Universidad Nacional Autónoma de México
Instituto de Investigaciones en Matemáticas Aplicadas y Sistemas

Protocolo Seguro en Redes Ad Hoc

Paola Díaz Flores

Tutor: Sergio Rajsbaum Gorodesky
Instituto de Matemáticas
Universidad Nacional Autónoma de México (UNAM)

Co-tutor: Carlos Mex Perera
Centro de Electrónica y Telecomunicaciones
Tecnológico de Monterrey (ITESM)

Dedicatoria

A mi hija, que es lo más preciado que tengo y que en esta etapa de mi vida fue un motivo de impulso, de ir hacia delante. Porque en aquellas noches de trabajo, ella lo ratificaba con una sonrisa.

A David, por todos los momentos inolvidables que hemos pasado.

A mis padres, porque he tenido su apoyo incondicional a lo largo de mi vida y se ha manifestado más en este proyecto, donde me han dado todo cuanto he necesitado.

Agradecimientos

A Dios, por haberme regalado la vida y que ésta se haya prolongado hasta éste momento.

A mis tutores, por haberme compartido sus conocimientos y por ser, además de grandes investigadores, grandes seres humanos, con los cuales fue una experiencia agradable trabajar.

A mis amigos, y en especial a Ana, por haberme brindado su amistad.

A la UNAM, por proporcionarnos excelentes investigadores para nuestra educación y además por el apoyo económico que se me proporcionó a lo largo de mi maestría

A CONACYT, por el apoyo económico otorgado en éstos dos años.

Índice general

1. Introducción **6**
- 1.1. Objetivo . . . 8
- 1.2. Definición del problema . . . 8
- 1.3. Justificación . . . 9
- 1.4. Distribución del trabajo . . . 10

2. Redes Inalámbricas Ad Hoc **11**
- 2.1. Redes Inalámbricas . . . 11
- 2.2. Redes Inalámbricas Ad Hoc . . . 13
- 2.3. Protocolos de ruteo para redes inalámbricas Ad Hoc . . . 13
  - 2.3.1. Protocolos de colonia de hormigas . . . 14
  - 2.3.2. El problema del camino más corto en una colonia de hormigas real 15
  - 2.3.3. Protocolos basados en colonia de hormigas . . . 17
  - 2.3.4. Protocolo *AntHocNet* . . . 19

3. Seguridad en redes inalámbricas y Criptografía **23**
- 3.1. Seguridad en Redes Inalámbricas Ad Hoc . . . 23
- 3.2. Ataques en redes inalámbricas Ad Hoc . . . 25
- 3.3. Ataque *Blackhole* . . . 26
- 3.4. Ataque *Blackhole* para el protocolo de ruteo *AntHocNet* . . . 27
- 3.5. Criptografía . . . 29

4. Métodos de defensa **34**
- 4.1. Mecanismos de detección de intrusos existentes contra el ataque *Blackhole* 34
- 4.2. Métodos de defensa propuestos . . . 42
  - 4.2.1. Primer método para el ataque *Blackhole*: *Método ACK* . . . 43
  - 4.2.2. Segundo método para el ataque *Blackhole*: *Método monitores* . . 46

5. Resultados **52**
- 5.1. Ambiente de simulación general . . . 52
  - 5.1.1. Consideraciones . . . 53
- 5.2. Análisis del efecto del ataque *Blackhole* en una red ad hoc con el protocolo *AntHocNet* . . . 54
- 5.3. Análisis del Método ack . . . 56

5.4. Análisis del Método Monitores. 58
5.5. Comparación de los métodos ack y Monitores 59
5.6. Simulaciones sin movilidad en los nodos 59

6. Conclusiones y Trabajo Futuro **62**
6.1. Conclusiones 62
6.2. Trabajo Futuro 63

Bibliografia **65**

Resumen

Esta tesis presenta dos mecanismos de defensa contra el ataque *Blackhole* en una red ad hoc. El ataque *Blackhole* consiste en que, cuando el intruso reciba paquetes de petición de ruta, contesta (de forma falsa) que el destino está a simplemente un salto de él, cuando el nodo origen use esta falsa ruta, los paquetes transmitidos no alcanzarán al destino si no que serían descartados por el intruso. Con cada método se describen una serie de acciones para poder mitigar dicho ataque. En otras palabras, nuestros métodos además de detectar que la red esta siendo atacada, toman acciones para poder reducir sus efectos. El primer método es sencillo de implementar, no detecta al nodo atacante pero elimina la ruta que va hacia él, cabe destacar que necesitamos para este primer método de una infraestructura de clave pública ya establecida. En el segundo método se propone una infraestructura adicional, es más eficiente que el primero, además una característica importante que tiene es que, es posible detectar exactamente cual nodo es el atacante, aislándolo de la red durante un periodo de tiempo predefinido. En los resultados obtenidos con las simulaciones de los dos métodos se vió, que el método monitores mitiga de mejor forma el ataque *Blackhole*, esto es porque en el primer método se sabe cual ruta lleva al intruso (la fuente sabe por cual vecino no están llegando los paquetes) pero no se sabe cual es el último nodo de esta ruta, en cambio en el segundo método al poder identificar al nodo intruso se puede proteger de mejor forma la red de éste nodo.

Capítulo 1

Introducción

En nuestros días pensar en trabajar aislados, sin estar conectado a una red es un poco complicado, esto es por servicios que ofrece una red, como podría ser Internet, impresión, transferencia de información, entre otros. Empezaremos por mencionar de forma muy simple el concepto de red, el cual podríamos decir que es un conjunto de dispositivos los cuales se conectan entre si para intercambiar información. Ahora bien con el paso del tiempo las comunicaciones que se daban entre los equipos por medio de cables, en nuestros días se han sustituido por un medio inalámbrico, como por ejemplo la radio, los infrarrojos o el láser, la cual permite que los usuarios tengan movilidad dentro de un rango de cobertura, y sigan conectados a la red. Una red inalámbrica se puede definir como dos o más dispositivos, los cuales intercambian información sin necesidad de tener un medio guiado de por medio. Este nuevo aspecto cambia completamente las características de red, pues ahora tener nodos móviles hace que la topología de red sea dinámica, que exista interferencia en las transmisiones, limitaciones en los recursos de los nodos al ser de tamaño reducido y con autonomía, entre otros.

Dentro de las redes inalámbricas existen dos configuraciones que son las más usadas, aquella con infraestructura que utiliza puntos de acceso, éstos coordinan las comunicaciones entre los dispositivos, además de ser el enlace entre la red cableada y la inalámbrica. Y la configuración ad hoc, donde los dispositivos móviles se comunican entre ellos sin ningún elemento extra que los controle, así la tarea de transmisión de paquetes es realizada por los mismos nodos.

Las redes inalámbricas se han hecho populares debido a su fácil instalación, pero cabe destacar que tienen ventajas y desventajas que son convenientes mencionar.

Entre las ventajas tenemos, que los usuarios de una red inalámbrica se pueden mover a lo largo de la zona de cobertura de la red, así en una empresa esta característica significa productividad, pues los empleados pueden tener acceso a los recursos de la red donde quiera que estén de la empresa (zona limitada). Además una red inalámbrica puede proveer servicios a una cantidad variable de usuarios, a diferencia de las redes

cableadas donde un usuario nuevo, requiere nueva infraestructura. Usuarios con dispositivos que tengan una antena de transmisión pueden hacer uso de redes inalámbricas por periodos cortos, como puede ser un café internet o redes privadas, sin necesidad de hacer un cambio en la infraestructura de la red.

Entre las desventajas podemos mencionar primero a la seguridad, esto es porque las transmisiones se hacen por medio inalámbrico, y cualquier dispositivo que se encuentre cerca y ponga su antena en modo promiscuo, puede escuchar las transmisiones de los demás equipos. Esto no sucede con las redes cableadas puesto que para un atacante, simplemente, estar conectado a la red ya representa un problema pues físicamente tendrá que tirar un cable y conectarse a la red para después escuchar los paquetes de los demás. Actualmente se han empleado técnicas de criptografía para evitar que nodos intrusos lean la información que no está dirigida a ellos, pero se ha comprobado que dichas técnicas son débiles cuando utilizan claves pequeñas, ya que es posible romper el cifrado. Cabe destacar que existen cifradores seguros e incluso si hacemos la clave más grande en algunos sistemas de cifrado, serán confiables, pero implica un procesamiento mayor en el dispositivo, donde en nuestro ambiente, de red inalámbrica no es conveniente, ya que los equipos tienen recursos limitados como podría ser el de computo, memoria y batería. Además dicha solución no es completa porque se podría dar el caso que el dispositivo fuera robado físicamente por un atacante, y que éste pueda conocer las claves del dispositivo. Es por esto que se dice que esta solución es una primera barrera a los atacantes y una segunda barrera seria la detección de intrusos, esto es cuando el atacante ya ha podido acceder a la red y hacer algún daño. Otra desventaja que podemos señalar es que la cobertura de los puntos de acceso, van del orden de decenas de metros, por lo que en algunas aplicaciones es insuficiente dicha cobertura y se tendrán que poner repetidores u otros puntos de acceso por lo que el costo total de la red incrementa. Se puede señalar también que como cualquier transmisión inalámbrica, la señal está expuesta a interferencia. Así los recursos importantes de la red como pueden ser los servidores nunca se conectan de forma inalámbrica. Y por último podemos mencionar como desventaja la tasa de transmisión de paquetes en la trasnmición de datos, pues está por debajo de aún la velocidad más baja de las redes cableadas.

Como ya se mencionó la seguridad es un aspecto que aún no está resuelto en las redes inalámbricas y se ha convertido en algo esencial para proveer comunicaciones protegidas entre nodos móviles en ambientes hostiles (como puede ser el inalámbrico). Distinto a las redes cableadas, la única característica que las redes inalámbricas ad hoc poseen, es un número de desafíos no triviales para el diseño de seguridad, tales como una arquitectura de red abierta punto a punto, medio inalámbrico compartido, recursos limitados, una topología altamente dinámica. Estos desafíos claramente hacen que las soluciones de seguridad tengan que ser multifacéticas para que proporcionen tanto una amplia protección como un buen desempeño.

A diferencia de otras redes donde usan nodos dedicados a realizar funciones básicas

de la red como puede ser la retransmisión de paquetes y el ruteo, en las redes ad hoc éstas funciones son realizadas por todos los nodos existentes en la red. Por esta razón, no se debe asumir que todos los nodos eventualmente cooperarán unos con otros ya que cualquier operación en la red consume energía, un recurso particularmente escaso en dispositivos pequeños que trabajan con batería. Por lo que una mala conducta que se da en los nodos es la falta de cooperación.

1.1. Objetivo

El objetivo de esta tesis es proponer mecanismos de defensa contra el ataque *Blackhole*, el cual consiste en, que el nodo intruso contesta a peticiones de ruta, diciendo que el nodo destino está a un salto de él, aún cuando esto no es cierto, esto lo hace para poder interceptar los paquetes que van hacia el destino, afectando así las comunicaciones entre el nodo fuente y el nodo destino. Ahora bien, cuando un nodo perteneciente a la red lanza el ataque *Blackhole*, el poder construir un mecanismo para identificar que la red esta siendo atacada, y de ser posible saber quien es el intruso y castigarlo. Además de saber que hay un intruso en la red lo que se desea es establecer un mecanismo de acción, el cual mitigue este ataque. Evaluar que tan buenos son nuestros mecanismos, que tanto ayudan para que el rendimiento de la red no decaiga, cuando está siendo atacada. Los mecanismos de detección fungen como una segunda barrera en lo que a seguridad respecta. Entonces se necesita construir mecanismos los cuales sean eficientes para que el ataque no continúe haciendo estragos en la red y a la vez que no necesite mucho procesamiento en los nodos, pues éstos tienen limitaciones al ser móviles, como puede ser de procesamiento, de batería, memoria, etc.

Estudiaremos las soluciones establecidas por otros investigadores para este mismo problema, que tan eficientes son. Pero lo que nosotros deseamos construir es un mecanismo que vaya de la mano con el protocolo de ruteo (*AntHocNet*) que aproveche características propias para poder enfrentar el ataque. Para esto estudiaremos a profundidad dicho protocolo para saber qué fases de éste nos pueden ayudar en nuestra tarea.

1.2. Definición del problema

El problema que se desea resolver en la tesis, es conocer a profundidad el ataque *Blackhole*, cómo trabaja, y definir que características de éste se pueden aprovechar para ser detectado, de forma pronta y efiente. Aunque de forma general las redes ad hoc presentan problemas, que deben ser tomadas en cuenta para cualquier solución de seguridad y se consideró necesario mencionar:

- La ausencia de una autoridad central en redes ad hoc, que controle las actividades en la red, como pueden ser el correcto comportamiento en los nodos, el acceso a los recursos de la red, distribución y mantenimiento de claves al utilizar criptografía, todo esto a beneficio de la seguridad de la red.

- La topología dinámica es otro problema, pues nodos que se confía en ellos, puede suceder que después no estén cerca, y se tenga que confiar en nuevos nodos.

- Conocer a profundidad el ataque a tratar, ya que éste puede tener variaciones y se debe proponer una solución que sea lo más completa posible.

- Que el mecanismo de detección sea compatible con el protocolo de ruteo, de ser posible que aproveche las características de éste, además de tomar en cuenta que una falla en el protocolo, provocará falla en el mecanismo.

- Que el mecanismo no debe de generar mucho procesamiento en los nodos, puesto que esto bajará el rendimiento de la red, ni que utilice excesivos mensajes, pues generará sobrecarga en la misma.

- Falta de control en los nodos nuevos, es decir, los nodos existentes confiarán ciegamente en los nuevos nodos que se incorporan a la red, confiándoles sus comunicaciones.

- Dificultad de identificar claramente al intruso, pues éste puede cambiar de actividad, presentar comportamientos anómalos de forma esporádica, o sin rebasar los umbrales establecidos para ser identificado.

- Asignación de una cota o umbral para decir que un nodo es intruso, para que no se dé el caso de que un nodo sea etiquetado como intruso cuando no lo es (falsos positivos), y viceversa que un nodo que sea intruso pase desapercibido su ataque (falsos negativos).

1.3. Justificación

Las redes ad hoc ofrecen ventajas con respecto a las redes cableadas e incluso a las redes inalámbricas con infraestructura, debido a que se pueden crear y configurar sin tener una autoridad que esté coordinando a los nodos, ya que ellos mismos hacen dicha tarea. Por esta característica se piensa que tienen un futuro prometedor, pensando en las aplicaciones que podrían tener. Ahora bien, simplemente por su naturaleza de ser inalámbricas y características adyacentes tienen problemas severos de seguridad, los cuales se tienen que resolver si se desea que tengan más difusión, ya que los usuarios desean tener su información y transmitirla de forma segura. Por lo que, con esta tesis quisimos aportar nuestra investigación a la seguridad de estas redes, pues consideramos que en esta área falta mucho trabajo por hacer, aún son inseguras y cuando se establecen

los protocolo de ruteo se suponen muchas cosas (como que los nodos participarán generosamente en todas las actividades de ruteo, que harán las cosas de forma correcta y de acuerdo al protocolo, etc.), donde muchas de ellas en ocasiones no suceden, y es ahí donde la red necesita detectar cuales son los nodos que no se están cumpliendo con los protocolos y políticas establecidas en perjuicio de la red y poder tomar medidas adecuadas para evitar su impacto en la red.

1.4. Distribución del trabajo

En los capítulos dos y tres se explicará las bases de esta investigación, se abordarán las redes inalámbricas ad hoc, se explican las características de los protocolos basados en colonias de hormigas y se extenderá la explicación al protocolo *AntHocNet*, el cual utilizamos como protocolo de ruteo para nuestra investigación (capítulo 2), en el tercer capítulo hablaremos de la seguridad en redes inalámbricas ad hoc, enfocándonos al ataque *Blackhole*, dicho ataque va a ser el que detectaremos en nuestros métodos. En el cuarto capítulo se plantearán los métodos de detección que propusimos, así como se revisarán las soluciones propuestas por otros investigadores para el mismo problema. En el capítulo 5 se incluyen los resultados de las simulaciones hechas, primero se presenta el impacto que tiene el ataque en el desempeño de la red, y después los resultados de las simulaciones para ver como mitigan nuestros métodos (ack y Monitores) dicho ataque. Por último en el capítulo 6 se muestran las conclusiones.

Capítulo 2

Redes Inalámbricas Ad Hoc

2.1. Redes Inalámbricas

Las redes inalámbricas utilizan ondas electromagnéticas (radio e infrarrojo) para permitir la comunicación entre los dispositivos conectados a la red, esto es, sustituyendo a los medios guiados (coaxial, fibra óptica, etc.) que se utilizan en las redes cableadas. Las redes inalámbricas más que verse como una sustitución de las redes cableadas deben verse como una extensión de las mismas. De esta forma proporcionan a los usuarios las ventajas de las redes inalámbricas, como podría ser la movilidad sin perder conectividad.

Las redes inalámbricas presentan dos configuraciones: ad hoc y con infraestructura.

- Redes inalámbricas basadas en infraestructura: en dicha configuración la red inalámbrica se crea como una extensión a la red cableada. Los elementos inalámbricos se conectan a la red cableada por medio de un punto de acceso, siendo estos los que controlan el tráfico entre las estaciones inalámbricas y la red cableada, como se muestra en la figura 2.1. Además esta configuración utiliza el concepto de celda, la cual se define como el área en que una señal es efectiva, esto es, cada punto de acceso dará una celda donde deberán estar los dispositivos que deseen conectarse a él. A pesar que las celdas suelen tener un área pequeña, si se desea tener mayor cobertura podrán ponerse varios puntos de acceso o repetidores, para ampliar dicha área. Además con la utilización de varios puntos de acceso podrá darse el servicio de *roaming*, esto es que los equipos pasen de una celda a otra sin perder conexión y sin sufrir cortes en la comunicación, esta es una característica importante en las redes inalámbricas.

- adhoc: a éstas redes también se le conocen como redes punto a punto. Dicha configuración es sencilla ya que, para que la red se establezca solamente se necesitan los dispositivos móviles con adaptadores para las comunicaciones inalámbricas. En estas redes el único requisito crítico es el rango de cobertura de la señal, esto es cuando dos equipos se quieran comunicar tendrán que estar dentro de

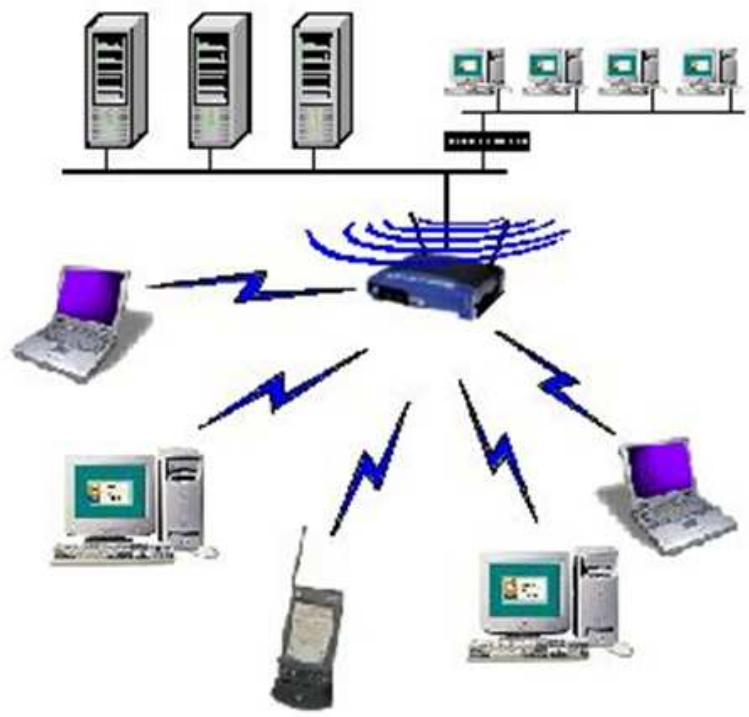

Figura 2.1: Red con Infraestructura

dicho rango para que la comunicación se lleve a cabo. Éstas redes se explicarán ampliamente a continuación, debido a la importancia en esta tesis.(Figura 2.2)

Figura 2.2: Red ad hoc

2.2. Redes Inalámbricas Ad Hoc

Una red inalámbrica ad hoc (*MANET (Mobile and Ad hoc NETworks)*) es una colección autónoma de dispositivos móviles que se comunican entre si. En una topología ad hoc, los propios dispositivos inalámbricos crean y mantienen la red y no existe ningún controlador central ni puntos de acceso que coordine las comunicaciones, cada dispositivo se comunica directamente con los dispositivos que se encuentren en su rango de transmisión (a un salto), y cuando desean comunicarse con nodos que se encuentran a más de un salto entonces los paquetes tendrán que ser retransmitidos por los nodos. Debido a que los nodos son móviles, la topología de la red puede cambiar rápidamente e imprevisiblemente con el tiempo. La red es descentralizada, dónde toda la actividad de la red, incluyendo el descubrimiento de la topología y la entrega de mensajes, deben ejecutarse por los nodos mismos, es decir, la función de ruteo se incorporará en los nodos móviles. Esta topología es práctica en lugares en los que pueden reunirse pequeños grupos de computadoras que no necesitan acceso a otra red. Ejemplos de entornos en los que podrían utilizarse redes inalámbricas ad hoc, una sala de conferencias donde los equipos se reúnen con regularidad para intercambiar ideas, así como para realizar operaciones de emergencia/rescate, ayuda en desastres y las redes militares. Tales escenarios de red no pueden depender de una conectividad centralizada y organizada. El conjunto de aplicaciones para *MANETs* es diverso, yendo desde redes pequeñas, estáticas que están limitadas por las fuentes de poder, a las redes de gran escala, móviles, muy dinámicas. El diseño de protocolos de ruteo para estas redes es un problema complejo. Sin tener en cuenta la aplicación, las *MANETs* necesitan algoritmos distribuidos eficaces para determinar la organización de la red, así como para llevar a cabo el ruteo de paquetes desde un nodo fuente hasta un nodo destino. Sin embargo, determinar caminos de ruta viables y la entrega de mensajes en un ambiente descentralizado donde la topología de red fluctúa no es un problema fácil. Mientras el camino más corto, (basado en una función de costo dada) desde una fuente a un destino en una red estática es, usualmente, la ruta más optima, normalmente, esta idea no se extiende fácilmente a *MANETs*, los factores como la calidad en los enlaces inalámbricos, la interferencia de multiusuario, el nivel de energía en los nodos y la topología cambiante, vuelven a las redes ad hoc complejas para establecer la ruta más optima.

2.3. Protocolos de ruteo para redes inalámbricas Ad Hoc

El protocolo de ruteo en redes ad hoc es una convención o estándar que controla como los nodos se ponen de acuerdo en la manera que rutean los paquetes entre los dispositivos móviles dentro de la red. En las redes ad hoc, los nodos no tienen, en primera instancia, conocimiento de la topología de la red alrededor de ellos, estos tienen que descubrirla. La idea básica es que un nodo nuevo (opcionalmente) anuncie su presencia y escuche los anuncios de sus vecinos. El nodo aprenderá de nodos cercanos

y maneras de alcanzarlos, y podría anunciar aquellos nodos que son alcanzables desde él. Al paso del tiempo, cada nodo sabe acerca de todos los demás nodos y una o más maneras de cómo alcanzarlos.

Los algoritmos de ruteo tienen que:

- Mantener las tablas de ruteo razonablemente pequeñas.
- Construir la mejor ruta para un destino dado (esta puede ser la mas rápido, la mas confiable, la de mayor rendimiento, la mas barata).
- Mantener la tabla actualizada cuando los nodos mueren, se mueven o entran en la red.
- Requierir una cantidad pequeña de mensajes/tiempo.

Las soluciones propuestas al problema de ruteo en redes ad hoc, pueden ser clasificadas en tres: proactivos, reactivos e híbridos.

- Protocolos proactivos: en dichos protocolos cada nodo mantiene una ruta en su tabla para cada otro nodo existente en la red. Dicha información es usada para transferir datos entre los nodos de la red. Para asegurar la frescura de las tablas de ruteo, estos protocolos adoptan diferentes tipos de mecanismos. Uno de ellos es mandar un mensaje especial "Hello", el cual, contiene información de direcciones y un intervalo de tiempo. Éstos protocolos no son considerados como una solución de ruteo efectiva para redes ad hoc, los nodos en dichas redes operan con bateria y ancho de banda limitada. La presencia de una alta mobilidad, grandes tablas de ruteo y baja escalabilidad resulta en el consumo de ancho de banda y tiempo de bateria en los nodos. Además actualizaciones continuas pueden crear carga innecesaria en la red.
- Protocolos reactivos: con estos protocolos, solo si un nodo fuente requiere de una ruta hacia un destino del cual no tiene una ruta vigente, inicia un proceso de descubrimiento de ruta, el cual va de un nodo a otro hasta que alcanza al destino o un nodo intermedio que tenga una ruta hacia el destino. El nodo fuente utiliza dicha ruta para la transmisión de datos hacia el nodo destino.
- Protocolos híbridos: mezclan características de los anteriores

2.3.1. Protocolos de colonia de hormigas

Los algoritmos de descubrimiento de rutas basados en colonias de hormigas están inspirados en el comportamiento de las hormigas en la naturaleza y aplicado al problema de ruteo en redes ad hoc. En un algoritmo de colonias de hormigas, múltiples agentes, representados por hormigas, cooperan unas con otras para encontrar las mejores rutas, usando comunicación indirecta mediante feromona. El primer algoritmo de

colonia de hormigas fue introducido para resolver el problema del viajero (Traveling Salesman Problem (TSP)) [6].

Una hormiga que se mueve coloca feromona (en cantidades variables) sobre la tierra, así marca el camino que sigue con el rastro de esta sustancia. Mientras una hormiga aislada se mueve en esencia aleatoriamente, una hormiga que encontrase un rastro de feromona previamente dejado, puede detectarlo y decidirse con una alta probabilidad a seguirlo, entonces refuerza el rastro con su propia feromona.

Una característica importante de estos algoritmos es que las hormigas pronto convergen en un subespacio del total de buenas soluciones. En otras palabras, todas las hormigas convergen no solo en una solución, pero en un subespacio de soluciones; después ellas van en búsqueda de mejoras de las soluciones encontradas.

2.3.2. El problema del camino más corto en una colonia de hormigas real

Una colonia de hormigas real es capaz de encontrar comida y seguir el camino mas corto desde el nido hasta la comida. La hormiga real al moverse deposita una sustancia llamada feromona sobre la tierra. Cuando una hormiga alcanza un punto donde tiene más de un camino para tomar, la probabilidad que vaya por uno de ellos dependerá de la cantidad de feromona depositada sobre ese camino. La hormiga seleccionará el camino y depositará más feromona; como resultado, la probabilidad de seleccionar este camino se incrementará. La feromona sobre los caminos más cortos hacia la comida crecerá rápidamente que, aquella de los otros caminos. La feromona se evapora a través del tiempo, permitiendo al sistema olvidar viejos caminos. Para mostrar como una colonia de hormigas encuentra el camino más corto, se tiene la Figure 2.3. Los círculos rojo y azul representan a la hormiga A y B, respectivamente.

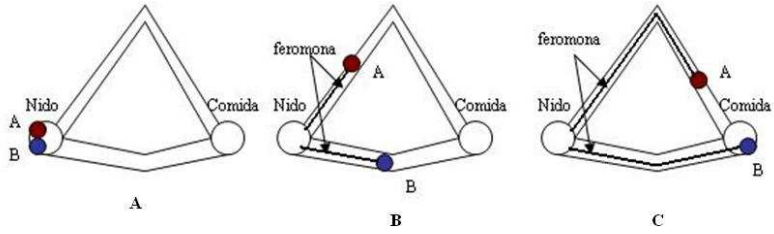

Figura 2.3: Problema del camino más corto

En 2.3a, dos hormigas que se encuentran en el nido necesitan ir a donde esta la comida. Ningún rastro de feromona esta en ninguno de los dos caminos. Cada hormiga selecciona uno de los dos caminos de forma aleatoria. Las hormigas depositan rastros de feromona mientras se mueven (figura 2.3b). La hormiga que selecciona el camino más corto llegará primero; tomará la comida y regresará a través del camino con la mayor cantidad de feromona. En este caso, la hormiga B llegará primero a la comida

(figura 9c) y cuando regrese, seleccionará el camino por donde llegó pues es el que tiene mayor valor de feromona y a la vez deposita más feromona como se muestra en (figura 2.4d). Ahora cuando la hormiga A alcance la comida, ella también seguirá el camino de B pues dicho camino es que el tiene mayor valor de feromona y además la hormiga A deposita feromona, así imponiendo más la selección de este camino. Gradualmente, la feromona sobre el camino más corto se incrementará como se muestra en (figura 2.4e).

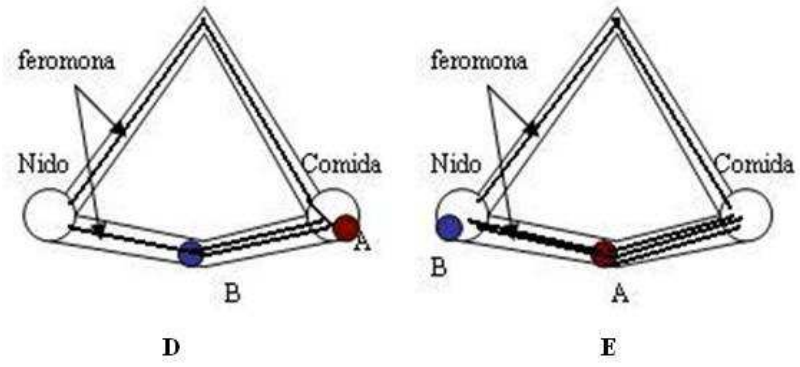

Figura 2.4: Problema del camino más corto (cont.)

Una sola hormiga no es inteligente, pero en conjunto la colonia de hormigas puede encontrar el camino más corto. Los caminos largos y sin explorar aún tienen la posibilidad de ser visitados. Si el camino mas corto falla, las hormigas seguirán un camino que hayan explorado recientemente. Aún si, las primeras hormigas usarán el camino más largo, la colonia de hormigas es capaz de encontrar el más corto pues la feromona se evapora con el tiempo y el camino más corto seguirá teniendo la posibilidad de ser visitado.

La colonia de hormigas real es un sistema dinámico autoconstruible y autoconfigurable, el cual es capaz de resolver sus problemas de forma eficiente. Estas características coinciden con los requerimientos de las *MANETs*.

Ahora bien, para aterrizar la analogía de la colonia de hormigas con los protocolos de ruteo, las hormigas en la naturaleza son paquetes de datos en redes y su caminar buscando comida, se refiere a la transmisión nodo a nodo buscando el nodo destino. Ahora bien, la feromona, la cual es utilizada para que las hormigas sepan por donde irse, en redes se refiere a Tablas de ruteo en cada nodo. Así, la tabla de ruteo en cada nodo se vería como la figura 2.5, donde dicha tabla es del nodo *i*, y en la primera columna esta los destinos nombrados como *D1, D2,...* etc. En el primer renglon se encuentran todos los vecinos del nodo *i*, los cuales están enumerados como *v1, v2,...* etc, que son los posibles siguientes saltos que pueden dar los paquetes (hormigas) estando en el nodo *i*. Así para ir al destino *D1*, a través del vecino *v1* se tiene el valor de feromona *f1*, y por el vecino *v2*, se tiene el valor de fereomona *f2*, y así sucesivamente. De esta forma, para un destino dado se tienen diferentes caminos para llegar a él, esto es, una forma

por cada vecino.

i	*v1*	*v2*	*v3*	. . .
D1	*f1*	*f2*	*f3*	
D2				
D3				
. . .				

Figura 2.5: Tabla de ruteo en cada nodo

2.3.3. Protocolos basados en colonia de hormigas

Para decidir que protocolo utilizar de los basados en colonia de hormigas se analizaron los protocolos siguientes, *ARAMA* [8] (*Ant Routing for Mobile Adhoc networks*), *ADRA* [18] (*An AntBased Distributed Rounting Algorithm*), *ARA* [1] (*The AntColony Based Routing Algorithm*) y *AntHocNet* [4], todos ellos comparten algunas ideas y otras no, estas últimas son las que hacen la diferencia y son las que analizamos.

Entre las ideas que comparten es el descubrimiento de ruta, esto es, cuando el nodo fuente desea tener una comunicación con algún otro nodo en la red y en su tabla de ruteo no existe ninguna ruta vigente para dicho destino, entonces el nodo fuente hace un descubrimiento de ruta mandando hormigas reactivas en su búsqueda. Cuando alguna de las hormigas reactivas llega al destino o algún nodo intermedio, el cual sabe una ruta hacia el destino, la hormiga reactiva se convierte en hormiga de retorno. La hormiga de retorno, regresa de forma inversa por la misma ruta que tomo la hormiga reactiva, hasta llegar a la fuente. Cabe destacar que en cada nodo al que llegan tanto la hormiga reactiva como la de retorno actualizan las tablas de ruteo, ya sea para el camino hacia la fuente o hacia el destino, respectivamente.

- *ARAMA* [8]: dicho protocolo propone que en la fase de descubrimiento y mantenimiento de ruta, la forma en que lo harán las hormigas sea la misma pero que la carga de hormigas en la red sea controlada, que no se haga completamente

la inundación de la red, si no que la tasa de generación de hormigas dependa de la calidad del servicio requerida por la fuente. Otro aspecto importante en éste protocolo es como los nodos actualizan las tablas de ruteo. Aquí, el valor de feromona se va degradando con respecto al tiempo, y en las entradas de las rutas que son utilizadas se incrementa el valor con cada paso de una hormiga o de un paquete de datos, así, el sistema olvidará viejos caminos a través del tiempo y conservará aquellas rutas que son utilizadas. Ahora bien, para escoger las mejores rutas se toma en cuenta el número de saltos y la cantidad de batería restante en los nodos para obtener una justa distribución de energía. Dichos valores pueden tomar diferente peso al ser considerados, y dejando al implementador esta decisión, así para algunos puede ser más importante el número de saltos que la batería en los nodos o viceversa

.

- *ADRA* [18]: en este protocolo, en el mantenimiento de ruta hacen mención de dos aspectos, el primero es el problema de la congestión, cuando un nodo sobre pasa el umbral de congestión, entonces el fabrica una hormiga llamada *antiant*, la cual se mandará a los demás nodos para informar que bajen sus valores de feromona para dicha ruta, pues ésta congestionada. El segundo aspecto es el problema del atajo, donde ocurre cuando una ruta recién descubierta es mas corta que la actualmente utilizada. Esto sucede por que las redes ad hoc tienen una topología dinámica y la característica que nodos entren y salgan de la red ocasiona que las rutas cambien. Esto es, si un nodo intermedio ya tiene una ruta establecida hacia un destino d, y en algún momento le llega un paquete de datos de otra ruta y ésta última es mejor que la anterior entonces a todos los nodos anteriores a él (de s hasta uno antes de él) mandará una hormiga *enforceant* para indicar que él ya sabe llegar al destino por una mejor vía.

- *ARA* [1]: Utiliza el concepto de ruteo bajo demanda. Se inunda la red de hormigas reactivas solamente cuando una ruta es necesaria hacia un destino, y cuando ya están establecidas las rutas, son los paquetes de datos quienes les dan mantenimiento a las rutas existentes para reducir la carga de hormigas en la red. Usa la semántica de tiempo como lo hace *AODV (Ad hoc On Demand Distance Vector)* [12], en vez del envejecimiento de feromonas. La estabilidad del algoritmo es cuestionable cuando se aplica a redes grandes.

- *AntHocNet* [4]: en este protocolo las mejores rutas son escogidas dependiendo del número de saltos y la congestión en la red. Así, si una zona de la red esta muy congestionada, será menos atractiva para los nodos y se optará por otras rutas, haciéndose así un balanceo de la carga en la red. La forma en que los nodos mandan los paquetes es de forma estocástica, es decir los paquetes se mandarán con una probabilidad de escoger un camino que responderá al valor de feromona contenida en la tabla de ruteo. Utiliza hormigas proactivas para el mantenimiento de las rutas y para explorar nuevos y mejores caminos.

Así analizamos las características de estos cuatro protocolos y llegamos a la conclusión que el mejor protocolo era *AntHocNet*, por las siguientes razones: Incluye una fase proactiva a diferencia de los otros protocolos, donde su tarea es buscar mejoras a los caminos existentes, así como evaluar si los caminos existentes se han congestionado y es necesario cambiar de rutas. Toma en cuenta para decidir cuales son las mejores rutas, el número de saltos (como los demás protocolos) y la congestión, éste último un problema difícil de atacar en dichas redes, haciendo así un balanceo de cargas en la red. El envío de paquetes es de forma estocástica, así el protocolo no satura las mejores rutas si no que distribuye sus paquetes entre las rutas existentes.

2.3.4. Protocolo *AntHocNet*

AntHocNet [4] es un algoritmo híbrido multiruta. Cuando una sesión de datos está por comenzar entre el nodo s y el destino d, s checa si tiene información de ruteo actualizada para el destino d. Si no, reactivamente envía hormigas como agentes, llamadas hormigas reactivas, para descubrir los caminos hacia d. Estas hormigas recogen información acerca de la calidad del camino que ellas siguen, y al llegar a d se convierten en hormigas de retorno las cuales regresan por el mismo camino y actualiza las tablas de ruteo. La tabla de ruteo Γ^i en el nodo i contiene para cada destino d y cada posible próximo salto n a un valor $\Gamma^i_{nd} \epsilon R$. Γ^i_{nd} es un estimado de lo bueno (en términos de número de saltos y congestión) que es el camino por n hacia d, lo cual llamaremos *feromona*. De esta forma, las tablas de ruteo en los diferentes nodos indican múltiples caminos entre s y d. Los paquetes de datos son enviados estocásticamente sobre los caminos: en cada nodo se selecciona el próximo salto con una probabilidad proporcional a su valor de feromona. Una vez que los caminos están establecidos y la sesión de datos esta corriendo, s comienza a enviar hormigas proactivas hacia d. Éstas hormigas siguen los valores de feromonas similar a como lo hacen los paquetes de datos. De esta forma monitorean la congestión de los caminos en uso. Además, ellas tienen una pequeña probabilidad de ser difundidas, así ellas pueden explorar nuevos caminos. En el caso de fallas de enlace, cualquier nodo trata de reparar el camino localmente, o mandar una advertencia a sus vecinos los cuales actualizan sus tablas de ruteo.

Establecer el camino reactivo

Las hormigas reactivas se mandan por un solo camino o son difundidas, buscando un destino d, de acuerdo a si el nodo tiene información en su tabla de ruteo con respecto al destino d o no. Debido a la difusión, las hormigas pueden proliferar rápidamente sobre la red, siguiendo diferentes caminos hacia el destino. Cuando un nodo recibe varias hormigas de una misma generación (es decir, que fueron generadas por la misma fuente, en un cierto intervalo de tiempo), se comparará el camino transcurrido por la hormiga con la previamente recibida de su generación: y solo si el número de saltos y el tiempo de viaje están los dos dentro de un cierto factor, se retransmitirá, con esto solo las mejores hormigas de la generación serán retransmitidas, eliminando así las que llevan

un mal camino. Usando esta política, la sobrecarga es limitada quitando las hormigas las cuales siguen malos caminos, mientras la posibilidad de encontrar múltiples buenos caminos no es impedido.

La principal tarea de las hormigas reactivas es encontrar un camino que conecte a *s* y *d*. La hormiga mantiene una lista *P*, *[1,. . . ,n]* de los nodos que ha visitado. Tras llegar al destino *d*, la hormiga reactiva se convierte en hormiga de retorno, la cual viaja hacia la fuente sobre *P*. La hormiga de retorno calcula un estimado Γ_p del tiempo que le llevará a un paquete de datos viajar sobre P hacia el destino y el cual es usado para actualizar las tablas de ruteo. Γ_p es la suma de estimaciones locales $\Gamma_{i\rightarrow i+1}$ en cada nodo $i \epsilon P$, es el tiempo que se requiere para alcanzar el próximo salto $i+1$: $\Gamma_p = \sum_{i=1}^{n-1} \Gamma_{i\rightarrow i+1}$. El valor de $\Gamma_{i\rightarrow i+1}$ está definido como $(Q^i_{mac}+1)\Gamma^i_{mac}$: el producto del estimado del tiempo promedio para enviar un paquete, Γ^i_{mac}, veces el número actual de paquetes en cola (mas uno) a ser enviados en la capa MAC, Q^i_{mac}. Γ^i_{mac} es calculado como un promedio del tiempo transcurrido entre la llegada de un paquete a la capa MAC y el fin de la transmisión exitosa. De esta manera, si τ^i_{mac} es el tiempo que toma enviar un paquete desde el nodo i, entonces el nodo i actualizará su estimado como sigue: $\Gamma^i_{mac} = \alpha\Gamma^i_{mac} + (1-\alpha)\tau^i_{mac}$ con $\alpha\epsilon[0,1]$. Desde Γ^i_{mac} es calculado en la capa MAC incluye actividades de acceso al medio, así toma en cuenta congestión local del medio compartido.

En cada nodo intermedio $i\epsilon P$, la hormiga de retorno virtualmente da un camino hacia el destino *d*, creando o actualizando las entradas en la tabla de ruteo Γ^i_{nd}. Cuando se llega a un nodo i desde su vecino n, la hormiga crea una entrada en su tabla de ruteo Γ^i, indicando que n es el próximo salto a tomar desde este nodo a fin de llegar a *d*. La entrada contendrá el valor de la feromona Γ^i_{nd}, el cual es el indicador de la calidad del camino yendo hacia el destino d sobre el próximo salto n. El valor de la feromona representa un promedio del tiempo que tarda y el número de saltos, para viajar a d a través de n. Si $\Gamma_{i\rightarrow d}$ es el tiempo estimado de viaje por la hormiga, y h es el número de saltos, el valor de la feromona esta definido como: $\tau_{id} = ((\Gamma_{s\rightarrow d} + h\Gamma_{hop})/2)^{-1}$, donde Γ_{hop} es un valor fijo, el cual representa el tiempo que toma un salto en condiciones ideales, es decir no toma en cuenta la congestión. Tomando este promedio es una forma de evitar posibles oscilaciones en el tiempo estimado obtenido por la hormiga (es decir, debido a las explosiones locales del tráfico) y tomar en cuenta tanto el retraso de extremo a extremo y el número de saltos. Si existiese una entrada Γ^i_{nd} en Γ^i, este valor es actualizado usando un promedio cargado: $\Gamma^i_{nd} = \gamma\Gamma^i_{nd} + (1-\gamma)\tau_{id}$, $\gamma\epsilon[0,1]$.

Ruteo de datos de forma estocástica

La fase de establecimiento de caminos descrito anteriormente crea a un número de buenos caminos entre la fuente y el destino, indicado en las tablas de ruteo de los nodos. Los datos son retransmitidos entre los nodos de acuerdo a las entradas en los valores de la feromona. Los nodos en *AntHocNet* envían los datos de forma estocástica. Cuando

un nodo tiene múltiples próximos saltos hacia el destino d, se seleccionará uno de ellos, con una probabilidad P_{nd} para el próximo salto n, donde $P_{nd} = \frac{\Gamma^2_{nd}}{\sum_{i \epsilon N_d} \Gamma^2_{id}}$. Se tomó el cuadrado para ser más ambiciosos con respecto a los mejores caminos. De acuerdo a esta estrategia, se escogerán los caminos en función de su calidad.

La estrategia probabilística de ruteo conduce que la carga de los datos se propague de acuerdo a un balance automático en la carga. Cuando un camino es claramente peor que otros (mayor congestión), éste será evitado, y su congestión será mitigada. Otros caminos tendrán más tráfico, resultando en más congestión, lo cual dará como resultado un incremento en el retraso de extremo a extremo. Con una adaptación continua del tráfico de datos, los nodos tratan de difundir la carga de datos de modo uniforme sobre la red. Esto es muy importante en redes Ad Hoc, porque el ancho de banda en el canal inalámbrico es muy limitado. Para hacer esto apropiadamente, es importante monitorear frecuentemente la calidad de los diferentes caminos. Para esto se usa las hormigas proactivas.

Mantenimiento proactivo de rutas y exploración

Mientras una sesión de datos está corriendo, el nodo fuente envía hormigas proactivas conforme a la tasa de envio de datos (una hormiga cada *n-ésimo* paquetes de datos). Ellas siguen el valor de la feromona en la misma forma que los datos (aunque el valor de la feromona no esta al cuadrado, así ellas prueban los caminos de forma mas uniforme), pero tienen una pequeña probabilidad de, en cada nodo, ser difundidas. De esta forma ellas sirven a dos propósitos. Si una hormiga alcanza al destino sin ninguna difusión simplemente muestra un camino existente, estimando la calidad de este camino y actualizando el valor de la feromona a lo largo del camino desde la fuente hasta el destino. Esto es, porque la congestión puede variar a través del tiempo, por lo que éstas hormigas actualizarán dicho valor. Una hormiga de retorno hace lo mismo en dirección del destino hacia la fuente. Por otro lado la hormiga que se difunde en cualquier punto, dejara el rastro de feromona conocido y explorara nuevos caminos.

Al difundirse llegará a todos los vecinos del nodo del cual se difundió. Es posible que en este vecindario no encuentre rastros de feromona hacia el destino, así tendrá que difundirse de nuevo. La hormiga, entonces rápidamente se proliferara e inundará la red, como una hormiga reactiva lo hace. Para evitar esto, se limitará el número de difusión a dos. Si la hormiga proactiva no encuentra información de ruteo con dos saltos, será borrada. El efecto de este mecanismo es la búsqueda de nuevas rutas concentradas alrededor de caminos actuales, así se busca mejoras en los caminos y variaciones.

Para orientar a las hormigas que se transmiten, se usan los mensajes *Hello*: usando estos mensajes, los nodos conocen acerca de sus vecinos inmediatos y tienen información de feromona acerca de ellos en su tabla de ruteo. Así cuando una hormiga llega a

un vecino del destino, puede directamente ir al objetivo. Regresando a inspiración de nuestro modelo, las colonias de hormigas, puede ser visto como difusión de feromona: la feromona depositada en la tierra se difunde, y puede ser detectada por las hormigas a cierta distancia. En un trabajo futuro se extenderá éste concepto, para dar una mejor orientación a la exploración por las hormigas proactivas. Además los mensajes *Hello* sirven a otros propósitos: permiten detectar las fallas de enlace. Esto ayuda a los nodos a quitar entradas viejas de feromonas de su tabla de ruteo.

Fallas de enlace

Los nodos pueden detectar fallas de enlace (es decir, un vecino que se ha movido y se ha salido del rango de transmisión del nodo), esto es, cuando la transmisión de un paquete u hormiga falla o cuando se esperaba un mensaje *Hello* y no es recibido. Cuando ésto sucede, un nodo puede perder un camino hacia uno o más destinos. Si el nodo tiene otro próximo salto alternativo hacia el mismo destino, o si el destino perdido no ha sido utilizado por datos, esta pérdida no es tan importante, y el nodo solamente actualiza su tabla de ruteo y envía una notificación para que sus vecinos se actualicen. Por otra parte, si el destino fue regularmente utilizado para tráfico de datos, y fue la única alternativa del nodo para dicho destino, la pérdida es importante y el nodo deberá tratar de reparar el camino. Ésta es la estrategia seguida por *AntHocNet*, con la restricción que el nodo solo reparará el camino si la pérdida de enlace fue descubierta con una falla en la transmisión de paquetes de datos.

Después de la falla de enlace, el nodo difunde una hormiga reparadora de ruta que viaja hacia el destino involucrado como una hormiga reactiva: sigue información de ruteo disponible cuando puede, y se difunde en otro caso. Una diferencia importante es que tiene un número máximo de difusiones, así su proliferación esta limitada. El nodo espera por un cierto tiempo, y si ninguna hormiga de retorno es recibida, se concluye que no fue posible encontrar un camino alternativo hacia el destino el cual es removido de la tabla de ruteo.

En el caso de que el nodo aun tenga otras entradas para los destinos involucrados en la falla de enlace, pero el próximo salto perdido era la mejor alternativa para el destino, o si la falla de enlace fue debido a la transmisión de una hormiga, el nodo solo enviará una notificación a sus vecinos. Además en el caso de un fallo en la reparación del camino se enviará una notificación similar. La notificación contiene una lista de los destinos que perdieron un camino hacia ellos, un estimado del retraso extremo a extremo y el número de saltos hacia este destino (si aún hay entradas para el destino). Todos sus vecinos reciben la notificación y actualizan sus tablas de feromona usando el nuevo estimado. Si ellos pierden el mejor o el único camino hacia un destino debido a la falla, ellos reenviaran la notificación, hasta que todos los nodos a lo largo de diferentes caminos son notificados de la nueva situación.

Capítulo 3

Seguridad en redes inalámbricas y Criptografía

En éste capítulo se abordará la seguridad en redes inalámbricas ad hoc, se puntualizarán las debilidades y porque son vulnerables a ataques. Se describirá en detalle el ataque *Blackhole*, el cual detectamos con los métodos que propusimos. En la segunda sección se hablará de criptografía con el fin de explicar como se lleva a cabo la criptografía de llave pública, esto es porque uno de nuestros métodos utiliza dicha criptografía. Además de citar las soluciones de seguridad que se han dado en redes ad hoc, utilizando criptografía de llave pública, esto es para mostrar que es muy común su uso en dicha área.

3.1. Seguridad en Redes Inalámbricas Ad Hoc

En años recientes las redes inalámbricas ad hoc (*MANETs*) han recibido una gran atención por sus sobresalientes capacidades (la más importante es, que la red es descentralizada, y para crear y mantener la red nada más se necesitan los nodos móviles). Generalmente las investigaciones asumen un ambiente de amistad y cooperación; y se han enfocado a problemas como acceso al medio o ruteo, pero la seguridad ha llegado a ser un asunto primario para proveer comunicación protegida entre nodos en un ambiente potencialmente hostil, como lo puede ser el inalámbrico. A pesar que la seguridad por mucho ha sido un tema de investigación en redes cableadas, la única característica que las *MANETs* presentan es un nuevo conjunto de desafíos en lo que a seguridad respecta. Estos desafíos incluyen una arquitectura de red abierta punto a punto, medio compartido inalámbrico, recursos limitados, una topología altamente dinámica. En consecuencia, algunas soluciones existentes para redes cableadas no aplican directamente en el dominio de las *MANETs*.

La limitación de recursos en *MANETs* constituye otro desafío no trivial para el diseño de la seguridad, como puede ser la capacidad de cómputo de los nodos móviles, así los *PDAs* difícilmente realizarán tareas como computación criptográfica asimétrica,

además los dispositivos móviles típicamente trabajan con batería, y podrían tener el recurso de energía muy limitado. El canal inalámbrico tiene un ancho de banda limitado y se comparte entre las entidades vecinas.

El medio inalámbrico y la movilidad de los nodos plantean aspectos dinámicos en *MANETs*, comparado con las redes cableadas. La topología de la red es altamente dinámica ya que los nodos frecuentemente entran, salen y deambulan en la red. El canal inalámbrico es además susceptible a interferencia y errores, exhibiendo características inestables en términos de ancho de banda y retraso.

Las características descritas sobre *MANETs* claramente hacen que al construirse soluciones de seguridad se tomen en cuenta tanto una protección amplia como un desempeño de la red deseada. La solución de seguridad debería extenderse a lo largo de muchos componentes individuales y confiar en su protección colectiva para proteger toda la red. El esquema de seguridad adoptado por cada dispositivo ha de trabajar dentro de sus propias limitaciones en recursos, en términos de capacidad de cómputo, memoria, capacidad de comunicación y suministro de energía.

Una característica distintiva en las *MANETs* desde una perspectiva de diseño de seguridad es la falta de una línea clara de defensa. Diferente a las redes cableadas que tiene routers dedicados, cada nodo móvil en redes ad hoc pueden fungir como router y transmitir paquetes para otros par de nodos. El canal inalámbrico está accesible tanto para usuarios legítimos como para intrusos maliciosos. No existe un lugar bien definido para el monitoreo del tráfico o el mecanismo de control de acceso pueda ser desplegado. Como resultado, el límite que separa la red interna del mundo exterior se hace borroso. Por otro lado, los protocolos de ruteo para redes ad hoc como *Ad Hoc On Demand Distance Vector (AODV)* [12] y *Dynamic Source Routing (DSR)* [10], y protocolos inalámbricos *MAC*, típicamente asumen un ambiente confiado y cooperativo. Como resultado, un intruso malicioso puede fácilmente ponerse como router y perturbar las operaciones de la red intencionalmente desobedeciendo las especificaciones del protocolo.

El fundamental objetivo de las soluciones para *MANETs* es proveer servicios de seguridad a los usuarios móviles, tales como autenticación (el sistema ha de asegurarnos que una tercera parte no pueda usurpar la identidad de alguna de las dos partes que intervienen en la comunicación), confidencialidad (el contenido de la comunicación ha de ser inútil para una tercera parte que lo pudiera interceptar), integridad (nos debe garantizar que la información transmitida, además de no ser interceptada, no pueda ser modificada por una tercera parte), no repudio (debe garantizar que ninguno de los participantes en una comunicación pueda negar parte de la misma. Es un concepto muy ligado al de autenticación).

Además, los dispositivos portátiles, así como los sistemas que almacenan informa-

ción son vulnerables a ponerse en peligro o ser capturados físicamente, especialmente dispositivos con protección débil. Los ataques pueden hacerse furtivamente dentro de la red a través de estos nodos.

La seguridad nunca viene gratis. En paralelo con el aumento de fuerza en seguridad esta el incremento en cómputo, comunicación y mantenimiento de la sobrecarga. Consecuentemente, para las soluciones de seguridad el desempeño de la red, en términos de escalabilidad, disponibilidad de servicio, robustez, y demás, se convierten en un asunto importante por los recursos limitados en *MANETs*. Mientras muchas propuestas contemporáneas se han enfocado sobre la fuerza de la seguridad para sus soluciones desde el punto de vista de criptografía, ellos dejaron el aspecto de desempeño de la red a un lado. De hecho, tanto la fuerza en la seguridad como el desempeño de la red son igual de importantes, y lograr un buen balance entre los dos extremos es un desafío fundamental en el diseño de seguridad para *MANETs*.

3.2. Ataques en redes inalámbricas Ad Hoc

Una *MANET* provee conectividad de red entre los nodos móviles a través de canales inalámbricos multisaltos principalmente por protocolos a nivel capa de enlace que aseguran la conectividad a un salto, y protocolos a capa de red que extienden esa conectividad a múltiples saltos. Estos protocolos distribuidos típicamente asumen que todos los nodos cooperan en el proceso de coordinación. Esta idea es desgraciadamente no cierta en ambientes hostiles. Porque la cooperación es asumida pero no forzada en *MANETs*, intrusos maliciosos pueden fácilmente perturbar las operaciones de la red violando las especificaciones del protocolo.

Las principales operaciones en la capa de red en *MANETs* son el ruteo y la transmisión de paquetes, donde los nodos interactúan para satisfacer la entrega de paquetes de una fuente a un destino. Basándose en las entradas de las tablas de ruteo, los paquetes son transmitidos por los nodos intermedios a lo largo de una ruta establecida hacia el destino. Sin embargo, tanto el ruteo como la transmisión de paquetes son vulnerables a ataques. Mientras un número considerable de ataques esta fuera del alcance de esta tesis para ser tratados, tales vulnerabilidades caen dentro de dos categorías: ataques de ruteo y ataques en la transmisión de paquetes, basados en la meta del ataque.

La familia de los ataques de ruteo se refiere a cualquier acción de anunciar actualizaciones de ruteo que no siguen las especificaciones del protocolo. El comportamiento especifico del ataque esta relacionado con el protocolo de ruteo usado por la *MANET*. Atacando el protocolo de ruteo, el intruso puede atacar el tráfico hacia ciertos destinos, tenerlos bajo su control y causar que los paquetes sean transmitidos a lo largo de una ruta que no sea la óptima o que no exista. Los intrusos pueden crear ciclos en la red,

e introducir congestión en ciertas áreas de la red. Múltiples intrusos coludidos pueden evitar que la fuente encuentre al destino y particionar la red, en el peor de los casos. Pueden ser que el intruso derribe nodos existentes en la red, o fabricar su identidad e suplantar otro nodo legítimo. Un par de nodos atacantes pueden crear un hoyo gusano (wormhole [7]) y hacer un atajo para el flujo normal entre cada uno. En el contexto de protocolos de ruteo bajo demanda, el intruso puede fijar como objetivo el proceso de mantenimiento de ruta y avisar que se ha perdido una liga en la red. Existen aun esfuerzos en investigación para identificar y vencer mas sofisticadamente e ingeniosamente los ataques de ruteo.

Además de ataques de ruteo, los adversarios pueden lanzar ataques en contra de la operación de transmisión de paquetes. Tales ataques no perturban el protocolo de ruteo, ni alteran las entradas en la tabla de ruteo de cada nodo. En cambio, causan que los paquetes de datos sean transmitidos de tal forma que es intencionalmente inconsistente a las entradas de las tablas. Por ejemplo, el intruso a lo largo de una ruta puede tirar los paquetes, modificar su contenido del paquete o duplicarlo. Otro tipo de ataque de transmisión de paquetes es el ataque de negación de servicio en donde el atacante inyecta un gran número de paquetes basura en la red. Estos paquetes gastan una porción significativa de recursos en la red, e introducen severos problemas de contención en el canal inalámbrico y congestión en la red.

3.3. Ataque *Blackhole*

Como ya lo hemos visto el conjunto de ataques que se pueden realizar contra redes ad hoc es innumerable, para acotar el problema a tratar, nosotros nos enfocaremos en el ataque *Blackhole* el cual, se describirá a continuación.

Éste ataque tiene dos propiedades:

1. El intruso se aprovecha del protocolo de ruteo, como en este caso *AntHocNet*, para comunicarle que él tiene una ruta válida hacia el nodo destino, aún cuando la ruta sea espuria, con la intención de interceptar los paquetes.

2. El atacante intercepta los paquetes no re-transmitiendo ningún paquete.

Sin embargo, el atacante corre el riesgo que sus vecinos monitoreen la red y se den cuenta del ataque que esta siendo realizado en su vecindario. Existe una forma mas sutil de este ataque, esto es, el intruso selecciona qué paquetes re-transmitirá y cuales trucará. Un intruso suprime o modifica los paquetes originados de algunos nodos, mientras deja pasar los de otros nodos, con lo cual limita la sospecha de su fechoría.

Existen además otras formas de lanzar el ataque *Blackhole*, como lo describen en [13; 3], utilizan el protocolo de ruteo *DSR*, el intruso lanza una petición de ruta, como

si viniera de otro nodo (el cual es el nodo victima), con un número de secuencia muy grande, entonces los nodos al introducir la ruta inversa (aquella que va al nodo fuente), introducirán que por la ruta del intruso pueden llegar a la fuente (victima), y como el número de secuencia es muy elevado, dicha ruta reemplazará a cualquier otra existente y es ahí donde el intruso atrae el trafico del nodo victima.

3.4. Ataque *Blackhole* para el protocolo de ruteo *AntHocNet*

El ataque *Blackhole* sobre el protocolo de ruteo *AntHocNet* será de la siguiente forma:

Cuando alguna hormiga reactiva llegue al nodo intruso, éste responderá con una hormiga de retorno la cual indique (de forma falsa) que el destino está a simplemente un salto de él. Cuando el nodo fuente empiece la transmisión de datos y lleguen al nodo intruso este tirara los paquetes. El nodo intruso no mandará hormigas proactivas. Y cualquier hormiga proactiva que llegue al nodo intruso será engañada al igual que las hormigas reactivas.

Para explicar de forma gráfica como se hará el ataque supongamos el siguiente ejemplo, se tiene la red de la figura 3.1. El nodo S desea establecer comunicación con el nodo D, el nodo intruso es el que está de color negro y los demás nodos son nodos intermedios.

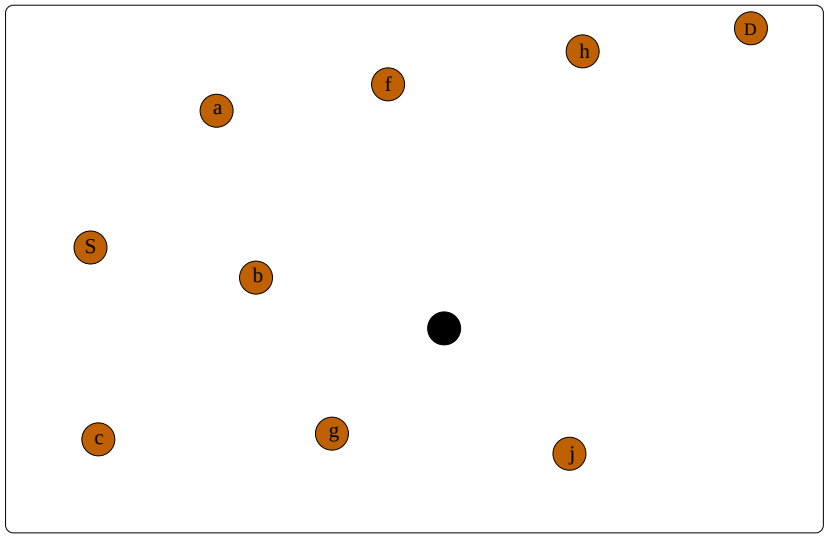

Figura 3.1: El nodo S desea una ruta hacia D.

El nodo S hace la inundación de paquetes RREQs para obtener una o varias rutas que lleven hacia D como lo muestra la figura 3.2.

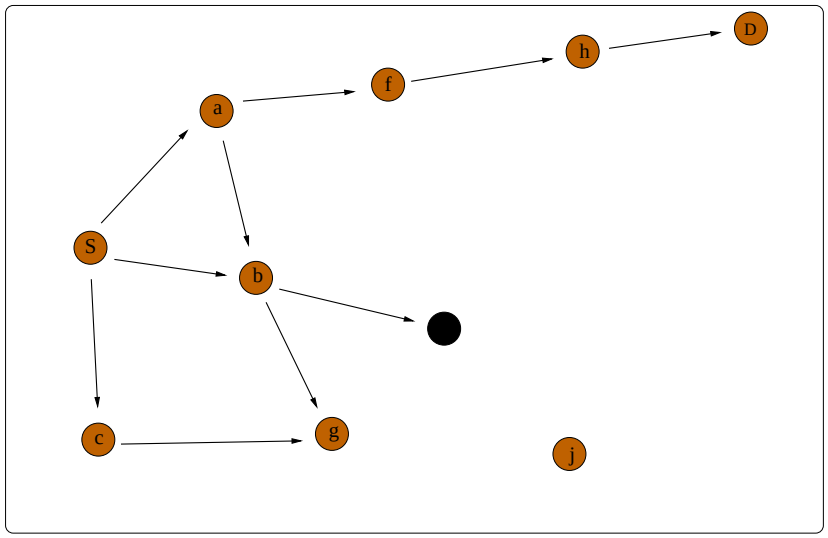

Figura 3.2: El nodo S realiza la inundación del paquete RREQ.

Entonces el nodo intruso le informará al nodo fuente S que el nodo destino D está a simplemente un salto de él, como lo muestra la figura 3.3. S registra en su tabla de ruteo que la ruta que pasa por el vecino a su valor de feromona es de 0.25, por el vecino b es de 0.333, entonces el nodo intruso ofertará una mejor ruta, aunque en realidad el nodo D ni siquiera es alcanzable desde él.

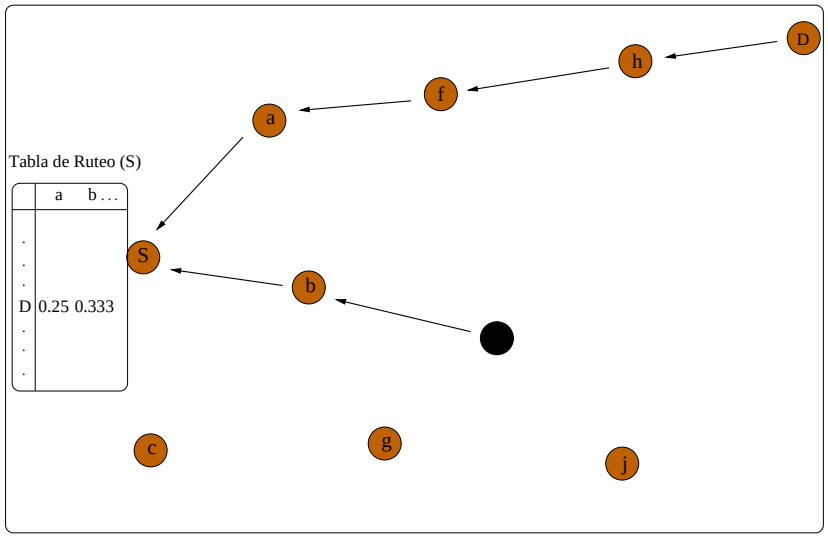

Figura 3.3: El nodo S registra en su tabla de ruteo una mejor ruta a través del intruso.

Debido a que los paquetes de datos son enviados de forma estocástica se tendrá mayor probabilidad que el nodo fuente tome la ruta equivocada. En la figura 3.4 se muestra la tabla de ruteo de transmisión para el nodo S, donde se muestra que los paquetes de

datos tienen el 36 % de probabilidad de tomar la ruta que pasa por el vecino *a* y el 63 % de tomar por el vecino *b*. y es aquí donde se da el ataque pues los datos tendrán mayor probabilidad de que sean interceptados por el intruso.

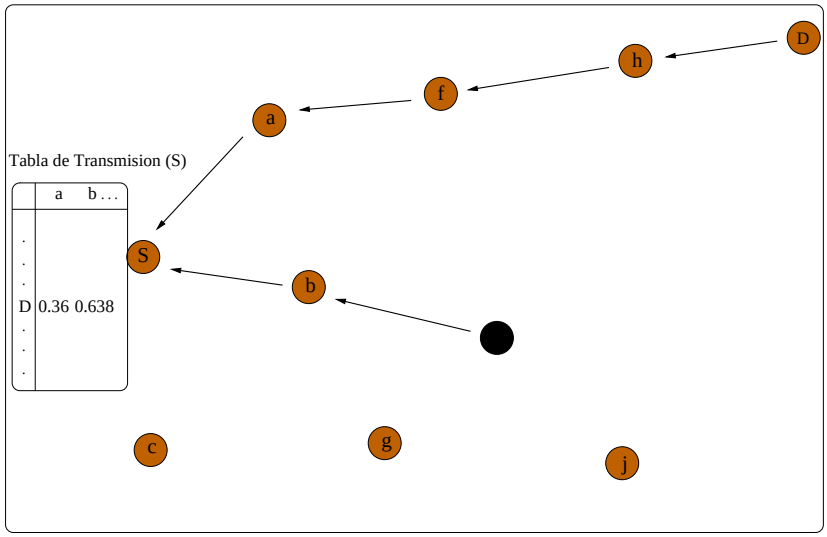

Figura 3.4: El intruso tiene mayor probabilidad de ser escogido para re-transmitir los paquetes de datos..

3.5. Criptografía

Cryptografía [5] se define (del griego *kryptos* que significa "ocultar" y *grafos*, que significa "escribir") literalmente como "escritura oculta". La criptografía estudia el cifrado y descifrado de la información.

Cifrar (tambien llamado encriptar), es transformar información original, llamada texto en claro, en información transformada, llamada texto cifrado, el cual tiene el aspecto de datos aleatorios, ilegibles.

El cifrado es reversible. Despues de transmitir, cuando la información alcanzó al destino, la operación inversa (descifrar, o desencriptar) transforma el texto cifrado en el original, texto en claro.

Las técnicas o reglas para cifrar -conocido como algoritmo de cifrado- determina que tan simple o complicado será el proceso de transformación. La mayoría de las técnicas de cifrado utiliza mejor fórmulas matemáticas simples que son aplicadas un número de veces en diferentes combinaciones. Además usan un valor secreto llamado clave para cifrar y descifrar. La clave es como un tipo de contraseña, usualmente conocido solo por el emisor y receptor de la información. El algoritmo de cifrado matemáticamente

aplica la clave, la cual es usualmente una cadena larga de números.

A diferencia de una contraseña convencional, la clave no te da directamente acceso a la información. En lugar de eso, es usada por el algoritmo para transformar la información en una forma particular. Con la clave, la información que ha sido bloqueada (cifrada) por ésta puede ser facilmente desbloqueada; sin la clave, la informanción es inaccesible.

El proceso de tratar de descifrar información sin la clave (o "romper" el mensaje cifrado) es llamado *criptoanálisis*.

El tipo de algoritmo de crifrado, la clave secreta, y otras características adjuntas forman lo que llamamos la *fuerza* de cifrado; la fuerza de cifrado es que tan dificil es leer el mensaje cifrado.

Recordar que una mala elección, o una protección inadecuada, abre la puerta a intrusos, tal como si se compartieran o robaran las claves. Si un intruso tiene acceso a las claves de cifrado, aún el algoritmo más fuerte no protegerá los datos. La figura 3.5 muestra un cifrado y descifrado simple.

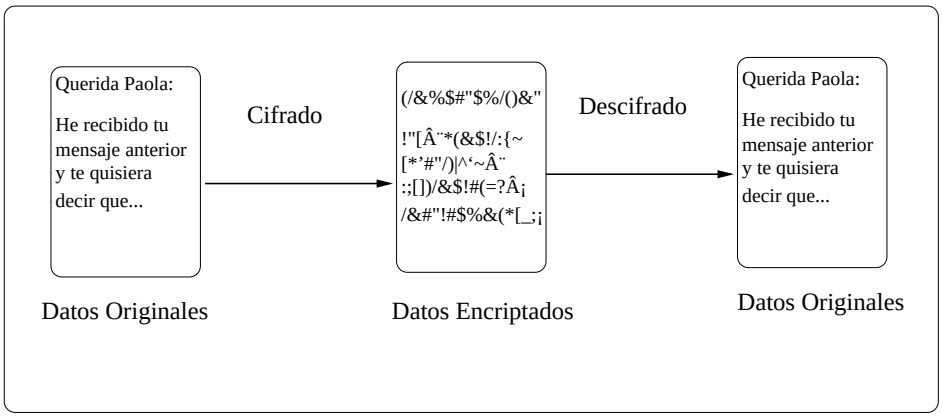

Figura 3.5: Cifrado y descifrado de forma simple

Los sistemas criptográficos modernos caen dentro de dos categorias generales (identificados por el tipo de clave que usan): criptografá simétrica y criptografía asimétrica.

Criptografía simétrica

Los sistemas de criptografía simétrica usan una sola clave. Ésta clave es usada tanto para cifrar como para descifrar información. Se necesita de una clave exclusiva para cada par de usuarios los cuales intercambian mensajes, y ambos usuarios deben conocer la clave secreta. La seguridad del método de cifrado depende completamente de que

tan bien sea protegida la clave. Así, como en la figura 3.6 se muestra que se tiene una carta en texto legible para todos, cuando se cifra con la clave, la carta se vuelve ilegible para un tercero que pudiera interceptar la información, y cuando llega al receptor, éste la descifra con la clave para poder leer la información.

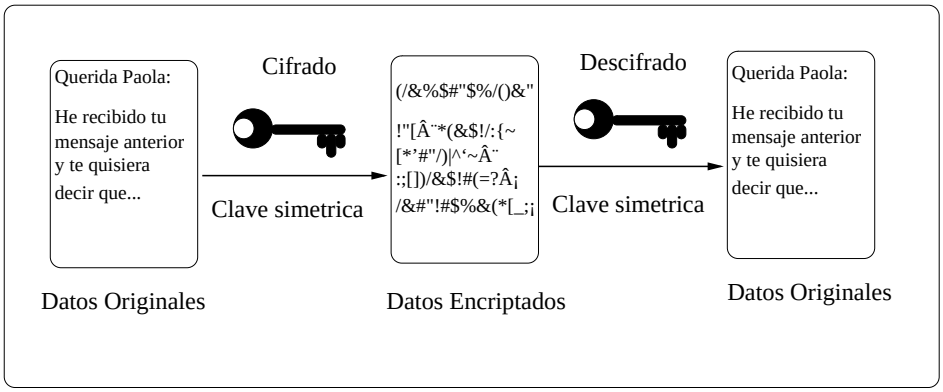

Figura 3.6: Criptografía de clave simétrica

Criptografía asimétrica

Los sistemas de criptografía asimétrica usan dos claves: una clave pública y otra privada. Dentro de un grupo de usuarios - por ejemplo, dentro de una red de computadoras - cada usuario tiene tanto una clave pública como una privada. Un usuario debe mantener su clave privada en secreto, pero la clave pública es conocida por todos los demás usuarios; las claves públicas deben además mantenerse en directorios electrónicos. Así como se muestra en la figura 3.7, se utilizan diferentes claves para cifrar que para descifrar.

Las claves pública y privada están relacionadas. Si tu cifras un mensaje con tu clave privada, el receptor del mensaje puede descifrarlo con tu clave pública. De forma similar, cualquiera puede enviar un mensaje cifrado a alguien más, simplemente con la clave pública del receptor del mensaje; el emisor no necesita conocer la clave privada del receptor. Cuando tu recibes un mensaje cifrado, tu y solo tu puedes descifrarlo con tu clave privada.

Además algunos sistemas de clave pública proveen una característica de autenticación el cual asegura que cuando el receptor descifra tu mensaje, el sabe que dicho mensaje viene de ti y no de alguien más.

Mantenimiento y distribución de claves

El problema más relevante con la criptografía como un método de seguridad es la

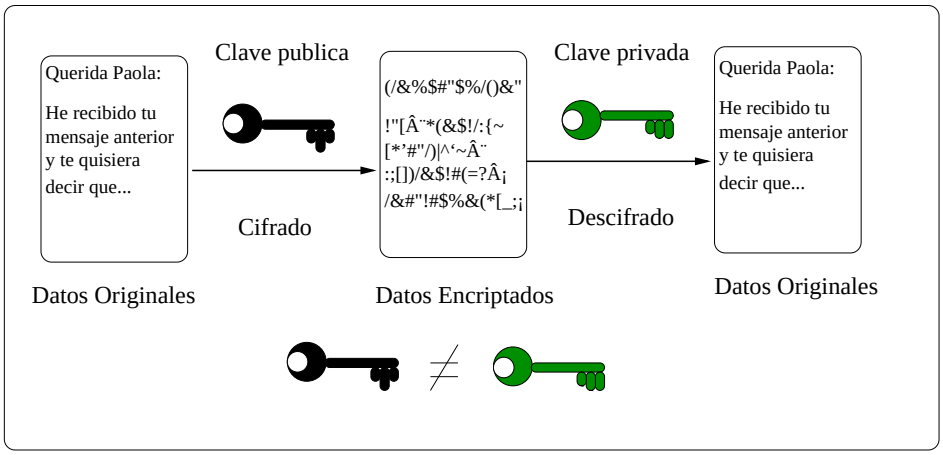

Figura 3.7: Criptografía de clave asimétrica.

distribución, almacenamiento y eventual disposición de las claves, el cual introduce una cara y considerable carga administrativa. Historicamente, las claves criptográficas fueron entregadas por mensajeros escoltados en cajas seguras. En algunos casos, esto es aún hecho. Con los más modernos productos de alta seguridad criptografíca, agencias de gobierno hacen la actual distribución de claves, entregando las claves sobre un medio magnético o en sitios individuales. Otra aproximación es distribuir una clave maestra, la cual es usada para generar claves de sesión adicional. Un sitio debe seguir estrictamente procedimientos rigurosos para proteger y monitorear el uso de claves y debe de haber una forma para cambiarlas. Aún con estas restricciones, existe siempre la posibilidad que la clave pueda ser robada o comprometida.

Por supuesto, si la clave es robada, existe otro problema. Porque el decifrar información cifrada depende solo de la disponibilidad de la clave, la información cifrada podría ser perdida si no se puede localizar la clave.

La dificultad de la distribución, almacenamiento, y disposición de la clave ha limitado el uso de criptografía en muchos productos en el pasado.

Criptografía y Redes ad hoc

Como ya se vio anteriormente, lo complicado de la criptografía es el mantenimiento y distribución de las claves y éste problema se agudiza más en los ambientes inalámbricos ad hoc, donde no hay una autoridad central que pueda dar dicho servicio. Además de que en un ambiente inalámbrico, los equipos pueden ser son móviles y podrían ser capturados físicamente por un intruso y tener acceso a las claves.

Pese a las deseventajas que se presentan en las redes inalámbricas ad hoc, la criptografía es actualmente utilizada para proveer seguridad en dichas redes, como se muestra

en *SAODV (Secure Ad hoc On-Demand Distance Vector)* [16] o en *Secure Position Aided Ad hoc Routing* [2], entre otras soluciones. El objetivo de presentar estas soluciones es para dar a conocer que el uso de la criptografía en estas redes, contrario a lo que se podría pensar, es común. En cierto modo, justificando así el uso de ésta en nuestro método.

Capítulo 4

Métodos de defensa

4.1. Mecanismos de detección de intrusos existentes contra el ataque *Blackhole*

Las soluciones que se han propuesto en lo que a métodos de detección para el ataque *Blackhole* respecta, son establecidas en protocolos de ruteo comunes como pueden ser *Ad hoc On Demand Distance Vector (AODV)* o *Dynamic Source Routing (DSR)*. El protocolo que utilizamos nosotros (*AntHocNet*) tiene poca o nula investigación en el área de seguridad, por lo que dichas soluciones no hacen referencia a nuestro protocolo.

Sergio Marti [11] propuso técnicas que mejoran el rendimiento de las *MANETs* en presencia de nodos comprometidos que lanzan ataques de negación de servicio como tirar paquetes. Para mitigar el descenso en el rendimiento de la red debido al ataque, los autores usan *watchdog*, para identificar a los nodos con mal comportamiento y un *pathrater* que ayudará al protocolo de ruteo a evitar estos nodos. Su solución es la siguiente; Cuando un nodo reenvia un paquete, el *watchdog* de este nodo verifica que el próximo nodo en el camino reenvie también el paquete. El *watchdog* hace esto escuchando de forma promiscua las transmisiones del nodo próximo. Si el nodo no reenvia el paquete dentro de un cierto periodo, esto se toma como un mal comportamiento, por lo que el *watchdog* define al nodo como mal portado. Además, si no se esta utilizando criptografía en la red, el nodo que escucha puede además verificar que el próximo nodo no modificó el paquete antes de transmitirlo. Cada vez que el nodo falla en las retransmisiones de paquetes, el *watchdog* incrementa la cuenta de fracasos. Si la cuenta excede un cierto umbral, se determinará que el nodo es intruso; por lo que este nodo será evitado al usar el *pathrater*. El *pathrater* corre en cada nodo de la red, seleccionando rutas para los paquetes basándose en el valor de confiabilidad asignado a los nodos, por el mecanismo *watchdog*.

La técnica de *watchdog-pathrater* tiene ventajas y desventajas. *Dynamic source routing (DSR)* [10] con el *watchdog-pathrater* tiene la ventaja que puede detectar malos

comportamientos al nivel de retransmisión, pero no al nivel de enlace. Las debilidades de *watchdog-pathrater* son que pueden no detectar malos comportamientos en presencia de:

- Colisiones ambiguas: esto previene que el nodo A escuche la transmisión del nodo B. Figura 4.1.
- Receptor de colisiones: el nodo A puede solo decir si B ha enviado un paquete, pero no si C recibió o no el paquete, Figura 4.2.
- Poder de transmisión limitada: un nodo que tenga un mal comportamiento puede limitar su poder de transmisión de tal forma que la señal es lo suficientemente fuerte para ser escuchada por el nodo previo pero lo suficientemente débil para llegar al próximo nodo.
- Mal comportamiento falso: esto ocurre cuando un nodo falsamente reporta que otro nodo es mal portado.
- Descartado parcial de paquetes: un nodo puede evadir el *watchdog-pathrater* tirando paquetes a una tasa tan baja que no rebasará el umbral establecido en la configuración del *watchdog-pathrater*.

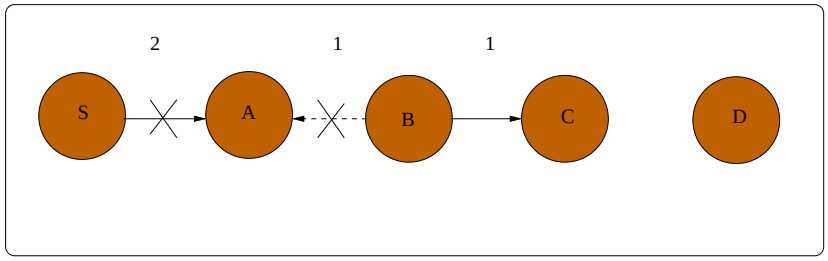

Figura 4.1: El nodo A no escucha la transmisión de B hacia C del paquete 1, porque la transmisión de B colisiona en A con el paquete 2 de la fuente.

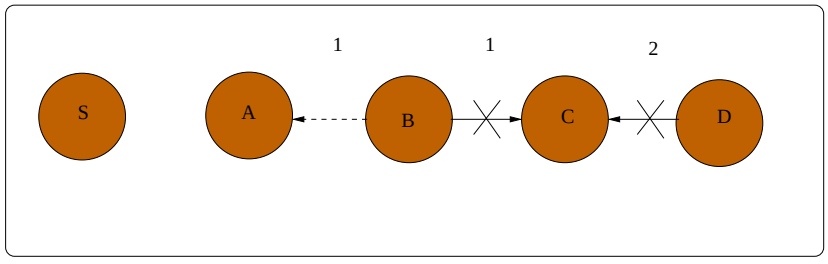

Figura 4.2: El nodo A cree que B ha transmitido el paquete 1 a C, a pesar que C nunca recibió el paquete debido a la colisión del paquete 2.

En [17] es la misma idea pero trabaja con el protocolo *AODV*. Añade un campo next_hop en los paquetes *AODV* así el nodo puede ser consciente del próximo salto correcto de entre sus vecinos. Además considera más tipos de ataques, como modificaciones en los paquetes, duplicación de paquetes y ataques de negación de servicios.

El protocolo *CONFIDANT (Cooperation Of Nodes: Fairness In Dynamic Ad Hoc NeTworks)* [9] ha sido otra solución que se ha propuesto, la cual consiste en un conjunto de extensiones de *DSR* que incluye los siguientes componentes: el *monitor*, el *sistema de reputación*, el *mantenimiento de camino*, y el *mantenimiento de confianza*. Un nodo que participa en el protocolo debe operar los cuatro componentes. Los caminos son escogidos basándose en los valores asignados a través de la observación directa o de reportes de ruteo y comportamiento en las transmisiones.

El componente *monitor* en un nodo es responsable de monitorear *acknowledgments* pasivos para cada paquete que transmite. Esto es similar a la función *watchdog*. Cuando un nodo transmite un paquete monitorea la transmisión del próximo salto, tratando de detectar desviaciones del comportamiento normal esperado. El componente de *mantenimiento de confianza* envía y recibe mensajes de alarmas. Estos mensajes son generados y enviados cuando el nodo localmente concluye que otro nodo es mal portado. Estos mensajes son intercambiados entre los nodos que son predefinidos como amigos. Las alarmas desde otros nodos toman sustancialmente menos peso. La conclusión es alcanzada basándose en el mecanismo de *acknowledgments* pasivos del componente *monitor* o de un mensaje de alarma recibido desde otro nodo. El componente de *sistema de reputación* mantiene una tabla de identidades de nodos y un valor asociado. Los valores son modificados de acuerdo a una función que utiliza pesos pequeños para alarmas reportadas de comportamientos incorrectos y pesos mayores para observaciones directas. Si el valor cae debajo de un cierto umbral el componente de *mantenimiento de camino* es llamado para remover el camino que contiene el nodo mal portado, además de ignorar los paquetes de ruteo que vienen del atacante y alerta a los nodos legítimos cuando ellos piden una ruta que utiliza una ruta comprometida.

Cada entrada en la lista de atacantes identificados por un nodo es asociado por un temporizador. Cuando expira la entrada en la lista es borrado dicho nodo y es de nuevo considerado como un participante legítimo de la red ad hoc.

Otro protocolo que también resuelve el problema de seguridad en redes ad hoc, es el *Security-Aware ad hoc Routing (SAR)* [15]. Dicho protocolo hace hincapié en que, en la mayoría de los protocolos de ruteo, los nodos confían implícitamente en los demás nodos, para la transmisión de sus paquetes, y este comportamiento permite a nodos intrusos lanzar ataques, de diferentes formas, como puede ser insertar paquetes de ruteo erróneos, contestar peticiones de antiguas rutas, cambiar actualizaciones de ruteo o avisos incorrectos de ruteo. Lo que este protocolo propone es cambiar la forma en cómo se establecen las rutas, en vez de que sean las mejores rutas, las que tienen

menor número de saltos (mas cortas), dicho protocolo sugiere que los nodos tengan niveles de confiabilidad, y cuando la fuente haga un descubrimiento incruste en el RREQ (paquete de petición de ruta) la jerarquia de seguridad que deben de tener los nodos que construyan la ruta. Si el nodo no tiene ese grado de confiabilidad necesaria para el RREQ, entonces tirara el paquete.

La figura 4.3 muestra un ejemplo de cómo el protocolo *SAR* establece la ruta entre los dos Generales. En esta figura, los Generales son los de mayor grado de confiabilidad, seguidos por los Oficiales y por último están los Soldados raso, donde se prefiere la ruta segura, la cual se compone de 5 saltos, pues esta compuesta de Oficiales; que la ruta de simplemente 2 saltos pero la cual se compone de Privados.

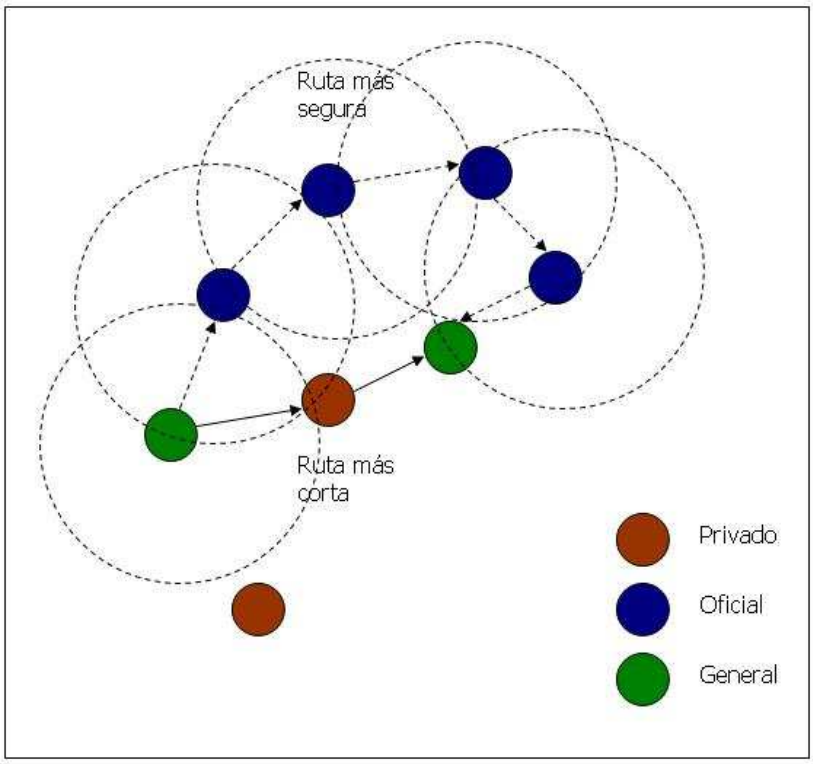

Figura 4.3: Establecimiento de rutas en el protocolo *SAR*.

Si la fuente encuentra dos rutas igualmente confiables, entonces lo que decidirá cual utilizar será la de menor número de saltos. Utiliza herramientas de criptografía para que ningún nodo altere el RREQ y el buen funcionamiento del protocolo *SAR*.

Una última solución que veremos es llamada *A Novel Intrusion Detection Method for Mobile Ad Hoc Networks* [14] (que en nuestra tabla resumen llamaremos *Maquina de Estados*, esto es por su nombre tan generico). Aquí se propuso la solución basada en maquinas de estado finitas para detectar ataques sobre *DSR*.

Primero, se diseñó la arquitectura de detección de intrusos de forma distribuida y cooperativa, la cual está compuesta por nodos monitores distribuidos. Cada nodo monitor corre independientemente y todos los monitores en su zona encuentran intrusos locales. Para seguir la pista de algún nodo intruso, los monitores tienen que intercambiar información con los demás. Considerando que los nodos tienen recursos limitados, solo algunos nodos son seleccionados como monitores en la red. Así, describen un algoritmo en el cual los nodos pueden ser seleccionados como monitores de su zona de forma periódica, aleatoria y justa. La figura 4.4 muestra como se ven los monitores y su zona.

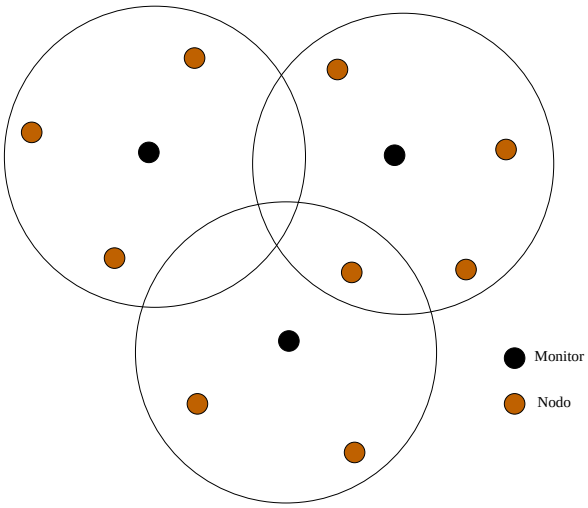

Figura 4.4: Monitores y su zona.

La característica de que, el proceso de votación sea de forma aleatoria da la seguridad al sistema.

Segundo, cada monitor emplea una máquina de estados finito (*FSM Finite State Machine*) para detectar el comportamiento incorrecto en los nodos. Los monitores mantienen maquinas de estados, para cada flujo de datos en cada nodo. En DSR, un nodo puede recibir y reenviar cuatro tipos de paquetes, paquetes *ROUTE REQUEST (RREQ), ROUTE REPLY (RREP), ROUTE ERROR (RRER)* y *DATA*.

En la figura 4.5 se muestra la máquina de estados para un RREQ. El estado de inicio es 1. Cuando el nodo recibe un paquete, la maquina de estados va al estado 2. Si el paquete es RREQ, la maquina de estados va al estado 3. Si dicho nodo es el objetivo del RREQ, el nodo responde con un RREP al generador del RREQ. La máquina de estados va al estado 4 y verifica si el RREP es de acuerdo al protocolo de ruteo. Si algunos campos del RREP, fueron modificados maliciosamente, la máquina de estados va al estado A1, y alerta de modificación al paquete, si no la máquina de estados va

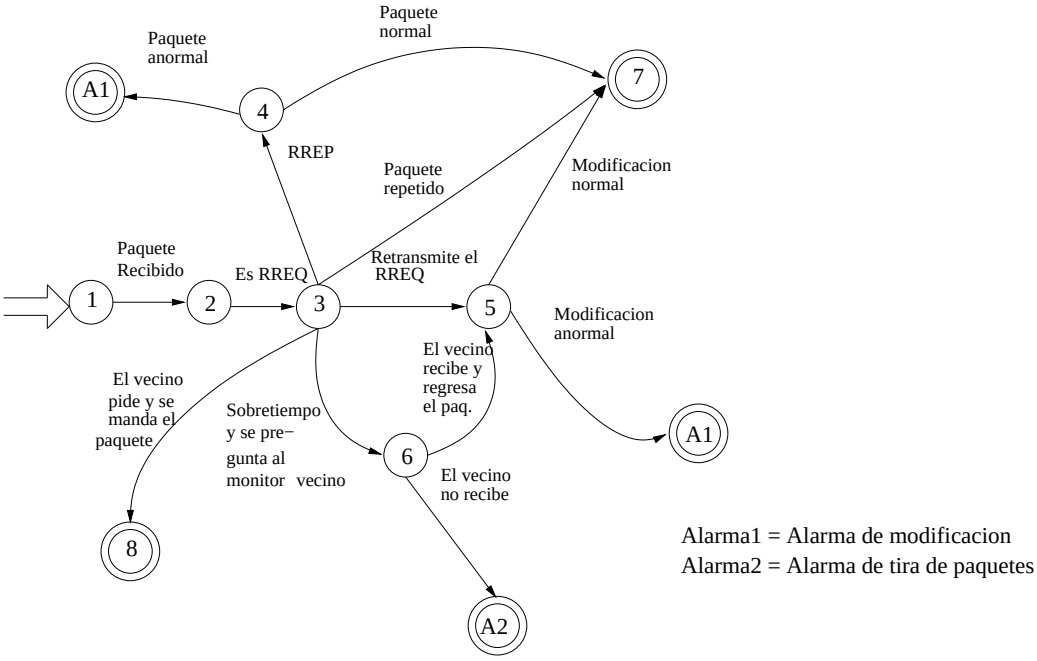

Figura 4.5: Máquina de estados finita cuando se recibe un paquete *ROUTE REQUEST*

al estado terminal 7. Si éste nodo ha visto recientemente el mismo RREQ, descarta el paquete y la máquina de estados va al estado terminal 7. Si el nodo reenvia el RREQ, la máquina de estados va al estado 5 y verifica si el reenvío del paquete es de acuerdo al protocolo de ruteo. Si algunos campos del RREQ fueron modificados maliciosamente, la máquina de estados va al estado A1 y alerta de modificación al paquete, si no va al estado terminal 7. Si el paquete no ha sido reenviado despues de un tiempo específico, la máquina de estados va al estado 6. Para ese tiempo el nodo pudo haberse movido de la zona del monitor y el monitor no pueda escuchar la retransmisión del paquete. Además el monitor pregunta a su vecino monitor si el paquete ha sido reenviado. Si el vecino ha recibido el paquete, le mandará al monitor el paquete para que lo compare. La máquina de estados irá al estado 5. Si ningún vecino ha recibido el paquete, la máquina de estados va al estado de alarma A2 y manda una alerta de tira de paquetes. Si el monitor vecino manda el paquete para comparar la máquina de estados va al estado terminal 8.

Se usa la misma máquina de estados finitos para los tres tipos de paquetes, es decir, ROUTE REPLY, ROUTE ERROR and DATA, pues ellos usan el mismo proceso. La figura 4.6 muestra su máquina de estados. El estado de inicio es el 1. Cuando un nodo recibe un paquete, la máquina de estados va al estado 2. Si el paquete es uno de los tres tipos de paquetes descritos anteriormente, la máquina de estados va al estado 3. Si es el objetivo del paquete, la máquina de estados va al estado terminal. Si el monitor vecino pregunta por el paquete y dicho paquete es enviado al vecindario, la máquina de estados va al estado terminal. Si el nodo reenvia el paquete, la máquina de estados va al

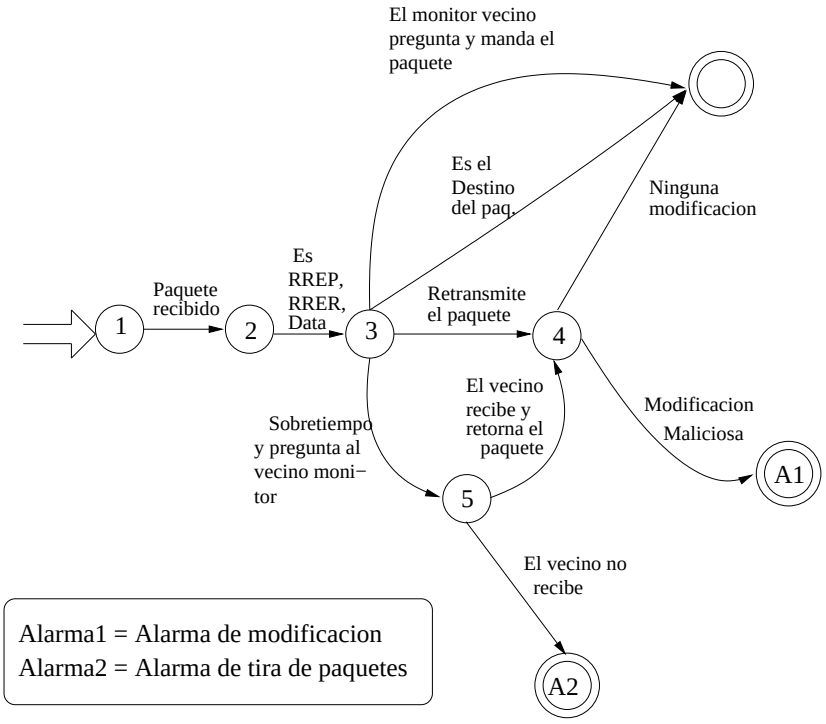

Figura 4.6: Máquina de estados finitos cuando se recibe un paquete *ROUTE REPLY, ROUTE ERROR, DATA*

estado 4 y verifica si la retransmisión del paquete es de acuerdo al protocolo de ruteo. Si algunos campos del paquete fueron modificados de forma maliciosa, la máquina de estados va al estado de alarma A1, y alerta sobre modificación al paquete, si no, la máquina de estados va al estado terminal. Cuando los nodos no reenvian el paquete dentro de cierto periodo, el monitor preguntará a sus vecinos monitores. Si algún vecino recibió el paquete, lo mandará al monitor para su comparación e irá al estado 4. De otra forma, la máquina de estados irá al estado de alarma A2.

Las figuras 4.5 y 4.6 muestran el proceso cuando un nodo recibe un paquete. La figura 4.7 muestra el proceso cuando un nodo manda un paquete. Y el paquete no es escuchado por el monitor. De otro modo el proceso es como el de la figura 4.5 o 4.6. El estado de inicio es el 1. Cuando el nodo recibe un paquete, la máquina de estados va al estado 2. Entonces, si el paquete es originado por el nodo, la máquina de estados va al estado 6. El monitor compara la dirección de la fuente del paquete con la dirección del nodo, el cual envió el paquete. Si las dos dirección concuerdan, la máquina de estados va al estado terminal. De otra forma, la máquina de estados va al estado de alarma A3 y alerta impersonalización. Esto implica que el nodo se está haciendo pasar por otro nodo. Si el paquete es retrasnmitido, la máquina de estados va al estado 3. Cuando el paquete es creado por el nodo, la dirección del nodo es la dirección de la fuente y la dirección del nodo es anexada en la lista de direcciones por las que se reenvia el paquete. Ya que los monitores no ven el paquete, ellos preguntan a sus vecinos monitores

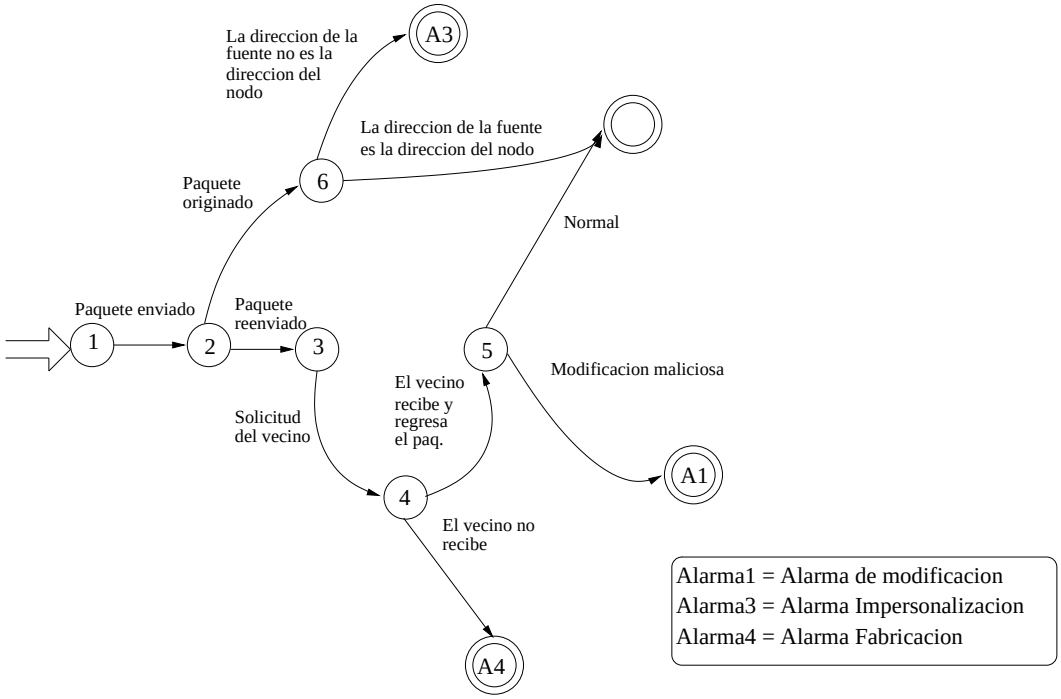

Figura 4.7: Máquina de estados finitos cuando manda un paquete

por el paquete. La máquina de estados va al estado 4. Si ninguno recibió el paquete, la máquina de estados va al estado de alarma A4, el nodo fabricó el paquete.

Si algún vecino recibió el paquete, lo mandará al monitor. La máquina de estados va al estado 5 y verifica si la retransmisión del paquete fue de acuerdo al protocolo de ruteo. Si algunos campos del paquete fueron modificados maliciosamente, la máquina de estados va al estado de alarma A1 y alerta por modificación al paquete, si no la máquina de estados va al esatdo terminal.

Tabla de resumen

A continuación para presentar la información de forma más clara, se presentará una tabla la cual, resume las características de los mecanismos de detección existentes contra el ataque *Blackhole*

En la tabla 4.1 se presentan las características de los mecanismos de detección existentes, se puede observar a simple vista en dicha tabla que todos estos mecanismos trabajan con protocolos bajo demanda, en el mecanismo *watchdog-pathrater*, el componente *watchdog* hace la detección y el componente *pathrater* el de reaccion evitando a los intrusos. En el mecanismo *CONFIDANT* el componente *monitor* es el que detecta quien es el intruso al igual que el *watchdog*, y el *mantenimiento de camino* junto con

Mecanismo de detección	**Detección**	**Reacción**	**Máquinas de estado**	**Pasivo**	**Comentario**
Watchdog-pathrater [11]	X	X			Utiliza la escucha promiscua para la detección y al identificar al intruso lo evita en las rutas
CONFIDANT [9]	X	X			Trabaja similar al *Watchdog-pathrater* pero esta más estructurado
SAR [15]				X	Utiliza niveles de confianza en los nodos
Maquinas de estado [14]	X		X		Utiliza nodos monitores y maquinas de estado en dichos nodos

Cuadro 4.1: Tabla resumen de los mecanismos de detección existentes

el *mantenimiento de ruta* son los que reaccionan, borrando las rutas existentes que incluyen al intruso y evitando los paquetes de éste, respectivamente. El mecanismo *SAR* lo clasificamos como pasivo, puesto que dicho mecanismo se basa en la construcción de rutas solo con nodos confiables, entonces no necesita de componentes de detección y reacción, pues como hay un nivel de confianza, se espera que los nodos se comporten de forma correcta. En el último método llamado *Maquinas de Estado*, por medio del nodo monitor, el cual vigila que los nodos vecinos hagan las cosas de acuerdo al protocolo, se detecta al intruso, y se generan alarmas para avisar a los demás nodos en la red.

4.2. Métodos de defensa propuestos

Se pensó en 2 soluciones para la detección de este tipo de intrusos (aquellos que lanzan ataques *Blackhole*). La primera solución se basa en la enumeración de los paquetes de datos y el envío de *acks* por parte del destino por cada paquete recibido, los cuales son encriptados con la clave pública de éste (supone una infraestructura de clave pública ya establecida). Al haber una ausencia de *acks* por el intruso la fuente baja el valor de feromona para dicha ruta, si el ataque continua puede llegar el valor de feromona hasta cero, mitigando así el ataque. El segundo método hace uso de nodos monitores los cuales se encargarán de vigilar el buen comportamiento de los nodos. Cuando los

monitores concluyan que un nodo está teniendo un mal comportamiento lo excluirán de la red negándole cualquier servicio de transmisión de paquetes. A continuación se abordarán las dos soluciones para ser explicadas a detalle.

4.2.1. Primer método para el ataque *Blackhole*: *Método ACK*

El protocolo de ruteo *AntHocNet* a fin de que mitigue el ataque se comportará de la siguiente manera:

- Se realizará la fase reactiva de la misma forma y se establecerán las diferentes rutas hacia el destino.
- La fuente al mandar los paquetes de datos, les asignará un número a cada paquete de forma consecutiva (figura 4.8) y los mandará por los diferentes caminos (figura 4.9).

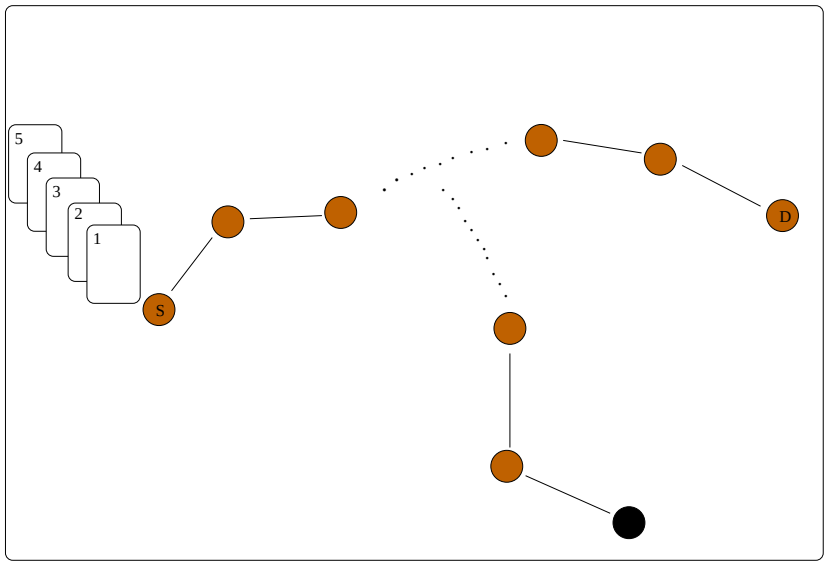

Figura 4.8: La fuente enumera los paquetes.

- Cuando los paquetes lleguen al destino, éste generará un *ack* por cada paquete recibido el cual contendrá el número del paquete y lo encriptará con su clave privada (supone una infraestructura de clave pública ya establecida, como la que se describió en el capítulo anterior) (figura 4.10).

- Cuando la fuente no reciba el correspondiente *ack* por parte del destino en cierta ruta, la fuente bajará el valor de feromona para dicho vecino. Como se muestra en la figura 4.11 donde el nodo fuente manda el paquete 2 por la ruta que lleva al intruso, y el intruso no podrá generar el ack pues no conoce la clave privada del destino.

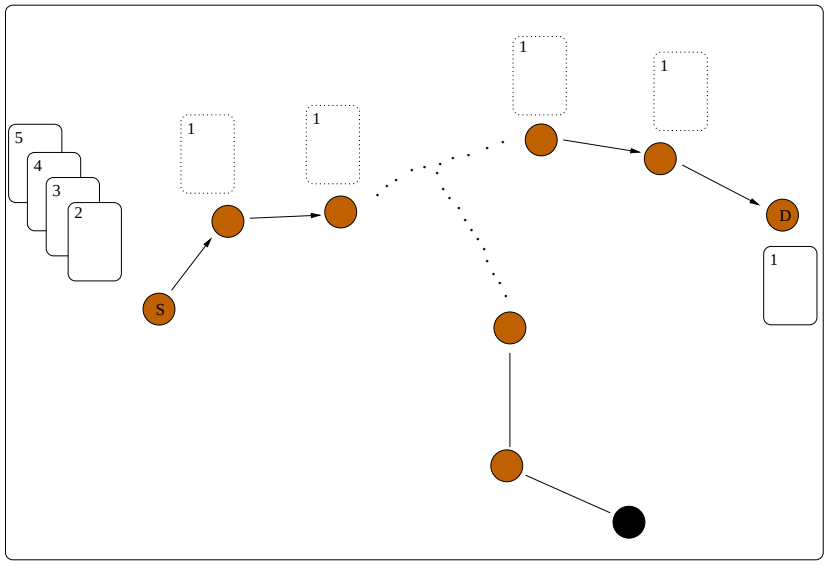

Figura 4.9: La fuente mandará los paquetes por las rutas establecidas.

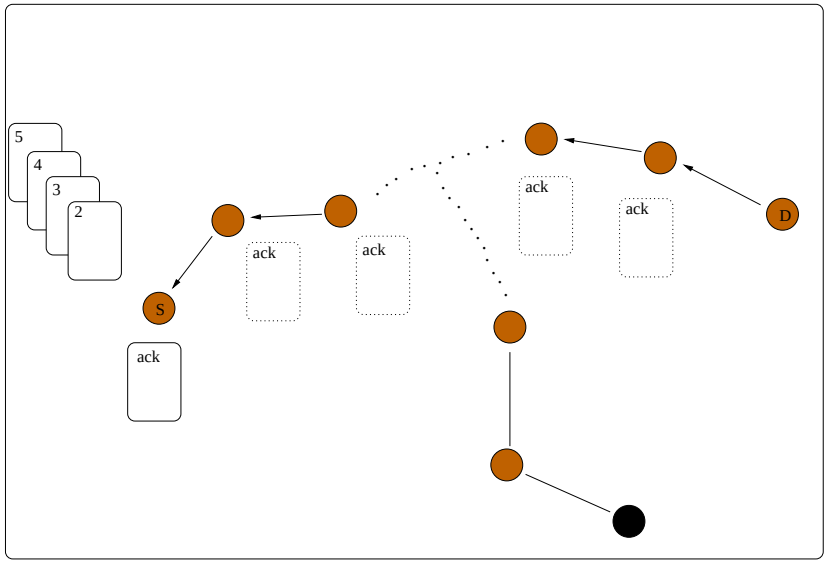

Figura 4.10: El nodo destino genera el ack y lo manda a la fuente

- La fuente mandará una notificación a los nodos que constituyen la ruta negativa, para que bajen su valor de feromona en sus tablas de ruteo hacia ese destino con esa ruta. Figura 4.12

- Fuera de estas modificaciones el protocolo se comportará de la misma forma en sus demás fases (proactiva, de mantenimiento, mensajes *hello*)

La fuente bajará el valor de feromona de acuerdo a la siguiente fórmula:

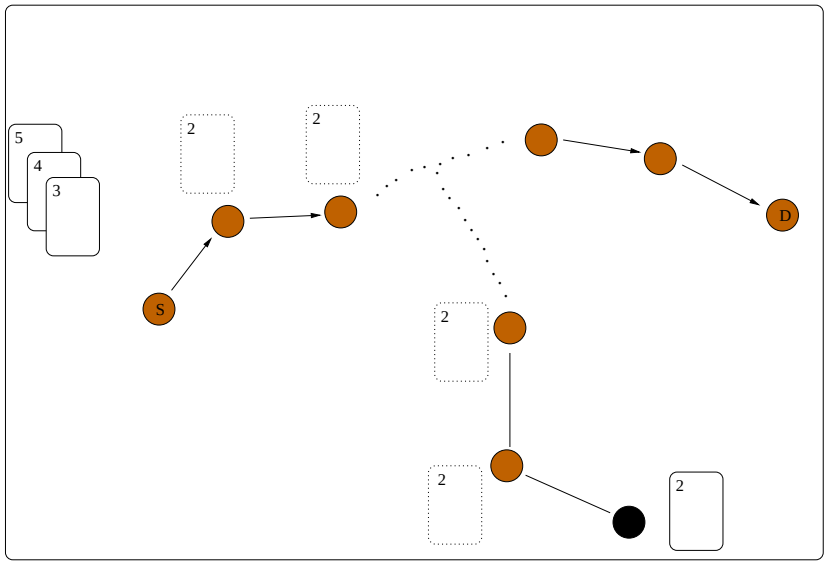

Figura 4.11: La fuente manda el paquete 2 por la ruta que lleva al intruso

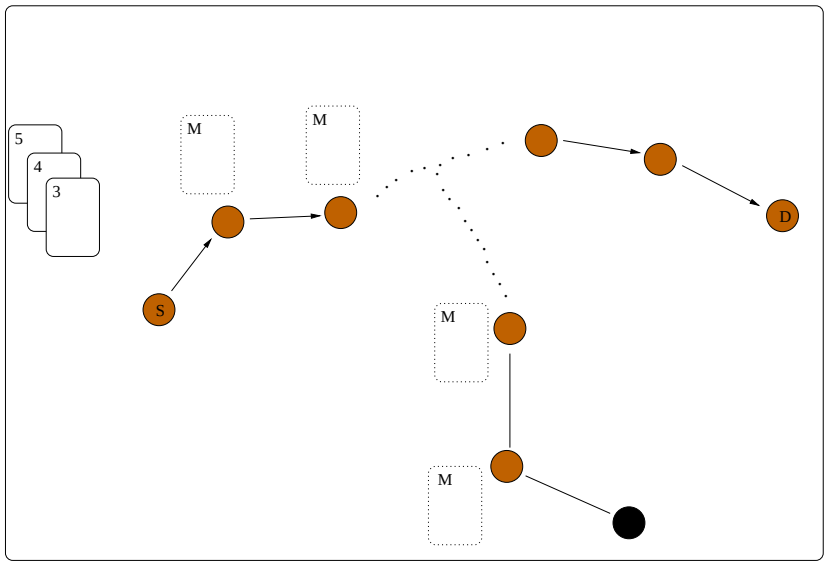

Figura 4.12: El nodo fuente manda una notificación para que los nodos bajen el valor de feromona en esta ruta

$$nuevo_{valor} = valor_{anterior} - (valor_{anterior} * \frac{reduccion}{100}) \tag{4.1}$$

la fórmula (4.1) representa una reducción en el valor de feromona de acuerdo a un porcentaje. Para nuestras simulaciones será la variable *reducción* la que estaremos modificando, que representa el porcentaje de reducción al que bajaremos el valor de feromona.

Si existiera una falla de enlace en un camino válido (que lleve al destino legítimo) y los *acks* no pudiesen llegar a la fuente o el paquete de datos al destino. Lo que sucederá es que el nodo que detecte la falla de enlace, como fue en un paquete de datos, tratará de reparar el camino. Si no fuera posible reparar el camino, mandará un mensaje diciendo que por esa ruta ya no es posible alcanzar al destino por lo que la fuente y los nodos intermedios la quitarán de su tabla de ruteo y si la reparación del camino se pudiera realizar de forma exitosa, entonces la fuente, así como los nodos intermedios re-establecerán el valor de la feromona con el mensaje que envíe el nodo que reparó el camino, el cual contendrá el nuevo valor de feromona.

En la fase proactiva, si se dá el caso que una hormiga proactiva, la cual ha salido del camino establecido para buscar nuevos caminos hacia el destino a dos saltos de distancia, se encuentre con el intruso dicha hormiga será engañada, pues el intruso le dirá que el destino es su vecino y establecerá la nueva ruta, pero cuando la fuente comience a mandar paquetes de datos y no reciba los *acks* entonces el valor de feromona para ésta ruta empezará a decrementarse.

Los mensajes *Hello* serán enviados de la misma forma.

4.2.2. Segundo método para el ataque *Blackhole*: *Método monitores*

En nuestra segunda propuesta introduciremos el concepto de *monitores*, los cuales van a fungir como *guardianes* y estarán a cargo de la seguridad de la red, es un sistema híbrido. Dicho sistema será como el de la figura 4.13.

Pondremos varios monitores (nodos verdes) cubriendo toda la zona de red. Dichos monitores podrían no tratar, simplemente, cuestiones de seguridad. En una implementación real, podrían dar otros servicios, como impresión o Internet. Se podrían ver como puntos de acceso. Cada monitor a su vez será conectado con los demás por medio de una estación base, teniendo así una comunicación independiente (fuera de banda) entre ellos (posiblemente cableada), como se muestra en la figura 4.14, la estación base (circulo azul) concentrará y analizará la información de los monitores, y ahí será donde se resuelva si existe un intruso o no.

El protocolo tendrá el siguiente comportamiento para éste segundo método de detección:

- El nodo fuente numerará los paquetes de datos y los mandará de forma estocástica como lo señala el protocolo.
- Debido a que el envío de paquetes es de forma estocástica, suponemos que en algún momento el destino recibirá un paquete, esto es si el destino es alcanzable

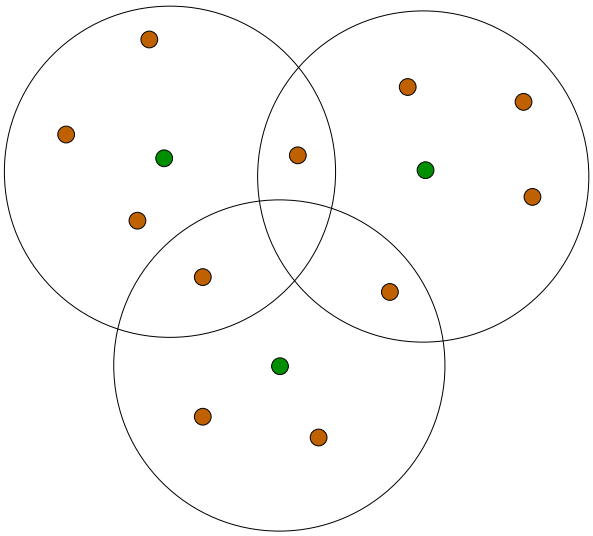

Figura 4.13: Monitores y su zona de monitoreo

desde la fuente. El destino verificará que el número de paquete sea consecutivo al anterior que recibió. Si no es así el nodo destino activará el método de detección. Así nuestro método será reactivo, evitando la sobrecarga en la red. Figura 4.15

- Cuando se active dicho método, cada monitor le pedirá a los nodos que están a su alcance una ruta hacia el destino (fabricando un RREQ que solamente tenga un salto (que solo llegue a sus vecinos), evitando así que inunden la red y provoquen sobrecarga). Figura 4.16.

- Todos los monitores mandarán la información a la estación base y ahi se analizará, para concluir si existe o no, un intruso en la red. Debido a que los monitores están cubriendo toda el área de simulación, algún monitor será vecino del destino. Además los demás vecinos a ellos confirmarán que el destino está cerca. El intruso por su parte dirá que el nodo destino está a un salto de él, dándose así la inconsistencia (que el destino se encuentra en dos áreas al mismo tiempo) entonces se podrá identificar claramente al intruso. Esto será cuando el destino no sea vecino del intruso, de lo contrario no se identificará al intruso. Esto se muestra en la figura 4.17, donde el intruso (nodo negro), dirá que el destino es su vecino, y a su vez los monitores sabrán que no existe la posibilidad de que el destino sea vecino del intruso.

- Si los monitores concluyen que la respuesta de un nodo no es congruente con la ubicación del destino entonces avisarán a todos en la red, que dicho nodo es intruso.

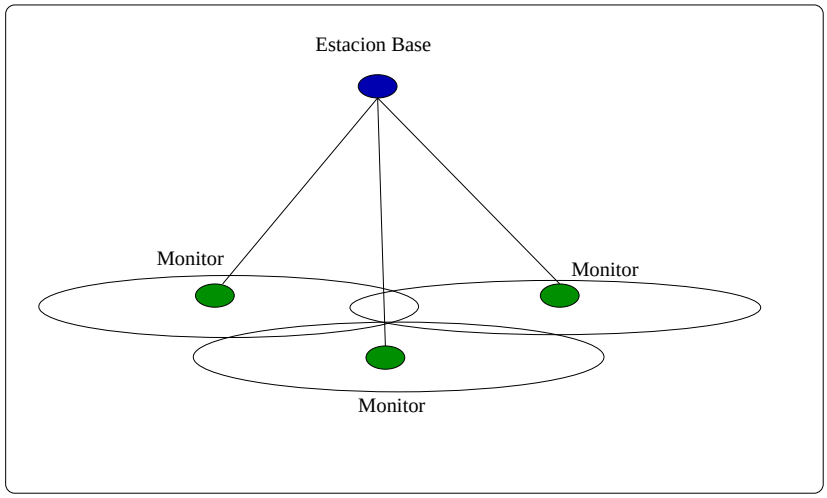

Figura 4.14: Comunicación entre monitores

- A su vez cada nodo tendrá una variable donde indique qué nodos son intrusos y un número asociado, el cual cuando se sepa que el nodo es intruso por primera vez se establecerá a un valor, el cual le llamaremos Ti (Tiempo de Inactividad), y si el nodo vuelve a reincidir se duplicará Ti y cada segundo se decrementará Ti a cierto valor, el cual llamaremos Rs (Recuperación por Segundo). Ti y Rs se variarán en las simulaciones. Figura 4.18.

- Mientras el valor de Ti asociado para cierto intruso sea mayor a 0, el nodo será aislado de la red. Cuando llegue a 0 los nodos volverán a confiar en él.

- Para evitar que el nodo intruso pueda identificar al monitor, activando el mecanismo y registrando quien manda un RREQ pidiendo una ruta hacia él, los monitores estarán cambiando de identidad. La estación base sabe cuantos nodos existen en la red, debido a que tiene comunicación con todos los monitores y éstos le pueden decir cuales nodos están en su área, entonces la estación base les proporcionará las nuevas identidades a los monitores. Así tomarán identidades no existentes, para no duplicar los nodos y generar errores.

Cabe aclarar en este punto, que se pensó en dos soluciones pues al tener el primer método planteado se observaron sus limitaciones como el que la red no puede identificar exactamente que nodo es el intruso, la fuente puede seguir siendo engañada si el nodo destino es inalcanzable desde la fuente, como también el requisito que se necesita de una infraestructura de clave pública (criptografía asimétrica) ya establecida, requisito que reconocemos que no es fácil resolver. La ventaja que este primer método muestra es que es fácil de implementar, teniendo ya resuelto el problema de establecer la clave pública. Nuestro segundo método es mas robusto, ofrece ventajas con respecto al primero como puede ser que se identifica al intruso claramente y se castiga. La desventaja es que

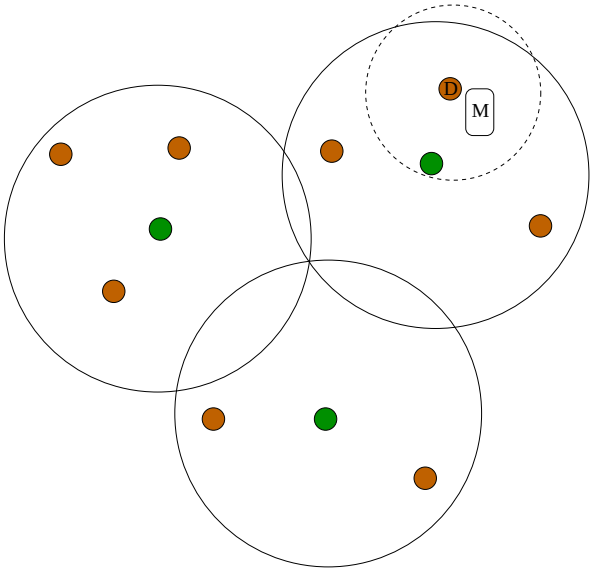

Figura 4.15: El nodo destino le manda un mensaje al monitor para activar el mecanismo

requiere de más elementos como podrían ser los monitores, más carga en la red, pues se envían más mensajes entre los nodos. Entonces con estas dos soluciones pensamos que hemos aportado soluciones tanto sencillas como completas.

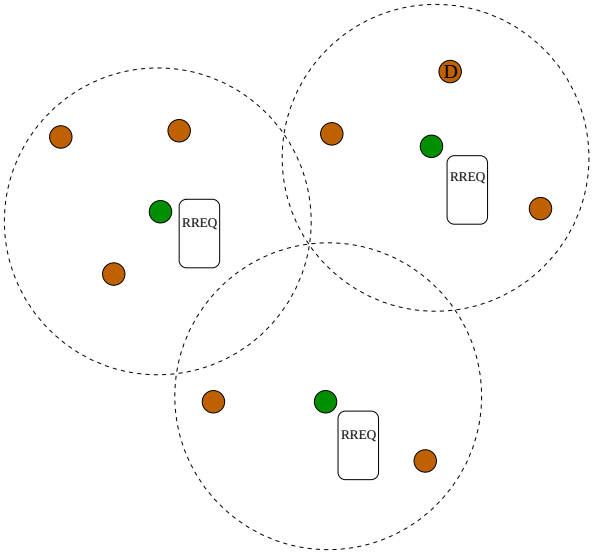

Figura 4.16: Los monitores piden a los nodos que tienen a su alcance una ruta hacia el nodo destino

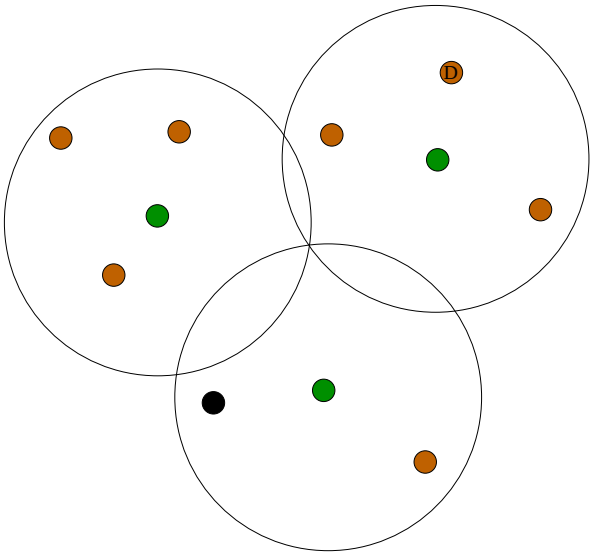

Figura 4.17: Los monitores se dan cuenta de que el nodo intruso no esta diciendo la verdad

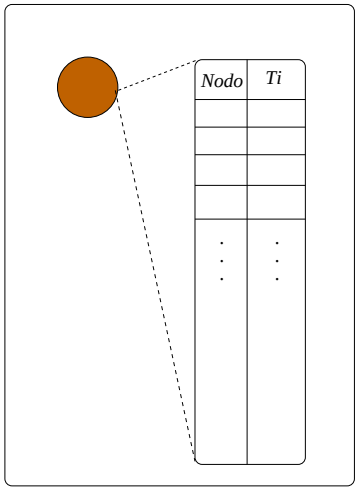

Figura 4.18: Arreglo en cada nodo donde indique cuales nodos son intrusos

Capítulo 5

Resultados

En este capítulo se mostrarán los resultados de los experimentos. Todas las implementaciones se hicieron en *MATLAB (Matrix Laboratory)*. El capítulo se divide en cinco secciones. En la primera se analizará el impacto que tiene el ataque *Blackhole* en el rendimiento de la red, en la segunda y tercera sección se analizará el método de detección y acción ack y Monitores respectivamente, mostrando con los resultados de las simulaciones como ayudan dicho métodos a mitigar el ataque y sus efectos, en la cuarta sección se compararán los resultados de dichos mecanismos, observando ventajas y desventajas de cada uno. Y por último en la quinta sección se realizarán de nuevo las simulaciones (ataque *Blackhole*, método ack y método monitores), pero los nodos no tendrán movimiento, así analizaremos como se comportan los métodos propuestos en una red estática.

5.1. Ambiente de simulación general

Todas las simulaciones tienen el siguiente escenario base, es un escenario de 25 nodos los cuales son situados aleatoriamente en un área de 100 x 100 m^2. Dentro de ésta área, los nodos se mueven de acuerdo al modelo de movimiento *random waypoint*: donde cada nodo aleatoriamente escoge un punto destino, una velocidad y un tiempo de pausa. Esto es el nodo se dirigirá hacia el punto destino con la velocidad escogida y se quedará ahí el tiempo de pausa. El tiempo de simulación es variable. El tráfico de datos es de 50 paquetes por segundo, y de una hormiga proactiva por cada 50 paquetes. Al primer momento se escoge una fuente y un destino aleatoriamente y continúan siéndolo a lo largo de la simulación, haya o no conectividad entre ellos. Existe un intruso, una o dos comunicaciones. En las simulaciones no tomamos en cuenta la congestión ya que de acuerdo al ataque implementado y el tema que nos compete (seguridad) no se considero necesario implementar todo el algoritmo de ruteo.

Cabe destacar que en las simulaciones tanto el nodo intruso, origen y destino son escogidos al azar, habiendo solamente la restricción que los nodos origen y destinos sean diferentes, pero el nodo intruso puede ser tanto el nodo origen como el destino,

permitiendo así, que dicho nodo pueda mandar paquetes como recibirlos, como se haría en un escenario real, es decir al intruso le interesa también que sus paquetes sean transmitidos de forma exitosa por los otros nodos.

5.1.1. Consideraciones

Cabe destacar desde este primer punto que suponemos una red ideal, con esto queremos decir que no tomaremos en cuenta la congestión, o en dado caso se podría decir que la carga de tráfico es constante, y por ende la congestión, las mejores rutas se regirán por el número de saltos. Así el valor de feromona que definimos en el capítulo 2, sección 2.3.4, quedará de la siguiente forma:

$$\tau_{id} = \frac{2}{\Gamma_{s\to d} + h\Gamma_{hop}} \tag{5.1}$$

Donde, Γ_{hop} es el valor ideal que le toma a un paquete dar un salto, h es el número de saltos. El tiempo que le llevará a un paquete de datos viajar desde s hasta d ($\Gamma_{s\to d}$), esta dado por:

$$\Gamma_{s\to d} = \sum_{i=1}^{n-1} \Gamma_{i\to i+1} \tag{5.2}$$

que es la sumatoria del tiempo que le toma a un paquete ir de un nodo i a otro i + *1*, $\Gamma_{i\to i+1}$, en toda la ruta (la ruta es de longitud n). El tiempo que le lleva a un paquete saltar de un nodo a otro , esta dado por

$$\Gamma_{i\to i+1} = (Q^i_{mac} + 1)\Gamma^i_{mac} \tag{5.3}$$

Donde Γ^i_{mac} es el estimado del tiempo promedio para enviar un paquete y Q^i_{mac} es el número actual de paquetes en cola a ser enviados en la capa MAC. Pero como suponemos que la congestión en la red es constante $Q^i_{mac} = cte$, entonces podemos no tomarla en cuenta y la ecuación (5.3) quedaría como sigue

$$\Gamma_{i\to i+1} = \Gamma^i_{mac} \tag{5.4}$$

Si sustituimos (5.4) en (5.2), la ecuación (5.2) quedaría de la siguiente forma

$$\Gamma_{s\to d} = \sum_{i=1}^{n-1} \Gamma^i_{mac} \tag{5.5}$$

Pero como n es el número de nodos que componen dicha ruta entonces la sumatoria no es más que el número de saltos:

$$\Gamma_{s\to d} = h\Gamma_{hop} \tag{5.6}$$

Hicimos esta igualación $\Gamma^i_{mac} = \Gamma_{hop}$ pues Γ_{hop} es el valor ideal que le toma a un paquete dar un salto, y como no es tomada en cuenta la congestión entonces Γ^i_{mac} no es mas que eso, un valor ideal.

Y por último sustituimos (5.6) en (5.1)

$$\tau_{id} = \frac{2}{2h\Gamma_{hop}} = \frac{1}{h\Gamma_{hop}} \tag{5.7}$$

Y como Γ_{hop} es el tiempo ideal que tarda un paquete en dar un salto de un nodo a otro, entonces dicho valor es constante y entonces el valor de feromona quedaría

$$\tau_{id} = \frac{1}{h} \tag{5.8}$$

El protocolo de ruteo utiliza la siguiente fórmula para actualizar el valor de feromona

$$\Gamma^i_{nd} = \gamma\Gamma^i_{nd} + (1-\gamma)\tau_{id}, \gamma\epsilon[0,1] \tag{5.9}$$

La fórmula anterior muestra que el nuevo valor se compone del valor anterior de feromona y el nuevo valor. γ indica cuanto de cada elemento constituirá el valor de feromona. Dicho valor en el artículo [4] se recomendó a 0.7. En la implementación tomamos la referencia del artículo y también lo dejamos a 0.7, esto quiere decir que tomará más peso el valor anterior de feromona que el nuevo valor, para la actualización.

5.2. Análisis del efecto del ataque *Blackhole* en una red ad hoc con el protocolo *AntHocNet*

Se implementó el ataque *Blackhole* y se realizaron simulaciones, variando el tiempo de simulación y evaluando cómo afectaba al rendimiento de la red, en términos de paquetes recibidos y paquetes truncados.

Se monitorearon los paquetes que eran interceptados por el intruso y los paquetes que llegaron al destino satisfactoriamente. La tabla 5.1 muestra que los valores convergen después de un tiempo a 60 % de los paquetes son descartados por el intruso y el 40 % son recibidos por el destino, el último dato es para dos comunicaciones donde se muestra que sigue la misma proporción.

Se pensó en otra forma de evaluar el impacto del intruso sobre la red, para esto se introducirá el concepto de una nueva variable, la cual llamaremos *ventaja del intruso* como se ve en la figura 5.1.

La variable *ventaja del intruso* responde a la siguiente ecuación:

$Ventajadelintruso = A - C$

Con 1000 seg. de simulación: Paquetes recibidos: 46 % Paquetes truncados: 53 %	**Con 8000 seg. de simulación:** Paquetes recibidos: 42 % Paquetes truncados: 57 %
Con 3000 seg. de simulación: Paquetes recibidos: 41 % Paquetes truncados: 58 %	**Con 9000 seg. de simulación:** Paquetes recibidos: 41 % Paquetes truncados: 58 %
Con 5000 seg. de simulación: Paquetes recibidos: 43 % Paquetes truncados: 56 %	**Con 10 000 seg. de simulación:** Paquetes recibidos: 41 % Paquetes truncados: 58 %
Con 7000 seg. de simulación: Paquetes recibidos: 43 % Paquetes truncados: 56 %	**Con 9000 seg. de simulación:** (dos conexiones) Paquetes recibidos: 42 % Paquetes truncados: 57 %

Cuadro 5.1: Tasa de paquetes recibidos y truncados cuando se lanza el ataque *Blackhole* sin ningún mecanismo de defensa

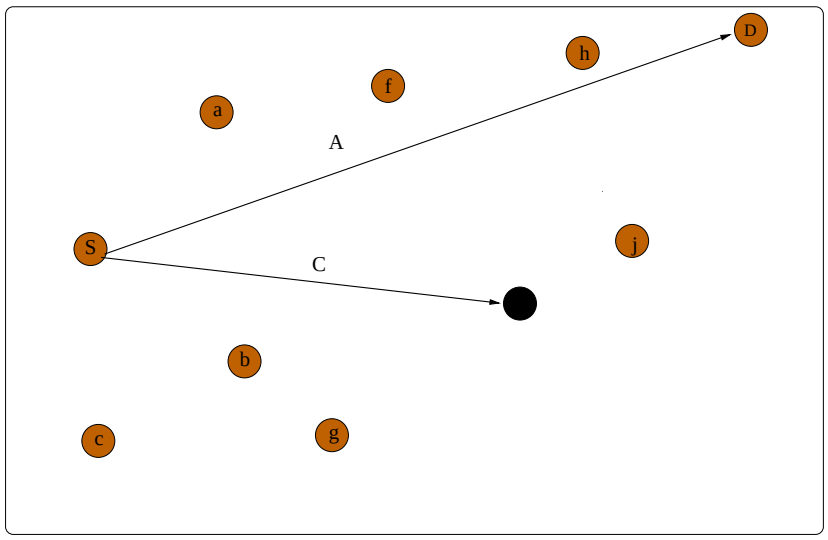

Figura 5.1: *Ventaja del intruso*

donde A es la distancia que existe de la fuente al destino y C es la distancia que existe de la fuente al intruso.

La llamamos *ventaja del intruso*, ya que cuanto mas positiva sea esta variable es porque el destino estará más lejos de la fuente con respecto al intruso, y por consiguiente el intruso tendrá mayor ventaja para interceptar los paquetes. Y cuanto mas pequeña e incluso negativa sea esta variable es porque el destino estará mas cerca de la fuente con respecto al intruso.

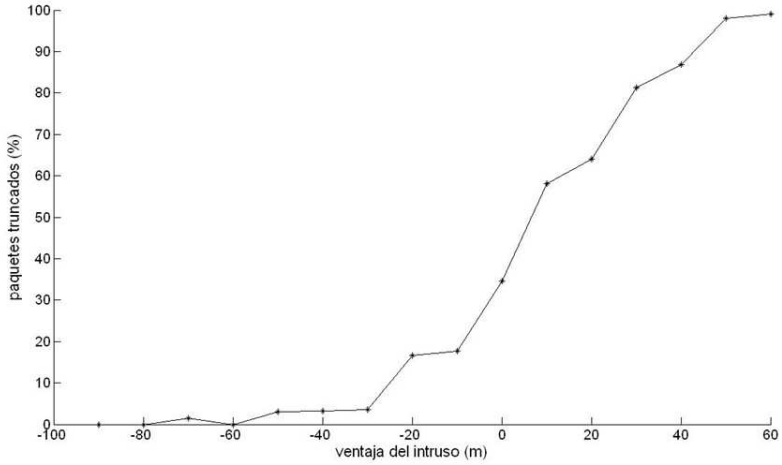

Figura 5.2: Gráfica de porcentaje de paquetes truncados contra *ventaja del intruso*

En la figura 5.2 graficamos el porcentaje de paquetes truncados contra *ventaja del intruso* donde se muestra que cuando el nodo intruso esté mas cerca del nodo fuente con respecto destino, existe mayor probabilidad que los paquetes sean truncados por el intruso y viceversa, debido a que éste ofertará una mejor ruta (menor número de saltos). Cuando la variable *ventaja del intruso* esta cerca del 0, se observa que el porcentaje de paquetes truncados oscila entre 50 %, esto es porque, tanto el intruso como el destino ofertarán una ruta igualmente atractiva (pues están a una misma distancia) y como los paquetes se mandan de forma estocástica existe la misma probabilidad que se manden por la ruta que lleva al destino como por la ruta que lleva al intruso.

5.3. Análisis del Método ack

Para medir que tan efectivo es el método 1, se realizaron diferentes simulaciones donde el tiempo de simulación se mantuvo fijo (a 8000 segundos), y se varió el porcentaje de reducción de la feromona (en la fórmula (4.1) se representa con la variable *reducción*, dicha fórmula se encuentra en la sección 4.2.1, del capítulo 4) al no recibir los acks.

Se puede ver claramente que el método de detección es mas efectivo cuanto más baja el valor de la feromona por cada ack no recibido, pero el método será menos tolerante a fallas normales de la red (principalmente en capa de enlace). En la tabla 5.2 se muestra cuantitativamente como es la relación entre las variables: reducción de feromona y paquetes truncados, es decir cuanto mayor es la reducción menor número de paquetes serán truncados y mas rápido se mitigará el ataque, aquí se dejara a decisión del administrador (una implementación real) que tan segura quiere la red.

Con 10 % de reducción Paquetes recibidos: 77 % Paquetes truncados: 22 %	**Con 40 % de reducción** Paquetes recibidos: 85 % Paquetes truncados: 15 %
Con 20 % de reducción Paquetes recibidos: 77 % Paquetes truncados: 22 %	**Con 50 % de reducción** Paquetes recibidos: 88 % Paquetes truncados: 12 %
Con 30 % de reducción Paquetes recibidos: 79 % Paquetes truncados: 20 %	

Cuadro 5.2: Tasa de paquetes recibidos y truncados con el método de detección ack para diferentes tasas de reducción de feromona

A continuación presentamos en la figura 5.3, la grafica de porcentaje de paquetes truncados vs. *ventaja del intruso*. Una línea (la cual en la gráfica se representa con asteriscos) es cuando la fuente baja el valor de la feromona en un 10 % por cada paquete que no recibe el ack y la otra (la cual en la gráfica se representa con círculos) es cuando la fuente baja el valor de la feromona un 50 % .

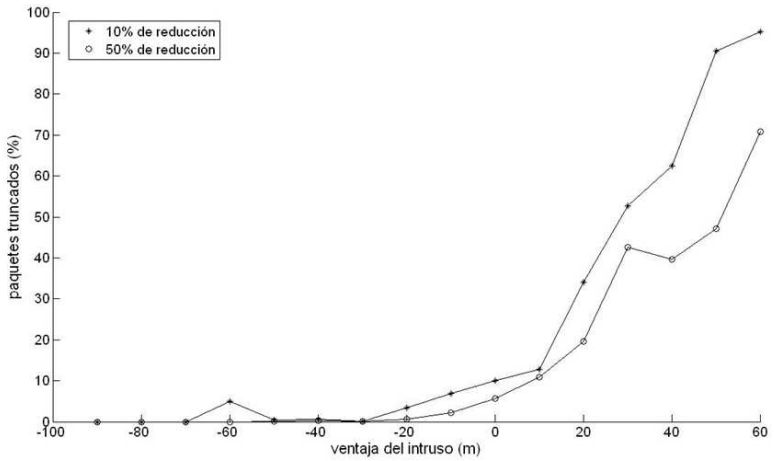

Figura 5.3: Método de detección ack

Los paquetes no son 100 % recibidos pues los paquetes que no reciben *ack* son tomados como paquetes truncados. Además si el destino es inalcanzable (que no existe un camino de la fuente hacia el destino, gráfica disconexa) y el intruso sí es alcanzable por la fuente. En el tiempo 0 el intruso dirá que tiene una ruta hacia el destino, la cual la fuente bajará el valor de feromona conforme no reciba *acks*, hasta llegar a cero, como la fuente no tiene otra ruta para el destino, entonces volverá a hacer un descubrimiento de ruta y la fuente será engañada por el intruso de nuevo y hasta que no mande paquetes se dará cuenta que dicha ruta es incorrecta, y así sucesivamente

hasta que por la topología dinámica, el destino se aproxime a la fuente y se vuelva alcanzable a ésta y así se puedan recibir los paquetes.

5.4. Análisis del Método Monitores.

Se implementaron las modificaciones mencionadas en la sección 4.2.2, para el método monitores y los resultados de las simulaciones son los que se muestran en la tabla 5.3. Aquí existen dos variables que se pueden modificar para la implementación que son: el tiempo que dejaremos inactivo al nodo intruso (Ti) y el tiempo de decremento por segundo (Rs). Aunque estas dos variables están relacionadas, y hasta cierto punto puede manejarse como una sola, pues es lo mismo si duplicamos el tiempo que esta inactivo y también duplicamos el tiempo de decremento por segundo, entonces para las simulaciónes dejaremos fija la variable de decremento por segundo (a una unidad) y variaremos el tiempo que el nodo esta inactivo. Todo lo simulamos a 9000 segundos, en la tabla 5.3 se muestran tres casos, enumerados como a, b y c, para referenciarlos después en las gráficas.

a) Cuando el intruso se detecta por primera vez It = 250, y subsecuentemente It= 500 Paquetes recibidos: 94 % Paquetes truncados: 6 %
b) Cuando el intruso se detecta por primera vez time It = 500, y subsecuentemente It= 1000 Paquetes recibidos: 98 % Paquetes truncados: 2 %
c) Cuando el intruso se detecta por primera vez time It = 1000, y subsecuentemente It= 2000 Paquetes recibidos: 98 % Paquetes truncados: 2 %

Cuadro 5.3: Tasa de paquetes recibidos y truncados con el método de detección monitores para diferentes tasas de tiempo, en el cual dejaremos inactivo al intruso

A continuación se presentan en la figura 5.4, las 3 gráficas de porcentaje de paquetes truncados vs. *ventaja del intruso*, referentes a los tres casos (a, b y c) de la tabla 5.3.

Una ventaja del sistema de detección monitores es que no es tan claro para los nodos, pues cuando el destino desea activar el mecanismo simplemente manda un mensaje a todos sus vecinos, pero los nodos no saben quienes son los monitores, y cuando éstos pregunten por rutas será igual que con los demás vecinos, no advirtiendo al intruso que se le esta cuestionando para verificar su respuesta. Esta es una forma de hacer

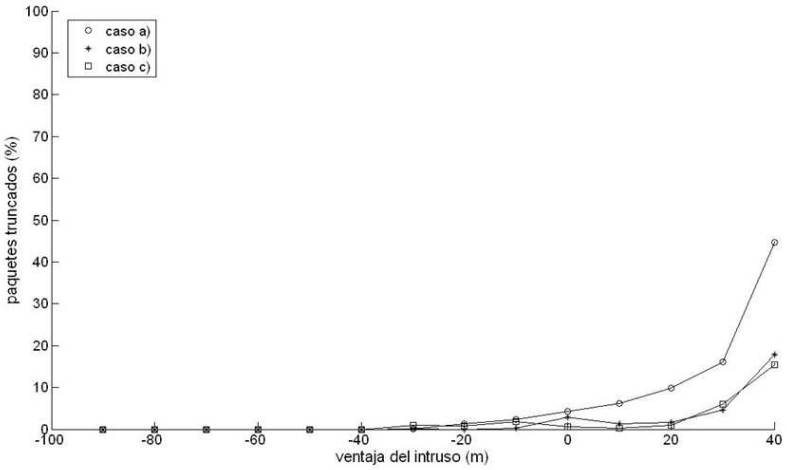

Figura 5.4: Método monitores

caer al intruso en la trampa, y poderlo identificar de forma clara.

5.5. Comparación de los métodos ack y Monitores

El método monitores, de acuerdo a los resultados, demuestra ser más efectivo que el ack, puesto que se identifica exactamente quien es el intruso y se puede castigar severamente no dejándolo participar en la red. Aunque su efectividad depende de cuanto tiempo dejaremos inactivo al intruso, ya que en estos lapsos de tiempo las comunicaciones serán seguras. Por otro lado se consideró que se necesita volver a confiar en el nodo puesto que podría retomar el "buen camino", y si volviera a reincidir en su mal comportamiento se castigará excluyéndolo de la red por un mayor tiempo. En cambio en el método ack la fuente simplemente sabe que por cierto camino no lo lleva al destino, pero no puede identificar quien exactamente es el intruso, pues éste se esconde entre los nodos que constituyen la ruta, es por ello que cuando vuelve a hacer un descubrimiento de ruta, vuelve a confiar en todos los RREP (paquetes de respuesta a la petición de ruta) que recibe, incluso si estos provienen del intruso.

5.6. Simulaciones sin movilidad en los nodos

En esta última sección de resultados se simularon de nuevo los tres casos, ataque *Blackhole*, método ack y método monitores, pero con la variación que los nodos no tendrán movimiento. Cabe destacar que en los tres casos, los nodos están en la misma posición como se muestra en la figura 5.5, simplemente se varió la asignación del nodo destino e intruso en la red y así variar la distancia al nodo fuente. La tabla 5.4 muestra los resultados de las simulaciones, además de mostrar cuales son los nodos que

representan el destino, intruso y origen en la figura 5.5.

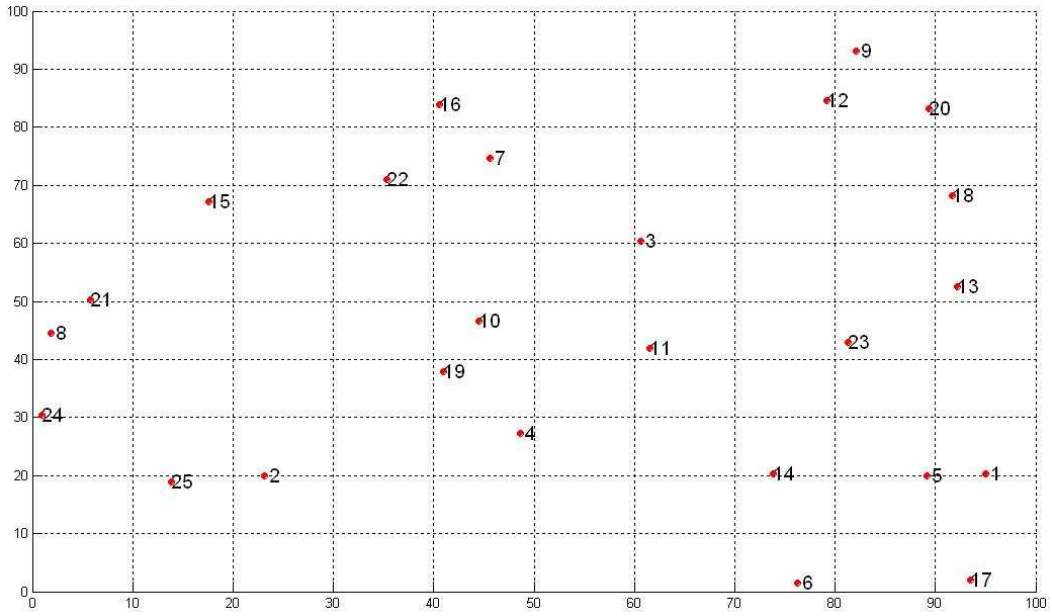

Figura 5.5: Red Estática

Ahora bien, observando los resultados de la tabla 5.4, vemos como en las simulaciones de *Blackhole*, la mayoría de los paquetes llegan a quien esta mas cerca de la fuente, ya sea el destino o el intruso, aunque un porcentaje pequeño se va por los otros caminos (los cuales conducen al intruso o destino, respectivamente). Esto es porque los paquetes no se van al 100 % por la mejor ruta, ya que el envió es de forma estocástica con una probabilidad que depende del valor de feromona existente en las tablas de ruteo. Con el método ack es muy sencillo, pues aunque el intruso este mas cerca que el destino con respecto a la fuente, ésta ultima al no recibir acks por parte del destino en la ruta que lleva al intruso, empezará a bajar el valor de feromona para dicha ruta hasta llegar a cero, mitigando así el ataque. Y como quedan rutas viables hacia el destino, la fuente no volverá a hacer un descubrimiento de ruta, evitando así que el intruso engañe de nuevo a la fuente. Por lo tanto la fuente se quedará solamente con las rutas legitimas (que llevan al destino). En el método monitores, cuando se descubra al intruso por primera vez el método lo dejara aislado por 500 seg., integrándolo después a la red. Pero después de ese momento al igual que en el método ack, la fuente no volverá a hacer un descubrimiento de ruta pues las rutas que tiene realmente llevan al destino. Entonces el intruso no podrá volver a reincidir porque la fuente no volverá a pedir rutas hacia el destino.

	Destino (nodo 16) a 3 saltos de distancia de la fuente e intruso (nodo 9) a 5 saltos de distancia de la fuente (nodo 11)	**Destino (nodo 9) a 5 saltos de distancia de la fuente e intruso (nodo 16) a 3 saltos de distancia de la fuente (nodo 11)**
Ataque *Blackhole*	Paquetes truncados: 10.10 % Paquetes recibidos: 89.89 %	Paquetes truncados: 94.68 % Paquetes recibidos: 5.31 %
Método ACK con 25 % de reducción de feromona	Paquetes truncados: 0.310 % Paquetes recibidos: 99.68 %	Paquetes truncados: 0.659 % Paquetes recibidos: 99.34 %
Método monitores con *Ti* por primera vez igual a 500, y subsecuentemente *Ti* = 1000	Paquetes truncados: 0.0015 % Paquetes recibidos: 99.99 %	Paquetes truncados: 0.0020 % Paquetes recibidos: 99.99 %

Cuadro 5.4: Simulaciones con los nodos estáticos

Capítulo 6

Conclusiones y Trabajo Futuro

6.1. Conclusiones

Los protocolos de ruteo basados en colonias de hormigas son suceptibles a los ataques comúnes, ya que la única caraterística que comparten estos protocolos es que para construir las mejores rutas, se toma en cuenta el número de saltos, así como lo hace *AODV* o *DRS*. Así, los ataques que aprovechan que las rutas son construídas en base al número de saltos pueden ser trasladados a los protocolos basados en colonias de hormigas. En el protocolo que utilizamos *AntHocNet*, los atacantes se pueden aprovechar de la fase proactiva del protocolo, haciendo que los nodos hagan actualizaciones erroneas. Debido a que la hormiga proactiva que actualiza las tablas de ruteo pasa por los nodos, alguno de ellos podría modificarla para reportar datos no correctos; que el intruso diga que tiene mucha congestión y que esa ruta se haga menos atractiva; entre otros.

Cabe destacar que nos enfocamos en el ataque *Blackhole*, para acotar el problema, ya que el área de seguridad es tan extensa, se necesitan poner límites para proponer una solución eficiente que resuelva por completo el problema.

En el desarrollo de la tesis se tomó en cuenta el trabajo que han realizado otros investigadores al respecto. Se observaron las debilidades que muestran dichas soluciones y se trató de dar una solución que no las incluyeran.

Propusimos dos métodos de detección llamados ack y Monitores. Se concluye que el método Monitores es más eficiente que el ack, debido a que en el método ack, se sabe qué ruta lleva al intruso, es decir el nodo fuente sabe por cual vecino los *acks* no están siendo recibidos pero no se sabe hasta donde se están truncando los paquetes de datos, y en realidad no se pueden tomar acciones pues el atacante se esconde entre los nodos que constituyen la ruta. En cambio en el método monitores no sucede lo mismo, aunque esta la restricción que si en realidad el intruso sí es vecino del destino éste no será identificado (ya que los monitores pensarán que si cabe la posibilidad que este diciendo la verdad). Pero si se da el caso contrario el intruso será descubierto por com-

pleto. Con esta estrategia lo que se buscó es que se le tendiera una trampa al intruso, y que éste no pudiera evitar caer, puesto que la forma en que se le pregunta por una ruta es igual a la que marca el protocolo, por lo que el intruso no sabe cuando se le está preguntando para evaluar su respuesta y cuando se le está pidiendo una ruta porque realmente se necesita. Además con la estrategia de que los monitores cambian de identidad el intruso no podrá identificarlos, puesto que cada vez tendrán una identidad diferente.

La característica de tener una topología dinámica en la red, en el método ack beneficia al intruso puesto que éste se puede mover y como la fuente solamente elimina la ruta que lleva al intruso, este puede hacer que, al siguiente descubrimiento de ruta, la fuente construya otra ruta por un diferente camino y que lo lleve a intruso de igual forma. En el método monitores dicha característica no afecta al método pues el intruso esta claramente identificado y donde quiera que él este todos los demás nodos sabran quien es.

6.2. Trabajo Futuro

La seguridad en redes ad hoc es difícil de establecer debido a las características propias de dichas redes. En esta tesis resolvimos el cómo detectar a un intruso para un ataque en particular (*Blackhole*), pero sabemos que falta mucho trabajo por hacer. Simplemente el pensar en todos los ataques que existen para éstas redes, además de aquellos ataques que son más estructurados, como pueden ser que más de un nodo lance el ataque, que los nodos atacantes conozcan el método de detección y lo evadan, dichos ataques necesitan investigación para ser resueltos.

En los métodos que propusimos incluso hay trabajo por mejorar, que queda como trabajo futuro y versiones posteriores a lo que se establece en éste documento. Como podría ser:

- En el método ack, pensar como autenticar al destino de una forma diferente a utilizar criptografía como nosotros lo hicimos, ya que ésta es difícil de establecer, aunque en la tesis supusimos una infraestructura de clave pública ya establecida, no dejamos de reconocer que para llegar a establecer dicha criptografía no es un problema trivial, ni sencillo, debido a la naturaleza de dichas redes, en las cuales no existe una autoridad centralizada, que podría fungir como Autoridad Certificadora. Además el procesamiento que genera la criptografía de llave pública es muy elevado, y recordemos que estamos trabajando con dispositivos con recursos limitados.

- En el segundo método de monitores, la mejora que se puede hacer al sistema es que los monitores fueran móviles, incluso que dicho papel se compartiera entre los mismos

nodos de una cierta área, como lo hacen en [14], no perdiendo así la característica de la red ad hoc, de no tener una autoridad centralizada, y que todos sean del mismo nivel.

Pero repetimos dichas mejoras se dejarán como trabajo futuro.

Bibliografía

[1] Imed Bouazizi. ARA - The Ant-Colony Based Routing Algorithm for Manets. In *Proceedings of the 2002 International Conference on Parallel Processing Workshops*, pages 79–85, Washington, DC, USA, 2002. IEEE Computer Society.

[2] Stephen Carter and Alec Yasinsac. Secure position aided ad hoc routing protocol. In *Proceedings of the IASTED International Conference on Communications and Computer Networks (CCN02)*, pages 329 – 334, 2002.

[3] Benjamin J. Culpepper and H. Chris Tseng. Sinkhole intrusion indicators in DSR manets. In *Proceedings of the First International Conference on Broadband Networks*, pages 681–688, Washington, DC, USA, 2004. IEEE Computer Society.

[4] Gianni Di Caro, Frederick Ducatelle, and Luca Maria Gambardella. AntHocNet: An ant-based hybrid routing algorithm for mobile ad hoc networks. In *Parallel Problem Solving from Nature*, pages 461–470, 2004.

[5] Philip fites and Martin P. J. Kratz. *Information Systems Security: A Practitioner's Reference*. VNR Computer Library, first edition, 1993.

[6] Luca Gambardella and M. Dorigo. Ant colony system: a cooperative learning approach to the travelingsalesman problem. *IEEE Transactions on Evolutionary Computation*, 1:53–66, 1997.

[7] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Packet leashes: A defense against wormhole attacks in wireless networks. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE*, pages 1976–1986, April 2003.

[8] O. Hussein and T. Saadawi. Ant routing algorithm for mobile ad-hoc networks (ARAMA). In *In Proceedings of the IEEE International Performance, Computing, and Communications Conference*, volume 1, pages 281–290, 2003.

[9] Tomasz Imielinski and Julio C. Navas. GPS-based geographic addressing, routing, and resource discovery. In *Commun. Association for Computing Machinery*, volume 42, pages 86–92, New York, NY, USA, 1999. Association for Computing Machinery Press.

[10] David B. Johnson, David A. Maltz, and Josh Broch. *DSR The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks*, chapter 5, pages 139–172. Addison-Wesley, 2001.

[11] Sergio Marti, T. J. Giuli, Kevin Lai, and Mary Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 255–265, New York, NY, USA, 2000. ACM Press.

[12] Charles E. Perkins and Elizabeth M. Royer. Ad-hoc On-demand Distance Vector Routing. In *Workshop on Mobile Computing System and Applications '99: Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications*, pages 90–100, New Orleans, Louisiana, USA, 1999. IEEE Computer Society.

[13] H. Chris Tseng and Benjamin Jack Culpepper. Sinkhole intrusion in mobile ad hoc networks: The problem and some detection indicators. In *Computers & Security*, volume 24, pages 561–570, October 2005.

[14] Ping Yi, YiPing Zhong, and Shiyong Zhang. A novel intrusion detection method for mobile ad hoc networks. In *European Grid Conference*, volume 3470/2005, pages 1183–1192. Springer Berlin / Heidelberg, July 2005.

[15] Seung Yi, Prasad Naldurg, and Robin Kravets. Security-aware ad hoc routing for wireless networks. In *MobiHoc '01: Proceedings of the 2nd Association for Computing Machinery international symposium on Mobile ad hoc networking & computing*, pages 299–302, New York, NY, USA, 2001. Association for Computing Machinery Press.

[16] Manel Guerrero Zapata and N. Asokan. Securing Ad hoc Routing Protocols. In *Proceedings of the 2002 Association for Computing Machinery (ACM) Workshop on Wireless Security (WiSe 2002)*, pages 1–10, September 2002.

[17] Yongguang Zhang and Wenke Lee. Intrusion detection in wireless ad-hoc networks. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 275–283, New York, NY, USA, 2000. Association for Computing Machinery Press.

[18] Xiangquan Zheng, Wei Guo, and Renting Liu. An ant-based distributed routing algorithm for ad-hoc networks. In *Communications, Circuits and Systems, 2004.*, volume 1, pages 412– 417, 2004.