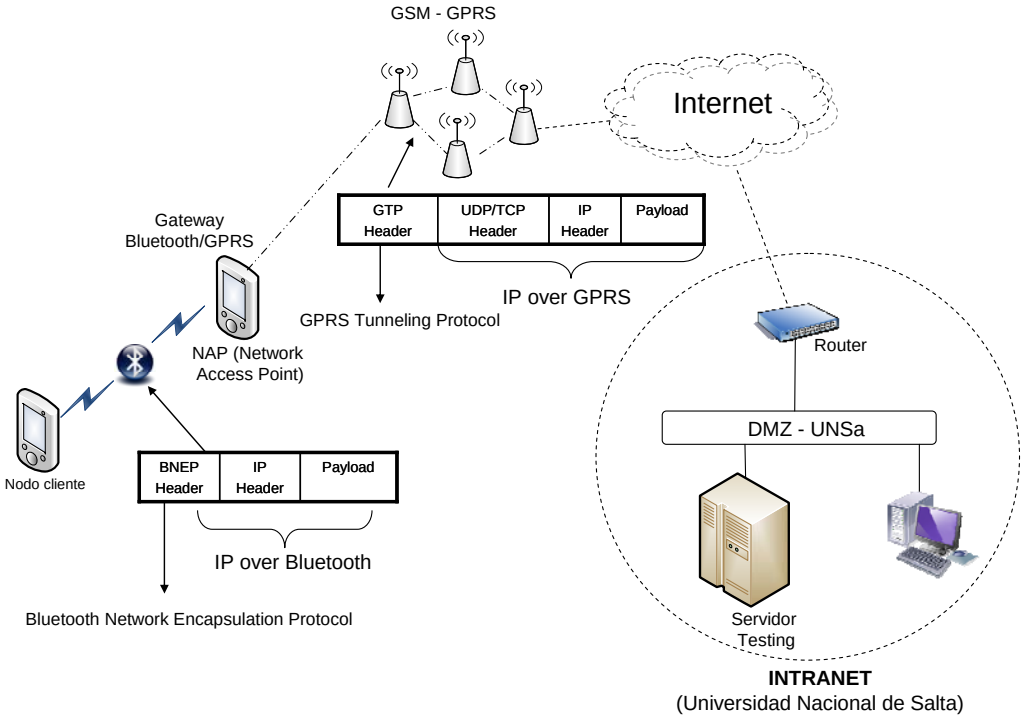

Figura 5-2: Comunicación entre el cliente y el servidor

5.2.4 Configuración de los componentes del escenario de pruebas

5.2.4.1 Configuración del servidor

El servidor de pruebas ("Testing") se instalo en la DMZ del campus de la Universidad Nacional de Salta (UNSa) y se le asigno un IP público (190.221.183.220). La DMZ de la UNSa se encuentra conectada a Internet a través de un router y un enlace sincrónico de 20Mbps.

La configuración del servidor de infraestructura utilizado en el escenario de pruebas, es la siguiente:

Servidor Testing	
Equipo	ACER ASPIRE AX3950
CPU	Intel core I3 540
Chipset	Intel H57
RAM	4 Gbytes DDR3-SDRAM
Disco rígido	1000 GBytes, Serial ATA, 10000 RPM
LAN	Gigabit Ethernet
WAN	Enlace sincrónico de 20Mbps
IP público	190.221.183.220

DNS Name	srvtesting.unsa.edu.ar
SO	Windows Server 2008 Service Pack 2 Standard Edition
Servicios Habilitados	- Internet Information Server (IIS 7.0). HTTP y HTTPs Server. - Routing and Remote Access Server (RRAS). VPN Server L2TP/IPSEC [120]
Software Instalado	- OpenSSL 1.0.1e [121] - OpenVPN Server 2.3.2 [122] (VPN Server SSL/TLS) - Filezilla Server 0.9 [123] (FTP y FTPs Server) - Iperf 3.0 [93] (Iperf Server)

5.2.4.2 Configuración del nodo Gateway

La configuración del dispositivo móvil utilizado como Gateway en el escenario de pruebas, es la siguiente:

Nodo Gateway	
Equipo	Samsung I9100 Galaxy S II
CPU	Dual-core 1.2 GHz Cortex-A9
Chipset	Exynos
RAM	1024MB RAM
SO	Android OS, v4.0.4 (Ice Cream Sandwich)
Root	SI
2G	GSM 850 / 900 / 1800 / 1900 MHz GPRS Up to 114 kbps; EDGE Up to 560 kbps
3G	HSDPA 850 / 900 / 1900 / 2100 MHz HSDPA, up to 21 Mbps; HSUPA, up to 5.76 Mbps
Bluetooth	v2.0
Batería	Lítio-íon, 2100 mAh, 3.7 v.
Aplicaciones Instaladas	- BusyBox (en /system/xbin) [86] - Jackpal Android terminal emulator [90] - Network Signal Info [124]

El equipo fue especialmente preparado para las pruebas, se procedió entonces a:

1. Desinstalar las aplicaciones no indispensables para su funcionamiento.
2. Deshabilitar dispositivos de hardware no utilizados en las pruebas (Ej: Wi-Fi).
3. Habilitar el modo de bajo consumo.
4. Habilitar el modo GSM [5]
   - Settings – More settings – Mobile networks – Network Mode -> GSM only
5. Habilitar el acceso a datos móviles
   - Settings – More settings – Mobile networks –Mobile data -> ON

[5] La integración de la MANET remota utiliza GSM/GPRS en lugar de UMTS/WCDMA.

6. Habilitar el anclaje Bluetooth (bluetooth *tether*)[6].
   - Settings – More settings – Tethering and portable hotspot – Bluetooth tethering -> ON

7. Relevar la dirección IP de la interfaz Bluetooth
   - Abrir el emulador de terminal y ejecutar:
     # su
     # cd /system/xbin
     # ifconfig -a

   De la salida que devuelve ifconfig, se debe obtener la dirección IP de la primera interfaz Bluetooth (bnep0), en nuestro dispositivo la dirección de la interfaz bnep0 es: **192.168.44.1**. Esta dirección se utiliza para la configuración del nodo cliente.

8. Relevar la dirección MAC de la interfaz Bluetooth
   - Settings – About device – Status – Bluetooth Address

   En nuestro dispositivo, la MAC es: **58:C3:8B:51:B8:06.** Esta dirección se utiliza para la configuración del nodo cliente.

9. Determinar la cantidad de routers desde el Gateway hasta el router DMZ
   - Abrir el emulador de terminal y ejecutar:
     # su
     # cd /system/xbin
     # traceroute 190.221.183.1 (IP del router DMZ)

   La salida de traceroute muestra la cantidad de saltos desde el Gateway hasta el Router DMZ. En nuestro escenario la cantidad de saltos es igual a 11. Este valor es utilizado para presentar la configuración de la MANET.

10. Relevar la información de la red GSM/GPRS
    - Utilizar la aplicación Network Signal Info para obtener la siguiente información: Distancia entre el Gateway y la estación base GSM, potencia de la señal y ubicación geográfica.

5.2.4.3 Configuración del nodo Cliente

La configuración del dispositivo móvil utilizado como cliente en el escenario de pruebas, es la siguiente:

Nodo Cliente	
Equipo	Samsung I9300 Galaxy S III
CPU	Quad-core 1.4 GHz Cortex-A9
Chipset	Exynos 4412 Quad
RAM	1024MB RAM
SO	Android OS ver. 4.1.2 (Jelly Bean)
Root	SI
2G	GSM 850 / 900 / 1800 / 1900 MHz GPRS Up to 114 kbps; EDGE Up to 560 kbps
3G	HSDPA 850 / 900 / 1900 / 2100 MHz HSDPA, up to 21 Mbps; HSUPA, up to 5.76 Mbps
Bluetooth	v3.0
Batería	Lítio-íon, 2100 mAh, 3.7 v.
Aplicaciones	- BusyBox (en /system/xbin) [86]

[6] Esta funcionalidad incorporada en el kernel de Android 4.0 permite al SO encaminar los paquetes BNEP a la red GPRS, sin necesidad de instalar aplicaciones adicionales.

Instaladas	- Jackpal Android terminal emulator [90] - Iperf [88] - HTTPing [87] - AndFTP [89] - OpenVPN installer [125] - OpenVPN settings [126] - Powertutor [100]

El equipo fue especialmente preparado para las pruebas, se procedió entonces a:

1. Desinstalar las aplicaciones no indispensables para su funcionamiento.
2. Deshabilitar dispositivos de hardware no utilizados en las pruebas (Ej: Wi-Fi).
3. Habilitar el modo de bajo consumo.
4. Deshabilitar el acceso a datos móviles.
   - Settings – More settings – Mobile networks –Mobile data -> OFF
5. Vincular el dispositivo cliente al gateway.
   - Settings – Bluetooth -> ON
   - Scan
   - Seleccionar el nodo Gateway ("Gateway Bluetooth-gprs")
   - Confirmar la clave secreta compartida (passkey) sugerida para la vinculación, la confirmación se debe efectuar en ambos dispositivos (cliente, gateway).
   - Una vez confirmada la vinculación, verificar que el nodo Gateway aparezca en el listado de dispositivos vinculados (Paired). En la Figura 5-3 se observa la vinculación del dispositivo cliente llamado "Samsung s3" al dispositivo Gateway llamado "Gateway bluetooth-gprs".

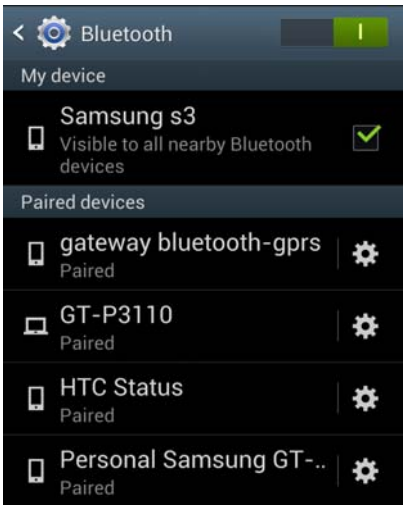

Figura 5-3: Vinculación Bluetooth entre el Cliente y el Gateway

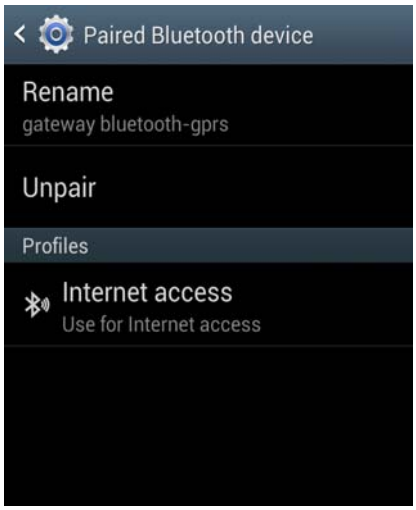

Figura 5-4: Conexión IP del nodo cliente al Gateway

6. Efectuar la conexión IP del dispositivo cliente al Gateway.

   - Seleccionar el icono de herramientas que se encuentra a la derecha del dispositivo Gateway y luego en perfiles la opción "*Internet access*" (Figura 5-4).

   Esta funcionalidad fue incorporada a partir de la versión 4.1 de Android, en versiones anteriores la configuración IP del cliente Bluetooth se realiza en forma manual, siguiendo los siguientes pasos:

- Settings – More settings – Tethering and portable hotspot – Bluetooth tethering -> ON
- Configurar el reverse tethering [7] para Bluetooth, ejecutando los siguientes comandos desde un emulador de terminal:

#su
(Pasa a modo supervisor)
#pand --connect 58:C3:8B:51:B8:06
(Crea una conexión PAN sobre la MAC bluetooth del nodo Gateway, que se obtuvo en la configuración del nodo gateway).
#ifconfig bnep0 192.168.44.47 netmask 255.255.255.0 broadcast 192.168.44.255
(Asigna un número de IP a la interfaz bnep0, el número debe pertenecer al rango de direcciones de la interfaz bnep0 del Gateway 192.168.44.0)
#route add default gw 192.168.44.1 dev bnep0
(Asocia el default router al número de IP de la Interfaz bnep0 del Gateway 192.168.44.1, que se obtuvo en la configuración del nodo gateway)
#setprop net.dns1 192.168.44.1
(Asocia el Name Server primario al número de IP de la Interfaz bnep0 del Gateway 192.168.44.1)

5.2.4.4 Configuración de la MANET

La MANET remota fue desplegada en el centro de estimulación y desarrollo escolar de la localidad de Yacones, perteneciente al municipio de Vaqueros de la Provincia de Salta. Esta localidad dispone de cobertura de red celular 2G (GSM/GPRS) muy limitada.

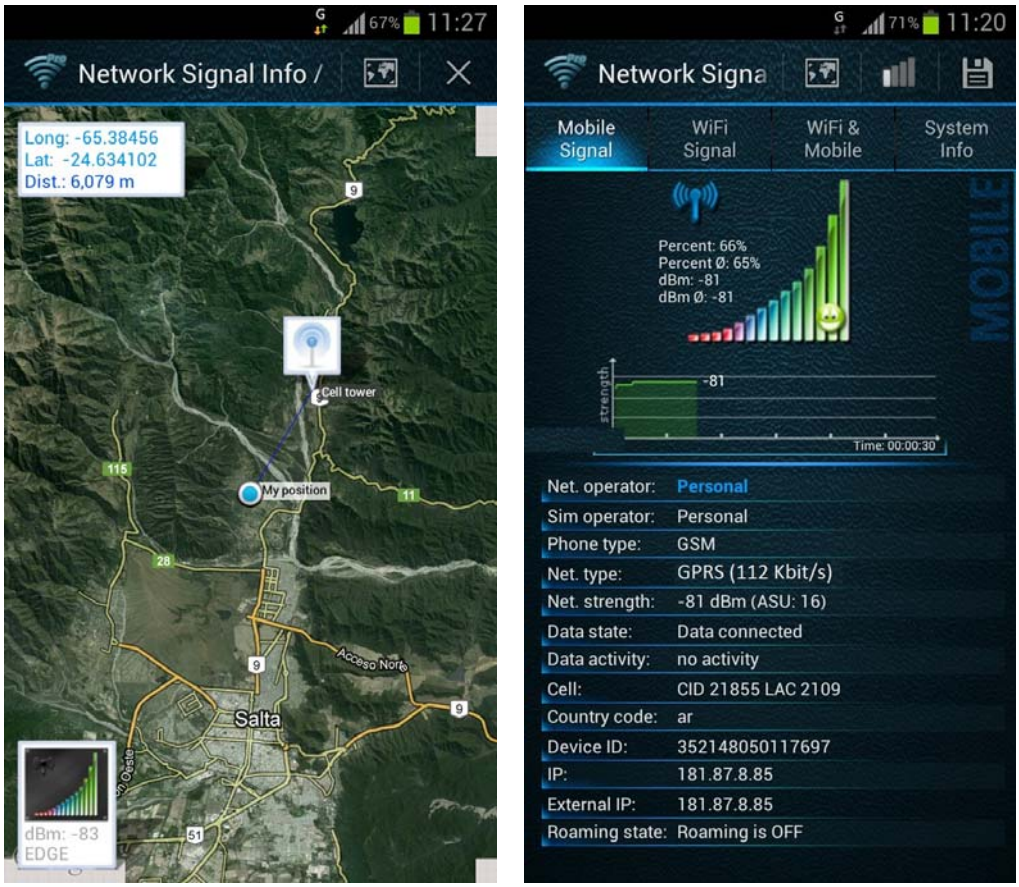

Figura 5-5: Gráficos generados por la aplicación Network Signal Info

[7] Esta configuración posibilita al nodo actuar como cliente del nodo Gateway.

En la Figura 5-5 se muestran la información de red generada por la aplicación Network Signal Info [124], ejecutada en el nodo Gateway. Además de la ubicación geográfica, se observa que la estación base GSM se encuentra a 6Km de distancia y la potencia de la señal es de -81 dBm[8].

En la Tabla 5-2 se resumen las características más importantes de la MANET remota.

MANET remota	
Tecnología elegida para el despliegue de MANET	Bluetooth 2.0
Interfaces Bluetooth utilizadas	Clase 1 (alcance teórico de hasta 100 mts y potencia máxima de 100mW)
Distancia entre el nodo Cliente y el Gateway	10 mts
Configuración de la piconet Bluetooth	Perfil PAN, protocolo BNEP
Cantidad de saltos desde el Gateway hasta el router DMZ.	11
Tecnología elegida para la integración de MANET	GPRS
Distancia entre el Gateway y la estación base GSM.	6 km
Potencia de señal	-81 dBm

Tabla 5-2: Configuración de la MANET remota

5.3 Métricas y aplicaciones utilizadas para efectuar las mediciones

Para medir el rendimiento del canal de comunicaciones extremo a extremo, entre el nodo cliente y el servidor, se eligieron las siguientes métricas: Latencia, Throughput y Consumo de energía (en el cliente). En la Tabla 5-3 se nombran las aplicaciones, del lado del cliente y del lado del servidor, que fueron utilizadas para obtener cada métrica.

Métrica	Aplicación Cliente	Aplicación Servidor
Latencia ICMP	Busybox Ping [86]	Windows Stack TCP/IP
Latencia http	HTTPing [87]	HTTP Server (IIS6)
Throughput TCP	Iperf [88]	Iperf Server
Throughput http	Busybox Wget [86]	HTTP Server (IIS6)
Throughput HTTPs	Busybox Wget [86]	HTTPs Server (IIS6)
Throughput FTP	ANDftp [89]	FTP Server (Filezilla)
Throughput FTPs	ANDftp [89]	FTPs Server (Filezilla)
Consumo de energía	Powertutor [100]	----------

Tabla 5-3: Aplicaciones utilizadas para obtener las métricas

En el capitulo anterior se realizo un descripción detallada de estas métricas y aplicaciones.

5.4 Establecimiento de canales de comunicación extremo a extremo.

Se implementaron las siguientes configuraciones para el canal de comunicación extremo a extremo, entre el cliente (en la MANET) y el servidor (en la intranet):

- Canal NO seguro.
- Canal seguro L2TP/IPsec [120].

[8] RSSI es una escala de referencia (en relación a 1 mW) para medir el nivel de potencia de las señales recibidas por un dispositivo móvil. La escala tiene al valor 0 (cero) como centro; representa 0 RSSI, o 0 dBm, generalmente la escala se expresa dentro de valores negativos; cuanto más negativo, mayor pérdida de señal. 0 dBm equivale a 1 mW de potencia, -10 dBm a 0.1 mW, -20 dBm a 0.01 mW, y sucesivamente. La señal mínima aceptable para establecer una conexión es de -85 dBm.

- Canal seguro OpenVPN (SSL/TLS) [127].
- Canal seguro OpenVPN (SSL/TLS) con compresión LZO [128].
- Canal seguro utilizando protocolos HTTPS [129] y FTPS [130].

5.4.1 Canal de comunicación NO seguro

Para establecer un canal de comunicación NO seguro entre el nodo cliente y el servidor, alcanza con brindar transporte IP entre el nodo cliente y el servidor, procedimiento descripto en el apartado 5.2.3. Una vez implementado el canal de comunicaciones NO seguro, se puede inyectar tráfico "no seguro" utilizando protocolos como ICMP, HTTP o FTP.

5.4.2 Canales de comunicación seguros

Los canales de comunicación entre el nodo cliente y el servidor (extremo a extremo) se pueden asegurar implementando VPNs basadas en SSL/TLS [111] (OpenVPN) o en IPSec [110] (L2TP/IPSec) o también utilizando protocolos de comunicación seguros como HTTPS [129] y FTPS [130]. En este apartado se describen los detalles de implementación de los canales de comunicación "seguros" sobre los que se realizaron mediciones.

5.4.2.1 OpenSSL para la gestión de Certificados Digitales

Los protocolos utilizados para establecer canales de comunicación seguros como SSL e IPSEC hacen uso de certificados digitales X.509 para autenticar a las entidades participantes (cliente y servidor). Estos certificados pueden ser emitidos por una CA externa (confiable) o ser auto firmados (self signed). En este trabajo se utilizaron certificados auto firmados generados utilizando la aplicación OPENSSL [121], a continuación se detallan los pasos para la generación de los certificados:

1. Configurar OpenSSL
   - Instalar OPENSSL.
   - Crear un directorio para almacenar la base de datos de la CA (Ej: c:\demoCA)
     mkdir \demoCA
   - Crear los siguientes subdirectorios:
     mkdir \demoCA\certs
     mkdir \democa\newcerts
     mkdir \democa\private
   - Crear el archivo de texto \demoCA\serial, incluir una línea con el número de serie para el primer certificado: 00.
   - Crear el archivo \democa\certindex.txt (debe estar vacío).
   - Copiar el archivo *openssl.cnf* a la carpeta demoCA, el archivo se encuentra en el directorio de instalación de OPENSSL.
   - Indicar la ruta para almacenar la información de la CA, modificar en el archivo *openssl.cnf* la línea: Dir = /democa

2. Generar un certificado root auto-firmado (self signed) para la CA
   - Ejecutar desde línea de comandos:
     openssl req -new -x509 -extensions v3_ca -keyout private/ca_testing_key.pem -out ca_testing_cert.pem -days 365 -config openssl.cnf

     C:\>cd \demoCA

```
C:\demoCA>c:\openSSL\bin\openssl req -new -x509 -extensions v3_ca -keyout
private/cakey.pem -out cacert.pem -days 365 -config openssl.cnf

Loading 'screen' into random state - done
Generating a 1024 bit RSA private key
…….
writing new private key to 'private/cakey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:AR
State or Province Name (full name) [Some-State]:Salta
Locality Name (eg, city) []:Salta
Organization Name (eg, company) [Internet Widgits Pty Ltd]:UNSa
Organizational Unit Name (eg, section) []:CIDIA
Common Name (eg, YOUR name) []:CA_testing
Email Address []:srocabad@cidia.unsa.edu.ar
```

- OpenSSL genera la clave privada de la CA (Private/ ca_testing_key.pem) y el certificado auto firmado de la CA (ca_testing_cert.pem). Se debe renombrar ca_testing_key.pem a ca_testing.key y ca_testing_cert.pem a ca_testing.crt respectivamente.

3. Crear una clave privada y una solicitud de firma para la entidad (servidor o cliente)
   - Ejecutar desde línea de comandos:
     openssl req -new -nodes -out name-req.pem -keyout private/name-key.pem -days 365 -config ./openssl.cnf (reemplazar “name” por el nombre de la entidad).

```
Generating a 1024 bit RSA private key
.......++++++
.....................++++++
writing new private key to 'private/name-key.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:AR
State or Province Name (full name) [Some-State]:Salta
Locality Name (eg, city) []:Salta
Organization Name (eg, company) [Internet Widgits Pty Ltd]:UNSa
Organizational Unit Name (eg, section) []:CIDIA
Common Name (eg, YOUR name) []:celular001
Email Address []:srocabad@cidia.unsa.edu.ar

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:123456
An optional company name []:
```

4. Firmar la solicitud generada en el paso 3.
   - Ejecutar desde línea de comandos:

     openssl ca -out name-cert.pem -days 365 -config openssl.cnf -infiles name-req.pem

   - OpenSSL genera la clave privada de la entidad (Private/name_key.pem) y el certificado firmado de la entidad (name_cert.pem).

- De ser necesario, se puede renombrar name_key.pem a name.key y name_cert.pem a name.crt respectivamente y se pueden empaquetar la clave privada y el certificado en formato PFX, ejecutando desde línea de comandos:

  openssl pkcs12 -export -out name.pfx -inkey name_key.key -in name.crt -certfile CA_testing.crt

5.4.2.2 L2TP/IPSEC

Servidor

Para la configuración del servidor L2TP/IPSEC, se procedió a:

1. Gestionar el certificado para el servidor IPSEC (pasos 3 y 4 del apartado 5.4.2.1)
   - Generar una solicitud de certificado para servidor IPSEC (solicitud serverIPSEC.csr y clave privada serverIPSEC.key).
   - Firmar la solicitud de certificado.
2. Instalar el certificado IPSEC (serverIPSEC.crt) y el certificado de la CA (CA_testing.crt) en el servidor "Testing".
3. Habilitar IPSEC
   - Abrir la consola de administración de políticas de seguridad (Secpol.msc)
   - Seleccionar IP security policies in local Computers
   - Seleccionar "server request security" y configurar los siguientes parámetros para la regla "All ip traffic":
     - Authentication method -> USE a certificate from certification authority (CA_testing).
     - Connection type -> Remote Access
     - Filter action -> Security methods -> Modo ESP: 3DES, SHA1. La Tabla 5-4 muestra las opciones de seguridad que se pueden configurar para el servidor IPSEC.

Modo	Autenticación	Cifrado
ESP	SHA1	3DES
ESP	MD5	3DES
AH	SHA1	N/A

Tabla 5-4: Opciones IPSec

   - Asignar la política
4. Configurar el acceso remoto.
   - Abrir la consola de administración del RRAS (Routing and Remote Acces Server), ejecutar desde consola de comandos: C:\WINDOWS\system32\rrasmgmt.msc /s
   - Seleccionar el servidor a configurar ("Testing").
   - Pulsar el botón derecho del mouse y seleccionar la opción "configure and enable routing and remote access". En el asistente de configuración de RRAS:
     - Pulsar Next en el mensaje de bienvenida.
     - Seleccionar la opción Virtual Private Network (VPN) access and NAT y pulsar Next
     - Seleccionar el adaptador de red asociado a la dirección IP pública del servidor, desmarcar la opción "enable security by firewall" y pulsar Next.
     - Seleccionar la opción "From a specified range of addresses" para la asignación de direcciones IP a los clientes y pulsar Next.
     - Pulsar New e introducir una dirección IP inicial y final para el nuevo rango y pulsar OK. Pulsar Next para continuar con el asistente. En nuestro escenario el rango de direcciones se configuro de 192.168.10.1 a 192.168.10.10, la primera dirección del rango se utiliza para el servidor.

- Seleccionar “No” para la autenticación por servidor Radius
- Pulsar Finish.

- Seleccionar la opción Ports del servidor (“Testing”), verificar que los WAN Miniport (L2TP) se encuentren creados (ver Figura 5-6).

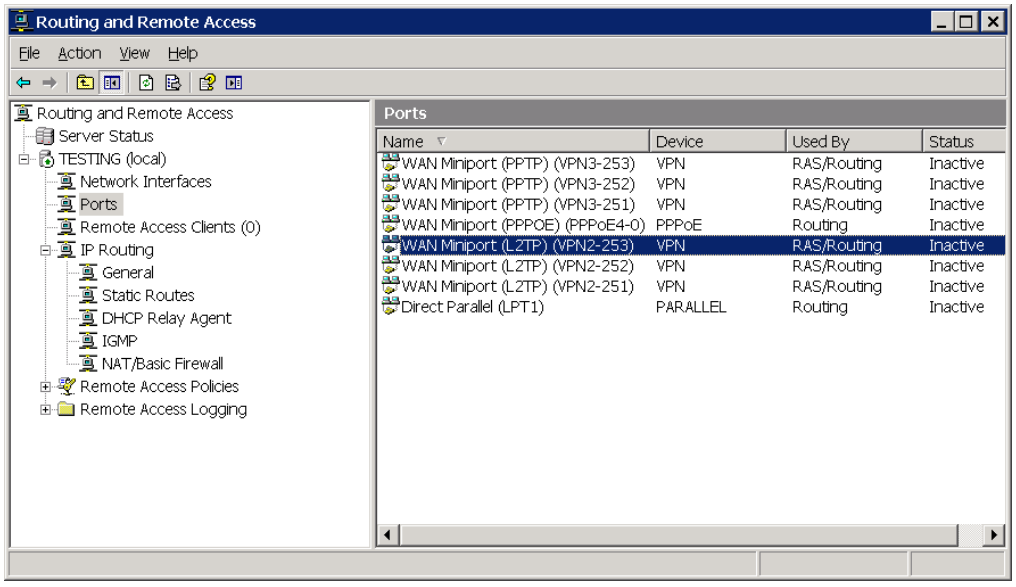

Figura 5-6: Mini puertos WAN en RRAS

5. Activar el acceso remoto.

   Ejecutar desde línea de comandos: net start "Routing and Remote Access".

6. Crear un usuario Windows para ser usado en el nodo cliente.

Cliente

Para la configuración del dispositivo móvil como cliente OpenVPN, se procedió a:

1. Gestionar el certificado para el cliente IPSEC (pasos 3 y 4 del apartado 5.4.2.1).

   - Generar una solicitud de certificado para cliente IPSEC (solicitud clienteIPSEC.csr y clave privada clienteIPSEC.key)
   - Procesar la solicitud y emitir el certificado (clienteIPSEC.pfx).

2. Instalar certificados

   - Copiar los certificados (CA_testing.crt, serverIPSEC.crt y clienteIPSEC.pfx) a la tarjeta SD del dispositivo cliente (/storage/sdcard0/certkey).
   - Seleccionar “Settings->Security->Credential storage->Install from device storage”. El asistente para instalar certificados de Android detectará los ficheros de certificados almacenados en la tarjeta SD.
   - Seleccionar “CA_testing”, introducir la contraseña de protección y pulsar OK.
   - Seleccionar “ServerIPSEC”, introducir la contraseña de protección y pulsar OK.
   - Seleccionar “ClienteIPSEC”, introducir la contraseña de protección y pulsar OK.

3. Crear una conexión OpenVPN

   - Seleccionar: Settings – more settings – VPN.

- Pulsar Add VPN network.
- Cargar los datos de la conexión VPN: Nombre del cliente (Cliente DMZ), Tipo de cliente (L2TP/IPsec PSK), dirección IP del servidor (190.221.183.220), certificado de usuario IPSec, certificado de la CA y certificado del servidor IPSec (Figura 5-7).
- Pulsar "Save" para guardar la conexión VPN.

4. Conectar al servidor [9]

   - Ir al menu Settings – more settings – VPN.
   - Pulsar sobre la conexión VPN creada (cliente DMZ).
   - Introducir el usuario y contraseña para el servidor.
   - Verificar que la conexión se establezca correctamente (Figura 5-8).

5. Comprobar el funcionamiento del túnel.

   - Desde el emulador de terminal ejecutar:

   ```
   # su
   # cd /system/xbin
   # ping –t 192.168.10.1
   ```

   - Comprobar que las respuestas del servidor lleguen al cliente.

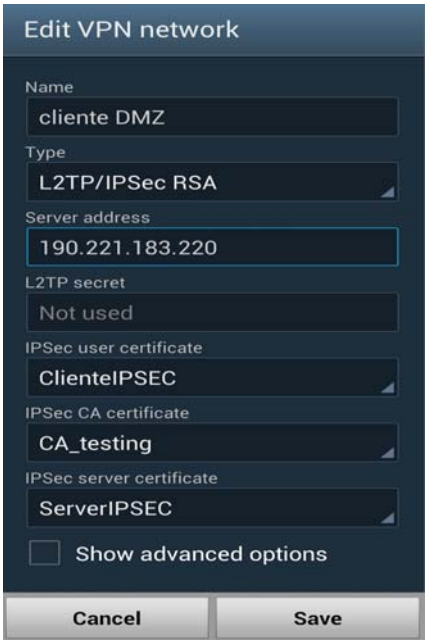

Figura 5-7: Configuración IPSec en el cliente Android

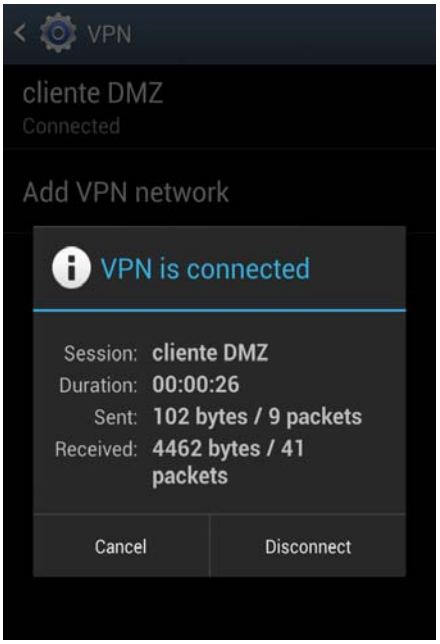

Figura 5-8:Conexión L2TP/IPSec en el cliente Android

Certificados y claves

La Tabla 5-5 muestra los archivos de certificados y claves utilizados para la configuración del canal L2TP/IPSEC.

Archivo	Descripción
Ca_testing.crt	Certificado de la CA

[9] L2TP/IPSEC crea un túnel extremo a extremo, entre el servidor (192.168.10.1) y el cliente (192.168.10.2)

ServerIPSEC.crt	Certificado del servidor IPSec
ServerIPSEC.key	Clave privada del servidor IPSec (1024 bits)
ClienteIPSEC.crt	Certificado del cliente IPSec
ClienteIPSEC.key	Clave privada del cliente IPSec (1024 bits)

Tabla 5-5: Archivos de certificados y claves privadas utilizados para el canal L2TP/IPSEC

5.4.2.3 OPENVPN (SSL/TLS)

Servidor

Para la configuración del servidor OpenVPN, se procedió a:

1. Instalar OpenVPN
   Instalar la aplicación OpenVPN Server para Windows v. 2.3.2 [122] en la carpeta "c:\program files\openvpn\config".
2. Configurar el servidor OpenVPN
   - Crear el archivo de configuración del servidor OpenVPN (server-DMZunsa.ovpn) con el contenido especificado en la Tabla 5-6.
   - Copiar el archivo a la carpeta "c:\program files\openvpn\config".

CLIENTE (cliente-DMZunsa.ovpn)	SERVIDOR (server-DMZunsa.ovpn)
remote srvtesting.unsa.edu.ar port 443 proto tcp-client dev tun client tls-client persist-key persist-tun #Certificado root de la CA ca CA_testing.crt #Certificado y claves para el cliente firmado por la CA cert VPN-client.crt key VPN-client.key # Algoritmo de cifrado # Triple DES con clave de 192 bits cipher DES-EDE3-CBC keysize 192 # Algoritmo para autenticación auth SHA1	 port 443 proto tcp-server dev tun server 10.3.0.0 255.255.255.0 client-to-client tls-server #Diffie Hellman Key dh keys/dh1024.pem #Certificado root de la CA ca CA_testing.crt #Certificado y claves para el servidor firmado por la CA cert VPN-server.crt key VPN-server.key # Algoritmo de cifrado # Triple DES con clave de 192 bits cipher DES-EDE3-CBC keysize 192 # Algoritmo para autenticación auth SHA1

Tabla 5-6: Configuración OpenVPN para el cliente y el servidor

Parameter	Options	Function	Usage
secret	<file>	Points to the file with the static key	--secret /kex.txt
cipher	<alg>	Specifies the algorithm to use for encryption of packets	--cipher AES-256-CBC
keysize	<n>	Specifies the size of the cipher key in bits	--keysize 128
auth	<alg>	Defines the message digest algorithm <alg> used by the HMAC authentication algorithm	--auth SHA1
tls-server		Uses SSL certificates and acts as TLS server during TLS handshake	--tls-server
tls-client		Uses SSL certificates and act as TLS client during TLS handshake	--tls-client
ca	<file>	Your generated CA file	--ca /CA.crt
dh	<file>	Your generated Diffie-Hellman key	--dh /DH.pem

Tabla 5-7: OPENVPN - Parámetros para la configuración de la criptografía
Fuente: [127]

3. Generar la clave Diffie-Hellman (Grupo 2 - 1024 bits)

   - Ejecutar desde línea de comandos:
     C:\OpenSSL\bin> openssl dhparam -outform PEM -out dh1024.pem 1024
   - Copiar la clave DH (dh1024.pem) a la carpeta "c:\program files\openvpn\config".

4. Especificar el algoritmo de cifrado

   - Editar el archivo de configuración del servidor OpenVPN (server-DMZunsa.ovpn)
   - Modificar el algoritmo de cifrado en la línea de parámetro "cipher" (ver Tabla 5-8)
   - Modificar el tamaño de clave de encriptación en la línea de parámetro "keysize".

Algoritmo	Parametro	Tamaño de clave
Triple DES	DES-EDE3-CBC[10]	192 bits[11]
Blowfish	BF-CBC	128 bits
AES	AES-256-CBC	256 bits

Tabla 5-8: OPENVPN - Algoritmos de cifrado

5. Especificar el algoritmo de autenticación.

   - Editar el archivo de configuración del servidor OpenVPN (server-DMZunsa.ovpn)
   - Modificar el algoritmo de cifrado en la línea de parámetro "auth" (ver Tabla 5-9)

Algoritmo	Parametro	Tamaño de compendio
HMAC-SHA1	SHA1	160 bits (default)
HMAC-MD5	MD5	128 bits (default)

Tabla 5-9: OPENVPN - Algoritmos para autenticación

6. Instalar el certificado de la CA
   - Copiar el certificado de la CA (CA_testing.crt) a la carpeta "c:\program files\openvpn\config".

7. Gestionar el certificado y clave privada para el servidor OpenVPN.
   Utilizar el mecanismo detallado en los pasos 3 y 4 del apartado 5.4.2.1, para:
   - Generar la clave privada (VPN-server.key) y solicitud de certificado (VPN-server.csr).

[10] CBC – Cipher Block Chaining Mode (Modo cifrado en bloque).

[11] Una clave DES tiene 64 bits de longitud, de los cuales solamente 56 bits son utilizados en el proceso de cifrado, los 8 bits restantes son utilizados para la detección de errores o paridad. Entonces, una clave 3DES tendrá 3 x 64 = 192 bits de longitud, de los cuales solo se utilizan 3 x 56 = 168 bits.

- Emitir y firmar la solicitud de certificado (VPN-server.crt).

8. Instalar el certificado y clave privada
   Copiar el certificado (VPN-server.crt) y la clave privada del servidor (VPN-server.key) a la carpeta "c:\program files\openvpn\config".
9. Levantar el servicio OpenVPNservice.
   - Abrir la consola de comandos de Windows Server (CMD.EXE).
   - Ejecutar "C:\Program Files\OpenVPN\bin\openvpnserv.exe" - start

Cliente

Para la configuración del dispositivo móvil como cliente OpenVPN, se procedió a:

1. Instalar el binario ejecutable (openvpn).
   - Ejecutar la aplicación aplicación "Openvpn installer".
   - Seleccionar la opción "install".
   - Seleccionar destino /system/xbin/.
2. Configurar el servidor OpenVPN
   - Crear el archivo de configuración del cliente OpenVPN (cliente-DMZunsa.ovpn) con el contenido especificado en la Tabla 5-6.
   - Copiar el archivo de configuración del cliente OpenVPN a la carpeta /storage/sdcard0/openvpn del nodo cliente.
3. Instalar el certificado de la CA
   - Copiar el certificado de la CA (CA_testing.crt) y del servidor (VPN_server.crt) en la carpeta "/storage/sdcard0/openvpn".
4. Gestionar el certificado y clave privada para el dispositivo cliente.
   Utilizar el mecanismo detallado en los pasos 3 y 4 del apartado 5.4.2.1, para:
   - Generar la clave privada (VPN-client.key) y solicitud de certificado (VPN-client.csr).
   - Emitir y firmar la solicitud de certificado ("VPN-client.crt").
5. Instalar el certificado y clave privada
   Copiar el certificado (VPN-client.crt) y la clave privada del cliente (VPN-client.key) a la carpeta "/storage/sdcard0/openvpn".
6. Conectar al servidor [12]
   - Ejecutar la aplicación "Openvpn settings".
   - Seleccionar la opción "Prerequisites" y verificar que todos los requisitos se cumplan (ver Figura 5-9).
   - Activar la opción OpenVPN y la configuración "cliente-DMZunsa.ovpn" (ver Figura 5-10).

[12] OpenVPN crea un túnel extremo a extremo, entre el servidor (10.3.0.1) y el cliente (10.3.0.2)

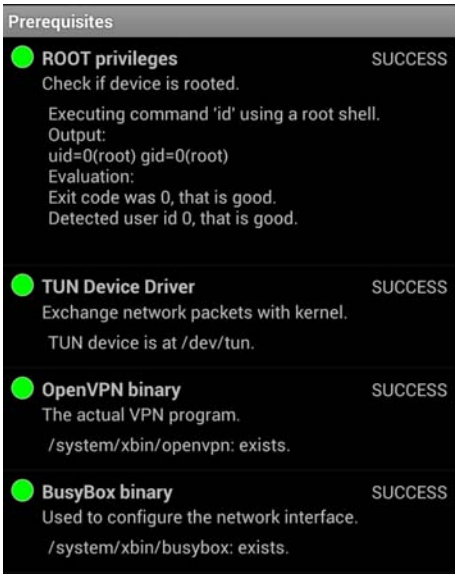

Figura 5-9: Pre-requisitos cliente OpenVPN

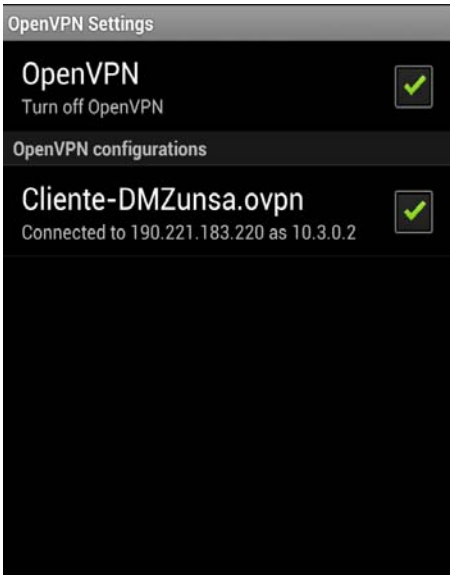

Figura 5-10:Conexión del cliente OpenVPN

7. Comprobar el funcionamiento del túnel.
   - Desde el emulador de terminal ejecutar:

     # su
     # cd /system/xbin
     # ping –t 10.3.0.1

   - Comprobar que las respuestas del servidor lleguen al cliente.

Certificados y claves

La Tabla 5-10 muestra los archivos de certificados y claves utilizados para la configuración del canal OpenVPN.

Archivo	Descripción
CA_testing.crt	Certificado root de la CA
Dh1024.pem	Clave Diffie-Hellman (1024 bits)
VPN-server.crt	Certificado del servidor Testing
VPN-server.key	Clave privada del servidor Testing (1024 bits)
VPN-client.crt	Certificado del nodo cliente
VPN-client.key	Clave privada del nodo cliente (1024 bits)

Tabla 5-10: Archivos de certificados y claves privadas utilizados para el canal OpenVPN

5.4.2.4 OPENVPN (SSL/TLS) con compresión LZO

Para introducir compresión LZO [128] al túnel VPN, se deben modificar los archivos de configuración del servidor (server-DMZunsa.ovpn) y del cliente (cliente-DMZunsa.ovpn) y agregar las siguientes líneas:

```
# enable LZO compression
comp-lzo
```

Estas modificaciones se deben realizar antes de levantar el servicio OpenVPNservice (en el servidor) y de establecer la conexión con OpenVPN settings (en el cliente).

5.4.2.5 HTTP over SSL (HTTPS)

Servidor

Para la configuración del servidor HTTPs, se procedió a:

1. Instalar el certificado de la CA (CA_testing.crt) en el servidor.
2. Generar una clave privada y un certificado firmado para el servidor, siguiendo los pasos 3 y 4 del apartado 5.4.2.1.
3. Importar el certificado y la clave privada.

   - Abrir la consola de administración de IIS (Internet Information Services), ejecutando desde la línea de comandos: C:\WINDOWS\inetsrv\iis.msc.
   - Seleccionar el nombre del servidor ("Testing").
   - Seleccionar Server Certificates. (Figura 5-11)
   - Importar el certificado generado en el punto anterior.

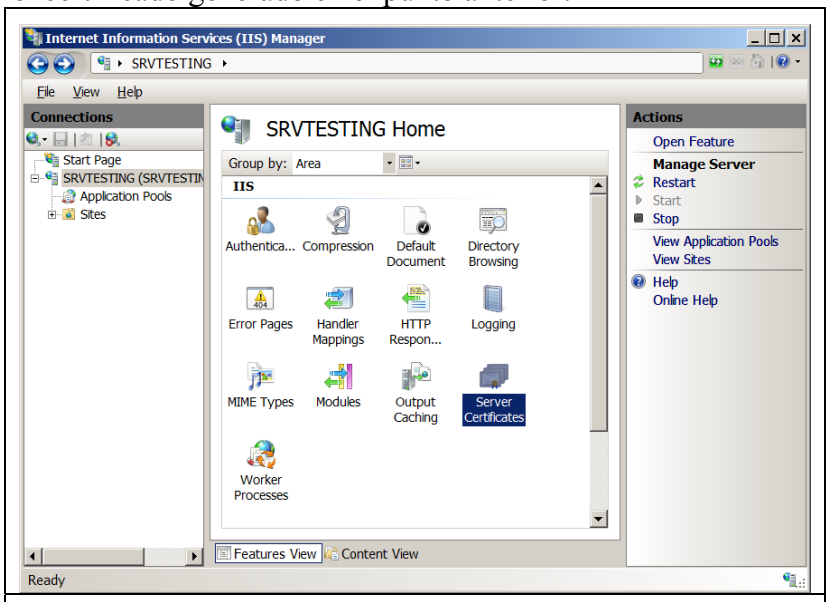

Figura 5-11: Importación de certificados en IIS7

4. Configurar el sitio Web.

   - Abrir la consola de administración de IIS (Internet Information Services), ejecutando desde la línea de comandos: C:\WINDOWS\inetsrv\iis.msc.
   - Seleccionar el nombre del servidor ("Testing").
   - Desplegar las opciones, pulsar el botón derecho del ratón sobre la opción "Web Sites" y seleccionar la opción "Add Web Site". Completar la siguiente información en el formulario (Figura 5-12):
     - Introducir el nombre del sitio ("WEB SSL").
     - Introducir la ruta completa del directorio donde se van a copiar los archivos del servidor web ("C:\Inetpub\wwwroot")
     - En la sección "Binding" seleccionar tipo de protocolo HTTPs y puerto (443)

- Seleccionar el certificado SSL importado en el punto anterior.
- Pulsar OK.

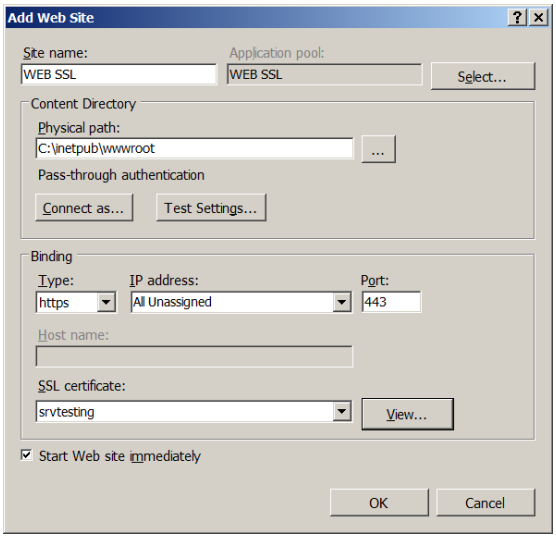

Figura 5-12: Configuración de sitio en IIS7

5. Habilitar canal seguro SSL

- Abrir la consola de administración de IIS.
- Desplegar los sitios webs del servidor ("Testing"->Web sites), marcar el sitio web creado en el punto 1 ("WEB SSL").
- Seleccionar "SSL settings" Secure Communications.
- Marcar las opciones "Require secure channel (SSL)" y "Require 128-bit encryption" y pulsar OK (Figura 5-13).

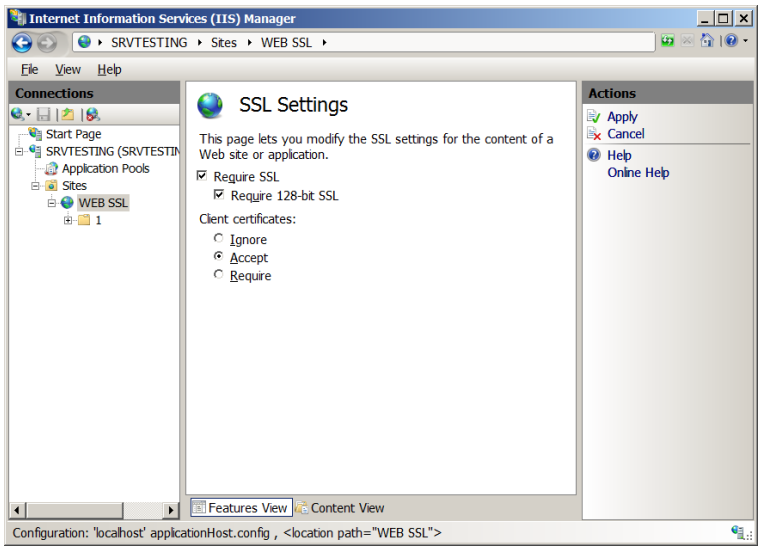

Figura 5-13: SSL settings en IIS7

6. Seleccionar algoritmos de autenticación y encriptación.

- Estos algoritmos se configuran utilizando la aplicación APPCMD desde la línea de comandos de Windows. Se pueden configurar algoritmos hash (MD5 y SHA1) y

algoritmos de encriptación (DES, 3DES y AES). Para el servidor "testing" se eligieron los algoritmos SHA1 y 3DES y se configuraron de la siguiente manera:

```
appcmd set config /commit:WEBROOT /section:machineKey /validation:SHA1
appcmd set config /commit:WEBROOT /section:machineKey /validation:3DES
```

7. Deshabilitar la compresión

   - Abrir la consola de administración de IIS.
   - Desplegar los sitios webs del servidor ("Testing"->Web sites), marcar el sitio web creado en el punto 1 ("WEB SSL").
   - Seleccionar "Compression"
   - Desmarcar "enable dynamic content compression"
   - Desmarcar "enable static content compression"

8. Cargar contenidos al servidor

   - Generar un archivo de 1024 Kbytes de tamaño con datos aleatorios (archivo1024.txt), se utilizaron archivos con diferentes relaciones de compresión para observar la variación de consumo energético en función de la relación de compresión elegida.
   - Copiar el archivo generado a la carpeta asignada al sitio web (C:\Inetpub\www).

Cliente

1. Instalar el certificado de la CA

   - Copiar el certificado (C:\certkey\CA_testing.crt) a la tarjeta SD del dispositivo cliente (/storage/sdcard0/certkey).
   - Seleccionar "Settings->Security->Credential storage->Install from device storage". El asistente para instalar certificados de Android detectará el fichero del certificado automáticamente en la tarjeta SD.
   - Introducir la contraseña de protección y pulsar OK
   - Introducir el nombre para el certificado (CA_testing), nos servirá para diferenciar este certificado de otros que tengamos instalados a la hora de utilizarlo.

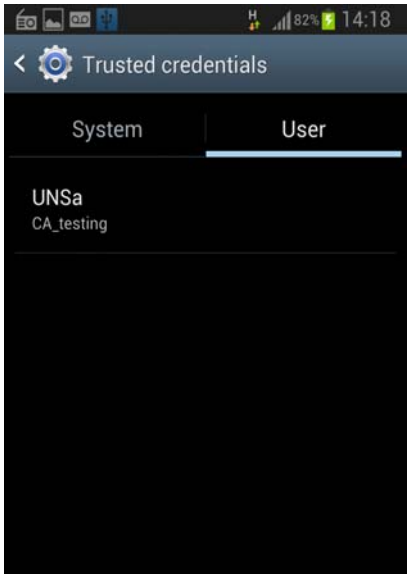

Figura 5-14: Credenciales almacenadas en Android

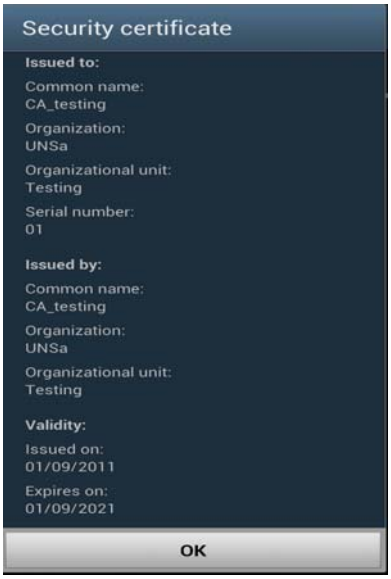

Figura 5-15: Información de certificado en Android

2. Verificar la información del certificado.

   - Seleccionar "Settings->Security->Credential storage->Trusted credentials->User" (Figura 5-14).
   - Pulsar sobre el certificado de la CA (UNSa/CA_testing) y comprobar que los datos sean correctos (Figura 5-15).

3. Comprobar el funcionamiento

   Realizar las siguientes pruebas para verificar el correcto funcionamiento del servidor HTTPs y del canal de comunicaciones:

   - Descargar un archivo, desde el servidor web "Testing" al nodo cliente, utilizando cualquier navegador instalado en el dispositivo móvil.
     Abrir el navegador y cargar la siguiente dirección: https://190.221.183.220/archivo1024.txt
   - Descargar un archivo, desde el servidor web "Testing" al nodo cliente, utilizando la aplicación "wget" del paquete Busybox.
     Desde el emulador de terminal ejecutar:
     # su
     # cd /system/xbin
     # wget https://190.221.183.220/archivo1024.txt

5.4.2.6 FTP over SSL (FTPS)

Servidor

1. Instalar el certificado de la CA (CA_testing.crt) en el servidor.
2. Configurar el servidor FTP
   - Crear la carpeta "home directory" para el sitio FTP (C:\Inetpub\ftproot)
   - Abrir la consola de administración de Filezilla Server, ejecutando desde línea de comandos "C:\Program Files\FileZilla Server\FileZilla Server Interface.exe".
   - Ir a la pantalla de configuración de usuarios (Menu Edit-> users).
   - Seleccionar la solapa "general" pulsar Add e introducir el nombre de usuario "anonymous", pulsar OK.
   - Seleccionar la solapa "shared folders" pulsar Add y seleccionar el directorio donde se alojaran los archivos del sitio FTP (C:\Inetpub\ftproot).
   - Tildar los permisos Read y List y pulsar OK (Figura 5-16).
3. Generar una clave privada (servidorftp.key) y un certificado firmado (servidorftp.crt) para el servidor FTP, siguiendo los pasos 3 y 4 del apartado 5.4.2.1.

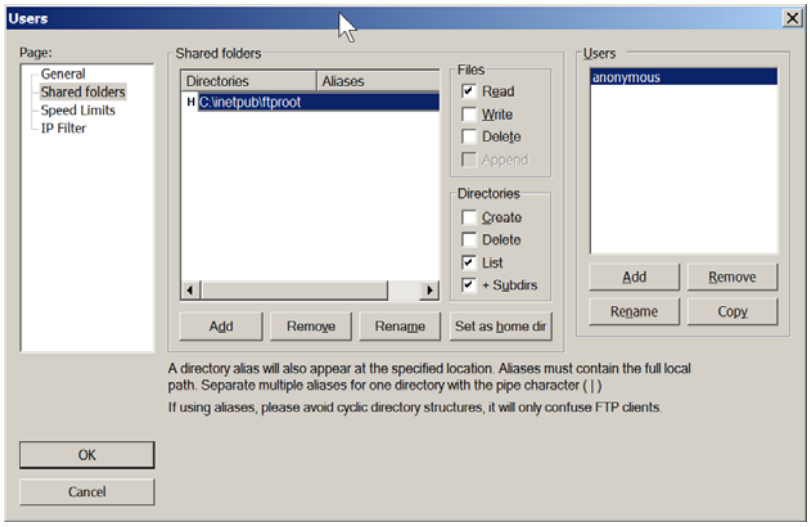

Figura 5-16: Configuración del servidor FTP

4. Instalar el certificado en el servidor FTP

   - Abrir la consola de administración de Filezilla Server, ejecutando desde línea de comandos "C:\Program Files\FileZilla Server\FileZilla Server Interface.exe".
   - Ir a la opción de configuración de parámetros generales (Menu Edit-> settings).
   - Seleccionar SSL/TLS settings (ver Figura 5-17).
   - Marcar “Enable FTP over SSL/TLS support” e indicar la ruta de la clave privada (c:\certkey\servidorftp.key) y del certificado (c:\certkey\servidorftp.cer).
   - Habilitar la opción “Allow explicit FTP over TLS”.
   - Introducir el número de puerto para conexiones SSL/TLS (990).
   - Pulsar OK.

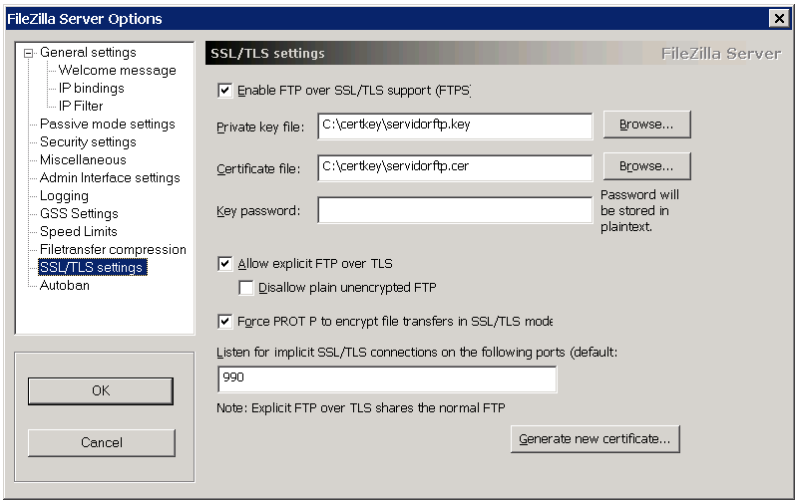

Figura 5-17: Instalación del certificado en el servidor FTP

5. Cargar contenidos en el servidor.

   - Generar un archivo de texto de 1024 Kbytes de tamaño con datos aleatorios (archivo1024.txt).
   - Copiar el archivo generado a la carpeta asignada al sitio web (C:\Inetpub\ftproot).

Cliente

1. Instalar el certificado de la CA
   Ídem a la configuración del cliente HTTPs

2. Verificar la información del certificado
   Ídem a la configuración del cliente HTTPs

3. Comprobar el funcionamiento
   - Abrir la aplicación AndFTP y pulsar Add.
   - Introducir los datos de configuración de conexión (ver Figura 5-18) y pulsar Save.
   - Seleccionar el servidor configurado (190.221.183.220) y pulsar Connect (Figura 5-19).
   - Seleccionar el archivo a descargar (archivo1024.txt) y pulsar "download" (Figura 5-20).
   - Verificar que el archivo se descargue correctamente.

Figura 5-18: Configuración del cliente FTPs

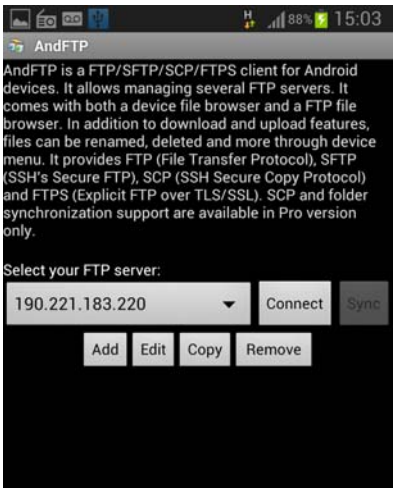

Figura 5-19: Conexión al servidor FTPs

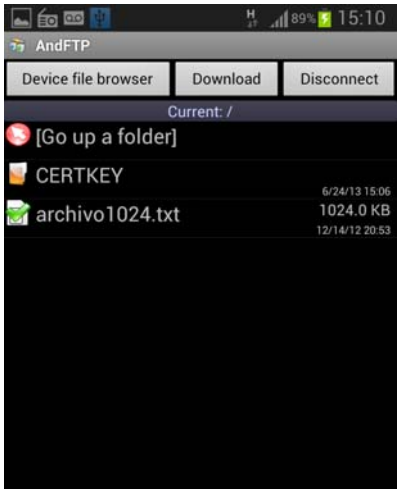

Figura 5-20: Descarga desde el servidor FTPs

5.4.3 Resumen de configuraciones de canal

La Tabla 5-11 muestra las características de los canales implementados sobre nuestro escenario de pruebas. Los canales fueron divididos en 3 grupos: 1. Canal no seguro, 2. Canal seguro utilizando protocolos seguros (HTTPs y FTPs) y 3. Canal seguro basado en VPN (L2TP/IPSEC, OPENVPN SSL/TLS y OPENVPN SSL/TLS con compresión LZO).

CANAL	Seguridad	Cliente (Android)	Servidor (Windows)	IP servidor	IP cliente
NO SEGURO	Sin seguridad	N/A	N/A	190.221.183.220	IP dinámico
SEGURO (PROTOCOLO SEGURO)	HTTPs	WGET	IIS6	190.221.183.220	IP dinámico
	FTPs	ANDFTP	FILEZILLA	190.221.183.220	IP dinámico
SEGURO (VPN)	IPSEC	CLIENTE L2TP/IPSEC	RRAS L2TP/IPSEC	192.168.1.1	192.168.1.xx
	SSL/TLS	OPENVPN Client	OPENVPN Server	10.3.0.1	10.3.0.xx
	SSL/TLS/LZO	OPENVPN Client	OPENVPN Server	10.3.0.1	10.3.0.xx

Tabla 5-11: Configuraciones de seguridad (Cliente/Servidor)

En la Tabla 5-12 se resumen los protocolos y algoritmos utilizados para la implementación de los canales seguros.

CANAL	Protocolo de seguridad	Algoritmo de Compresión	Algoritmo de Cifrado	Algoritmo de Autenticación	Algoritmo para Calculo de compendio
HTTPS	SSL/TLS	n/a	3DES (192 bits)	HMAC	SHA1 (160 bits)
FTPS	SSL/TLS	n/a	3DES (192 bits)	HMAC	SHA1 (160 bits)
L2TP/IPSEC	IPSEC	n/a	3DES (192 bits)	HMAC	SHA1 (160 bits)
OPENVPN	SSL/TLS	n/a	3DES (192 bits)	HMAC	SHA1 (160 bits)
OPENVPN LZO	SSL/TLS	LZO	3DES (192 bits)	HMAC	SHA1 (160 bits)

Tabla 5-12: Protocolos y algoritmos utilizados para la implementación de canales seguros

Adicionalmente, el canal OPENVPN se configuro utilizando diferentes algoritmos de encriptación e integridad. Los archivos de configuración del servidor y el cliente (ver Tabla 5-6) fueron modificados para habilitar/deshabilitar la compresión LZO y seleccionar diferentes algoritmos para cifrado, autenticación y calculo de compendio. En la Tabla 5-13 se presentan las configuraciones de canal OPENVPN utilizadas para la implementación del canal seguro.

Algoritmo de Compresión	Algoritmo de Cifrado	Algoritmo de Autenticación	Algoritmo para Calculo de compendio
n/a	3DES (192 bits)	HMAC	SHA1 (160 bits)
n/a	3DES (192 bits)	HMAC	MD5 (160 bits)
n/a	AES (128 bits)	HMAC	SHA1 (160 bits)
n/a	AES (128 bits)	HMAC	MD5 (128 bits)
LZO	3DES (192 bits)	HMAC	SHA1 (160 bits)
LZO	3DES (192 bits)	HMAC	MD5 (160 bits)
LZO	AES (128 bits)	HMAC	SHA1 (160 bits)
LZO	AES (128 bits)	HMAC	MD5 (128 bits)

Tabla 5-13: Algoritmos de encriptación e integridad utilizados con OPENVPN

5.5 Mediciones realizadas

Para cada configuración de canal se efectuaron las siguientes mediciones:

Medición	**Aplicación Cliente**	**Aplicación Servidor**	**Mecanismo para generar tráfico entre el Cliente y el Servidor.**
Latencia ICMP	Busybox ping	Windows Stack TCP/IP	Echo Request/Reply (32 bytes)
Latencia HTTP	HTTping	HTTP Server (IIS7)	HTTP GET
Throughput TCP y consumo de energía	Iperf Client y Powertutor	Iperf Server	Inyección de tráfico TCP aleatorio (1024 Kbytes)
Throughput HTTP y consumo de energía	Busybox wget y Powertutor	HTTP Server (IIS7)	Descarga de archivo (de 1024 Kbytes) utilizando el protocolo HTTP.
Throughput HTTPs y consumo de energía	Busybox wget y Powertutor	HTTPs Server (IIS7)	Descarga de archivo (de 1024 Kbytes) utilizando el protocolo HTTPs.
Throughput FTP y consumo de energía	Busybox wget y Powertutor	FTP Server (Filezilla)	Descarga de archivo (de 1024 Kbytes) utilizando el protocolo FTP.
Throughput FTPs y consumo de energía	AndFTP y Powertutor	FTPs Server (Filezilla)	Descarga de archivo (de 1024 Kbytes) utilizando el protocolo FTPs.

Tabla 5-14: Mecanismos utilizados para efectuar las mediciones

5.5.1 Metodología de medición

Las mediciones se automatizaron, utilizando scripts y aplicaciones que corrieron de forma continua durante 7 días en la franja horaria 6:00 am a 11:00 pm, de esta manera se contemplaron diferentes niveles de carga de la red GPRS. Los resultados obtenidos se promediaron para determinar el valor final de cada medición.

A continuación se enumeran los pasos necesarios para efectuar una medición:

- Establecer comunicación extremo a extremo, según la configuración de canal utilizada (ver Tabla 5-11).
- Ejecutar la aplicación Powertutor.
- Arrancar el monitoreo de consumo de energía (Pulsar "Start Profiler")
- Generar tráfico entre el Cliente y el Servidor, utilizando la aplicación y el mecanismo según el tipo de medición a efectuar (ver Tabla 5-14)
- Detener Powertutor (Pulsar "Stop Profiler")
- Guardar el "log" de Powertutor [13] (Pulsar Menú -> Save Log)
- Copiar el "log" generado por "Powertutor".
- Copiar el "log" generado por la aplicación utilizada para la medición.
- Analizar y procesar los archivos de "logs".
- Promediar resultados.

Una vez obtenidos los resultados promedio de todas las pruebas, se procedió a generar los gráficos comparativos, los mismos se presentan en el capitulo de resultados.

5.5.2 Resumen de las mediciones

La Tabla 5-15 resume las diferentes configuraciones de canal implementadas, las mediciones realizadas y las aplicaciones utilizadas para medir.

[13] Powertutor almacena en un archivo de texto (*log*) el consumo de energía detallado por aplicación y componente del sistema. En el capitulo 4 se detalla el funcionamiento de esta aplicación y en el apéndice 4 se describe detalladamente el análisis de un archivo log.

MEDICIONES	CONFIGURACION DE CANAL					
	NO SEGURO	L2TP IPSEC	OPENVPN (SSL/TLS)	OPENVPN LZO (SSL/TLS)	HTTPs (SSL/TLS)	FTPs (SSL/TLS)
Latencia ICMP (Busybox Ping)	SI	SI	SI	SI	N/A	N/A
Latencia HTTP (HTTPing)	SI	SI	SI	SI	N/A	N/A
Throughput TCP (Iperf)	SI	SI	SI	SI	N/A	N/A
Throughput HTTP (Busybox Wget)	SI	SI	SI	SI	SI	N/A
Throughput FTP (Wget - ANDftp)	SI	SI	SI	SI	N/A	SI

Consumo de energía (Powertutor)

Tabla 5-15: Resumen de mediciones realizadas

Complementando la información presentada en la Tabla 5-15, las siguientes tablas muestran como fueron utilizadas las aplicaciones cliente en cada configuración de canal y medición efectuada:

Latencia ICMP (Busybix ping)	
CANAL	CLIENTE
NO seguro	Ping 190.221.183.220
Seguro L2TP/ IPSEC	Ping 192.168.1.1
Seguro SSL/TLS	Ping 10.3.0.1
Seguro SSL/TLS/LZO	Ping 10.3.0.1

Latencia HTTP (HTTPing)	
CANAL	CLIENTE
NO seguro	HTTPing http://190.221.183.220
NO seguro - HTTPs	HTTPing https://190.221.183.220
Seguro – L2TP/ IPSEC	HTTPing http://192.168.1.1
Seguro - SSL/TLS	HTTPing http://10.3.0.1
Seguro - SSL/TLS/LZO	HTTPing http://10.3.0.1

Throughput TCP (Iperf)	
CANAL	CLIENTE
NO seguro	iperf -c 190.221.183.220 -n 1024k –l 8k –w64k
Seguro – L2TP/ IPSEC	iperf -c 192.168.1.1 -n 1024k –l 8k –w64k
Seguro - SSL/TLS	iperf -c 10.3.0.1 -n 1024k –l 8k –w64k
Seguro - SSL/TLS/LZO	iperf -c 10.3.0.1 -n 1024k –l 8k –w64k

Throughput HTTP	
CANAL	CLIENTE
NO seguro	wget http:\\190.221.183.220\archivo1024.txt
NO seguro - HTTPs	wget https:\\190.221.183.220\archivo1024.txt
Seguro - L2TP/ IPSEC	wget http:\\192.168.1.1\archivo1024.txt
Seguro - SSL/TLS	wget http:\\10.3.0.1\archivo1024.txt
Seguro - SSL/TLS - LZO	wget http:\\10.3.0.1\archivo1024.txt

Throughput FTP	
CANAL	CLIENTE
NO seguro	wget ftp:\\190.221.183.220\archivo1024.txt
NO seguro - FTPs	AndFTP ftps:\\190.221.183.220\archivo1024.txt
Seguro - L2TP/ IPSEC	wget ftp:\\192.168.1.1\archivo1024.txt
Seguro - SSL/TLS	wget ftp:\\10.3.0.1\archivo1024.txt
Seguro - SSL/TLS/LZO	wget ftp:\\10.3.0.1\archivo1024.txt

"Esta página se dejó en blanco intencionalmente".

Capítulo 6: Resultados y conclusiones

6.1 Resultados

En este apartado se presentan los resultados obtenidos en las mediciones, utilizando gráficos comparativos que resumen los aspectos estudiados.

6.1.1 Latencia ICMP

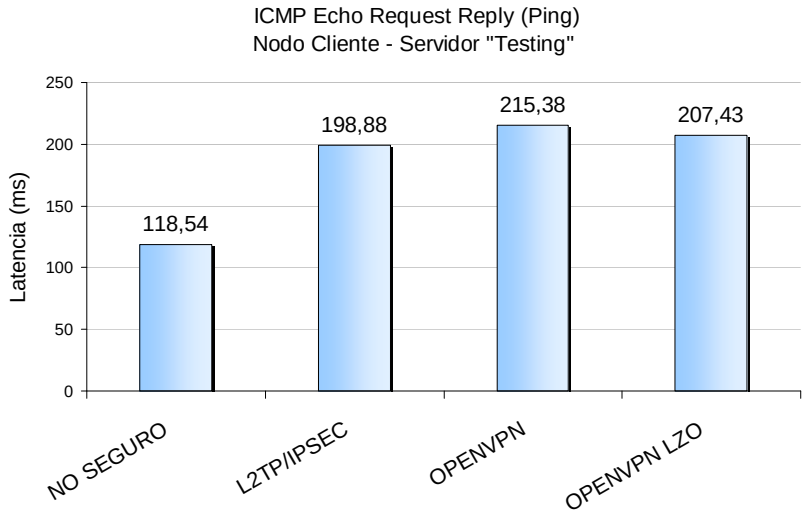

Gráfico 6-1: Latencia ICMP para solicitudes ECHO Request/Reply generadas por Ping

En el Gráfico 6-1 se evidencia que la implementación de un canal seguro introduce un incremento considerable en la latencia: 70% en promedio. Asimismo, se observa que los valores de latencia ICMP para solicitudes echo resultan elevados en OpenVPN respecto a las otras medidas y la compresión no consigue una mejora significativa, lo cual se debe a que las pruebas se realizaron enviando una cantidad de datos de carga útil muy reducida (32 bytes).

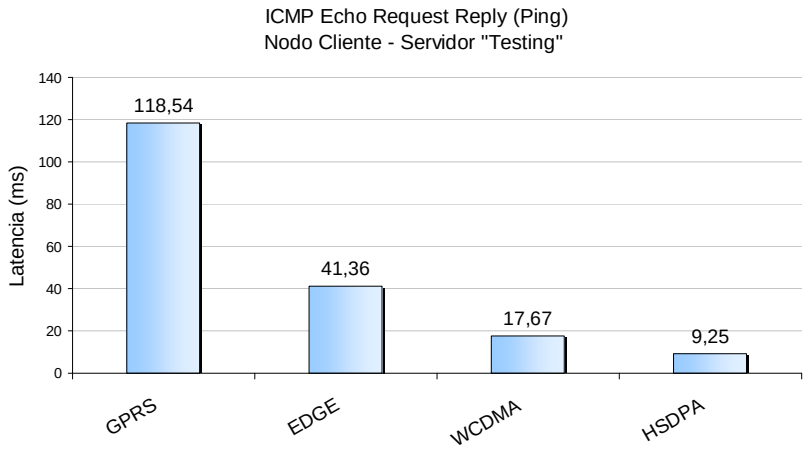

Gráfico 6-2: Latencia ICMP echo Request Reply para diferentes redes celulares

A modo de referencia y para ilustrar el bajo rendimiento de GPRS frente a las redes celulares disponibles en zonas urbanas, en el Gráfico 6-2 se muestran los resultados de las mediciones de latencia (Ping) sobre un canal no seguro utilizando diferentes tecnologías. Si comparamos GPRS con la tecnología de mejor rendimiento HSDPA, observaremos una reducción en la latencia de casi un 130%.

6.1.2 Latencia HTTP

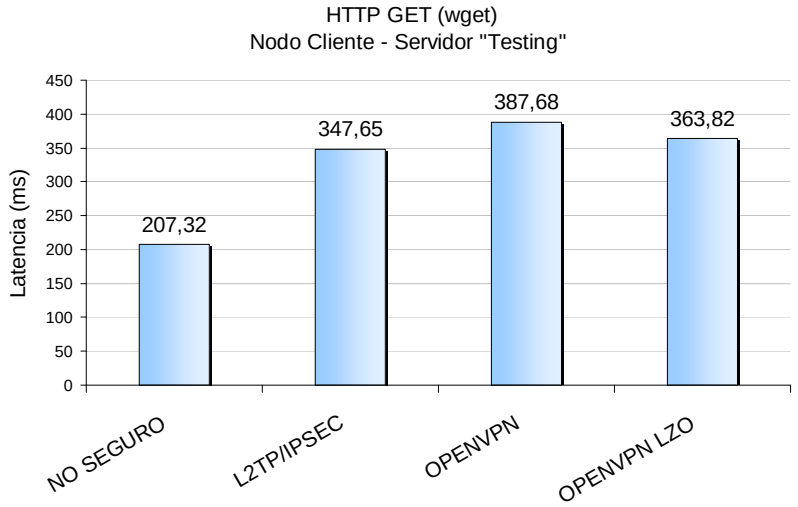

Gráfico 6-3: Latencia HTTP para solicitudes GET generadas por HTTPing

En el Gráfico 6-3 puede verse que en OpenVPN sigue manteniéndose elevada la latencia, en este caso para solicitudes HTTP.

6.1.3 Throughput TCP

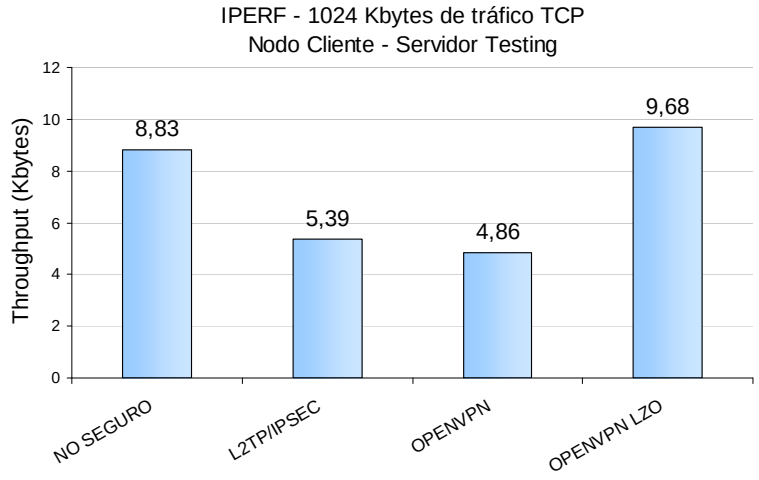

Gráfico 6-4: Throughput para tráfico TCP generado por IPerf

En el Gráfico 6-4 se observa que el uso de un canal como OpenVPN, disminuye casi en un 50% los valores obtenidos para el throughput TCP, respecto del canal no seguro. La compresión provoca un aumento del orden del 100% en el rendimiento del canal OpenVPN.

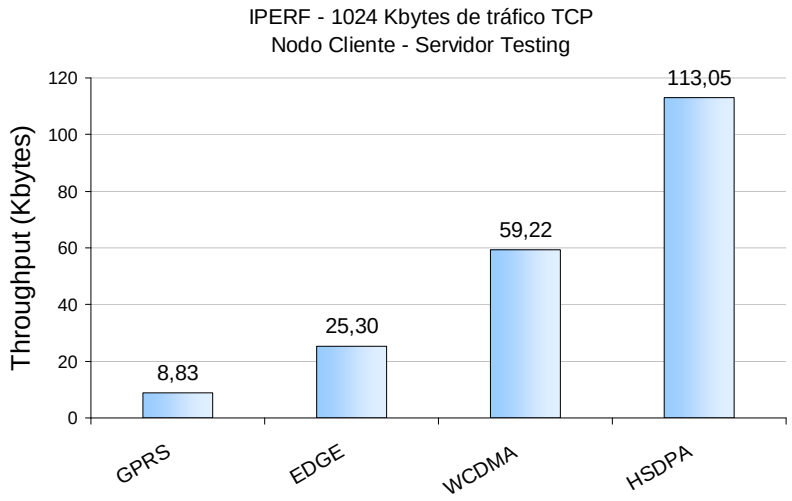

Gráfico 6-5: Throughput para tráfico TCP Iperf en diferentes redes celulares, utilizando un canal no seguro.

La comparación de los valores de throughput obtenidos utilizando canales no seguros implementados sobre distintas tecnologías celulares se muestra en el Gráfico 6-5, del mismo se desprende que las diferencias son altamente significativas dependiendo de la tecnología utilizada.

6.1.4 Throughput HTTP

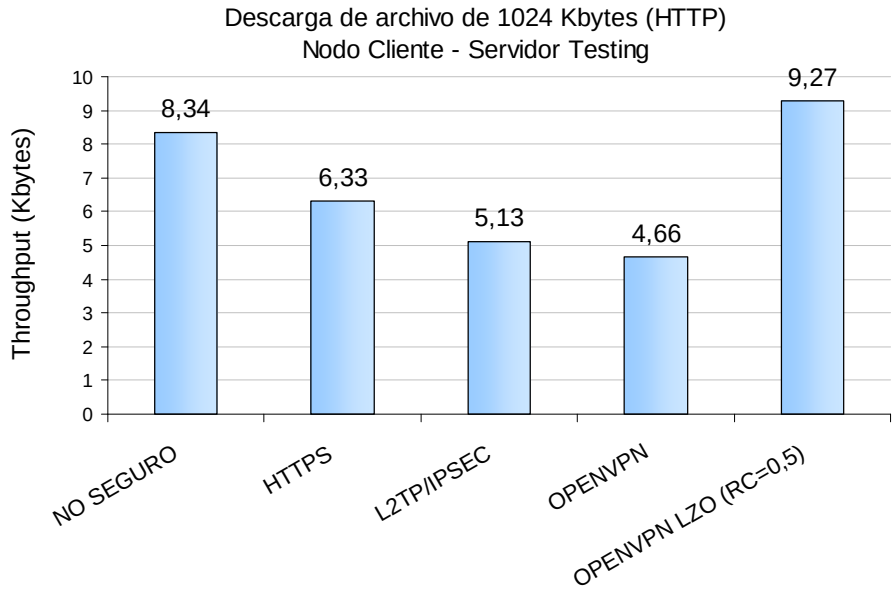

Gráfico 6-6: Throughput para tráfico HTTP generado por Busybox Wget

HTTPS introduce una mejora en el throughput respecto de HTTP sobre canales OpenVPN o L2TP/IPSec. Como se observa en el Gráfico 6-6, el uso de la compresión LZO con un *ratio*[14] promedio del 0.5 (50%) en el canal OpenVPN mejora el rendimiento del protocolo HTTP, superando incluso al del canal "NO" seguro.

6.1.5 Throughput FTP

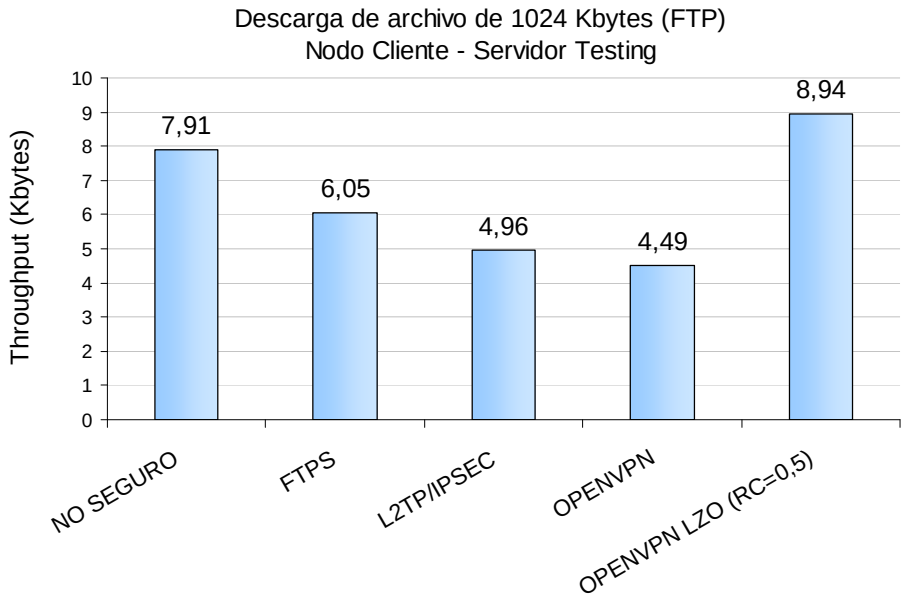

Gráfico 6-7: Throughput para tráfico FTP generado por Busybox Wget

Del Gráfico 6-7 se observa que el uso de la seguridad en el protocolo FTP presenta un comportamiento análogo al del protocolo HTTP, con pequeñas diferencias a favor de HTTP en el promedio de los valores medidos.

[14] La relación o *ratio* de compresión (RC) indica en qué proporción ha sido reducida la carga útil de datos. Por ejemplo: un RC=0,1 indica que la información a transmitir se redujo un 10%.

6.1.6 Comparativo de Throughput entre HTTP y FTP

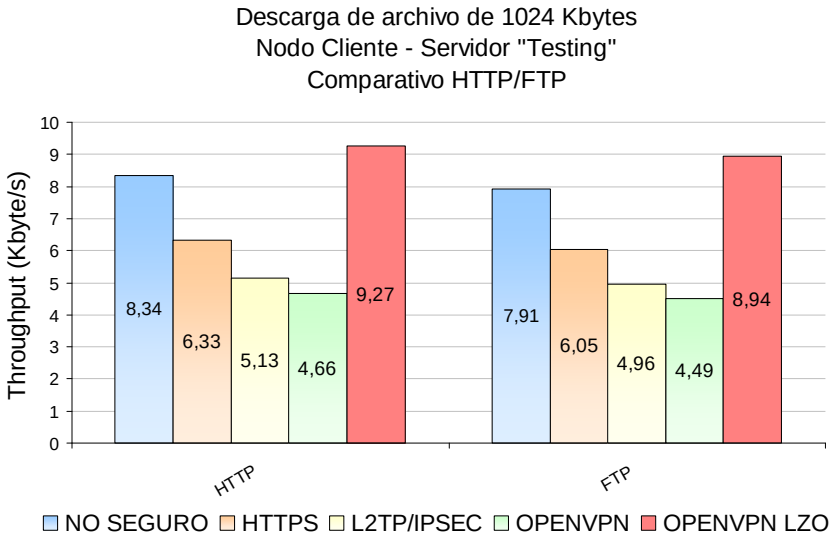

Gráfico 6-8: Comparativo de Throughput HTTP/FTP para tráfico generado por Busybox Wget

De los valores obtenidos en las mediciones que se vuelcan en el Gráfico 6-8, se observa que el protocolo HTTP presenta un mejor rendimiento respecto del protocolo FTP, para todas las configuraciones de canal implementadas.

6.1.7 Consumo de energía TCP

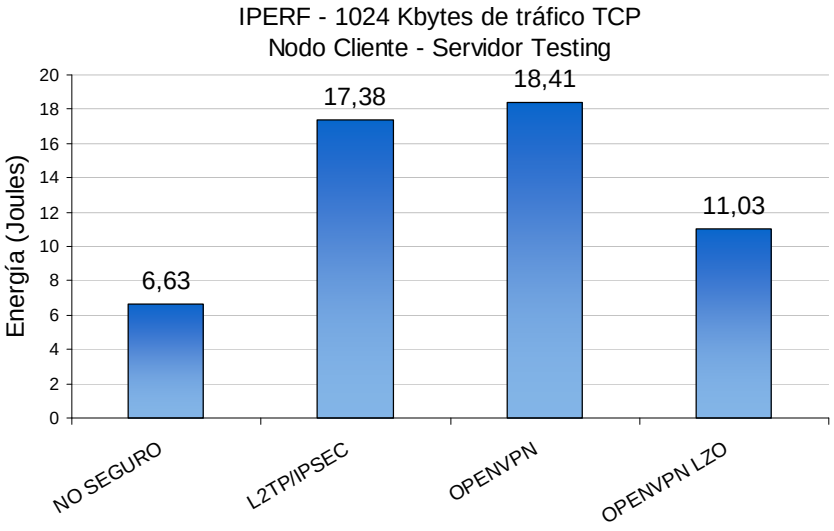

Gráfico 6-9: Consumo de energía para tráfico TCP generado por IPerf

Los resultados presentados en el Gráfico 6-9 muestran que el aseguramiento de un canal prácticamente triplica el consumo de energía en el cliente, el mismo puede disminuirse considerablemente si se habilita la compresión LZO.

6.1.8 Consumo de energía HTTP

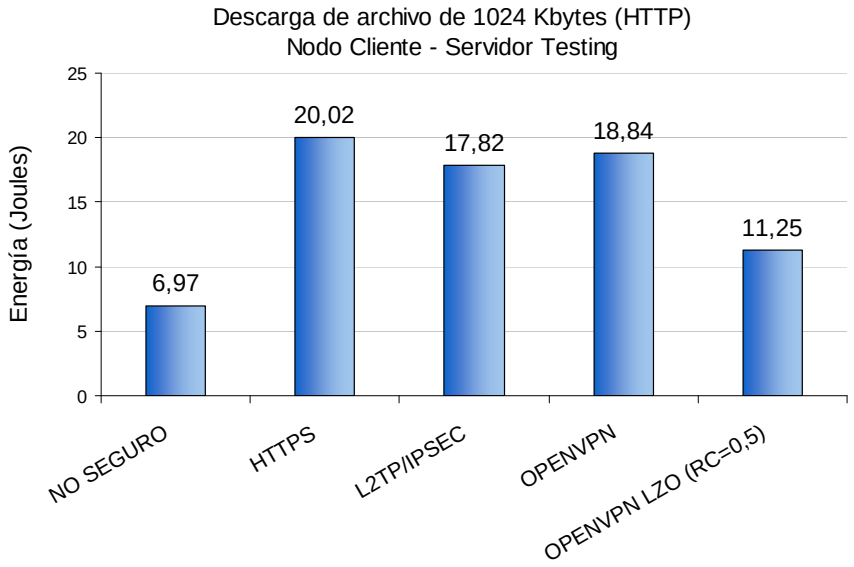

Gráfico 6-10: Consumo de energía para tráfico HTTP generado por Busybox Wget

El consumo de energía provocado por la descarga de un archivo de 1024Kbytes utilizando el protocolo HTTP aumenta significativamente si se realiza a través de un canal seguro. Como lo muestran los datos presentados en el Gráfico 6-10, este consumo puede ser reducido si se utiliza compresión LZO.

6.1.9 Consumo de energía FTP

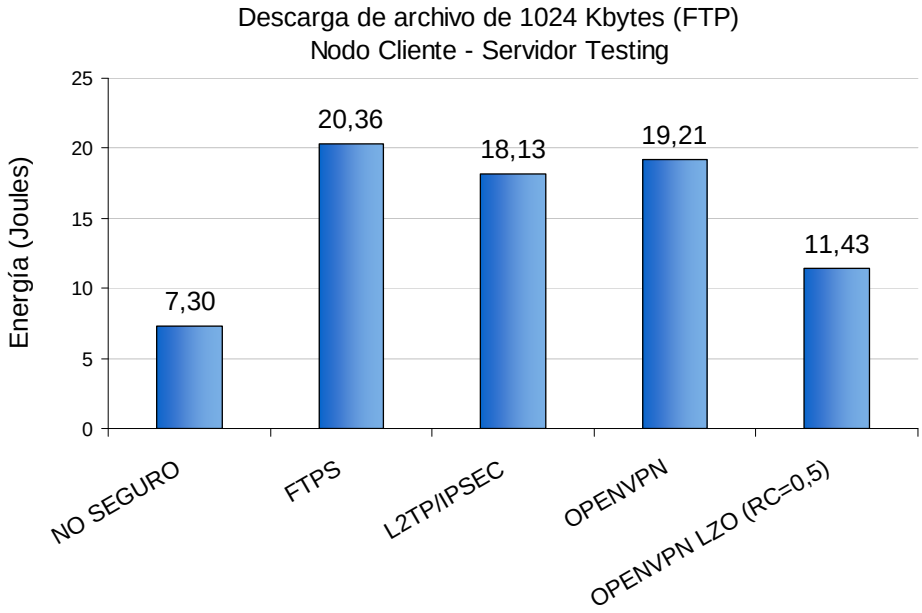

Gráfico 6-11: Consumo de energía para tráfico FTP generado por Busybox Wget

Los resultados ilustrados en el Gráfico 6-11, muestran que el uso del protocolo FTP sobre canales seguros provoca un importante consumo adicional de energía, que se puede reducir si se habilita la compresión LZO. Este mismo comportamiento se observó en el Gráfico 6-10 para el protocolo HTTP.

6.1.10 Comparativo de consumo de energía entre HTTP y FTP

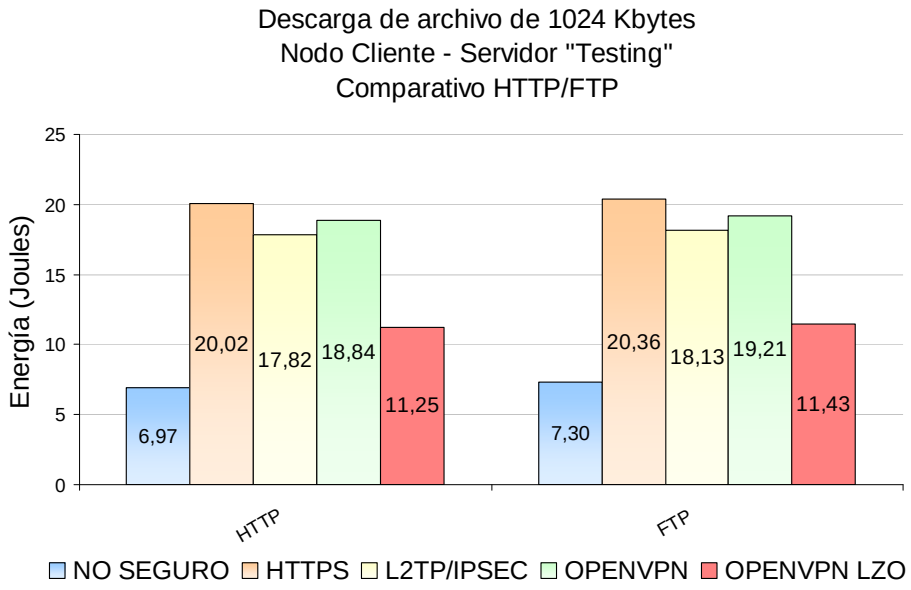

Gráfico 6-12: Comparativo de Throughput HTTP/FTP para tráfico generado por Busybox Wget

El uso del protocolo FTP en lugar de HTTP para la descarga de un archivo de 1024kbytes, provoca un mayor consumo de energía. Este consumo adicional es en promedio un 3% superior, como puede verse en el Gráfico 6-12.

6.1.11 Impacto de la seguridad en el rendimiento

Los resultados de las mediciones presentados en los gráficos de este apartado, ilustran al lector sobre como el uso de un canal seguro introduce un consumo de energía adicional y una reducción del rendimiento. Si no se utiliza compresión se observa un importante incremento en el consumo de energía y una disminución del throughput, y además que la variación en el consumo energético es proporcionalmente superior a la del throughput. El uso de la compresión mejora el throughput, superando incluso al alcanzado en un canal no seguro, con un aumento menos significativo del consumo de energía.

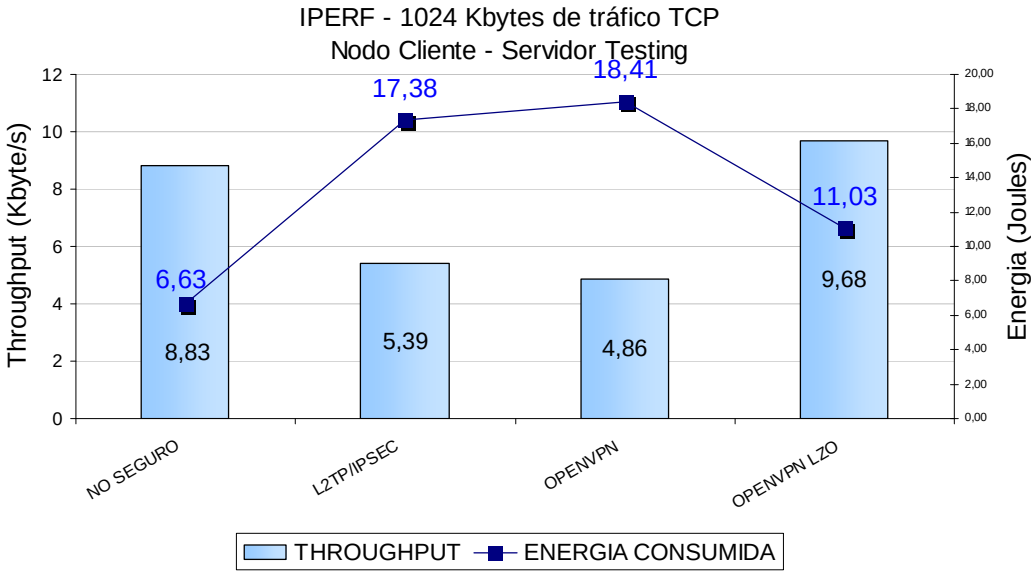

Gráfico 6-13: Throughput y consumo de energía para tráfico TCP generado por Iperf

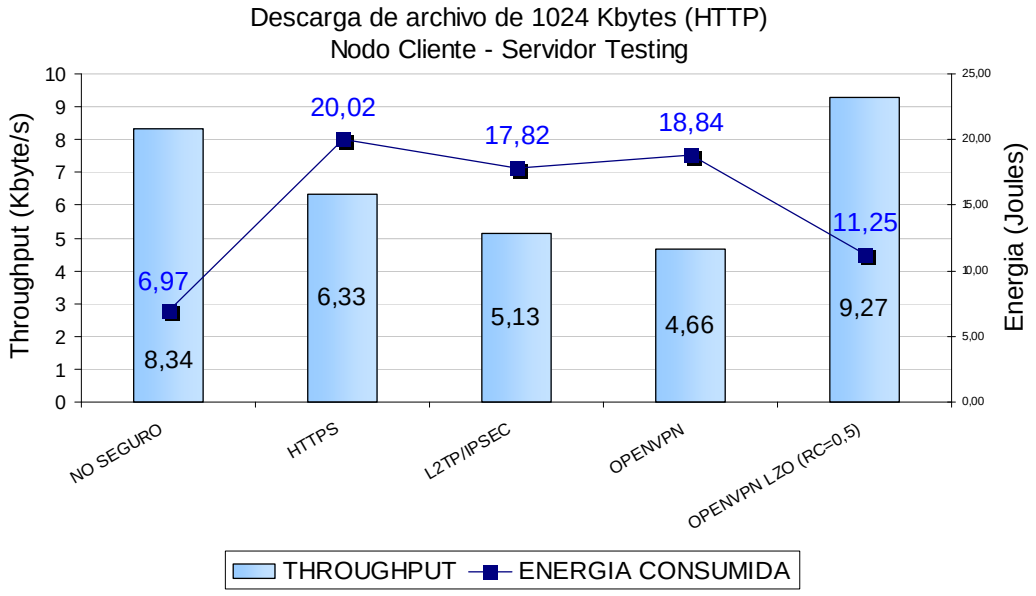

Gráfico 6-14: Throughput y consumo de energía para tráfico HTTP generado por Busybox Wget

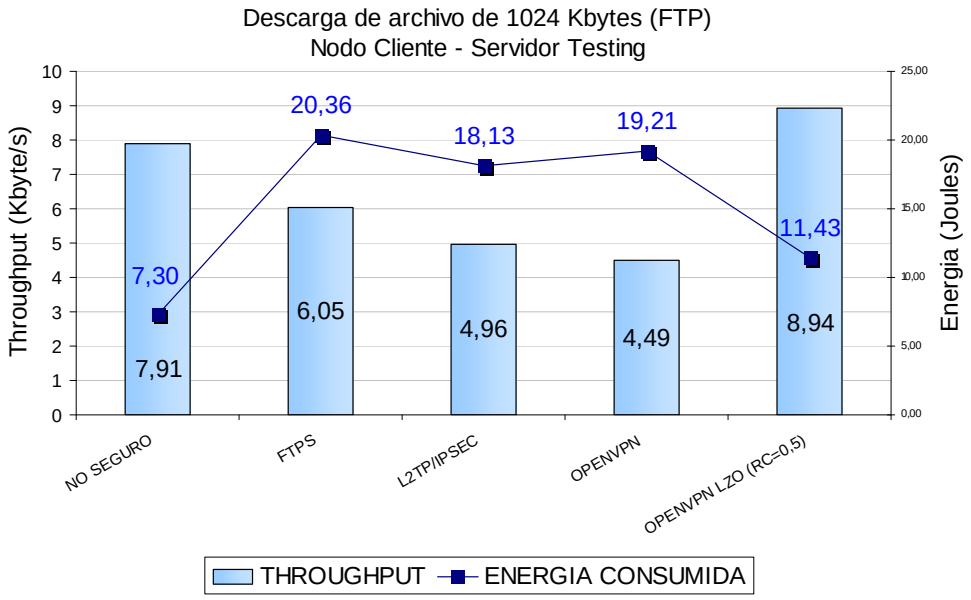

Gráfico 6-15: Throughput y consumo de energía para tráfico FTP generado por Busybox Wget

6.1.12 Impacto de la relación de compresión en el rendimiento

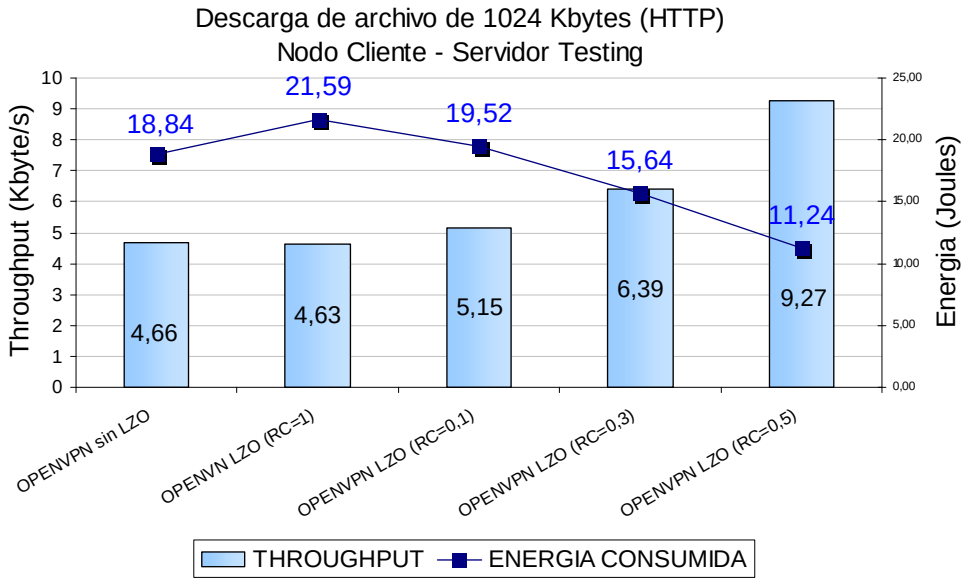

Gráfico 6-16: Throughput y consumo de energía para tráfico HTTP, sobre un canal OpenVPN con diferentes ratios de compresión.

En el Gráfico 6-16 se comparan canales OpenVPN sin compresión y con compresión utilizando diferentes *ratios*. Los resultados presentados evidencian que la disminución de la relación de compresión provoca un crecimiento del throughput y un descenso del consumo de energía. Se observa que, un ratio de compresión de 0.5 reduce el gasto energético en un 60% (de 18,84 a 11,24 Joules) e incrementa el throughput en un 100% (de

4,66 a 9,27 Kbytes/s). Mientras que, un ratio de compresión igual a 1 degrada el rendimiento ya que introduce un consumo adicional de energía (de 18,84 a 21,59 Joules) y una disminución del throughput (de 4,66 a 4,63 Kbytes/s).

6.1.13 Impacto de los algoritmos de encriptación, integridad y compresión en el rendimiento

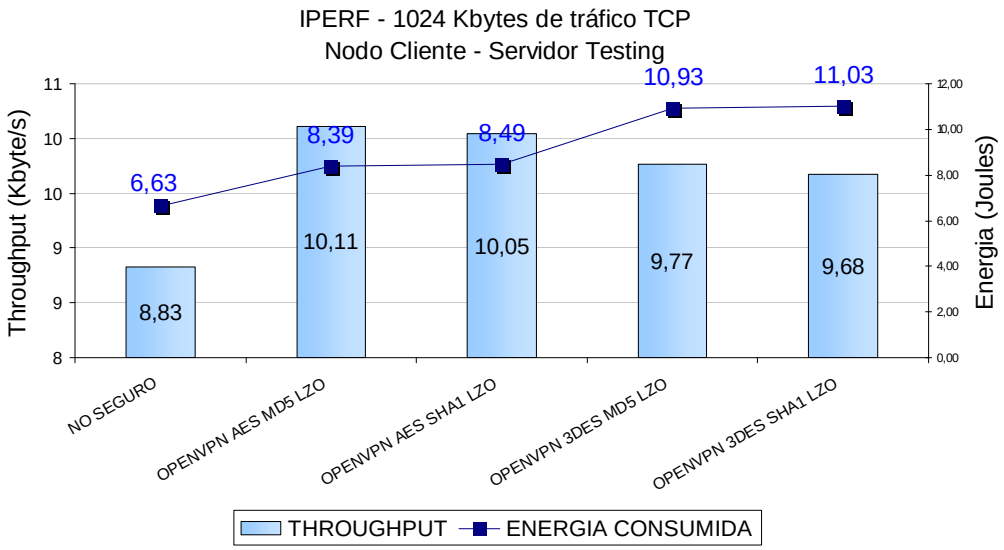

Gráfico 6-17: Throughput y consumo de energía para tráfico TCP, utilizando diferentes algoritmos de encriptación y cifrado

En el Gráfico 6-17 se compara el rendimiento de un canal no seguro respecto de canales seguros configurados utilizando compresión LZO y diferentes algoritmos para el cifrado y la integridad. Se destaca la mejora del throughput debida al uso de LZO, haciendo que el rendimiento de un canal seguro sea superior al de un canal no seguro en todos los casos. También se observa que la opción con el nivel más elevado de seguridad (ESP – SHA-1 – 3DES) es la que peor rendimiento consigue en términos de throughput y consumo de energía.

6.2 Conclusiones

La seguridad implica un consumo adicional de recursos que puede variar, dependiendo de los protocolos y algoritmos que se elijan para el establecimiento del canal seguro, en un amplio rango que en promedio es del orden de 35%.

El uso de un canal "seguro" introduce una disminución en el rendimiento del ancho de banda y en algunos casos duplica el consumo de energía, como puede verse de los datos obtenidos en este trabajo y que se muestran en los gráficos del apartado anterior.

La elección de un nivel de seguridad dependerá de la disponibilidad de recursos: cobertura de la red celular, ancho de banda y energía; existentes en la zona de despliegue de la MANET.

La compresión es una opción a considerar siempre y cuando existan limitaciones de energía en zona de despliegue, esta afirmación se sustenta en los resultados presentados, en estos se pone en evidencia que el incremento en el consumo de energía introducido por la compresión es proporcionalmente bajo en relación a la mejora que se consigue en el rendimiento (throughput y latencia).

La relación de compresión ("*ratio*") es una variable que debe ser analizada antes de habilitar la compresión. El compromiso entre seguridad, gasto energético y rendimiento logra un equilibrio utilizando compresión con una relación que este por encima del 30% (RC=0.3).

Observamos, de los gráficos presentados en el apartado anterior, que el canal L2TP/IPSec consigue un mejor rendimiento y un menor consumo de energía comparado con el canal OpenVPN SSL/TLS sin compresión. Mientras que, el canal OpenVPN proporciona un mejor nivel de seguridad respecto de L2TP/IPSec, pero a su vez introduce un mayor overhead y tráfico de control.

OpenVPN es más flexible y simple de configurar, proporciona la posibilidad de elegir entre varios algoritmos de cifrado e integridad, lo cual permite bajar el nivel de seguridad en beneficio de un mejor rendimiento.

L2TP/IPSec presenta algunas desventajas que se deben tomar en cuenta antes de su elección, algunas de ellas son: mayor dificultad en la configuración, gran parte del protocolo IPsec reside en el kernel del SO lo cual hace que en muchas implementaciones todavía no este soportada la compresión y que la configuración de los algoritmos de cifrado e integridad sea muy compleja.

Respecto al protocolo HTTPS, si analizamos la latencia y el throughput sobre HTTPS ambos tienen un mejor rendimiento comparado con HTTP sobre OpenVPN o L2TP/IPSEC, aunque también presenta un mayor consumo de energía, esto es debido a que OpenVPN y L2TP/IPSEC realizan un solo *handshake* para el establecimiento del canal, luego todas las

transferencias se llevan a cabo por el mismo sin necesidad de un nuevo handshake, mientras que HTTPS efectúa un handshake por cada transferencia.

El mismo análisis se aplica también con referencia a la comparación entre FTPS y FTP sobre OpenVPN o L2TP/IPSEC.

En relación al rendimiento de los protocolos HTTP y FTP, en todas las pruebas realizadas HTTP mostró un mejor rendimiento promedio que FTP, incluyendo mejor latencia (10% menos), Throughput superior (5% más) y menor consumo de energía (3% menos). Esto se debe a que HTTP envía una sola solicitud y recibe el archivo como respuesta, mientras que FTP utiliza un flujo de datos de solicitud/respuesta que introducen retardos adicionales, y además a que FTP utiliza un canal para controlar el estado de la transferencia y otro para realizar la transferencia de los archivos, mientras que HTTP no requiere del canal de control.

A continuación se exponen los aspectos a tener en cuenta para la elección del nivel de seguridad teniendo en cuenta el equilibrio entre seguridad y consumo de recursos:

- Utilizar el protocolo HTTP en lugar de FTP
- Utilizar OpenVPN con compresión LZO, siempre y cuando la relación de compresión sea superior al 30%.
- Si se va utilizar solamente HTTP y lo que se busca optimizar el rendimiento la opción a elegir es HTTPS.
- Si se va utilizar solamente HTTP y se busca un menor consumo de energía la mejor opción es HTTP sobre OpenVPN o L2TP/IPSEC.
- Si además de HTTP se van a utilizar otros protocolos que requieran transporte TCP y cuando la prioridad sea el rendimiento (throughput, latencia y consumo de energía), la mejor alternativa es L2TP/IPSEC.
- Si además de HTTP se van a utilizar otros protocolos que requieran transporte TCP y cuando la prioridad es la seguridad, se debe asegurar el canal utilizando OpenVPN.
- El canal NO seguro es una opción a considerar cuando el ancho de banda y/o la energía sean muy limitados, ya que el uso de un canal seguro puede ocasionar que algunas aplicaciones dejen de funcionar correctamente por falta de ancho de banda o que la energía almacenada en la batería del nodo cliente se consuma muy rápidamente.

Para finalizar y con base en los niveles de rendimiento y seguridad alcanzados durante el trabajo experimental realizado, se puede afirmar que "*el despliegue de MANETs y su integración segura a redes de infraestructura, constituye una alternativa tecnológica que posibilita el acceso seguro a información digital, desde zonas remotas que disponen de cobertura de red celular reducida y se encuentran fuera del alcance de los centros de distribución de energía*". La propuesta presentada en esta tesis es aplicable a zonas rurales aisladas del país que tengan características similares a las del escenario de estudio.

Apéndice 1: Contribuciones

Publicaciones

Los resultados obtenidos durante el desarrollo del trabajo de investigación asociado a esta tesis, fueron presentados en las siguientes publicaciones con referato:

1. **"Integración Segura de MANETs con Limitaciones de Energía a Redes de Infraestructura" [33].** CACIC 2011, La Plata - Buenos Aires - Argentina. (http://sedici.unlp.edu.ar/handle/10915/18771).

   En este trabajo, se realizo el estudio de un caso de integración de una MANET Bluetooth *indoor* a una red de infraestructura, sin considerar condiciones externas como distancia, interferencias y otras. El punto de acceso a la red se implementó utilizando las características de enrutamiento de Linux y habilitando la pila de protocolos BlueZ. Se efectuaron mediciones extremo a extremo sobre un canal "no seguro" y luego sobre un canal "seguro", el aseguramiento del canal se implemento utilizando diferentes configuraciones del protocolo IPSec. Entre los resultados se presentan gráficos comparativos de consumo de energía entre las diferentes configuraciones de seguridad.

2. **"Integración Segura de MANETs, desplegadas en zonas de recursos limitados, a Redes de Infraestructura" [131].** CACIC 2012, Bahía Blanca - Buenos Aires - Argentina. (http://sedici.unlp.edu.ar/handle/10915/23762).

   Continuando con esta línea de investigación, esta vez se trabajo sobre un escenario de pruebas *outdoor* afectado por factores externos que disminuyen el rendimiento e incrementan el consumo de recursos en los nodos de la red ad hoc. En el desarrollo de la publicación, se fundamenta la elección de Bluetooth como tecnología de soporte para la formación de la MANET remota y de GSM/GPRS para la integración de la misma a la red de infraestructura. Entre los resultados se presentan gráficos que muestran el consumo de energía para cada configuración de canal y la distribución del consumo entre los siguientes ítems: Establecimiento de sesión, encriptación, autenticación y transmisión.

   La exposición de este trabajo resulto distinguida y premiada como "mejor exposición" en el Workshop de Arquitectura, Redes y Sistemas Operativos (WARSO).

3. **"M-learning en zonas de recursos limitados" [132]**. TE&ET 2013, Santiago del Estero - Argentina. (http://sedici.unlp.edu.ar/handle/10915/27585)

   En esta publicación, se describe una experiencia del uso de MANETs en zonas rurales de recursos limitados (energía y ancho de banda). En el trabajo de campo realizado, se desplegaron MANETs de bajo consumo en escuelas rurales, con la finalidad de facilitar a docentes y alumnos el acceso a contenidos m-learning instalados en un servidor de infraestructura. Se consiguió mantener el rendimiento de la MANET dentro niveles aceptables de eficiencia y seguridad, sin comprometer los recursos, lo que posibilito un funcionamiento correcto de las estrategias de m-learning en este tipo de zonas. Entre las conclusiones de esta publicación se destaca la siguiente: "El uso de las MANETs es

efectivo y eficiente para el desarrollo de experiencias de m-learning en zonas de recursos energéticos limitados".

4. **"Caso de estudio de comunicaciones seguras sobre redes móviles ad hoc" [133]**. CACIC 2013, Mar del Plata - Buenos Aires – Argentina. (http://sedici.unlp.edu.ar/handle/10915/31244).

   En este trabajo se presenta un escenario de estudio similar al utilizado en esta tesis. Inicialmente se brindan detalles de implementación del escenario, luego se describen las pruebas y mediciones realizadas sobre el mismo, y finalmente se publican los principales resultados con sus correspondientes discusiones, así como también las conclusiones más relevantes que surgieron de la investigación realizada.

A continuación se mencionan otras publicaciones con referato generadas por esta tesis:

- "Integración Segura de MANETs a Redes de Infraestructura" [134]. Jornadas científicas interdisciplinarias 2011, Orán - Salta - Argentina.
- "Enseñanza de Redes Móviles Ad Hoc basada en simulación" [135]. Jornadas Nacionales de TIC e Innovación en el Aula 2011, La Plata – Argentina.
- "Despliegue de MANETs para M-learning en zonas de recursos limitados" [136]. WICC 2013, Entre Rios – Argentina.

Exposiciones en congresos y jornadas

- Exposición del trabajo "Integración Segura de MANETs a Redes de Infraestructura" [6], presentado en las "V jornadas científicas interdisciplinarias" de la Sede regional Orán de la UNSa. Expositor: Sergio H. Rocabado Moreno. Orán - Salta - Argentina. Noviembre de 2011.
- Exposición del trabajo "Integración Segura de MANETs con Limitaciones de Energía a Redes de Infraestructura" [1], presentado en el "XVII Congreso Argentino de Ciencias de la Computación", CACIC 2011. Expositor: Sergio H. Rocabado Moreno. La Plata – Argentina. Octubre de 2011.
- Exposición del trabajo "Integración Segura de MANETs, desplegadas en zonas de recursos limitados, a Redes de Infraestructura" [131]. Presentado en el "XVIII Congreso Argentino de Ciencias de la Computación", CACIC 2012. Expositor: Sergio H. Rocabado Moreno. Bahía Blanca - Buenos Aires – Argentina. Octubre de 2012.
- Exposición del trabajo "Despliegue de MANETs para M-learning en zonas de recursos limitados" [136]. Presentado en el "XV Workshop de Investigadores en Ciencias de la Computación", WICC 2013. Expositor: Sergio H. Rocabado Moreno. Entre Rios – Argentina. Abril de 2013.
- Exposición del trabajo "M-learning en zonas de recursos limitados" [132], presentado en el "VIII Congreso de Tecnología en Educación y Educación en Tecnología", TE&ET 2013. Expositores: Sergio H. Rocabado Moreno y Susana Herrera. Santiago del Estero - Argentina. Junio de 2013.
- Exposición del trabajo "Caso de estudio de comunicaciones seguras sobre redes móviles ad hoc" [133], presentado en el "XIX Congreso Argentino de Ciencias de la

Computación", CACIC 2013. Expositor: Sergio H. Rocabado Moreno. Mar del Plata – Buenos Aires - Argentina. Octubre de 2013.

Actividades de extensión

Dictado del curso de extensión "Redes Móviles Ad Hoc" avalado por la Facultad de Ciencias Exactas de la Universidad Nacional de Salta según resolución Res.CD.Cs.Ex. 680/2011 y EXP-EXA: 8749/2011. Director: Sergio H. Rocabado Moreno. Instructores: Sergio H. Rocabado Moreno y Ernesto Sanchez. Duración: 20 horas. Salta – Argentina. Febrero de 2012.

Transferencia de tecnología

Despliegue de MANETs en zonas rurales de recursos limitados

En [132] y [137] se presentan los resultados del trabajo experimental realizado en el departamento Pellegrini de la Provincia de Santiago del Estero, cuya ciudad cabecera es Nueva Esperanza. En este trabajo se utilizo una solución tecnológica, "el despliegue e integración de MANETs a redes de infraestructura", como alternativa para, "posibilitar el acceso a la información y al conocimiento a pobladores de zonas rurales aisladas que se encuentran fuera del alcance de los centros de distribución de energía y que disponen de cobertura celular muy reducida".

Se diseñaron estrategias para el despliegue de MANETs en este tipo de zonas, manteniendo el rendimiento de las MANETs dentro de niveles aceptables de eficiencia y sin comprometer recursos que son limitados en la zona de despliegue (energía y ancho de banda).

El trabajo de campo más relevante se desarrolló en la Escuela Rural N° 348 Narciso Vera, ubicada en Pozo Nuevo, a 20 km de Nueva Esperanza. El despliegue de una MANET y su integración a la red de infraestructura de la UNSE, hizo posible que profesores y alumnos de esta escuela accedan a objetos de aprendizaje, almacenados en un servidor de recursos m-learning, desde teléfonos celulares.

Proyecto PROCODAS

El proyecto "Aprendizaje mediado por dispositivos móviles en zonas rurales con recursos energéticos limitados", fue presentado en la Convocatoria de Proyectos de Tecnología para la Inclusión Social 2013 de Proyectos Complementarios en el área Hábitat Social, perteneciente al Programa Consejo de la Demanda de Actores Sociales (PROCODAS) del MINCYT. El objetivo principal de este proyecto es la alfabetización de alumnos de escuelas rurales, ubicadas en zonas de recursos limitados, a través de medios digitales y estrategias de enseñanza-aprendizaje. Para ello, se propone el despliegue de MANETs de bajo consumo (en escuelas rurales) y equipamiento basado en energía solar para la recarga de los dispositivos móviles que forman parte la MANET. El proyecto se encuentra admitido y en proceso de evaluación.

Apéndice 2: Becas y Premios

Beca PROFITE

Para la conclusión de este trabajo, el autor fue seleccionado y accedió a una beca del programa de becas para la finalización de tesis de posgrado para docentes de universidades nacionales (PROFITE). Resolución 2559 de la SPU.
URL: http://portales.educacion.gov.ar/spu/profite/

Premio a la mejor exposición

La exposición del trabajo "Integración Segura de MANETs, desplegadas en zonas de recursos limitados, a Redes de Infraestructura" [131], realizada por el autor de esta tesis, resulto distinguida y premiada como mejor exposición del Workshop de Arquitectura, Redes y Sistemas Operativos del CACIC 2012. Bahía Blanca, Argentina. Octubre de 2012.
URL: http://cs.uns.edu.ar/cacic2012/index.php/es/component/content/article/30

Apéndice 3: Tecnologías celulares

Generaciones de la Telefonía Celular

Las dos principales familias de tecnologías a nivel mundial son CDMA y TDMA [138]. Cada familia fue evolucionando en el tiempo y esas evoluciones son las denominadas generaciones: segunda generación (2G), tercera generación (3G) y cuarta generación (4G) (Figura A - 1).

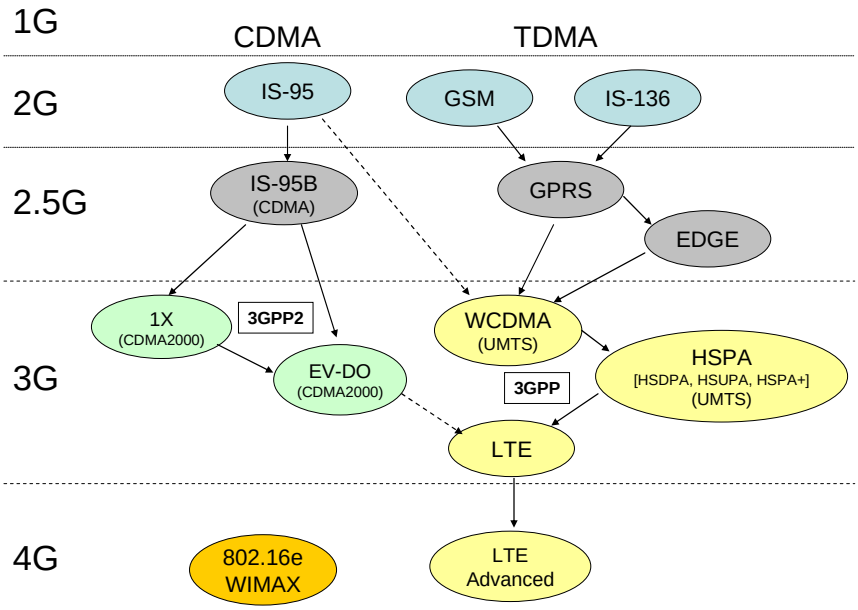

Figura A - 1: Diagrama de evolución de las tecnologías de celular
Fuente: Elaboración propia

- Primera generación (1G)

La primera generación fue introducida a finales de la década de 1970, y comprende los primeros celulares analógicos para transmisión de voz en sistemas como el AMPS. Algunos celulares de primera generación podían conectarse a una PC por una interfaz serial y configurarse como si fueran MODEMs (*dial-up*), de esta manera proporcionaban servicios de transmisión de datos con tasas de transferencia muy reducidas (<9kbps).

- Segunda generación (2G)

La segunda generación se refiere a los sistemas de voz digitales (GSM, IS-95) de la década de 1990, que reemplazaron a los teléfonos analógicos basados en interfaces TDMA y CDMA.

Los sistemas digitales como el GSM o IS-95 poseen extensiones para brindar servicios de datos orientados a paquetes sin requerir el establecimiento de una conexión (*connectionless*): GPRS y EDGE para GSM e IS-95B para IS-95. Estas extensiones se denominan comúnmente tecnologías 2.5G y posibilitan el acceso a Internet con velocidades reducidas de descarga (<200 Kbps).

- Tercera generación (3G)

La tercera generación de redes celulares ofrece servicios de datos orientados a paquetes con altas tasas de transmisión, el objetivo inicial establecido por el IMT-2000 (ITU [139]) fue de 2 Mbit/s en ambientes "*indoor*" de baja movilidad.

Las principales tecnologías de 3ra generación son: UMTS y CDMA 2000. Siendo UMTS (WCDMA/HSPA) la evolución de las operadoras que utilizan el estándar GSM y se encuentra estandarizado por 3GPP [140], mientras CDMA 2000 (EV-DO) es la evolución de CDMA y esta estandarizado por 3GPP2 [141]

WCDMA

Es el estándar 3G para redes GSM, proporciona una mayor eficiencia espectral lo que permite proporcionar mayores tasas de transmisión (hasta 2 Mbps), además posibilita el transporte de diferentes tipos de servicios (voz y datos) por el mismo canal y a diferentes tasas. Si bien esta tecnología esta orientada a elevadas velocidades de datos, también soporta aplicaciones simples como una llamada de voz, o el envío de un SMS.

WCDMA es un sistema CDMA. CDMA significa en inglés “acceso múltiple por división de código”, lo que significa que el canal de frecuencia disponible se divide en diferentes secuencias de códigos que son multiplexadas por las señales de usuario de los abonados individuales. Todos los abonados trasmiten sobre la misma frecuencia y al mismo tiempo [140].

HSPA

Evolución de la tecnología WCDMA, presenta las siguientes revisiones:

- HSDPA (rel5) - Mejora la velocidad para el enlace descendente (Downlink).
- HSDUPA (rel6) - Proporciona mayor velocidad para el enlace ascendente (Uplink), y la posiciona muy cercana a la velocidad del enlace descendente (Downlink).
- HSPA+ (rel7) - Aumenta significativamente la capacidad del HSPA, dobla la capacidad de datos y aumenta en hasta tres veces la capacidad de voz a través de VoIP. Mejora aun más las velocidades para el canal ascendente y descendente.

HSPA+ Se transformo en la principal tecnología de banda ancha móvil en el mundo con un 43,8% de las redes 3G ya actualizadas en el segundo cuatrimestre de 2012. Estimaciones realizadas en indican que más del 80% de las redes 3G serán HSPA+ en menos de 3 años [4].

CDMA 2000 (EV-DO)

EV-DO es el estándar 3G para IS-95, en redes CDMA2000 es significativamente más rápido que EDGE en redes GSM. Provee mayores tasas de transferencia compactando múltiples canales, 2.4 Mbit/s con Rev. 0 y hasta 3.1 Mbit/s con Rev. A. Aprovecha de una manera más eficaz el uso de la batería incrementando el tiempo de uso y de espera del celular, e incorpora servicios para aplicaciones que tienen requerimientos asimétricos de transmisión como intercambio de archivos, navegación web, entre otros.[141]

LTE

Se presenta como evolución para las tecnologías EV-DO y HSPA. Inicialmente, la tecnología UMB fue presentada como la evolución de EV-DO para CDMA, a fines de 2008 Qualcomm, principal patrocinador de UMB, anunció que ponía fin el desarrollo de la tecnología UMB, favoreciendo a LTE.

LTE aumenta la capacidad de datos en áreas Urbanas con densidad elevada brindando hasta 100Mbps para downlink y 50 Mbps para uplink, al mismo tiempo ofrece una importante reducción de la latencia (<10 ms). Inicialmente se utiliza solo para la transmisión de datos, aliviando de esta manera el tráfico de las redes actuales.

- Cuarta generación (4G)

La ITU [139] ha definido las especificaciones para que una tecnología sea denominada de cuarta generación, una determinada tecnología es considerada 4G cuando sea reconocida como un sistema IMT-Advanced(4G).
LTE-A y Mobile WiMAX 802.16e son las tecnologías aceptadas por ITU como 4G.

LTE-Advanced

En octubre de 2009, LTE-Advanced fue evaluada como una candidata a la tecnología 4G, y en octubre de 2010 ITU anunció oficialmente a LTE-Advanced como un sistema IMT-Advanced(4G). LTE-A proporciona mayores tasas de datos y reducción en la latencia y posee una mayor eficiencia espectral con un ancho de banda de hasta 100MHz. Fue proyectado para ofrecer tasas bajada (Downlink) de 100Mbps con el usuario en movimiento y 1Gbps con el usuario detenido. Posee también una tasa de subida (Uplink) de hasta 500Mbps.

WiMAX

Comprende una familia de estándares (802.16 de IEEE [142]) diseñados para cubrir las necesidades de las redes inalámbricas de área metropolitana (WMANs). El principal objetivo de WiMAX es el de proporcionar acceso a Internet de alta velocidad en un amplio rango de cobertura y con tasas de transferencia elevadas.

El grupo de trabajo del IEEE 802.16 desarrolló dos estándares para la tecnología WiMAX: 802.16 (2004) para suscriptores fijos y 802.16e (2005) para suscriptores móviles.

WiMAX Fijo (802.16)

- Extensión de banda ancha inalámbrica (punto a multipunto), permite conectar suscriptores (nodos remotos) a una estación base (BTS) perteneciente a la red de infraestructura cableada. Cada suscriptor (punto fijo inalámbrico) se conecta a la estación base utilizando una antena "fija" instalada en un lugar estratégico de su ubicación.
- No se requiere visibilidad directa entre los suscriptores y la estación base (BTS).
- Proporciona tasas de hasta 134 Mbps en un rango de 8 Km.

WiMAX móvil (802.16e):

- Evolución del acceso inalámbrico de banda ancha del IEEE 802-16. Combina servicios fijos y móviles en una arquitectura de red similar a un sistema celular. La estación base (BTS) puede comunicarse con nodos fijos (punto fijo inalámbrico) y nodos móviles (punto móvil inalámbrico).
- Proporciona tasas de hasta 75 Mbps en un rango de 3 Km.

Mientras que las tecnologías 3G se adaptan mejor a dispositivos móviles como teléfonos celulares o PDA´s, WiMAX fue concebido para ser utilizado en computadoras portátiles con requerimientos de Internet, como Notebooks o Netbooks. En la Tabla A - 1**¡Error! No se encuentra el origen de la referencia.** se presenta un sumario de las principales características de WiMAX fijo y WiMAX móvil, además se realiza una comparación con las tecnologías 3G.

Parameter	Fixed WiMAX	Mobile WiMAX	HSPA	1x EV-DO
Standards	IEEE 802.16(2004)	IEEE 802.16e (2005)	3GPP Release 6	3GPP2
Peak downlink	9.4Mbps in 3.5MHz with 3:1 DL-to-UL ratio TDD; 6.1Mbps with 1:1	46Mbpsa with 3:1 DL-to-UL ratio TDD; 32Mbps with 1:1	14.4Mbps using all 15 codes; 7.2Mbps with 10 codes	3.1Mbps; Rev. B will support 4.9Mbps
Peak uplink	3.3Mbps in 3.5MHz using 3:1 DL-to-UL ratio; 6.5Mbps with 1:1	7Mbps in 10MHz using 3:1 DL-to-UL ratio; 4Mbps using 1:1	1.4Mbps initially; 5.8Mbps later	1.8Mbps
Bandwidth	3.5MHz and 7MHz in 3.5GHz band; 10MHz in 5.8GHz band	3.5MHz, 7MHz, 5MHz, 10MHz, and 8.75MHz initially	5MHz	1.25MHz
Coverage (typical)	3–5 miles (4,8 – 8 Km)	< 2 miles (< 3,2 Km)	1–3 miles (1,6 – 4,8 Km)	1–3 miles (1,6 – 4,8 Km)
Mobility	Not applicable	Middle	High	High

Tabla A - 1: Tecnologías inalámbricas de banda ancha
Fuente: IEEE 802.16 [142], 3GPP [140] y 3GPP2 [141]

Evolución tecnológica por generación

A continuación se presentan tablas que contienen información técnica relacionada con las diferentes tecnologías:

Generación	**2G**			**3G**					**4G**
Tecnología	GSM	GPRS	EDGE	WCDMA (UMTS)	HSDPA (UMTS)	HSUPA	HSPA+	LTE	LTE-Advanced
Tasa de datos máx. teórica (Downlink)	14,4 Kbps	171,2 Kbps	473,6 Kbps	2,0 Mbps	7,2 Mbps	7,2/14,4 Mbps	21/42 Mbps	100 Mbps	1,0 Gbps
Tasa de datos máx. teórica (Uplink)	-	-	473,6 Kbps	474 Kbps	384 Kbps	5,76 Mbps	7,2/11,5 Mbps	50 Mbps	0,5 Gbps
Latencia (ms)	500	500	300	250	130	~ 70	~ 30	~ 10	< 5
Especificación (Releases)				Rel-99	Rel-5	Rel-6	Rel-7 y 8	Rel-8 y 9	Rel-10

Tabla A - 2: Evolución de la tecnología TDMA (GSM)
Fuente 3GPP [140] y GSM europe [143]

Generación	2 G	2,5 G	3 G		
Tecnología	CDMAONE (IS-95-A)	CDMA2000 1X	CDMA 1xEV-DO	CDMA 1xEV-DO Rev. A	CDMA 1xEV-DO Rev. B
Tasa de datos máx. teórica (downlink)	14,4 kbit/s	153,6 kbit/s	2,4 Mbit/s	3,1 Mbit/s	4,9 Mbit/s

Tabla A - 3: Evolución de la tecnología CDMA (IS-95)
Fuente 3GGP2 [141]

Resumen de la familia TDMA-GSM, de 2G a 4G

GPRS	**General Packet Radio Service** • Tasa máxima encontrada en la práctica: 26 a 40 kbit/s. • Disputa los mismos slots de tiempo ya existentes para el tráfico de voz, lo que es un limitante para que los operadores ofrezcan tasas mayores.
EDGE	**Enhanced Data rates for Global Evolution** • Puede aumentar en 3 veces la tasa de transmisión por la utilización de un nuevo esquema de modulación. • Tasa máxima encontrada en la práctica de 384Kbps y un promedio de 110 a 120 kbps en una red cargada.
WCDMA	**Wideband CDMA** • También conocido como CDMA DS (Direct Sequence). • WCDMA proporciona una mayor eficiencia espectral, lo que permite proporcionar mayores tasas de transmisión (hasta 2 Mbps). • Posibilita el transporte de diferentes tipos de servicios (voz y datos) por el mismo canal y a diferentes tasas.
HSPA	**High Speed Packet Service** • Evolución de la tecnología WCDMA, tiene a su vez diferentes revisiones: HSDPA (rel5), HSUPA (rel6) y HSPA+ (rel7).
HSDPA	**High Speed Downlink Packet Access** • Es un servicio de paquetes de datos, basado en el WCDMA, que optimiza la transmisión de datos hacia el teléfono celular (downlink o enlace de bajada). • Los operadores lanzaron HSDPA inicialmente con tasas de bajada (downlink) máximas de 3,6 Mbps, actualizándola posteriormente para los 7,2 Mbps ó 14,4 Mbps.
HSUPA	**High Speed Uplink Packet Access** • Primera evolución del WCDMA/HSDPA que optimiza la transmisión de datos en la dirección a la Estación Radio Base (enlace de subida, uplink). Esta mejora incluye mayor rendimiento, menor latencia y mayor eficiencia espectral.
HSPA+	**High Speed Packet Access Plus** • Evolución del HSPA que puede ofrecer tasas de datos de downlink de hasta 84 Mbps y Uplink de hasta 23 Mbps. • También conocida como HSPA Evolution y HSPA Evolved.
LTE	**Long Term Evolution** • Evolución de las tecnologías EV-DO y HSPA/HSPA+. • Ofrece altas tasas de datos (100Mbps para downlink y 50Mbps de para uplink) y reducción de la latencia (< 10ms).
LTE-A	**Long Term Evolution Advanced**

	• Evolución de LTE, forma parte de la cuarta generación. • Ofrece tasas de bajada (Downlink) de 100Mbps con el usuario en movimiento y 1Gbps con el usuario detenido. • Brinda tasas de subida (Uplink) de hasta 500Mbps.

Tabla A - 4: Familia de tecnologías celulares TDMA-GSM, de 2G a 4G
Fuente 3GPP [140]

Sitios Web con información sobre tecnología celular

3GPP [140]	**3rd Generation Partnership Project** Responsable por la estandarización del UMTS. http://www.3gpp.org
ETSI [144]	**European Telecommunications Standards Institute** Desarrolló las normas para el GSM. Es posible hacer descarga gratuita de las normas. http://www.etsi.org
GSM World [143]	**GSM World** Sitio Web de la asociación mundial del GSM. Estadísticas del GSM en el mundo. http://www.gsmworld.com
3GPP2 [141]	**3rd Generation Partnership Project 2** Responsable por la estandarización del CDMA 2000. http://www.3gpp2.org/Public_html/specs
TIA [145]	**Telecommunications Industry Association** Desarrolló las normas para el CDMA. http://www.tiaonline.org
ITU[139]	**International Telecommunication Union** Define las especificaciones para las tecnologías de cuarta generación (4G). http://www.itu.int

Tabla A - 5: Sitios Web con información sobre tecnología celular

Apéndice 4: Logs de Powertutor

A continuación se presenta un ejemplo de archivo de LOG generado por la aplicación Powertutor y mas adelante se realiza una interpretación de su contenido.

phone-service in-service
phone-network GPRS
signal 75
phone-call idle
data connected
battery-change 2 91/100 4235453
batt_current 4.66E-4
batt_temp 45.300000000000004
setting_brightness automatic
setting_screen_timeout 30000
time 1383186995944
localtime_offset -10800000
model dream
associate 10162 edu.umich.PowerTutor@15

begin 0
total-power 600
meminfo 852416 42776 2508 97856
LCD 600
LCD-brightness 130
LCD-screen-on true
LCD-10162 600
Wifi 0
Wifi-on false
GPS 0
GPS-state-times 1.0 0.0 0.0
GPS-sattelites 0
Audio 0
Audio-on false
associate 10001 android.uid.sec.activitywidget:10001@1
associate 10004 com.sec.pcw@3183
associate 10009 com.android.bluetooth@16
associate 10014 com.samsung.map@16
associate 10015 com.google.uid.shared:10019@16
associate 10016 com.sec.android.app.clockpackage@1
associate 10017 com.samsung.android.providers.context@1
associate 10018 android.media:10032@16
associate 10019 com.sec.android.fotaclient@1
associate 10020 com.google.android.apps.uploader@40000
associate 10021 com.google.android.apps.plus@330400495
associate 10022 com.sec.spp.push@12
associate 10023 com.osp.app.signin@140363
associate 10024 com.sec.phone@1
associate 10124 com.android.smspush@16
associate 10125 com.sec.android.app.ics.networkinfo@105
associate 10126 com.ideashower.readitlater.pro@252
associate 10130 com.magicandroidapps.iperf@206
---
begin 1
total-power 643
LCD 600
LCD-brightness 130
LCD-screen-on true
LCD-10162 600
CPU 33

CPU-sys 10
CPU-usr 0
CPU-freq 200.0
CPU-0 13
CPU-1000 37
CPU-1001 1
CPU-1001 0
CPU-1004 0
CPU-1009 0
CPU-10001 0
CPU-10004 0
CPU-10009 0
CPU-10014 0
CPU-10015 0
CPU-10016 0
CPU-10017 0
CPU-10018 0
CPU-10019 0
CPU-10020 0
CPU-10021 0
CPU-10022 0
CPU-10023 0
CPU-10024 0
CPU-10025 0
CPU-10026 0
CPU-10030 0
Wifi 0
Wifi-on false
2G 10
2G-on true
2G-uplinkBytes 0
2G-downlinkBytes 0
2G-packets 0
2G-state IDLE
2G-oper Personal
GPS 0
GPS-state-times 1.0 0.0 0.0
GPS-sattelites 0
Audio 0
Audio-on false
……..
……..
begin 5
total-power 638
LCD 600
LCD-brightness 130
LCD-screen-on true
LCD-10162 600
CPU 29
CPU-sys 7
CPU-usr 0
CPU-freq 400.0
CPU-0 1
CPU-1001 0
CPU-1004 0
CPU-1009 0
CPU-10001 0
CPU-10004 0
CPU-10009 0
CPU-10014 0
CPU-10015 0
CPU-10016 0

CPU-10017 0
CPU-10018 0
CPU-10019 0
CPU-10020 0
CPU-10021 0
CPU-10022 0
CPU-10023 0
CPU-10024 0
CPU-10025 0
CPU-10026 0
CPU-10030 21
Wifi 0
Wifi-on false
2G 10
2G-on true
2G-uplinkBytes 0
2G-downlinkBytes 0
2G-packets 0
2G-state IDLE
2G-oper Personal
GPS 0
GPS-state-times 1.0 0.0 0.0
GPS-sattelites 0
Audio 0
Audio-on false
……..
begin 6
total-power 616
LCD 600
LCD-brightness 130
LCD-screen-on true
LCD-10162 600
CPU 6
CPU-sys 2
CPU-usr 0
CPU-freq 200.0
CPU-0 2
CPU-1001 0
CPU-1004 0
CPU-1009 0
CPU-10001 0
CPU-10004 0
CPU-10009 0
CPU-10014 0
CPU-10015 0
CPU-10016 0
CPU-10017 0
CPU-10018 0
CPU-10019 0
CPU-10020 0
CPU-10021 0
CPU-10022 0
CPU-10023 0
CPU-10024 0
CPU-10025 0
CPU-10026 0
CPU-10030 74
Wifi 0
Wifi-on false
2G 10
2G-on true
2G-uplinkBytes 0

2G-downlinkBytes 0
2G-packets 0
2G-state IDLE
2G-oper Personal
GPS 0
GPS-state-times 1.0 0.0 0.0
GPS-sattelites 0
Audio 0
Audio-on false

Interpretación del contenido del archivo LOG

Al principio del log se puede encuentrar información relacionada con la red celular, como ser la tecnología utilizada (phone-network), la intensidad de la señal (signal), el estado de la conexión de datos (data), y otros.

En el instante 0 (begin 0) se inicializa una marca de tiempo para diferenciar las muestras, esto se debe a que los resultados de diferentes pruebas se pueden almacenar en un solo archivo de salida.

time 1377634014953

También en este instante se identifican a las aplicaciones con un numero único (UID), el consumo de energía de cada aplicación durante el resto de la traza se etiqueta con el UID. Por ejemplo, en el archivo log examinado se asocia el proceso Iperf al numero 10130 (UID)

associate 10130 com.magicandroidapps.iperf@206

Las muestras se toman una por segundo comenzando de uno. El número después de begin representa el tiempo transcurrido en segundos desde el inicio de la traza, algunos ejemplos:

begin 1 (1 seg)
….
….
begin 5 (transcurrieron 5 segundos a partir del inicio de la prueba)
LCD-10030 629 (indica que el uso de LCD por parte del proceso 10030 consumió 629mW en el quinto segundo)
……..
begin 6 (transcurrieron 6 segundos a partir del inicio de la prueba)
CPU-10030 74 (indica que el uso de CPU por parte del proceso 10030 consumió 74mW en el sexto segundo)

Joule es watt por unidad de tiempo, entonces los 74 mW utilizados en el sexto segundo, equivalen a 74 mj de energía (74 mW x seg = 74 mj).

El consumo total de energía del dispositivo es asociado al componente de hardware sin utilizar ningún número de proceso (UID). Ej:

begin 5 (transcurrieron 5 segundos a partir del inicio de la prueba)

CPU 29 (indica que el consumo total de energía de la CPU en el quinto segundo es de 29mW).

Este valor es el resultante de la suma de los consumos de CPU de los procesos:

begin 5
CPU 29 (7+1+21=29)
CPU-sys 7
CPU-0 1
CPU-10030 21

Ejemplo de estimación de la energía consumida por proceso y componente de hardware

A continuación se enumeran los pasos necesarios para calcular la estimación de la energía consumida por el uso de CPU por parte de la aplicación Iperf.

1. Identificar el UID asociado a la aplicación Iperf (10130).

   associate 10130 com.magicandroidapps.iperf@206

2. Recorrer el archivo de log para identificar "todas" las muestras que informen consumos de energía por uso de CPU del proceso 10030 (CPU-10030).

   begin 1
   CPU-10030 0

   begin 2
   CPU-10030 0

   begin 3
   CPU-10030 20

   begin 4
   CPU-10030 0

   begin 5
   CPU-10030 21

   begin 6
   CPU-10030 74

3. Sumar los consumos de las muestras

   Total= 0+0+20+21+74= 115 mj
   (En 6 segundos, el uso de CPU por parte del proceso asociado a la aplicación Iperf consumió 115 mj de energía)

Del mismo modo se pueden estimar los consumos de energía de otros componentes de hardware.

Lista de Figuras

Lista de Tablas

Lista de Acrónimos y Abreviaciones

2G	Second generation
3G	Third generation
3GPP	3rd Generation Partnership Project
3GPP2	3rd Generation Partnership Project 2
AKM	Autonomous Key Management
AMPS	Advanced Mobile Phone System
AODV	Ad Hoc On Demand Distance Vector routing protocol
ARAN	A Secure Routing Protocol for Ad Hoc Wireless Networks
AWDS	Ad Hoc Wireless Distribution Service.
BFTR	Best-Effort Fault Tolerant Routing
BNEP	Bluetooth Networking Encapsulation Protocol
CA	Certification
CASHnet	Cooperation and Accounting Strategy for Hybrid Networks

CC	Common Criteria
CDMA	Code Division Multiple Access
CONFIDANT	Cooperation of nodes, Fairness in dynamic ad-hoc networks
CORE	A COllaborative REputation mechanism to enforce node cooperation in Mobile Ad Hoc Networks
CPU	Central Processing Unit
CRL	Certificate Revocation List
DES	Data Encryption Standard
DICTATE	DIstributed CerTification Authority with probabilisTic frEshness for Ad Hoc Networks
DMZ	Demilitarized Zone
DoS	Denial of Service
DPA	Differential Power Analisys
DSDV	Highly Dynamic Destination Sequenced Distance Vector Routing Protocol
DSDV	Destination-Sequenced Distance Vector protocol
DSR	Dynamic Source Routing
DYMO	Dynamic Manet On-Demand Routing
EDGE	Enhanced Data rates for GSM Evolution
EVDO	Evolution Data Only/Optimized
FIPS	Federal Information Processing Standard
FLAC	FLow-based route Access Control
FTP	File Transfer Protocol
FTPS	Implicit FTP over TLS/SSL
GGSN	Gateway GPRS Support Node
GMSK	Gaussian Minimum Shift Keying
GPRS	General Packet Radio Service
GSM	Global System Mobile
GTP	GPRS Tunneling Protocol
HSDPA	High Speed Downlink Packet Access
HSPA	High Speed Packet Service
HSUPA	High Speed Uplink Packet Access
HTTP	Hypertext Transfer Protocol
HTTPS	HTTP over SSL
IBC	Identity Based Cryptography
IDA	Information Dispersal Algorithm
IDS	Intrusion Detection System
IEC	International Electrotechnical Commission
IETF	Internet Engineering Task Force
IIS	Internet Information Services
IP	Internet Protocol
IPSEC	Internet Protocol SECurity
IRS	Intrusion Response System
IS-95	Interim Standard 95
ISO	International Organization for Standardization
IT	Intrusion Tolerance
ITU	International Telecommunication Union
L2TP	Layer 2 Tunneling Protocol
L2TP/IPSEC	L2TP over IPSec
LAN	Local Area Network
LCD	liquid crystal display
LED	Light-Emitting Diode
LTE	Long Term Evolution
LZO	Lempel–Ziv–Oberhumer compression
MAC	Media Access Control
MAC	Message Authentication Code
MANET	Mobile Ad Hoc Network

MDP	Mobile Development Platform
M-LEARNING	Mobile Learning
MMRP	Mobile Mesh Routing Protocol
MOCA	Mobile Certificate Authorities
NAP	Network Access Point
NEP	Nokia Energy Profiler
OFDM	Orthogonal Frequency Division Multiplexing Access
OLED	Organic Light-Emitting Diode
OS	Operating System
PDA	Personal Digital Assistant
PEM	Privacy-enhanced Electronic Mail
PGP	Pretty Good Privacy
PKCS	Public-Key Cryptography Standards
PKG	Private Key Generator
PKI	Public Key Infraestructure
RDP	Route Discovery Process
RFC	Request For Comments
RSA	Rivest, Shamir y Adleman
RSP	Recorded Shortest Path
SA	Security Association
SAODV	Secure Ad-hoc On-demand Distance Vector Routing Protocol
SAR	Security Aware Ad-hoc Routing
SEAD	Secure Efficient Distance Vector Routing for Ad Hoc Networks
SEGK	Simple and Efficient Group Key Management
SEKM	Secure and efficient key management in mobile ad hoc networks
SFTP	SSH Secure FTP
SG-PKM	Secure Group-Based Public Key Management
SGSN	Serving GPRS Support Node
SMT	Secure Message Transmission
SPA	Simple Power análisis
SPC	Shortest Path Confirmation
SPREAD	Secure Protocol for Reliable Data Delivery
SRP	Secure Routing Protocol
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TIA	Telecommunications Industry Association
TIARA	Techniques for Intrusion-resistant Ad Hoc Routing Algorithms
TIK	Instant Key Disclosure
TLS	Transport Layer Security
TTP	Trusted Third Party
UMB	Ultra Mobile Broadband
UMTS	Universal Mobile Telephone System
URSA	Ubiquitous and Robust Access Control for Ad hoc Network
VoIP	Voice over IP
VPN	Virtual Private Network
WAN	Wide Area Network
WCDMA	Wideband CDMA
WiFi	Wireless Fidelity
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network
WMANs	Wireless Metropolitan Area Networks

Bibliografía

1. JOHANSSON, Per. (2001). Bluetooth – an Enabler for Personal Area Networking. *IEEE Network (Ericsson Research)*.
2. CORDEIRO DE MORAIS, Carlos and AGRAWALL Dharma. (2011). Integrating MANETs, WLANs and Cellular Networks. In World Scientific Publishing (Ed.), *Ad Hoc and Sensor Networks - Theory and Applications* (pp. 587-620). Singapore: World Scientific Publishing.
3. ALDABBAS, Hamza; ALWADAN, Tariq and JANICKE, Helge. (February 2012). Data Confidentiality in Mobile Ad hoc Networks. *IJWMN International Journal of Wireless & Mobile Networks, Vol. 4*, pp. 225-236.
4. CORSON, S. and MACKER, J. (1999). *Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations. RFC 2501*. IETF. Retrieved from http://tools.ietf.org/html/rfc2501
5. BONNEFOI, Pierre Francois and SAUVERON, Damien. (2008). MANETS: An exclusive choice between use an security? *Computing and Informatics,, Vol. 27*, pp. 799-821.
6. KAUR, Amandeep and SINGH, Hardeep. (February 2013). A Study of Secure Routing protocols. *International Journal of Application or Innovation in Engineering & Management, Vol. 2*(3).
7. DATTA, Amitava and SOUNDARALAKSHMI, Subbiah. (2010). A Survey of State-of-the-Art Routing Protocols for Mobile Ad Hoc Networks. In Laurence T. Yang & Agustinus Borgy Waluyo (Ed.), Mobile Intelligence: John Wiley & Sons.
8. PERKINS, Charles & ROYER Elizabeth. (1997). *Ad-hoc On-Demand Distance Vector Routing*. Paper presented at the 2nd IEEE Workshop on Mobile Computing Systems and Applications.
9. JOHNSON, David and MALTZ, David. (1998). DSR: The Dynamic Source Routing Protocol forMulti-Hop Wireless Ad Hoc Networks. In Charles E. Perkins (Ed.), *Ad Hoc Networking*.
10. DUGAN, Jon. (2008). Overview of Network Measurement Tools. Brooklyn, NY: Energy Sciences Network Lawrence Berkeley National Laboratory.
11. Threats and security requirements for VANETs secure vehicle communication. *C2C-CC Sec.Workshop*.
12. NOGUEIRA LIMA, Michele; DOS SANTOS, Aldri and PUJOLLE, Guy. (2009). A Survey of Survivability in Mobile Ad Hoc Networks. *IEEE Communications Surveys & Tutorials, 11*(1), 66 - 77.
13. ZHU, B.; WAN, Z.; KANKANHALLI, M. S.; BAO, F. and DENG, R.H. (2004). *Anonymous secure routing in mobile ad-hoc networks.* Paper presented at the 29th IEEE Int. conf. on Local Computer Networks, Tampa, USA.
14. GOYAL, Priyanka; PARMAR, Vinti and RISHI, Rahul. (January 2011). MANET: Vulnerabilities, Challenges, Attacks, Application. *IJCEM International Journal of Computational Engineering & Management, Vol. 11*.
15. GOKHALE, Vikrant; GHOSH, S.K.; GUPTA; Arobinda. (2010). Classification of Attacks on Wireless Mobile Ad Hoc Networks and Vehicular Ad Hoc Networks. In

KHAN Pathan AL Sakib (Ed.), *Security of Self-Organizing Networks MANET, WSN, WMN, VANET* (pp. 195 - 225): Auerbach Publications.

16. SASAN, Adibi and GORDON, Agnew. (March 2008). Security Measures for Mobile Ad-Hoc Networks. In Y. Zhang J. Zheng, and M. Ma (Ed.), *Handbook of research on Wireless security* (pp. 500-513). Hershey - New York: Information Science Reference.
17. HAO, Yang; HAIYUN, Luo and FAN, Ye. (2004). Security in mobile ad hoc networks:challenges and solutions. *IEEE Wireless Communications Magazine, 11*(1), pp. 38 -47.
18. GOYAL, Priyanka; BATRA, Sahil and SINGH, Ajit. (November 2010). A Literature Review of Security Attack in Mobile Ad-hoc Networks. *International Journal of Computer Applications, 9*(12).
19. PRADIP, Jawandhiya; MANGESH, Ghonge and ALI, M.S. . (2010). A Survey of Mobile Ad Hoc Network Attacks. *International Journal of Engineering Science and Technology, 2*(9), 4063-4071.
20. MISHRA Amitabh. (2008). Threats and attacks. In Amitabh Mishra (Ed.), *Security and Quality of Service in Ad Hoc Wireless Networks* (pp. 43-59). New York: Cambridge University Press.
21. GOKHALE Vikrant , GHOSH S.K. and GUPTA Arobinda. (october 2011). Classification of Attacks on Wireless Mobile Ad Hoc Networks and Vehicular Ad Hoc Networks. In Al-Sakib Khan Pathan (Ed.), *Security of Self-Organizing Networks MANET, WSN, WMN, VANET* (pp. 195-225). USA: Auerbach Publications.
22. PARUL, Tomar; SURI, P.K. and SONI M. K. (2010). A Comparative Study for Secure Routing in MANET. *International Journal of Computer Applications, 4*(5), pp. 17-22.
23. WU, Bing; WU, Jie; CHEN, Jianmin and CARDEI, Mihaela. (2006). A Survey on Attacks and Countermeasures in Mobile Ad Hoc Networks. In X. Shen Y. Xiao, and D.-Z. Du (Ed.), *Wireless Network Security* (pp. 103-136). New York: Springer.
24. RAJA, Datta and NINGRINLA, Marchang. (2012). Security for Mobile Ad Hoc Networks. In Krishna Kant Sajal K Das, Nan Zhang (Ed.), *Handbook on Securing Cyber-Physical Critical Infrastructure* (pp. 147-190). USA: Morgan Kaufmann.
25. ZAFER, Murtaza A.;AGRAWAL, Dakshi and SRIVATSA, Mudhakar. (2009). Bootstrapping Coalition MANETs: Physical-Layer Security under Active Adversary. NY, USA: IBM Research, T. J. Watson Center.
26. RAVI, Srivaths; RAGHUNATHAN, Anand and CHAKRADHAR, Srimat (2004). *Tamper Resistance Mechanisms for Secure Embedded Systems.* Paper presented at the 17th International Conference on VLSI Design (VLSID'014).
27. MARKANTONAKISA, Konstantinos; TUNSTALLB, Michael and HANCKEA Gerhard. (2009). Attacking smart card systems: Theory and practice. *information security technical report, 14*, pp 46-56.
28. KOCHER, Paul; JAFFE, Joshua and JUN, Benjamin. (1995). *Introduction to Differential Power Analysis and Related Attacks*. White Paper. Cryptography Research, Inc.
29. BATINA, L. (July 2005). *Side-channel issues for designing secure hardware implementations*. Paper presented at the On-Line Testing Symposium, 2005. IOLTS 2005. 11th IEEE International, Leuven-Heverlee, Belgium

30. COMMON CRITERIA. (2010). The Common Criteria for Information Technology Security Evaluation (CC). from http://www.commoncriteriaportal.org/
31. NIST - National Institute of Standards and Technology. (2010). Federal Information Processing Standard (FIPS). from http://www.nist.gov/itl/fips.cfm
32. SIVAKUMAR, K. and SELVARAJ, G. (January 2013). Overview of various attacks in manet and countermeasures for attacks. *International Journal of Computer Science and Management Research, Vol. 2*(1).
33. ROCABADO, Sergio; SANCHEZ, Ernesto; DIAZ, Javier y ARIAS FIGUEROA, Daniel. (2011). *Integración Segura de MANETs con Limitaciones de Energía a Redes de Infraestructura*. Paper presented at the CACIC 2011, La Plata - Buenos Aires - Argentina. http://sedici.unlp.edu.ar/handle/10915/18771
34. WU, Bing; WU, Jie and CARDEI, Mihaela. (2008). A Survey of Key Management in Mobile Ad Hoc Networks. In Y. Zhang J. Zheng, and M. Ma (Ed.), *Handbook of research on Wireless security* (pp. 479-499). Hershey - New York: Information Science Reference.
35. SUDHIR, Agrawal; SANJEEV, Jain and SANJEEV, Sharma. (Jan 2011). A Survey of Routing Attacks and Security Measures in Mobile Ad-Hoc Networks. *Journal of computing, 3*(1), 41-48.
36. MANDALA, Satria; NGADI, Asri and ABDULLAH, Hanan (2008). A Survey on MANET Intrusion Detection. *International Journal of Computer Science and Security, Vol 2* (1), pp. 1-11.
37. RASHA, T.K and SHWETHA, Vincent. (October 2012). A Survey on Intrusion Response Mechanism for MANET. *International Journal of Emerging Technology and Advanced Engineering, vol 2*(10).
38. KUMAR, Sathish. (2010). Classification and Review of Security Schemes in Mobile Computing. *Wireless Sensor Network, Vol.2*.
39. SUDIP, Misra and SUMIT, Goswami. (october 2011). Key Management in Mobile Ad Hoc Networks. In Al-Sakib Khan Pathan (Ed.), *Security of Self-Organizing Networks MANET, WSN, WMN, VANET* (pp. 147-172). USA.
40. HOEPER, K. and GONG, G. (2004). Models of Authentications in Ad Hoc Networks and Their Related Network Properties. *International Association for Cryptologic Research*.
41. DALAL, Renu;SINGH, Yudhvir and KHARI, Manju (2012). *A Review on Key Management Schemes in MANET*. Paper presented at the International Journal of Distributed and Parallel Systems.
42. SHAMIR, Adi. (1979). How to Share a Secret. *Communications of the ACM, 22*(11).
43. HOU, L. and HAAS, Z.J (1999). Securing Ad Hoc Networks. *IEEE Network, Vol. 13*(4), pp. 24-30.
44. YI, S. and KRAVETS, R. (2003). MOCA: Mobile Certificate Authority for Wireless Ad Hoc. *The 2nd Annual PKI Research Workshop*
45. WU, Bing; WU, Jie and FERNANDEZ, Eduardo (August 2007). Secure and efficient key management in mobile ad hoc networks. *Journal of Network and Computer Applications, vol. 30*(3), pp. 937-954
46. HEGLAND, Anne Marie; WINJUM, Eli and SPILLING, Pal (2006). A survey of Key management in ad hoc networks. *IEEE Communications Surveys & Tutorials, 8*(3).

47. LUO, H.; KONG, J. and ZERFOS P. (2004). URSA: ubiquitous and robust access control for mobile ad hoc networks. *IEEE/ACM Trans. Netw, Vol. 12*(6), pp. 1049-1063.
48. ZHU, Bo; BAO, Feng; DENG, Robert and WANG, Guilin. (2005). Efficient and robust key management for large mobile ad hoc networks. *The International Journal of Computer and Telecommunications Networking 48*(4).
49. KAPIL, Anil and RANA, Sanjeev (2007). Identity-Based Key Management in MANETs using Public Key Cryptography. *International Journal of Security, 3*(1).
50. LI, Jingfeng; WEI, Dawei and KOU Hongzhao. (2006). *Identity-Based and Threshold Key Management in Mobile Ad Hoc Networks*. Paper presented at the International Conference on Wireless Communications, Networking and Mobile Computing. WiCOM 2006.
51. CAPCUN, S.; BUTTY, L. and HUBAUX, J. (2003). Self-Organized Public-Key Management for Mobile Ad Hoc Networks. *IEEE Trans. Mob. Comput, vol. 2*(1), pp. 52-64.
52. LUO, J.; HUBAUX, J. and EUGSTER, P.T. (2007). DICTATE: DIstributed CerTification Authority with probabilisTic frEshness for Ad Hoc Networks. *IEEE Transactions on Dependable and Secure Computing, vol. 2*(3), pp. 311-323.
53. YI, S. and KRAVETS, R. (2004). Composite Key Management for Ad Hoc Networks. *MobiQuitous, IEEE Computer Society*, pp. 52-61.
54. SANZGIRI, Kimaya; DAHILL, Bridget; LEVINE, Brian; ROYER, Elizabeth and SHIELDS, Clay. (November 2002). A Secure Routing Protocol for Ad Hoc Networks. *In 10 Conference on Network Protocols (ICNP)*.
55. YI, Seung; NALDURG, Prasad and KRAVETS, Robin. (2001). *Security-Aware Ad hoc Routing for Wireless Networks*. Paper presented at the Second ACM Symposium on Mobile Ad Hoc Networking & Computing.
56. PAPADIMITRATOS, P. and HASS, Z. J. (Jan. 2002). *Secure Routing for Mobile Ad Hoc Networks*. Paper presented at the Communication Networks and Distributed Systems Modeling and Simulation Conf.
57. HU, Yih-Chun; JOHNSON, David B. and PERRING, Adrian. (2002). *Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks*. Paper presented at the 4th IEEE Workshop on Mobile Computing Systems & Applications.
58. ZHU, Sencun; XU, Shouhuai and SETIA, Sanjeev. (nov. 2003). *Establishing Pairwise Keys for Secure Communication in Ad Hoc Networks: A Probabilistic Approach*. Paper presented at the 11th IEEE International Conference on Network Protocols.
59. PERRING, Adrian; CANETTI, Ran; TYGAR, J. D. and SONG, Dawn. (2002). *The TESLA Broadcast Authentication Protocol*.
60. HU, Yih-Chun; PERRING, Adrian and JOHNSON, David B. (2002). *Packet Leashes: A Defense against Wormhole Attacks in Wireless Ad Hoc Networks*. Technical Report TR01-384. Rice University Department of Computer Science.
61. ALLEN, James; GAUGHAN, Patrick; SCHIMMEL, David and YALAMANCHILI, Sudhakar. (Sep. 2002). *Ariadne- An Adaptive Router for Fault-tolerant Multicomputers*. Paper presented at the MobiCom'02.

62. ZAPATA GUERRERO, Manel and ASOKAN, N. (July 2002). Secure Ad hoc On-Demand Distance Vector (SAODV) Routing. *ACM Mobile Computing and Communications Review, 3*(6), pp. 106-107.
63. MICHIARDI, P. and MOLVA, R. (2002). *CORE: A COllaborative REputation mechanism to enforce node cooperation in Mobile Ad Hoc Networks*. Paper presented at the IFIP - Communication and Multimedia Security Conference 2002.
64. BUCHEGGER, S. and LE BOUDEC, J. (July 2005). Self-policing Mobile Ad Hoc Networks by Reputation Systems. *IEEE Communications Magazine*.
65. BUTTYAN, L. and HUBAUX, J. (October 2003). Stimulating Cooperation in Self-Organizing Mobile Ad hoc networks. *Mobile Networks and Applications, vol. 8*(5).
66. WEYLAND, Attila and BRAUN, Weyland. (2009). CASHnet - Cooperation and Accounting Strategy for Hybrid Networks. Bern, Switzerland: Institute of Computer Science and Applied Mathematics, University of Bern.
67. ANANTVALEE, Tiranuch and WU, Jie. (2006). A Survey on Intrusion Detection in Mobile Ad Hoc Networks. In X. Shen Y. Xiao, and D.-Z. Du (Ed.), *Wireless/Mobile Network Security* (pp. 170 - 196): Springer.
68. JACOBY, GA and DAVIS, NJ. (2007). Mobile host-based intrusion detection and attack identification. *IEEE Wireless Communications Magazine, 14*(4), pp. 53-60.
69. NADKARNI, K; MISHRA, A. (2004). *A novel intrusion detection approach for wireless ad hoc networks.* Paper presented at the IEEE wireless communications and networking conference (WCNC, 2004).
70. LAUF, A.; PETERS, R. and ROBINSON, W.H. (2010). A distributed intrusion detection system for resource-constrained devices in ad hoc networks. *Ad Hoc Networks, 8*(3), pp. 253-256.
71. BOSE, S; BHARATHIMURUGAN, S and KANNAN, A. (February 2007). *Multi-layer integrated anomaly intrusion detection system for mobile ad hoc networks.* Paper presented at the IEEE ICSCN 2007, Chennai, India.
72. RAZAK, SA; FURNELL, SM; CLARKE, NL and BROOKE, PJ. (September 2008). Friend-assisted intrusion detection and response mechanisms for mobile ad hoc networks. *Ad Hoc Networks, vol. 6*(7), pp. 1151-1167.
73. WANG, Wei; MAN, Hong and LIU, Yu. (April 2009). A framework for intrusion detection systems by social network analysis methods in ad hoc networks. *Wiley Security and Communication Networks, vol. 2*(6).
74. DENG, H; XU, R; LI, J; ZHANG, F; LEVY, R and LEE,W. (2006). *Agent-based cooperative anomaly detection for wireless ad hoc networks.* Paper presented at the 12th conference on parallel and distributed systems.
75. MA, Chuan Xiang and FANG, Ze Ming. (Jan 2009). *A Novel Intrusion Detection Architecture Based on Adaptive Selection Event Triggering for Mobile Ad-hoc Networks.* Paper presented at the Intelligent Information Technology and Security Informatics, 2009. IITSI '09, Moscow.
76. OTROK, H; Mohammed, N; Wang, L; Debbabi, M; Bhattacharya, P,. (2008). A game-theoretic intrusion detection model for mobile ad hoc networks. *Elsevier Computer Communications, 31*(4), pp. 708 - 721.
77. STERBENZ, J.P.G.; KRISHNAN, R. and HAIN, R.R. (September 2002). *Survivable Mobile Wireless Networks: Issues, Challanges, and Research Directions*. Paper presented at the WiSe'02, Atlanta, Georgia, USA.

78. RAMANUJAN, R.; KUDIGE, S. and NGUYEN, T. (2003). *Techniques for intrusion-resistant ad hoc routing algorithms TIARA.* Paper presented at the In DARPA information survivability conference and exposition (DISCEX), Los Alamitos, CA, USA.
79. XUE, Y. and NAHRSTEDT, K. (2004). Providing fault-tolerant ad hoc routing service in adversarial environments. *Wireless personal communications: an international journal of computing, vol. 29*(3).
80. LOU, W; LIU, W. and FANG, Y. (2004). *SPREAD: enhancing data confidentiality in mobile ad hoc networks.* Paper presented at the IEEE computer and communications societies (INFOCOM).
81. PAPADIMITRATOS, P. and HASS, Z. J. (September 2003). *Secure Data Transmission in Mobile Ad Hoc Networks*. Paper presented at the International Conference on Web Information Systems Engineering (WISE'03). Atlanta, Georgia, USA.
82. RABIN, M. O. (1989). Efficient Dispersal of Information for Security, Load Balancing, and Fault Tolerance. *Journal of the ACM, 36*(2), pp. 335–348.
83. DA SILVA, Eduardo; LIMA, Michele N.; DOS SANTOS, Aldri and ALBINI, Luiz Carlos,. (2012). Secure Group-Based Public Key Management for Mobile Ad Hoc Networks. *Journal of Selected Areas in Telecommunications (JSAT).*
84. WU, Bing; WU, Jie; and DONG, Yuhong. (2008). An efficient group key management scheme for mobile ad hoc networks. *Int. Journal of Security and Networks*.
85. GOOGLE. (2011). Google Play - Android Application Market. from https://play.google.com/store
86. STERICSON, Stephen. (2011). BusyBox. from https://play.google.com/store/apps/details?id=stericson.busybox
87. FOLKERT van Heusden. HTTPing for Google Android mobile phones. Retrieved from http://www.vanheusden.com/Android/HTTPing
88. MAGICANDROIDAPPS.COM. (2011). Iperf for Android. from https://play.google.com/store/apps/details?id=com.magicandroidapps.iperf
89. LYSESOFT. AndFTP. Retrieved from http://www.lysesoft.com/products/andftp/index.html
90. JACKPAL. Android Terminal Emulator. Retrieved from https://github.com/jackpal/Android-Terminal-Emulator
91. NLANR/DAST. (2012). Iperf project. from http://sourceforge.net/projects/iperf
92. NLANR/DAST. (May 2012). Iperf 2.0.5 Windows executable. from http://code.google.com/p/iperf-cygwin/
93. NLANR/DAST. (Jul 2012). Iperf 3 Windows executable. from https://code.google.com/p/iperf3-cygwin/
94. WIKIPEDIA. (2012). The free encyclopedia. from http://en.wikipedia.org/wiki
95. SHEARER, Findlay. (2008). Hierarchical View of Energy Conservation. In Elsevier (Ed.), *Power Management in Mobile Devices*. United States of America: ELSEVIER.
96. LESS WATTS. (2011). Powertop. Retrieved from https://lesswatts.org/projects/powertop/

97. MICROSOFT RESEARCH. (2011). Joulemeter: Computational Energy Measurement and Optimization. Retrieved from http://research.microsoft.com/en-us/projects/joulemeter/
98. GOOGLE Training. (2012). Android PowerManager API Guides. from http://developer.android.com/reference/android/os/PowerManager.html
99. MICROSOFT RESEARCH. (2012). Energy Profiler. from http://research.microsoft.com/en-us/projects/eprof
100. GORDON, Mark; ZHANG, Lide and TIWANA, Birjodh. PowerTutor. University of Michigan. Retrieved from http://ziyang.eecs.umich.edu/projects/powertutor
101. NOKIA. (2009). Nokia energy Profiler (NEP). from http://www.developer.nokia.com/Resources/Tools_and_downloads/Other/Nokia_Energy_Profiler/
102. QUALCOMM. (2012). Trepn Profiler. from https://developer.qualcomm.com/mobile-development/development-devices/trepn-profiler
103. PATHAK, Abhinav; HU, Charlie and ZHANG, Ming. (2011). *Fine-Grained Power Modeling for Smartphones Using System Call Tracing*. Paper presented at the EuroSys.
104. BALASUBRAMANIAN, Niranjan; BALASUBRAMANIAN, Aruna and VENKATARAMANI, Arun (2009). *Energy consumption in mobile phones: a measurement study and implications for network applications*. Paper presented at the 9th ACM SIGCOMM conference on Internet measurement conference, New York, USA. http://dl.acm.org/citation.cfm?id=1644893.1644927
105. QUALCOMM. (2012). Snapdragon™ Mobile Processors. from https://developer.qualcomm.com/discover/chipsets-and-modems/snapdragon
106. QUALCOMM. (2012). Snapdragon™ MDP & DragonBoard Mobile Development Devices. from https://developer.qualcomm.com/mobile-development/development-devices
107. SPATH, Chris. (2011). *Optimizing Apps For Power and Network Efficiency Using Trepn™ Profiler*. Paper presented at the UPLINQ Conference. https://www.uplinq.com/2011/sites/default/files/slides/Optimizing-Apps-For-Power-Network-Efficiency-Snapdragon-MDP2.pdf
108. ZHANG, Lide; TIWANA, Birjodh; QIAN, Zhiyun and WANG, Zhaoguang. (2010). *Accurate online power estimation and automatic battery behavior based power model generation for smartphones*. Paper presented at the 2010 IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS), Scottsdale, AZ. http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5751489
109. GOOGLE. (2012). Our Mobile Planet. from http://www.thinkwithgoogle.com/mobileplanet
110. KENT, S.; ATKINSON, R. (1998). *Security Architecture for the Internet Protocol*. IETF. Retrieved from http://www.ietf.org/rfc/rfc2401.txt
111. DIERKS, T.; RESCORLA, E. (2008). *The Transport Layer Security (TLS) Protocol (ver 1.2)*. IETF. Retrieved from http://tools.ietf.org/html/rfc5246
112. ETSI EN 301 344. (2000). *Digital cellular telecommunications system, General Packet Radio Service (GPRS), Service description*. Retrieved from http://www.etsi.org/index.php/technologies-clusters/technologies/mobile/gprs

113. CORDEIRO DE MORAIS, Carlos and AGRAWALL Dharma. (2011). Wireless PANs. In World Scientific Publishing (Ed.), *Ad Hoc and Sensor Networks - Theory and Applications* (pp. 196-258). Singapore: World Scientific Publishing.
114. LEE, Jin-Shyan; SU, Yu-Wei and SHEN, Chung-Chou. (2007). *A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi.* Paper presented at the The 33rd Annual Conference of the IEEE Industrial Electronics Society.
115. SPECIAL INTEREST GROUP (SIG) Bluetooth. (2001). Bluetooth Network Encapsulation Protocol (BNEP) Especification.
116. RUSELL, Paul and NETFILTER Core Team. THE NETFILTER.ORG PROJECT. Retrieved from http://www.netfilter.org/
117. BALASUBRAMANIAN, Niranjan; BALASUBRAMANIAN, Aruna and VENKATARAMANI, Arun. (2009). *Energy Consumption in Mobile Phones: A Measurement Study and Implications for Network Applications*. Paper presented at the 9th ACM SIGCOMM conference on Internet measurement conference.
118. PERRUCCI, Gian Paolo; FITZEKY, Frank and SASSOY, Giovanni. (2009). *On the Impact of 2G and 3G Network Usage for Mobile Phones Battery Life*. Paper presented at the European Wireless 2009.
119. 3GPP. (2011). Specification 29060 - GPRS Tunneling Protocol, release 11.0. from http://www.3gpp.org/ftp/Specs/html-info/SpecVsWi--29060.htm
120. PATEL, B.; ABOBA, B y Otros (2001). *L2TP/IPsec, RFC 3193*. IETF. Retrieved from http://tools.ietf.org/html/rfc3193
121. OpenSSL for Windows. (2011). from http://slproweb.com/products/Win32OpenSSL.html
122. OpenVPN for Windows. (2010). from http://openvpn.net/index.php/download.html
123. Filezilla Server (2012). from https://filezilla-project.org/
124. KAIBITS Software. Network Signal Info. Retrieved from http://www.kaibits-software.com/product_netwotksignaldonate.htm
125. SCHÄUFFELHUT, Friedrich. OpenVPN Installer. Retrieved from http://code.google.com/p/android-openvpn-installer
126. SCHÄUFFELHUT, Friedrich. OpenVPN Settings. Retrieved from http://code.google.com/p/android-openvpn-settings
127. FEILNER, Markus (2006). *OpenVPN - Building and Integrating Virtual Private Networks*. Birmingham, B27 6PA, UK.: Packt Publishing.
128. OBERHUMER, Markus F.X.J. (2010). LZO compression. from http://www.oberhumer.com/opensource/lzo/
129. RESCORLA, E. (2000). *HTTP Over TLS, RFC 2818*. IETF. Retrieved from http://tools.ietf.org/html/rfc2818
130. HUTCHINSON and FORD, P. (2005). *Securing FTP with TLS, RFC 4217*. IETF. Retrieved from http://tools.ietf.org/html/rfc4217
131. ROCABADO, Sergio; SANCHEZ, Ernesto; DIAZ, Javier y ARIAS FIGUEROA, Daniel. (2012). *Integración Segura de MANETs, desplegadas en zonas de recursos limitados, a Redes de Infraestructura*. Paper presented at the CACIC 2012, Bahia Blanca - Buenos Aires - Argentina. http://sedici.unlp.edu.ar/handle/10915/23762
132. ROCABADO, Sergio; HERRERA, Susana y Otros. (2013). *M-LEARNING EN ZONAS DE RECURSOS LIMITADOS*. Paper presented at the TE&ET 2013, Santiago del Estero - Argentina. http://sedici.unlp.edu.ar/handle/10915/27585

133. ROCABADO, Sergio; SANCHEZ, Ernesto; DIAZ, Javier y ARIAS FIGUEROA, Daniel. (2013). *Caso de estudio de comunicaciones seguras sobre redes móviles ad hoc*. Paper presented at the CACIC 2013, Mar del Plata - Buenos Aires - Argentina. http://sedici.unlp.edu.ar/handle/10915/31244
134. ROCABADO Sergio. (2011). *Integración Segura de MANETs a Redes de Infraestructura*. Paper presented at the V jornadas científicas interdisciplinarias, Sede regional Orán de la UNSa. Orán - Salta - Argentina.
135. ROCABADO, Sergio; SANCHEZ, Ernesto y ARIAS FIGUEROA, Daniel. (2011). *Enseñanza de Redes Móviles Ad Hoc basada en simulación*. Paper presented at the I Jornadas Nacionales de TIC e Innovación en el Aula y III Jornadas de Experiencia en EaD, UNLP - La Plata - Buenos Aires.
136. ROCABADO, Sergio; HERRERA, Susana; ARIAS FIGUEROA, Daniel. (2013). *Despliegue de MANETs para M-learning en zonas de recursos limitados*. Paper presented at the XV Workshop de Investigadores en Ciencias de la Computación (WICC), Universidad Autónoma de Entre Ríos (UADER) - Entre Rios - Argentina. http://sedici.unlp.edu.ar/handle/10915/27096
137. ROCABADO, Sergio; HERRERA, Susana; CAMPOS, Matias y CORONEL, Adrián. (Octubre 2013). *Redes móviles ad hoc para zonas de recursos limitados*. Paper presented at the IX Jornadas de Ciencia y Tecnología de Facultades de Ingeniería del NOA Santiago del Estero.
138. COMPUTER LANGUAGE COMPANY. (2010). Computer Desktop Encyclopedia. from http://www.computerlanguage.com
139. ITU. (2011). International Telecommunication Union. from http://www.itu.int
140. 3RD GENERATION PARTNERSHIP PROJECT "3GPP". (2012). 3GPP specifications. from http://www.3gpp.org/specifications
141. 3RD GENERATION PARTNERSHIP PROJECT 2 "3GPP2". (2012). 3GPP2 specifications. from http://www.3gpp2.org/Public_html/specs
142. IEEE 802.16 Working Group. (Dec. 2005). IEEE Standard for Local and Metropolitan Area Networks. IEEE, New York, USA.
143. GSM WORLD. (2012). from http://www.gsmworld.com
144. ETSI. (2012). European Telecommunications Standards Institute. from http://www.etsi.org/standards
145. TIA. (2011). Telecommunications Industry Association. from http://www.tiaonline.org