Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación
Laboratorio de Redes Móviles, Inalámbricas y Distribuidas (ICARO)

Implementación de un Sistema del Protocolo de Configuración Dinámica de Hosts con Mensajes de Autenticación

Trabajo Especial de Grado
presentado ante la Ilustre
Universidad Central de Venezuela
por el Bachiller:

David Rubel
C.I.: 17.124.006
E-mail: davidrubels@gmail.com

para optar al título de Licenciado en Computación

Tutora: Profa. María Elena Villapol

Octubre, 2012.

Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación
Laboratorio de Comunicación y Redes

ACTA DEL VEREDICTO

Quienes suscriben, Miembros del Jurado designado por el Consejo de la Escuela de Computación para examinar el Trabajo Especial de Grado, presentado por el Bachiller David Rubel Salas C.I.: 17.124.006, con el título **"Implementación de un Sistema del Protocolo de Configuración Dinámica de Hosts con Mensajes de Autenticación"**, a los fines de cumplir con el requisito legal para optar al título de Licenciado en Computación, dejan constancia de lo siguiente:

Leído el trabajo por cada uno de los Miembros del Jurado, se fijó el día 31 de Octubre de 2012, a la 1 PM, para que sus autores lo defendieran en forma pública, en Sala PB3, lo cual estos realizaron mediante una exposición oral de su contenido, y luego respondieron satisfactoriamente a las preguntas que les fueron formuladas por el Jurado, todo ello conforme a lo dispuesto en la Ley de Universidades y demás normativas vigentes de la Universidad Central de Venezuela. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobarlo.

En fe de lo cual se levanta la presente acta, en Caracas el 31 de Octubre de 2012, dejándose también constancia de que actuó como Coordinador del Jurado el Profesor Tutor María Elena Villapol.

Prof. María Elena Villapol
(Tutor)

Prof. David Perez
(Jurado Principal)

Prof. Robinson Rivas
(Jurado Principal)

Resumen

Título:
Implementación de un Sistema del Protocolo de Configuración Dinámica de Hosts con Mensajes de Autenticación.

Autor:
David Rubel

Tutor:
María Elena Villapol

El auge que ha tenido Internet ha creado la necesidad en la gente de estar conectados a esta red. Para estar conectado a Internet es necesario que un dispositivo de comunicación posea configurados en forma apropiada varios parámetros de de la pila TCP/IP. El común denominador de la gente posee poco o ningún conocimiento de cómo configurar estos parámetros, por lo que se ven en la necesidad de depender de algún administrador de red que configure dichos dispositivos de comunicación. El Protocolo de Configuración Dinámica de Hosts solventa este inconveniente, permitiendo a los dispositivos en una red TCP/IP obtener información de configuración automáticamente. Sin embargo, esta facilidad también se convierte en una desventaja cuando existen personas malintencionadas que quieran acceder a una red ya que DHCP no distingue usuarios, abriéndoles el paso para realizar cualquier cantidad de ataques a otros usuarios o sistemas, incluso para el mismo DHCP.

Actualmente existe un mecanismo de autenticación para DHCP descrito en el RFC 3118. En dicho RFC se mencionan dos métodos de autenticación llamados *Configuration Token* y *Delayed Authentication*, con los cuales se permite otorgarle direcciones IP sólo a aquellos usuarios que poseen una llave y se autentiquen correctamente. Lamentablemente no se ha encontrado ninguna implementación de DHCP con soporte para dicho mecanismo. Para crear una solución a esta problemática, en este trabajo se ha modificado una implementación existente de un cliente y un servidor DHCP para poder hacer uso de la opción de autenticación en mensajes DHCP cumpliendo con los lineamientos indicados en el RFC 3118.

La solución desarrollada fue sometida a diversas pruebas de funcionamiento, estrés y rendimiento para evaluar el comportamiento del sistema DHCP luego de las modificaciones. Los resultados obtenidos durante las pruebas demostraron que se logró desarrollar una solución funcional sin añadir una sobrecarga significativa al comparar el rendimiento de la solución desarrollada con el de la implementación original.

Palabras Claves: DHCP, Autenticación, Configuration Token, Delayed Authentication, Implementación.

Tabla de contenido

Índice de Figuras

Índice de Tablas

1. Introducción

El auge que ha tenido Internet ha creado la necesidad en la gente de estar conectados a esta red. Para estar conectado a Internet es necesario que un dispositivo de comunicación posea configurados en forma apropiada varios parámetros de de la pila TCP/IP. El común denominador de la gente posee poco o ningún conocimiento de cómo configurar estos parámetros, por lo que se ven en la necesidad de depender de algún administrador de red que configure dichos dispositivos de comunicación. Por los motivos expuestos anteriormente sería deseable tener un mecanismo que pueda ser usado por un dispositivo para obtener automáticamente los parámetros de configuración apropiados.

El protocolo de configuración dinámica de hosts o *Dynamic Host Configuration Protocol* (DHCP) [1] es un protocolo que le permite a los dispositivos en una red TCP/IP obtener información de configuración automáticamente, eliminando el trabajo en forma manual por parte de un administrador de red. Sin embargo, esta facilidad también se convierte en una desventaja cuando existen personas malintencionadas que quieran acceder a una red ya que DHCP no distingue usuarios, abriéndoles el paso para realizar cualquier cantidad de ataques a otros usuarios o sistemas, incluso para el mismo DHCP.

Existen unas pocas herramientas fuera de DHCP que permiten controlar el acceso a una red, mitigando en cierta forma la problemática expuesta anteriormente, pero la implementación de estos controles adicionales puede traer consigo una inversión económica adicional y una configuración que puede aumentar la complejidad de una red. Por ello se hace deseable la implementación de un mecanismo embebido en el mismo DHCP que permita controlar a quien se le permite recibir información de configuración de red.

1.1. Planteamiento del Problema

DHCP fue desarrollado al comienzo de los años 1990, cuando la seguridad en la Internet, o incluso en redes locales no era una gran preocupación, por lo que en DHCP no se había implementado ningún mecanismo de seguridad. Además dentro del modelo de amenazas para DHCP hay dos problemas de seguridad bien conocidos. En primer caso, un servidor DHCP no autorizado pudiera proveer información de configuración incorrecta a un cliente con el fin de establecer un ataque de "hombre en el medio" o un ataque de "denegación de servicio". Segundo, un cliente DHCP no autorizado pudiera obtener información de configuración de un servidor con la intención de comprometer posteriormente a la red o pudiese causar el agotamiento de direcciones IP a ser otorgadas enviando múltiples peticiones de configuración al servidor, causando otro tipo de ataque de denegación de servicio.

Afortunadamente existe un mecanismo de autenticación que pudiese prevenir esta problemática, el cual fue descrito en el RFC 3118 [5] en el año 2001. Lamentablemente a pesar de ello no se han encontrado implementaciones de DHCP que soporten dicho mecanismo o peor aún, no ha sido implementado en todos estos años. Por ello la implementación de tal mecanismo podría ser beneficiosa tanto para los usuarios de DHCP como para el sistema en su totalidad.

1.2. Objetivo General

El objetivo de este trabajo es el de realizar la modificación de un cliente y un servidor DHCP para agregar los diferentes mecanismos de autenticación propuestos en el RFC 3118 [5], con la finalidad de evitar la posible explotación de vulnerabilidades por parte de agentes no autorizados en una red.

1.3. Objetivos Específicos

Los objetivos específicos de este trabajo son los siguientes:

- Elegir una implementación software libre de DHCP (cliente y servidor).
- Extender la funcionalidad y añadir los métodos de autenticación descritos en el RFC 3118 a una implementación existente de DHCP.
- Realizar diversas pruebas de funcionamiento, estrés y rendimiento para evaluar el comportamiento del sistema DHCP luego de las modificaciones.
- Documentar los cambios realizados en el código fuente del cliente y servidor DHCP.
- Analizar los resultados obtenidos.

1.4. Justificación

El protocolo DHCP es ampliamente utilizado para proveer una forma sencilla y automatizada de configurar apropiadamente una interfaz de red de algún dispositivo. Esto hace que sea una herramienta casi necesaria en muchos casos, ya sea para facilitar labores de administración en ambientes con gran cantidad de computadores o para cubrir la necesidad de usuarios con pocos conocimientos en el área de obtener una configuración que le permita acceder a Internet. Esta facilidad también se puede convertir en una desventaja cuando existen personas malintencionadas que quieran acceder a una red ya que DHCP no distingue usuarios, abriéndoles el paso para realizar cualquier cantidad de ataques a otros usuarios o sistemas. Por lo tanto, la implementación de un mecanismo que permita configurar sólo aquellos dispositivos autenticados es un método viable para mitigar dicho problema.

1.5. Estructura del Documento

El presente trabajo está estructurado de la siguiente forma:

- **Capítulo 1 - Introducción:** Se describe brevemente el propósito de DHCP, la problemática con la implementación actual y la justificación de este trabajo.

- **Capítulo 2 - DHCP:** Se describe el funcionamiento de DHCP, los distintos tipos de mensajes enviados, el formato de las tramas y los pasos a seguir para obtener una dirección IP a través de DHCP.

- **Capítulo 3 - Autenticación en Mensajes DHCP:** Se describe la opción de autenticación, el formato de la opción y los posibles métodos de autenticación propuestos para esta opción.

- **Capítulo 4 - Metodología y Herramientas** Se describe brevemente las actividades a realizar para la implementación de la solución y las herramientas utilizadas para lograr ese fin.

- **Capítulo 5 - Procedimiento para la Implementación de la Solución:** Se describe en detalle las actividades realizadas que permitirán el desarrollo de la solución.

- **Capítulo 6 - Desarrollo de la Solución:** Se describe en detalle las consideraciones que fueron tomadas para la modificación del código fuente y para la implementación de los métodos de autenticación.

- **Capítulo 7 - Pruebas:** Se describe las pruebas a realizar, los escenarios planteados para los mismos y el correspondiente análisis de los resultados obtenidos en cada prueba.

- **Capítulo 8 - Conclusiones:** Se plantea las conclusiones alcanzadas en este trabajo.

2. DHCP

Con el pasar del tiempo se ha ido incrementando exponencialmente el número de computadoras utilizadas en todo el mundo y por lo tanto, el tamaño en general de las redes de computadoras. Esto ha traído como consecuencia que las labores de administración o configuración realizadas de forma manual sea cada vez menos factible, por lo que surge la necesidad de crear un mecanismo para configurar automáticamente los equipos.

2.1. Descripción General

DHCP es un protocolo definido en el RFC 2131 [1] que le permite a los equipos pertenecientes a una cierta red TCP/IP obtener parámetros de configuración en forma automática tales como dirección IP, máscara de subred, puerta de enlace predeterminada, servidores de sistemas de nombres de dominio o *Domain Name System* (DNS), entre otros. Su funcionamiento se divide en dos componentes: un protocolo para el envío de parámetros de configuración y un mecanismo para la asignación de direcciones de red.

El protocolo DHCP funciona bajo un modelo cliente-servidor en el cual los equipos que deseen obtener la configuración necesaria para comunicarse con cualquier otro equipo a través de la red o Internet realizan una petición al servidor o los servidores DHCP presentes en la red para obtener una configuración adecuada que les permita realizar dicha tarea. Por éste motivo, también se puede decir que DHCP permite cierta centralización respecto a la configuración debido a que basta con realizar cambios únicamente en la información enviada por el servidor para que los cambios se vean reflejados en cada uno de los clientes, facilitando así la administración de redes que poseen gran cantidad de equipos.

Los mensajes intercambiados por el cliente y el servidor están basados en los mensajes utilizados por el protocolo BOOTP [15], el cual se usaba previamente a la existencia de DHCP para obtener la dirección IP de un equipo, obtener la ubicación de red de su imagen de arranque en el caso de terminales sin discos duros, entre otros. Sin embargo, DHCP introduce una mejora notable la cual es que una dirección IP puede ser reutilizada por otro cliente si la dirección está disponible, a diferencia de BOOTP en el que cada dirección IP es asignada previamente y corresponde sólo a un cliente.

Tanto los mensajes de BOOTP como los mensajes DHCP son enviados a través del protocolo de transporte *User Datagram Protocol* (UDP) hacia los puertos 67 y 68 que corresponden al puerto del servidor y el cliente, respectivamente. De este modo se permite cierta compatibilidad hacia atrás con BOOTP.

Existen tres tipos de asignación para las direcciones IP:

- **Asignación automática:** DHCP asigna una dirección IP permanente a un cliente.

- **Asignación dinámica:** DHCP asigna una dirección IP a un cliente por una cantidad de tiempo limitada o hasta que el cliente avise que desea liberar la dirección.

- **Asignación manual:** La dirección IP del cliente es asignada por el administrador de la red y se utiliza DHCP para comunicarle la dirección asignada al cliente.

El servidor DHCP está encargado de asegurarse que no exista más de un cliente con la misma dirección IP, sin importar el tipo de asignación que se haya utilizado, aunque el cliente también puede notificarle por medio de un mensaje al servidor si se ha obtenido una dirección duplicada luego de haber realizado una verificación de que no haya otro cliente presente con la misma dirección.

Además de asignar direcciones IP, el servidor DHCP también provee configuración adicional, aunque no todos los parámetros de configuración son requeridos por un cliente para poder intercambiar paquetes con cualquier host en Internet. El cliente y el servidor pueden negociar la transmisión de sólo aquellos parámetros requeridos por el cliente o aquellos específicos para una subred en particular.

DHCP no debe requerir un servidor en cada subred, ya que el mismo debería funcionar a través de enrutadores y agentes de relevo BOOTP donde existan, los cuales son dispositivos destinados a reenviar mensajes BOOTP a través de distintas subredes. Se puede dar la posibilidad de tener más de un servidor DHCP funcionando en la red, en ese caso un cliente podría recibir múltiples respuestas a una petición de parámetros de configuración y deberá aceptar solamente una de ellas.

A pesar de que una de las funciones principales de DHCP es permitir la configuración automática de los clientes, debe permitir la asignación de parámetros permanente o fija de clientes específicos y coexistir con host configurados estáticamente.

Un servidor DHCP debe retener la configuración del cliente sin importar que el cliente o el servidor se hayan reiniciado y de ser posible siempre se le debería asignar la misma configuración a un cliente cuando sea requerida.

2.2. Formato de Trama DHCP

Los mensajes que intercambian cliente y servidor siguen el mismo formato que los mensajes de BOOTP, como se muestran en la Figura 2.1 y en la Figura 2.2, donde se indican una breve descripción de cada campo de la trama, la cantidad de bytes que contienen y el desglosamiento del campo "options":

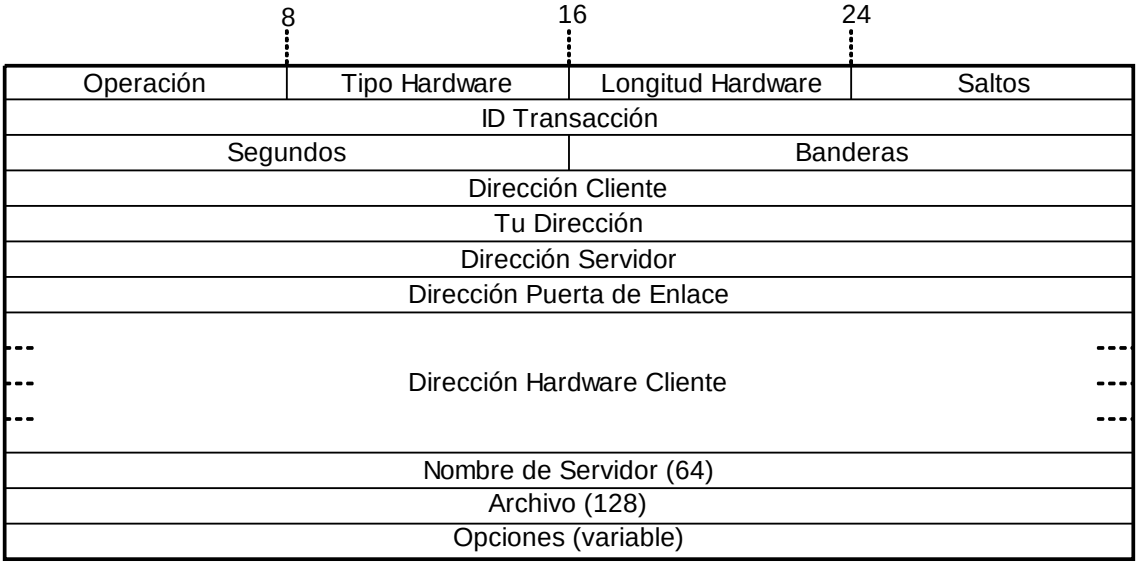

Figura 2.1 Formato de trama DHCP.

- Operación: Tipo de mensaje (1= BOOTREQUEST - 2= BOOTREPLY).
- Tipo Hardware: Código que indica el tipo de interfaz de red.
- Longitud Hardware: Longitud de dirección de hardware.
- Saltos: Cantidad de saltos realizados a través de enrutadores.
- ID Transacción: ID de transacción para asociar mensajes y respuestas.
- Segundos: Segundos transcurridos desde que el cliente inició la petición.
- Banderas: El bit más significativo denota el envío de mensajes por difusión (broadcast) y el resto de los bits son reservados.
- Dirección Cliente: Dirección IP del cliente (solo se coloca si ya posee una).
- Tu Dirección: Dirección IP que será asignada al cliente.
- Dirección Servidor: Dirección IP del servidor.
- Dirección Puerta de Enlace: Dirección IP del agente de relevo.
- Dirección Hardware Cliente: Dirección de hardware del cliente (Dirección MAC).
- Nombre de Servidor: Indica un nombre de host opcional para el servidor.
- Archivo: Nombre del archivo de arranque. Usado en BOOTP y mantenido por motivos de compatibilidad hacia atrás.
- Opciones: Campo de longitud variable que contiene parámetros opcionales.

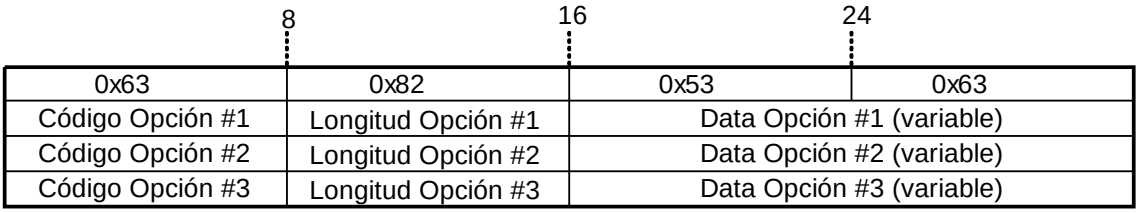

Figura 2.2 Formato del campo Options.

- Código Opción: Código de la opción.
- Longitud Opción: Longitud en bytes del campo "Data Opción" de los datos correspondientes a la opción "Código Opción".
- Data Opción: Datos enviados, como por ejemplo parámetros de configuración.

2.3. Tipos de Mensaje

Los tipos de mensaje DHCP que existen son los siguientes:

- DHCPDISCOVER: Es enviado por el cliente para descubrir la presencia de servidores DHCP disponibles en la red local. Son identificados con el tipo de mensaje 1.

- DHCPOFFER: Es enviado por el servidor en respuesta a un mensaje DHCPDISCOVER, ofreciendo parámetros de configuración. Son identificados con el tipo de mensaje 2.

- DHCPREQUEST: Es enviado por el cliente para pedir la asignación de los parámetros ofrecidos por un servidor DHCP y rechazar las ofertas de los demás servidores, para confirmar la correctitud de una dirección asignada previamente, o para extender el tiempo de préstamo de una dirección. Son identificados con el tipo de mensaje 3.

- DHCPDECLINE: Es enviado por el cliente para indicar que la dirección de red ofrecida por el servidor está siendo usada por otro cliente. Son identificados con el tipo de mensaje 4.

- DHCPACK: Es enviado por el servidor para confirmar la asignación de los parámetros indicados al cliente. Son identificados con el tipo de mensaje 5.

- DHCPNACK: Es enviado por el servidor para indicar que no se puede asignar la dirección pedida por el cliente, ya sea porque esté incorrecta o porque haya expirado el tiempo de préstamo. Son identificados con el tipo de mensaje 6.

- DHCPRELEASE: Es enviado por el cliente para indicarle al servidor que se desea liberar la dirección de red y cancelar el tiempo de préstamo restante. Son identificados con el tipo de mensaje 7.

- DHCPINFORM: Es enviado por el cliente para pedir únicamente parámetros de configuración local en caso que el cliente ya tenga alguna dirección de red asignada. Son identificados con el tipo de mensaje 8.

2.4. Funcionamiento de DHCP

DHCP básicamente cumple dos funciones. La primera es la de proveer almacenamiento persistente de parámetros de red para los clientes: Por cada cliente, el servidor DHCP almacena una entrada del tipo llave-valor, donde la llave es un ID único compuesto del número de la subred y un identificador único dentro de la misma y el valor contiene los parámetros de configuración para el cliente.

Por defecto, la llave es de la forma "Número de subred - Dirección de hardware", como por ejemplo la dirección MAC, pero la llave bien pudiera ser de la forma "Número de subred - Nombre de host", proporcionando de ésta forma una asignación más eficiente ante eventos como reemplazo de una tarjeta de red o que el cliente haya sido movido a otra subred. El cliente también puede especificar algún identificador que lo distinga de forma única en la red por medio de la opción "client identifier", de ser así, el cliente deberá incluir siempre dicha opción con el mismo valor en cada mensaje DHCP que vaya a transmitir.

La segunda función de DHCP es la de asignación temporal o permanente de direcciones IP a los clientes, en la cual el cliente hace una petición de uso de una dirección por un período de tiempo y el servidor garantiza no asignar dicha dirección durante ese tiempo de préstamo e intentará asignar esa misma dirección cada vez que el cliente vuelva a hacer una petición, permitiendo de ésta forma extender de alguna manera el tiempo de préstamo.

Una dirección que haya sido asignada de forma permanente solamente podrá ser reutilizada si el cliente envía un mensaje al servidor DHCP indicando que se desea liberar la dirección, aunque también para evitar este inconveniente, el servidor podría asignar la dirección no infinitamente sino por períodos de tiempo muy largos para poder detectar si el cliente se encuentra usando esa dirección. En cambio, en algunos ambientes en los que se haga una asignación dinámica de direcciones y haya más clientes que direcciones disponibles, se deberá reutilizar las direcciones cuyos tiempos de préstamo ya hayan expirado.

Al momento de asignar una dirección a un cliente, el servidor DHCP previamente debería verificar que otro cliente no esté utilizando dicha dirección, como por ejemplo enviando un Echo Request ICMP a la dirección a asignar y si no se recibe respuesta alguna, se puede asignar esa dirección. Adicionalmente, el

cliente también debería verificar que la dirección que ha recibido no esté siendo usada por otro cliente, como por ejemplo enviando un ARP Request [2] para esa dirección y si no recibe respuesta alguna es porque es el único cliente con esa dirección IP.

Para llevar a cabo las funciones descritas anteriormente, debe existir un intercambio de mensajes entre el cliente y el servidor DHCP, los cuales están formados por el mensaje DHCP seguido de un conjunto de opciones.

Los primeros cuatro bytes del campo de opciones deben corresponder a los valores decimales "99, 130, 83, 99" como es indicado en el RFC 1497 [3] y seguido a esto se incluyen las opciones de DHCP definidas en el RFC 2132 [4] para indicar los parámetros de configuración que sean de interés. Hay una opción que es obligatoria en todo mensaje DHCP, que corresponde a la opción "DHCP message type" con "Código Opción= 53", el cual indicará en un campo el tipo de mensaje DHCP correspondiente y determinará algunas opciones que pueden necesitarse y otras que pueden aparecer o no dependiendo del tipo de mensaje DHCP.

2.5. Proceso de Configuración por Medio de DHCP

Para que un cliente que no tiene dirección de red pase a estar completamente configurado de forma automática, se debe realizar la siguiente secuencia de pasos, tal como se ilustra en la Figura 2.3:

- El cliente manda por difusión un mensaje DHCPDISCOVER a su subred local, el cual puede tener o no opciones sugiriendo valores para la dirección de red y/o tiempo de préstamo.

- Cada servidor puede responder con un mensaje DHCPOFFER indicando la dirección de red disponible para el cliente en el campo "Tu Dirección" y otros parámetros a través de las opciones del mensaje DHCP.

- El cliente recibe uno o más mensajes de DHCPOFFER dependiendo si hay varios servidores DHCP y escoge uno de ellos para pedir la asignación de los parámetros de configuración ofrecidos por ese servidor. El cliente envía por difusión un mensaje DHCPREQUEST que debe incluir la opción "server identifier" para indicar el servidor que ha sido seleccionado, la opción "requested address" debe ser colocada con el valor "Tu Dirección" enviado en el mensaje DHCPOFFER y se puede incluir otras opciones pidiendo valores específicos para su configuración.

- Los servidores reciben el mensaje DHCPREQUEST enviado por difusión y verifican si el "server identifier" corresponde al suyo. En caso de ser así, el servidor almacena en un medio persistente la asignación de parámetros

realizada hacia el cliente y envía un mensaje DHCPACK conteniendo los parámetros de configuración, sino, el servidor tomará el mensaje como una notificación de que fue rechazada su oferta. Si el servidor es incapaz de satisfacer los requerimientos que el cliente haya solicitado en el mensaje DHCPREQUEST, el servidor debería responder con un mensaje DHCPNACK.

- El cliente recibe el mensaje DHCPACK con los parámetros de configuración, toma nota del tiempo de préstamo y procede a revisar que la dirección otorgada no esté siendo usada por otro cliente por medio de ARP por ejemplo. Si el cliente detecta que la dirección está siendo usada, debe enviar un mensaje DHCPDECLINE rechazando la dirección y reiniciar el proceso de configuración, sino, el cliente ya está totalmente configurado. Si el cliente recibe más bien un mensaje DHCPNACK, el cliente reinicia el proceso de configuración.

- El cliente puede liberar voluntariamente la dirección que le fue asignada enviando un mensaje DHCPRELEASE al servidor.

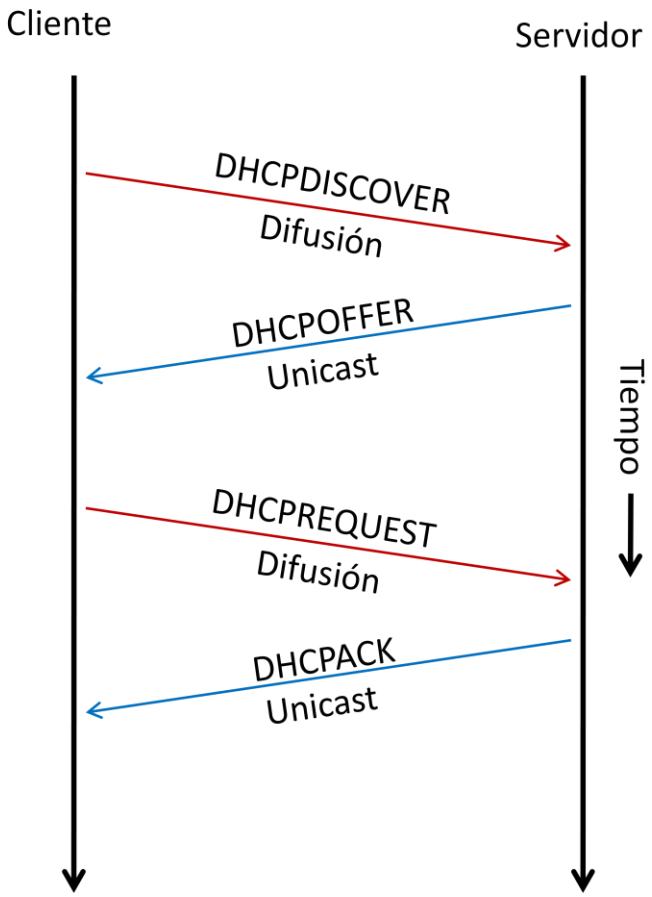

Figura 2.3 Ejemplo de intercambio de mensajes DHCP.

3. Autenticación en Mensajes DHCP

En algunos ambientes es posible que no sea deseable que cualquier host conectado a la red sea capaz de obtener una dirección IP, por ejemplo, debido a políticas de seguridad como otorgar una dirección IP solamente a aquellos hosts permitidos. Por ello surge la necesidad de idear algún mecanismo para controlar la entrega de direcciones IP.

3.1. Descripción General

La autenticación en los mensajes DHCP es una opción definida en el RFC 3118 [5] que permite generar un tipo de ticket de autenticación para que sólo aquellos hosts que estén autorizados sean configurados automáticamente por el servidor DHCP. La autenticación no se limita solamente a los hosts, sino que también se puede utilizar para autenticar los mensajes enviados por el servidor DHCP.

3.2. Formato de la Trama

El formato que sigue la opción de autenticación en mensajes DHCP se puede apreciar en la Figura 3.1 y se describen los campos a continuación:

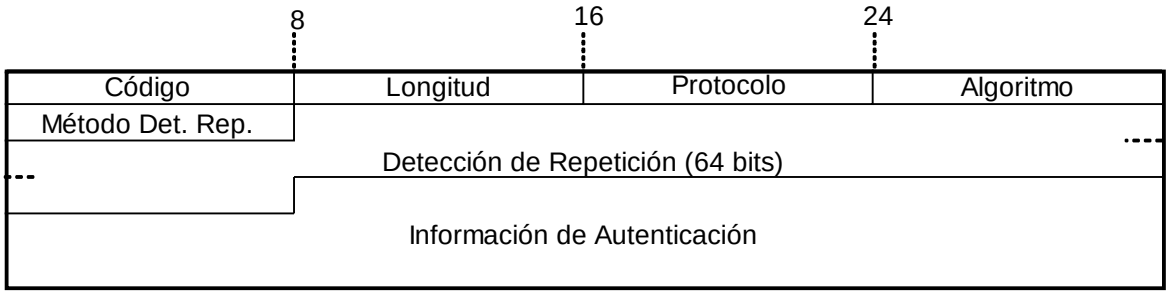

Figura 3.1 Formato de opción de autenticación.

- Código: El código que corresponde a éste tipo de mensajes es el número 90.
- Longitud: Contiene la longitud en bytes de los campos "Protocolo", "Algoritmo", "Método Det. Rep.", "Detección de Repetición" e "Información de Autenticación".
- Protocolo: Indica la técnica en particular para la autenticación usada en esa opción.
- Algoritmo: Indica un algoritmo a utilizar en base al protocolo indicado en el campo anterior.
- Método Det. Rep. (Método de Detección de Repetición, o RDM): Indica el método a utilizar para detectar posibles mensajes duplicados.
- Detección de Repetición: Contiene un valor utilizado para detectar ataques de repetición que varía según el "Método Det. Rep.". Cabe destacar que

sea cual sea el valor del campo "Protocolo", si el valor del campo "Método Det. Rep." es 0, el valor del campo "Detección de Repetición" debe ser un contador monótonamente creciente como por ejemplo la hora actual.

Según el RFC 3118, existen dos alternativas para enviar mensajes con información de autenticación llamadas *Configuration token* y *Delayed Authentication*, los cuales serán descritos a continuación.

3.3. Configuration Token

Los mensajes de este tipo son identificados por tener un valor de "0" en el campo "Protocolo", "Algoritmo" y "Método Det. Rep.".

En los mensajes del tipo *Configuration Token* se hace envío de un valor conocido previamente por el cliente y el servidor que es utilizado como un método básico de autenticación semejante al provisto por una contraseña. Si la contraseña presente en el mensaje es distinta a la clave compartida, el receptor debe descartar este mensaje. Por este motivo, la protección ofrecida por este mecanismo es algo débil aunque bastante sencilla ya que no es necesario generar y almacenar constantemente claves. En la Figura 3.2 se puede observar un diagrama de flujo donde se detalla el conjunto de acciones realizadas durante el proceso de validación de un mensaje DHCP usando este método de autenticación.

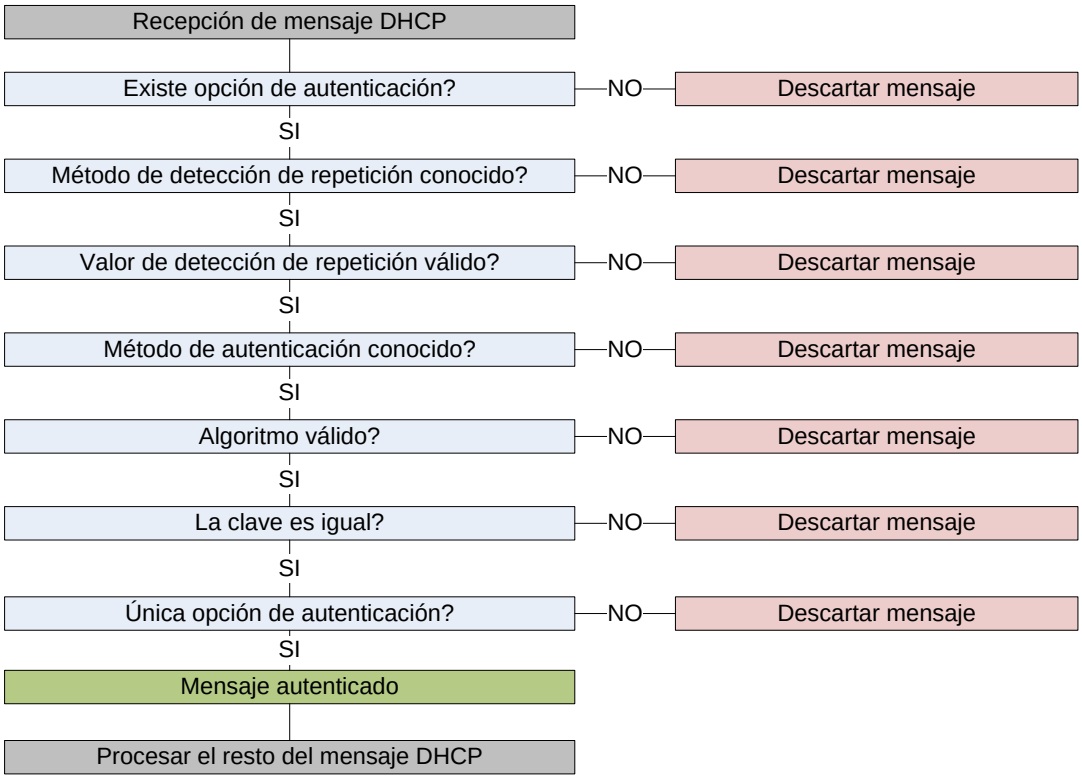

Figura 3.2 Diagrama de flujo de validación de mensajes usando Configuration Token.

3.4. Delayed Authentication

Los mensajes de este tipo son identificados por tener un valor de “1” en el campo "Protocolo". En la autenticación retardada, el cliente hace una petición de autenticación en su mensaje DHCPDISCOVER al servidor, el cual responde con un mensaje que incluye la información de autenticación. Esta información de autenticación consiste en un valor generado al momento para proveer autenticación de mensaje y de host llamado “Código de autenticación de mensaje” (*Message Authentication Code* o MAC).

Para la autenticación retardada se necesita que se tenga una llave secreta compartida entre el cliente y cada servidor DHCP al cual se quiere hacer petición de una dirección IP. Cada llave tiene un identificador único que es usado para determinar cuál llave fue utilizada para generar el código MAC en el mensaje DHCP.

El formato de los mensajes DHCPDISCOVER con autenticación se puede apreciar en la Figura 3.3 mientras que el formato de los mensajes DHCPOFFER, DHCPREQUEST y DHCPACK se puede apreciar en la Figura 3.4.

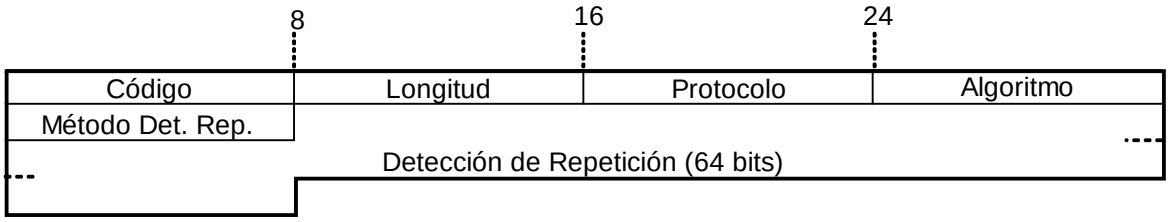

Figura 3.3 Delayed Authentication DHCPDISCOVER.

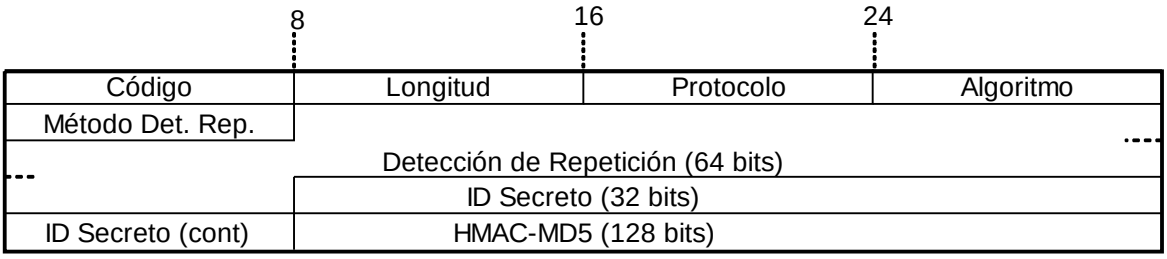

Figura 3.4 Delayed Authentication DHCPOFFER, REQUEST, ACK.

- Detección de Repetición: Valor definido según el campo "Método Det. Rep.".
- ID Secreto: Identificador único para un valor secreto compartido o llave entre emisor y receptor para generar el código MAC de este mensaje.
- HMAC-MD5: Hash MD5 obtenido de una función generadora de código MAC, definida en el RFC 2104 [6].

Para realizar el cálculo del código MAC se utiliza como entrada el mensaje DHCP incluyendo la cabecera y los campos de opciones, además de la llave compartida. Cabe destacar que los campos “Dirección Puerta de Enlace” y “Saltos” de la cabecera DHCP y el campo “MAC” de la opción de autenticación son colocados en cero para el cálculo del código MAC.

Un mensaje es tomado como válido por el receptor si el valor del campo “Detección de Repetición” es aceptable de acuerdo al método indicado en el campo “Método Det. Rep.” y si el código MAC calculado por el receptor es igual al código MAC contenido en la opción de autenticación. En la Figura 3.5 se puede observar un diagrama de flujo donde se detalla el conjunto de acciones realizadas durante el proceso de validación de un mensaje DHCP usando este método de autenticación.

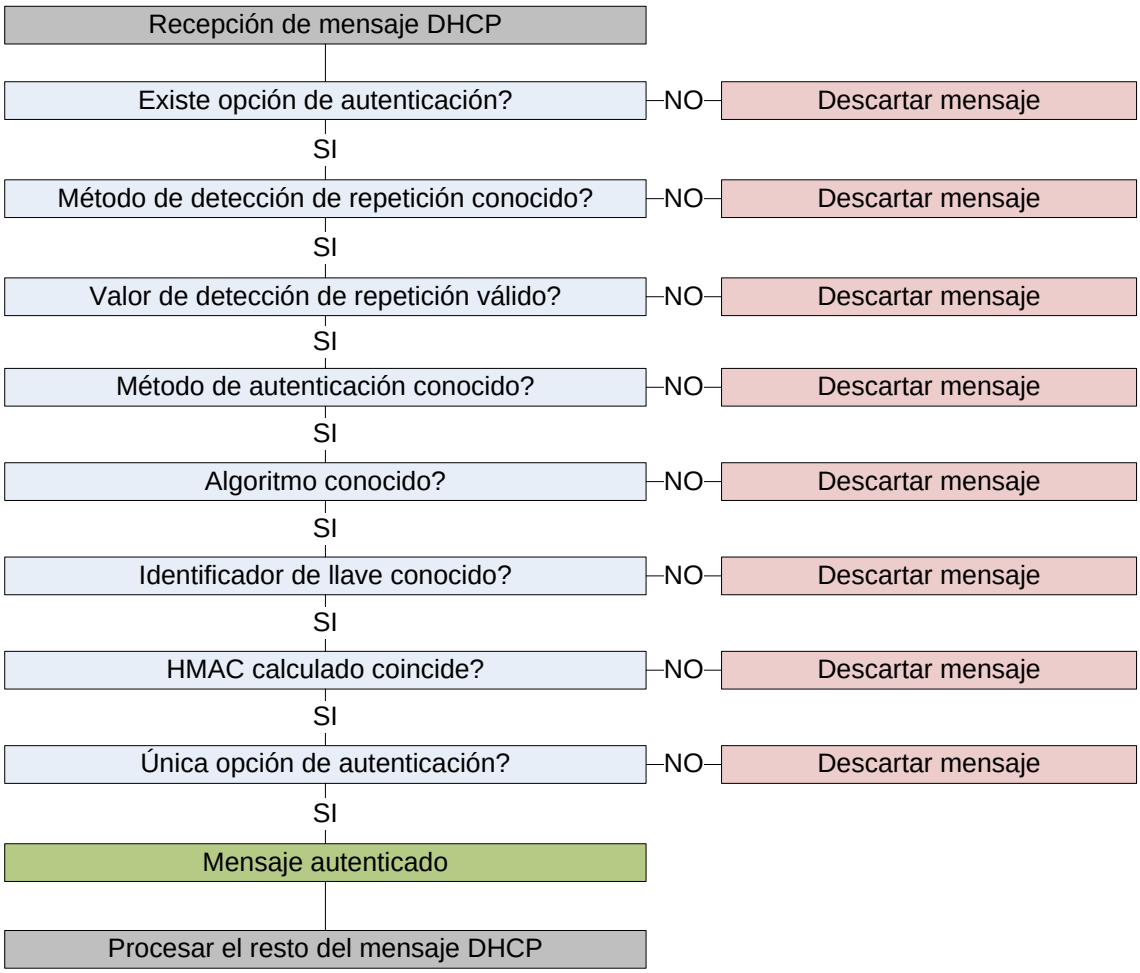

Figura 3.5 Diagrama de flujo de validación de mensajes usando Delayed Authentication.

4. Metodología y Herramientas

En este capítulo se describen las fases en las que se dividió la realización de este trabajo para cumplir con los objetivos propuestos. Además se describen las herramientas necesarias para llevar a cabo la elaboración del mismo.

4.1. Fases para la Elaboración del Presente Trabajo

Para la realización de este trabajo, se dividió el desarrollo del mismo en las siguientes fases:

- Investigación acerca de implementaciones existentes de DHCP.
- Selección de implementación de DHCP.
- Análisis de código fuente de la implementación seleccionada.
- Implementación de DHCP con autenticación.
- Diseño de los escenarios de prueba.
- Realización de pruebas.
- Análisis de los resultados.

4.1.1. Investigación de Implementaciones Existentes de DHCP

En esta fase se realiza la búsqueda de distintas implementaciones de clientes y servidores DHCP que sigan la filosofía de "código abierto". Esto con la finalidad de poder agregar el soporte a la autenticación en mensajes DHCP y llevar a cabo la solución propuesta en este trabajo.

4.1.2. Selección de Implementación de DHCP

Esta fase consiste en la elección de una de las implementaciones que hayan sido encontradas durante la fase anterior. Luego se procede a seleccionar una de ellas basado en varios criterios establecidos previamente.

4.1.3. Análisis de Código Fuente

Una vez seleccionada una implementación de DHCP sobre la cual trabajar, se procedió a revisar el código fuente de la misma para entender cómo se encuentra estructurada la aplicación, cómo es el flujo de la misma y qué elementos interactúan para llevar a cabo el proceso de concesión de una dirección IP usando esta implementación de DHCP. Posteriormente, se puede determinar cómo llevar a cabo la solución propuesta en este trabajo, es decir el envío, recepción y validación de mensajes DHCP con autenticación.

4.1.4. Implementación de DHCP con autenticación

Luego de haber analizado la implementación seleccionada, se procedió a extender el código fuente de la misma. Esto con la finalidad de añadirle las funcionalidades necesarias para llevar a cabo la autenticación en mensajes DHCP tal como se describe en el RFC 3118. Además se agregan ciertos controles para llevar a cabo la validación de los mensajes. De esta forma se puede determinar si un mensaje debería ser aceptado o descartado si el ente que envió el mensaje provee información suficiente que demuestre que se encuentra debidamente autenticado o no respectivamente.

4.1.5. Diseño de los Escenarios de Prueba

En esta fase se plantean los diversos escenarios de prueba que se configuraron para evaluar varios aspectos de la solución desarrollada en este trabajo. Estas pruebas son diseñadas teniendo en cuenta cómo debería de ser el comportamiento de la solución desarrollada ante casos de autenticación exitosos, casos de fallo por distintos motivos, topología de la red en la que se encuentran los elementos que componen al sistema DHCP, entre otros.

4.1.6. Realización de Pruebas

Se realizan las pruebas planificadas para comprobar la estabilidad y el correcto funcionamiento del sistema ante los varios casos donde es usada la opción de autenticación. Durante la realización de las pruebas se recopilan ciertos registros y datos de la aplicación para su futuro análisis

4.1.7. Análisis de los Resultados

Finalmente, se podrán examinar e interpretar los datos obtenidos en la fase anterior para concluir si se lograron los objetivos planteados en este trabajo.

4.2. Herramientas Utilizadas

Para la realización del presente trabajo se usaron distintas herramientas de software que se describen a continuación:

- **Debian GNU/Linux:** Debian [16] es una distribución del sistema operativo Linux y está compuesto de varios paquetes de software siguiendo la filosofía de software libre bajo la licencia GNU GPL.

- **Ubuntu:** Ubuntu [17] es una distribución del sistema operativo Linux y basada en "Debian". Ubuntu se destaca por ser una de las más populares distribuciones de Linux usadas en computadores personales.

- **Internet Systems Consortium (ISC) DHCP:** Es la implementación código abierto de DHCP más usada en la Internet. Posee un grupo de trabajo que constantemente realiza actualizaciones y una lista de correos donde una comunidad de desarrolladores y usuarios pueden realizar aportes o preguntas acerca del mismo. Es distribuido bajo los términos de la Licencia ISC [7].

- **GCC (GNU Compiler Collection):** Es un sistema de compilación producido por el Proyecto GNU y usado como el compilador estándar por la mayoría de los sistemas operativos derivados de Unix, entre ellos Linux. Mayormente utilizado para compilar programas en lenguaje C/C++, pero también posee soporte para manejar otros lenguajes de programación como Objective-C, Fortran, entre otros [8].

- **Make (GNU Make):** Es una utilidad usada en varios sistemas operativos que permite la creación automatizada de programas ejecutables y bibliotecas a partir de un conjunto de archivos que contengan el código fuente del mismo [9].

- **SU (Substitute User):** Es una utilidad que permite cambiar de cuenta de usuario a través del intérprete de comandos. Es generalmente utilizado para elevar los privilegios del usuario actual a aquellos de un superusuario o administrador del sistema local [10].

- **VMware Workstation:** Es un hipervisor que permite la creación y ejecución de una o varias máquinas virtuales en una máquina física, denominada huésped. Cada máquina virtual puede ejecutar su propio sistema operativo (por ejemplo Windows, Linux, BSD, entre otros), no necesariamente siendo el mismo en ejecución bajo el huésped. Desarrollado por VMware, Inc [11].

- **Wireshark:** Es un analizador de paquetes y protocolos de red usado para capturar y analizar el tráfico en una red de computadoras. Comúnmente usado para solucionar problemas en redes y en el desarrollo de protocolos de software y protocolos de comunicación [12].

5. Diseño de la Solución

En este capítulo se describirá el conjunto de tareas que se realizarán para poder implementar la solución a la problemática presentada en este trabajo y así cumplir con los objetivos planteados.

5.1. Investigación de Implementaciones Existentes de DHCP

El objetivo principal de este trabajo es la de lograr la implementación de un cliente y servidor DHCP que tenga soporte para los mensajes de autenticación. Para ello se sigue una investigación previa en el tema [13] en la que se especifica la necesidad de implementar dicho mecanismo, la carencia de dicho mecanismo en las implementaciones actuales de DHCP y cómo implementar el mismo.

Ya que el objetivo principal a alcanzar en este trabajo es netamente lograr la autenticación en mensajes DHCP y que se puede encontrar una variedad de implementaciones de DHCP ya bien establecidas en la Internet, se consideró innecesario desarrollar la solución desde cero y se optó por seleccionar alguna implementación que permitiera lograr los objetivos con mayor rapidez.

Durante la búsqueda, se observó que la cantidad de implementaciones código abierto de clientes DHCP son algo escazas, además de que en su gran mayoría están orientadas para sistemas Linux. Por ello se recopiló una lista de posibles implementaciones a elegir para el sistema operativo antes mencionado y desarrollar la solución sobre el mismo.

5.2. Selección de Implementación de DHCP

Luego de realizar la investigación descrita en el paso anterior, se procedió a seleccionar a una entre todas las implementaciones encontradas, las cuales se nombran a continuación:

5.2.1. Udhcpc y Udhcpd (Busybox)

Udhcpc y Udhcpd (micro cliente - micro servidor DHCP) son unas implementaciones de DHCP desarrolladas en el lenguaje de programación C y generalmente distribuidas dentro del paquete de utilidades Busybox [18], que consta de herramientas pensadas para ser ejecutadas en dispositivos con sistemas embebidos. Esto hace que estas implementaciones sean sumamente ligeras y sencillas, sin embargo esto también representa un inconveniente ya que sólo proveen un conjunto de funcionalidades básicas y carecen de diversos

mecanismos de controles o validaciones de mensajes, por lo que el producto final pudiese no cumplir con algunos aspectos deseables de seguridad.

5.2.2. LoosyDHCP, dhcp4java y JDHCPD

Las implementaciones LoosyDHCP [19], dhcp4java [20] y JDHCPD [21] están desarrolladas bajo el lenguaje de programación Java, lo cual podía representar un beneficio a favor debido a la portabilidad que provee dicho lenguaje de programación, es decir el mismo código funcionaría indistintamente del sistema operativo. Sin embargo, se encontró que estas implementaciones consistían únicamente de servidores DHCP y en una de ellas se indicaba el motivo que restringía la implementación de clientes en este lenguaje: Java, al ser un lenguaje de muy alto nivel, no poseía ningún método para acceder a la información de la capa de enlace por lo que el cliente no podría construir correctamente los mensajes iniciales. Además, Java coloca una dirección IP autogenerada cuando la dirección IP fuente de un mensaje es la dirección 0.0.0.0 y este es un aspecto necesario para el cliente en el proceso de DHCP, por lo que el desarrollo de la solución bajo este lenguaje de programación quedó descartado rápidamente.

5.2.3. ISC DHCP

Esta implementación código abierto de DHCP es desarrollada por el Internet Systems Consortium [7] bajo el lenguaje de programación C y es actualmente la de mayor uso en la Internet, incluso siendo la implementación utilizada por defecto en varias distribuciones de Linux. Recibe mantenimiento constante por parte de un grupo de trabajo y también por medio de aportes de la comunidad de usuarios y desarrolladores.

Cuenta con una lista de correos donde el equipo del ISC y la comunidad intercambian comentarios, dudas, soluciones, entre otros, lo que representaba una ventaja sobre las otras implementaciones que fueron evaluadas. Esta es la implementación que fue seleccionada para añadirle soporte para mensajes de autenticación en DHCP y se eligió la versión 4.2.4-P1.

5.3. Análisis de Código Fuente

Para poder implementar los cambios necesarios y añadir el soporte a los mensajes de autenticación en DHCP, hay que primero analizar el código fuente de la implementación para entender las piezas en las que está estructurado el ISC DHCP y comprender la forma en la que los desarrolladores del ISC DHCP realizan el proceso de construcción, procesamiento, envío y recepción de los mensajes. De esta forma se puede obtener una idea del enfoque a seguir para modificar el código adecuadamente, intentando reutilizar la mayor cantidad de código posible.

5.3.1. Estructura de la Implementación

El código del ISC DHCP se encuentra dividido a lo largo de más de 100 archivos fuentes y bibliotecas, cada uno ubicado dentro de algún directorio en particular dependiendo del papel que cumpla, por ejemplo, archivos que son usados únicamente por el cliente son ubicados en un directorio llamado *client*, del mismo modo, el servidor posee un directorio *server*, utilidades en común se encuentran en el directorio *common*, las bibliotecas en general se encuentran en el directorio *includes*, y así sucesivamente.

Para poder adaptar la implementación actual a las indicaciones descritas en el RFC 3118, se observó que los archivos a los que principalmente había que realizarle modificaciones eran los siguientes:

- **dhcpd.c:** Este archivo es el punto inicial del servidor. Contiene el código respectivo para el arranque del demonio y sus parámetros de ejecución, rutas de archivos de configuración, registros de clientes e inicio de rutinas de escucha y despacho de mensajes, entre otras.

- **dhcp.c:** Contiene todo lo que concierne a la recepción, procesamiento, clasificación y envío de mensajes DHCP por parte del servidor.

- **dhclient.c:** En este archivo se encuentra en general todo lo relacionado al cliente, como lo es el arranque del demonio del cliente y sus parámetros de ejecución, rutas de archivos de configuración, chequeo de direcciones IP adquiridas previamente, inicio de rutinas de recepción, procesamiento, clasificación y envío de mensajes DHCP por parte del cliente, entre otras.

- **dhcpd.h:** Esta biblioteca es una de las piezas centrales del ISC DHCP. En ella se encuentra una gran parte de las estructuras utilizadas en muchas de las funciones de toda la implementación. También se encuentran los prototipos de muchas de las funciones ubicadas en distintos archivos que se encuentran en los demás directorios para que puedan ser utilizados desde casi cualquier parte del código que compone al DHCP.

- **dst_all.c:** Este archivo en sí no es original del código fuente provisto por el ISC, pero es una recopilación de los archivos contenidos en el directorio *dst*, que corresponde al Digital Signature Toolkit desarrollado por Trusted Information Systems. Consiste en un conjunto de utilidades para la inicialización y uso de funciones de cifrado de datos por medio de algoritmos de hashing y llaves. El motivo por el cual se realizó tal recopilación será descrito más adelante.

5.3.2. Enfoque a Seguir Para la Modificación de la Implementación

Con la información obtenida a partir del análisis del código fuente, se pudo idear dónde había que realizar cambios al código y cómo sería el enfoque a seguir para realizar dichos cambios. En particular se deseaba hacer los cambios de tal forma de que el código añadido fuese lo más similar al código desarrollado por el grupo de trabajo del ISC, es decir usar las mismas funciones de construcción y procesamiento de las distintas opciones para incluir así la opción de autenticación en los mensajes DHCP. Sin embargo, se tuvo que optar por modificar directamente el mensaje DHCP a nivel de bytes ya que no se encontró suficiente documentación acerca de las funciones y estructuras ya existentes que permitieran hacer un uso correcto de las mismas, como por ejemplo:

- **Complejidad del formato de la opción de autenticación:** Cada opción conocida por el ISC DHCP está declarada en un archivo llamado *tables.c*, en el que se indica el número de código asignado a la opción, un "universo" al que pertenecen y el formato o los tipos de datos que componen dicha opción. Los tipos de datos ya se encuentran predefinidos y son similares a los encontrados en los lenguajes de programación, como por ejemplo "entero de 32 bits", "cadena de texto", "dirección IPv4", entre otros. El problema radica en que la opción de autenticación cuenta con una variedad de campos de distintas longitudes e incluso un número variable de campos. Esto hace que sea difícil designarle una especie de plantilla acerca de cómo se encuentra estructurada la opción de autenticación. Para estos casos particulares es posible crear un *universo* únicamente para ser usado por una opción en particular pero esto acarrea el siguiente inconveniente.

- **Complejidad de la estructura "universe":** La estructura *universe*, o como se ha llamado anteriormente el *universo*, se encuentra declarada en *tree.h*. En ella se determina cómo va a ser el manejo de las opciones que pertenecen a ese universo, como lo son las funciones que estarán encargadas de la búsqueda de opciones, agregación, encapsulación, eliminación, análisis, entre otras funciones. Lamentablemente, esta estructura es sumamente compleja, no posee documentación que ayude a comprender su funcionamiento y por falta de tiempo no se pudo invertir el esfuerzo necesario para entender cómo hacer uso de ella.

Dado estos inconvenientes se intentó obtener respuestas por parte de la comunidad o el grupo de trabajo del ISC a través de la lista de correos pero lamentablemente ninguno de los dos ofreció apoyo alguno. Por ello se planteó modificar a nivel de bytes el mensaje DHCP luego de que fuese construido por el código original de DHCP y justo antes de ser transmitido a través del medio de comunicación. De igual forma, al recibir un mensaje con la opción de autenticación presente, se procedería a verificar la autenticidad del mensaje antes de que el mensaje fuese procesado por el código original de DHCP.

De esta manera el código a añadir funcionaría como una especie de módulo que funcionase sobre el proceso de DHCP pero sin alterar en ninguna forma el comportamiento original del mismo, asegurando que el código original de DHCP nunca encontrase rastros de que existe o haya existido la opción de autenticación en el mensaje.

Para ello en el lado del emisor una función se encargaría de modificar el mensaje original DHCP, donde se le agregaría la opción de autenticación y el mensaje resultante sería enviado a través de la red, tal como se ilustra en la Figura 5.1. De modo similar, en el lado del receptor se ejecutaría un segmento de código que se encargaría de validar y eliminar la opción de autenticación en un mensaje recibido antes de que el mismo sea procesado por el código original de DHCP, tal como se ilustra en la Figura 5.2.

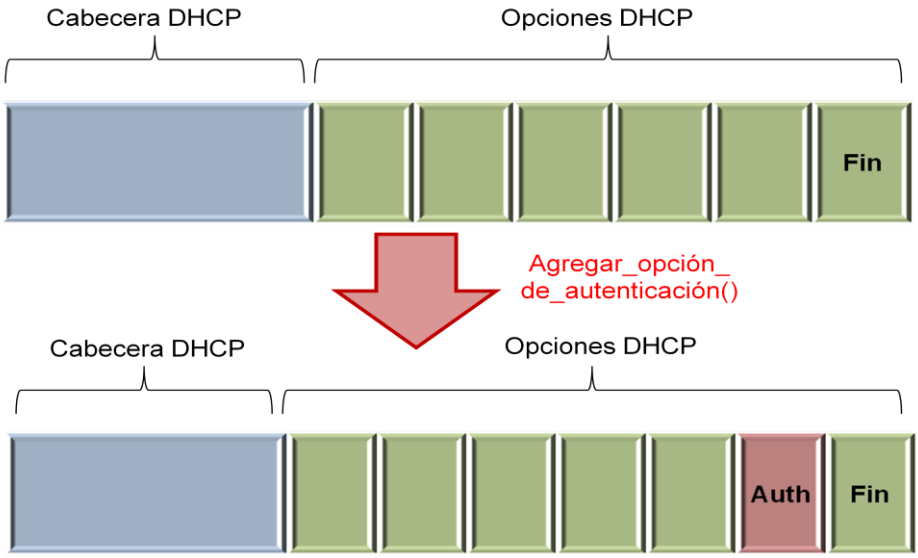

Figura 5.1 Inserción de opción de autenticación a un mensaje DHCP.

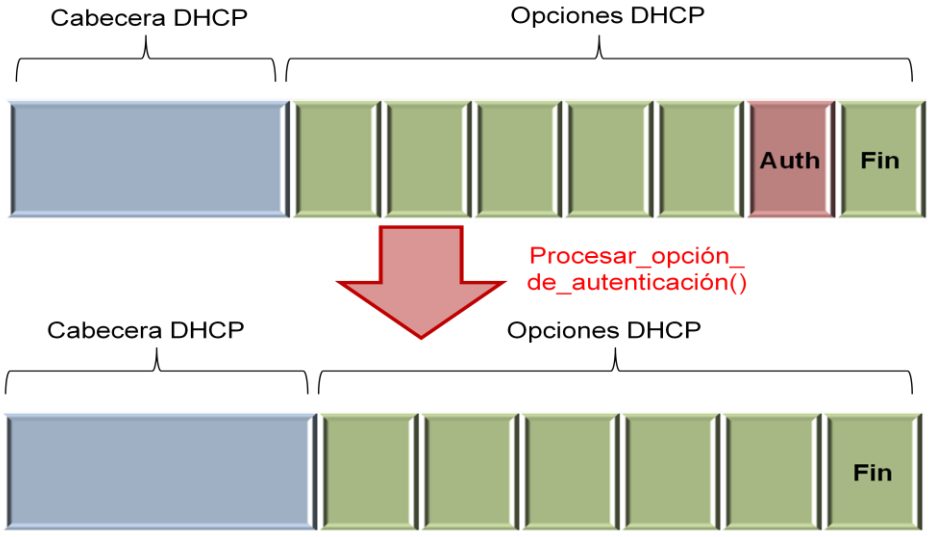

Figura 5.2 Validación y eliminación de la opción de autenticación en un mensaje DHCP.

6. Desarrollo de la Solución

Partiendo del análisis realizado en el capítulo anterior se pudo idear en líneas generales cómo realizar cada acción que nos permitiría cumplir con las indicaciones descritas en el RFC 3118 y con los objetivos planteados para este trabajo. Para lograr los objetivos planteados, la implementación de la solución seguirá una metodología de desarrollo incremental en el que progresivamente fue agregando nuevas funcionalidades hasta lograr un resultado satisfactorio y comprobando que estas no alteran de ninguna forma a las ya existentes.

A continuación se describirá ciertas consideraciones generales que se tomaron en cuenta al momento de la implementación de la solución y posteriormente se entrará en detalle en lo que concierne al desarrollo de cada uno de los métodos de autenticación que fueron implementados.

6.1. Consideraciones de Diseño

Durante el desarrollo de la solución, se tenía cierta libertad para implementar muchos de los elementos que compondrían a la aplicación. Ya que no se pudo reutilizar ciertos fragmentos del código del ISC DHCP, todo lo que correspondía al análisis de la opción de autenticación, marcas de tiempo para prevenir ataques de repetición, secretos o llaves utilizadas durante el proceso de autenticación, entre otros, tenía que ser manejado completamente por el código a agregar. Para ello se dispuso a crear varios archivos donde se almacenará toda esa información. En el Anexo Nº A se puede obtener más información acerca del contenido de los archivos.

Además, por ambigüedad del RFC 3118 en algunos aspectos de la inclusión de la opción de autenticación en los mensajes DHCP, se tuvo que asumir o decidir cómo actuar en unas determinadas situaciones que no mencionaba qué hacer explícitamente.

Las consideraciones que se tomaron son:

- El procesamiento de la opción de autenticación deberá ser transparente para el código del DHCP. Luego de ser procesado, la opción de autenticación es removida del mensaje DHCP. El mensaje resultante será procesado por el código original del ISC DHCP como si nunca hubiese existido dicha opción.

- Un mensaje DHCP debe contener solamente una opción de autenticación. aún cuando no hayan indicaciones al respecto en el RFC 3118. Sin embargo en el RFC 3315 [14] acerca de DHCPv6 se encuentra una sección dedicada a la autenticación donde sí indican que sólo debe estar presente

una opción de autenticación por mensaje. Se optó por cumplir esa indicación por ser información más reciente y probablemente más madura.

- Los archivos que contienen los registros de llaves utilizadas y marcas de tiempo para prevenir ataques de repetición sólo serán actualizados luego de que un mensaje DHCP sea totalmente y satisfactoriamente procesado. Esto hará que no se realicen cambios por mensajes inválidos, evitando realizar tareas innecesarias del lado del receptor.

- Al usar el método de *Configuration Token* no será obligatorio el uso de la opción *Client Identifier*. En el método *Delayed Authentication* sí es obligatorio, pero se consideró que el método de *Configuration Token* debía permanecer lo suficientemente simple para no requerir mayor intervención o configuración por parte del administrador del sistema.

- Al usar el método de *Configuration Token*, la opción de autenticación deberá estar presente en cada mensaje DHCP. En el RFC 3118 no se profundiza mucho en este método a diferencia de *Delayed Authentication*, en el que en casi todo mensaje DHCP se agrega la opción de autenticación porque ya es conocida la llave, por lo que se decidió aplicar el mismo principio para ambos métodos de autenticación.

- Al usar el método de *Delayed Authentication* si no se especificó previamente en el archivo de configuración del cliente algún *Client Identifier*, añadiremos dicha opción al momento de construir la opción de autenticación. El valor del *Client Identifier* en este caso corresponderá a la dirección MAC de la interfaz por la que el cliente esté enviando el mensaje, ya que dicho valor es usualmente utilizado por defecto en otras implementaciones de DHCP. Se consideró que esto pudiese facilitar el proceso de configuración de cada cliente porque así se evitaría el descarte de mensajes DHCP por ausencia de un *Client Identifier* en los mensajes del cliente, por ejemplo por algún descuido en la configuración por parte del administrador.

- El único método de detección de ataques de repetición soportado hasta los momentos será el contador monótonamente creciente, ya que es el único especificado en la bibliografía [5].

- Las llaves a utilizar por cada uno de los dispositivos serán distribuidas siguiendo el enfoque de pre-distribución de llaves. La distribución manual de cada conjunto de llaves con sus respectivos identificadores estará a cargo del administrador del sistema debido a que es sumamente difícil la obtención de llaves para los dispositivos que no poseen una dirección IP.

6.2. Configuration Token

Para comenzar con el desarrollo de la solución, se procedió primero con el soporte al método de autenticación *Configuration Token*, ya que era el más sencillo de implementar y además serviría de esqueleto para el desarrollo del segundo método de autenticación. Para ello se agregó una entrada adicional a los argumentos aceptados al llamar al demonio por el intérprete de comandos, que corresponde al tipo de método de autenticación a utilizar. En este caso corresponde al método *Configuration Token* seguido de la contraseña compartida, la contraseña se guardará en una variable para su posterior uso. De activarse este método de autenticación también se lee un archivo que contiene los valores de detección de ataques de repetición respectivos a los dispositivos con los que se haya realizado un intercambio de mensajes DHCP.

Luego cada vez que se fuera a enviar un mensaje DHCP se haría una llamada a una función justo antes de enviar el mensaje, la cual se encargará de construir y agregar la opción de autenticación con la información necesaria. El motivo por el que se eligió hacer las modificaciones en este punto en particular es para asegurar la transparencia del código añadido para el resto del sistema DHCP y porque en este punto el contenido del mensaje DHCP es fácilmente manipulable ya que consiste en una cadena de caracteres o bytes para efectos del lenguaje C.

Al usar *Configuration Token* en el lado del emisor, los valores de los campos *protocol*, *algorithm* y *RDM* se colocan en 0, se obtiene la hora actual en formato *Network Time Protocol* (NTP) a través de la función "gettimeofday()" para colocarla en el campo *replay detection* y se copia la contraseña en el campo *authentication information*. Luego se verifica el nuevo tamaño del paquete y se ajusta la longitud acordemente, dependiendo si se encuentra todavía por debajo de la longitud mínima o si es mayor. Finalmente se envía el mensaje.

Del lado del receptor, lo primero que se verifica es que esté presente la opción de autenticación en el mensaje DHCP. Luego se verifica que el valor de *replay detection* asociado al emisor del mensaje DHCP sea estrictamente mayor al anteriormente registrado en caso de existir, o en caso de no existir este sería el primer mensaje recibido de parte de ese emisor y se toma como válido. Se verifica que el campo *protocol* sea 0 (el código asignado a *Configuration Token*), al igual que los campos *algorithm* y *RDM* ya que este método no posee ningún algoritmo de cifrado y no admite ningún otro método de detección de ataques de repetición. Finalmente se verifica que la contraseña enviada sea exactamente igual a la contraseña provista al arranque del demonio por línea de comando. Si alguna de esas validaciones falla el mensaje es descartado y se registra un mensaje de error. Si el mensaje está debidamente autenticado se elimina la opción de autenticación y se vuelve a verificar la presencia de alguna opción de autenticación, la cual en caso de resultar positiva también se descarta el paquete por motivos anteriormente descritos. Si todas esas validaciones culminan exitosamente, se pasa el mensaje resultante al código original de DHCP. Si el

mensaje no es descartado por validaciones realizadas en el código original, se actualiza y se guarda en un archivo el valor del campo *replay detection* asociado al respectivo cliente. Esto se hace luego de que haya sido procesado el mensaje y antes de que sea enviado a alguna función despachadora de respuestas al mensaje actual.

El diagrama de secuencia en la Figura 6.1 resume el procesamiento de la opción de autenticación en mensajes DHCP intercambiados entre el cliente y el servidor.

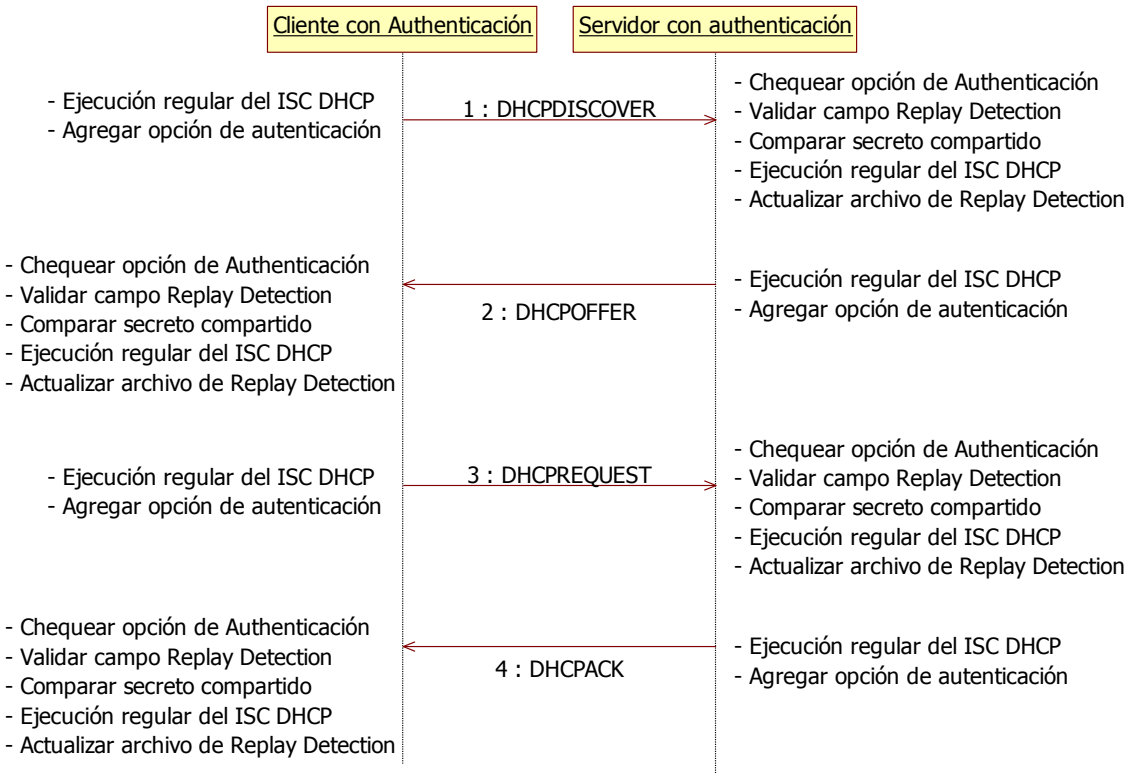

Figura 6.1 Proceso de autenticación usando el método *Configuration Token*.

6.3. Delayed Authentication

Para implementar el método de autenticación de *Delayed Authentication* se tomará como base el código ya desarrollado para el primer método de autenticación y se añadirán las funcionalidades necesarias, las cuales se describen a continuación.

De igual forma que el método anterior se añadió al código del demonio otra entrada a la lista de argumentos aceptados que correspondería al método de autenticación *Delayed Authentication*, seguido del algoritmo de hashing a usar, aunque actualmente sólo está soportado el algoritmo MD5. Al activarse este

método de autenticación, se guardan los valores del método usado para la autenticación, el algoritmo de hashing a usar, se carga la lista de valores de detección de ataques de repetición, una lista predistribuída a cada dispositivo que contiene las llaves y sus identificadores respectivos y una lista con las llaves que han sido utilizadas en intercambios de mensajes con otro dispositivo en particular. Además, algunos procedimientos, estructuras y algoritmos de hashing utilizados durante el proceso de autenticación deben ser inicializados previamente a su utilización, por lo que se hará dicha acción la primera vez que se tenga que calcular un hash, ya sea para el envío de un mensaje o para la validación de un mensaje recibido.

Al usar el método de *Delayed Authentication*, en el lado del emisor se coloca el campo *protocol* en 1, el campo *algorithm* con el valor respectivo del algoritmo seleccionado, el campo *RDM* con el valor respectivo al método de detección de repetición seleccionado, se obtiene la hora actual en formato NTP a través de la función "gettimeofday()" para colocarla en el campo *replay detection* y dependiendo del tipo de mensaje DHCP enviado se colocan datos o no en el campo de *Authentication information*. Si el mensaje es del tipo DHCPDISCOVER el campo permanece vacío, pero en caso contrario se coloca el identificador de la llave a usar en el campo *Secret-ID,* se calcula el código de autenticación del mensaje (HMAC) como se especifica en el RFC 2104 [6] y se coloca en el campo *HMAC*. Luego se verifica el nuevo tamaño del paquete y se ajusta la longitud acordemente, dependiendo si se encuentra todavía por debajo de la longitud mínima o si es mayor. Finalmente se envía el mensaje.

Del lado del receptor, lo primero que se verifica es que esté presente la opción de autenticación en el mensaje DHCP. Luego se verifica que el valor de *replay detection* asociado al emisor del mensaje DHCP sea estrictamente mayor al anteriormente registrado en caso de existir, o en caso de no existir este sería el primer mensaje recibido de parte de ese emisor y se toma como válido. Se verifica que el campo *protocol* sea 1 (el código asignado a *Delayed Authentication*), que el campo *algorithm* concuerde con el algoritmo en uso por el receptor y que el campo *RDM* pertenezca a un método de detección de ataques de repetición conocido por el receptor. A partir de este punto pueden ocurrir dos situaciones.

- Si un servidor recibe un mensaje del tipo DHCPDISCOVER, selecciona una llave al azar de entre todas las llaves conocidas, la registra como la llave a utilizar en los mensajes intercambiados con ese cliente en particular y el mensaje se toma como autenticado.

- Si se recibe otro tipo de mensaje, se almacena el HMAC presente en el mensaje y se procede a calcular el HMAC del mensaje con la llave indicada, la cuál debe ser conocida por el receptor. Posteriormente se verifica que el HMAC proporcionado y el HMAC calculado sean iguales. Si alguna de esas validaciones falla el mensaje es descartado y se registra un mensaje de error.

Si el mensaje está debidamente autenticado se elimina la opción de autenticación y se vuelve a verificar la presencia de alguna opción de autenticación, la cual en caso de resultar positiva también se descarta el paquete por motivos anteriormente descritos. Si todas esas validaciones culminan exitosamente se pasa el mensaje resultante al código original de DHCP. Si el mensaje no es descartado por validaciones realizadas en el código original, se actualizan y se guardan en sus respectivos archivos los valores del campo *replay detection* y la llave asociada al respectivo cliente. Esto se hace luego de que haya sido procesado el mensaje y antes de que sea enviado a alguna función despachadora de respuestas al mensaje actual.

El diagrama de secuencia en la Figura 6.2 resume el procesamiento de la opción de autenticación en mensajes DHCP intercambiados entre el cliente y el servidor.

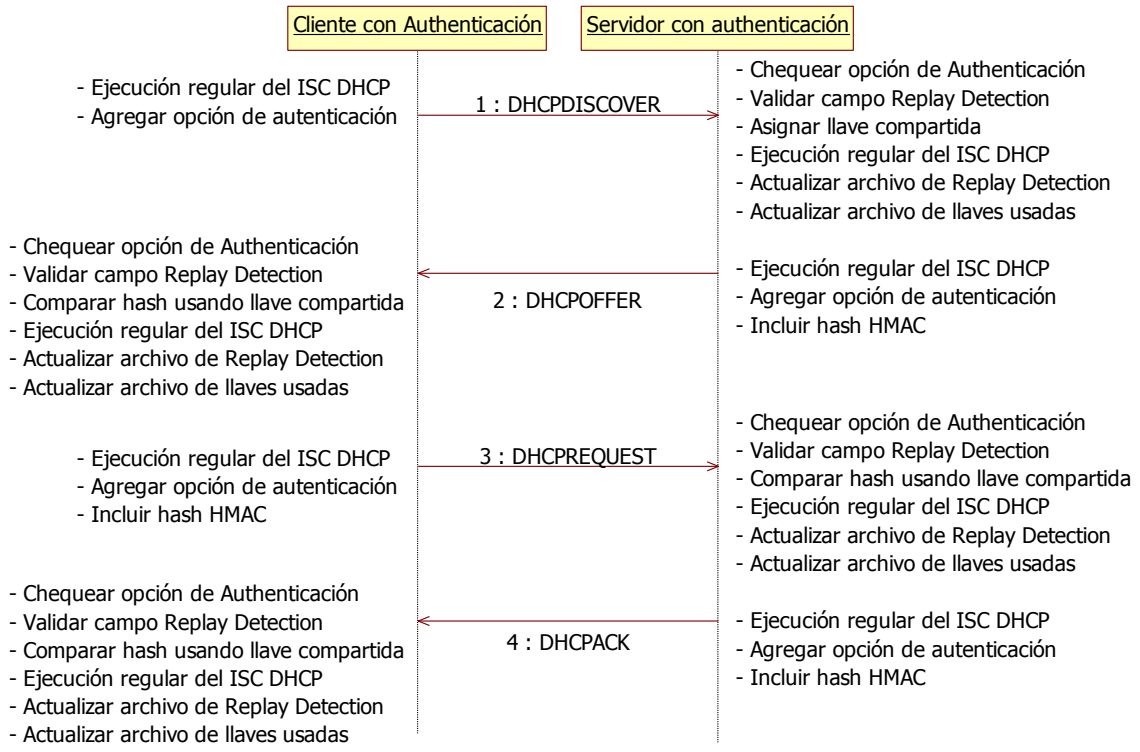

Figura 6.2 Proceso de autenticación usando el método *Delayed Authentication*.

7. Pruebas

En este capítulo se describen los escenarios de pruebas que fueron diseñados y realizados para evaluar varios aspectos de la implementación desarrollada. Para ello se prepararon básicamente tres conjuntos de pruebas.

7.1. Pruebas de Corrección

En estas pruebas se evalúa que el comportamiento de la implementación cumple con las indicaciones establecidas en el RFC 3118 en lo que respecta a la correcta validación de mensajes DHCP.

7.1.1. Preparación de las Pruebas

Para la realización de las pruebas de corrección se preparó un esquema de red donde sólo se encuentra un cliente y un servidor en la misma sub red como se ilustra en la Figura 7.1.

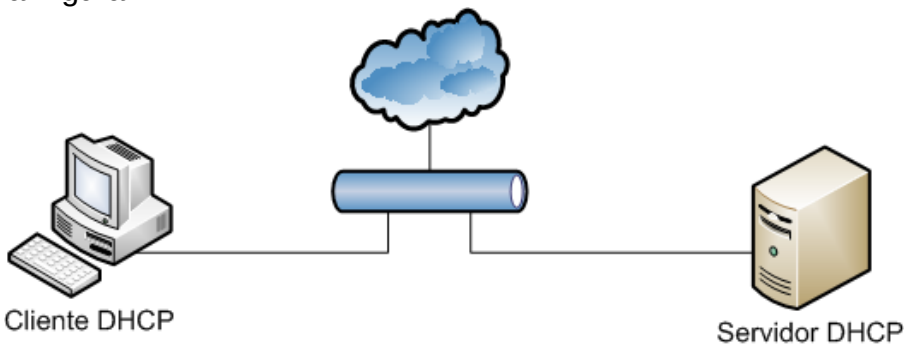

Figura 7.1 Topología de red para pruebas de corrección.

En este primer caso se prepararon diversos escenarios en los que el cliente envió mensajes autenticados correctamente y envió mensajes con variados errores en la opción de autenticación para que los mismos fueran descartados apropiadamente por el servidor. Un segundo caso es el escenario donde se encontraba un agente de relevo DHCP entre cliente y servidor como se ilustra en la Figura 7.2, para comprobar que la solución funciona a través de agentes de relevo.

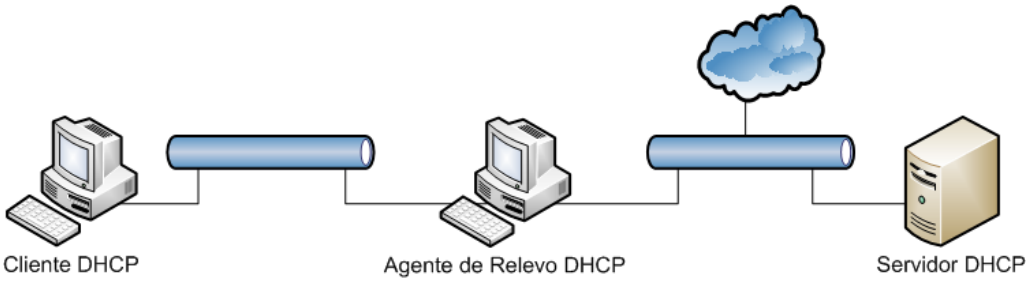

Figura 7.2 Topología de red para pruebas de corrección con agentes de relevo DHCP.

Por último se realizó una prueba que consistió en colocar varios clientes y servidores en la misma sub red como se muestra en la Figura 7.3, cada uno usando o no algún método de autenticación y compartiendo algún secreto o llaves que permitía el intercambio de mensajes con otro dispositivo. Es decir, los clientes podían adquirir una dirección IP sólo de aquellos servidores con los que se compartía el mismo secreto o conjunto de llaves y utilizaban el mismo método de autenticación, o bien si no utilizaban ningún método de autenticación.

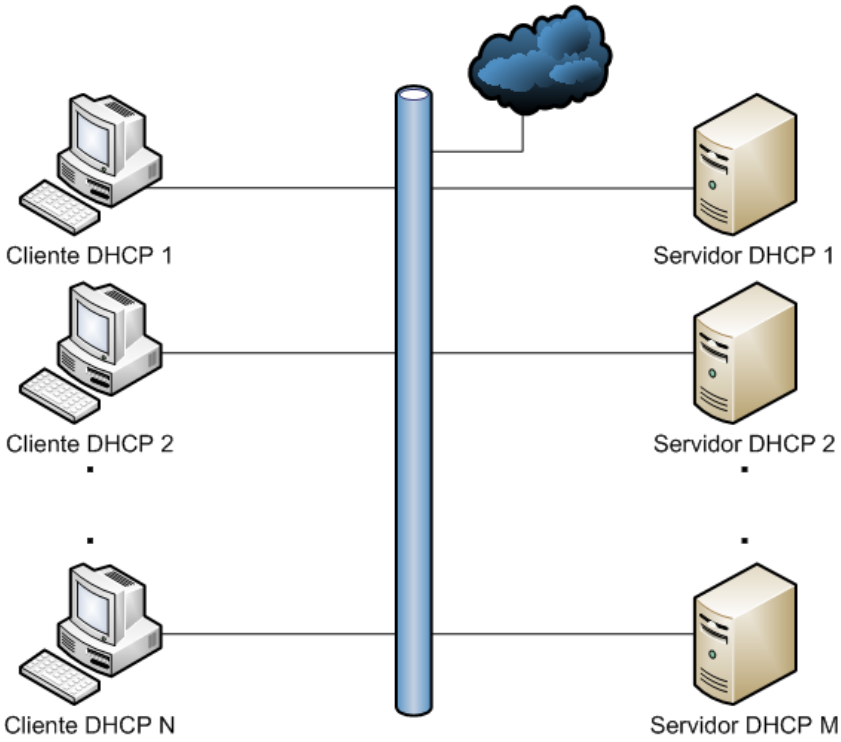

Figura 7.3 Topología de red para pruebas de corrección con varios clientes y servidores DHCP.

7.1.2. Ejecución de las Pruebas

Los escenarios que se plantearon para evaluar el correcto funcionamiento de la solución son los siguientes:

7.1.2.1. Opciones de autenticación válidas

En este escenario el cliente y el servidor DHCP se iniciaron usando el mismo método de autenticación y utilizando el mismo secreto compartido o conjunto de llaves, por lo que el cliente enviaba mensajes DHCP autenticados satisfactoriamente y podía adquirir información de configuración de red por un período de tiempo. En la Figura 7.4 y Figura 7.5 se pueden apreciar partes de una captura de Wireshark donde se muestra la opción de autenticación válida usando los métodos *Configuration Token* y *Delayed Authentication* respectivamente.

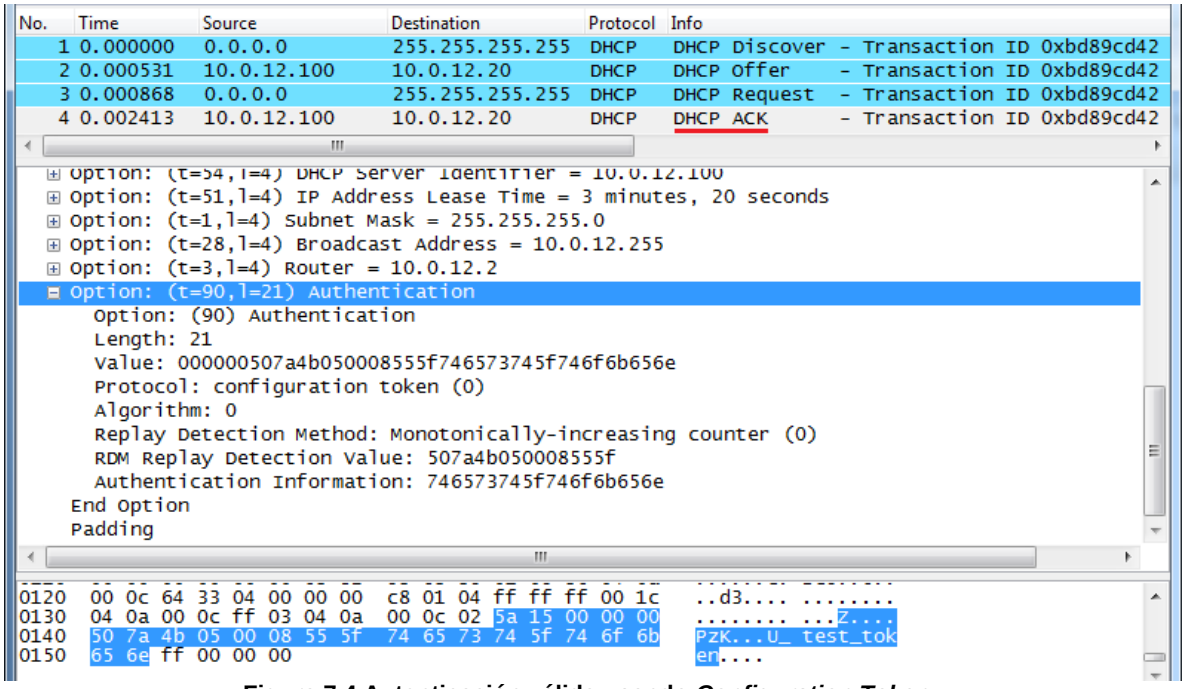

Figura 7.4 Autenticación válida usando *Configuration Token*.

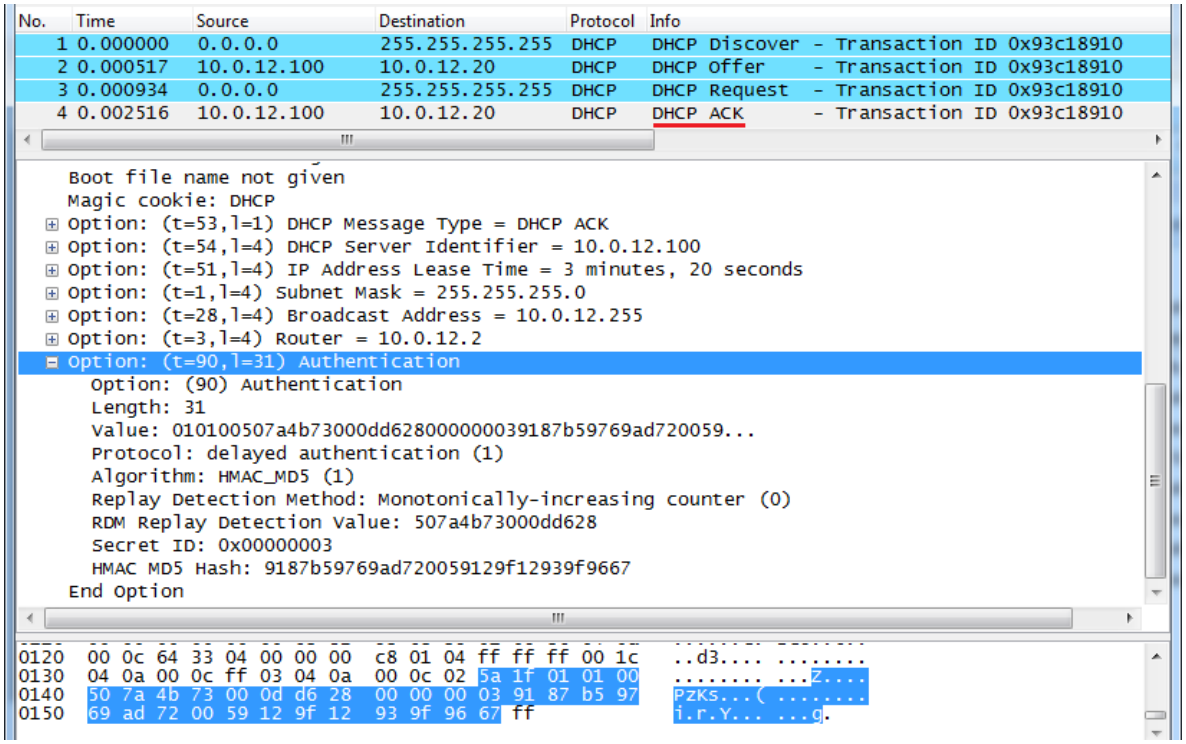

Figura 7.5 Autenticación válida usando *Delayed Authentication*.

7.1.2.2. Opción de autenticación no presente

En este escenario el cliente se inició sin usar algún método de autenticación. El servidor recibió un mensaje DHCPDISCOVER sin opción de autenticación, por lo que descartó el mensaje y el cliente no pudo obtener información de configuración de red. De forma similar, un cliente que recibió un mensaje sin la opción de autenticación descartó dicho mensaje. En la Figura 7.6 se puede apreciar una captura de un mensaje que no posee la opción de autenticación.

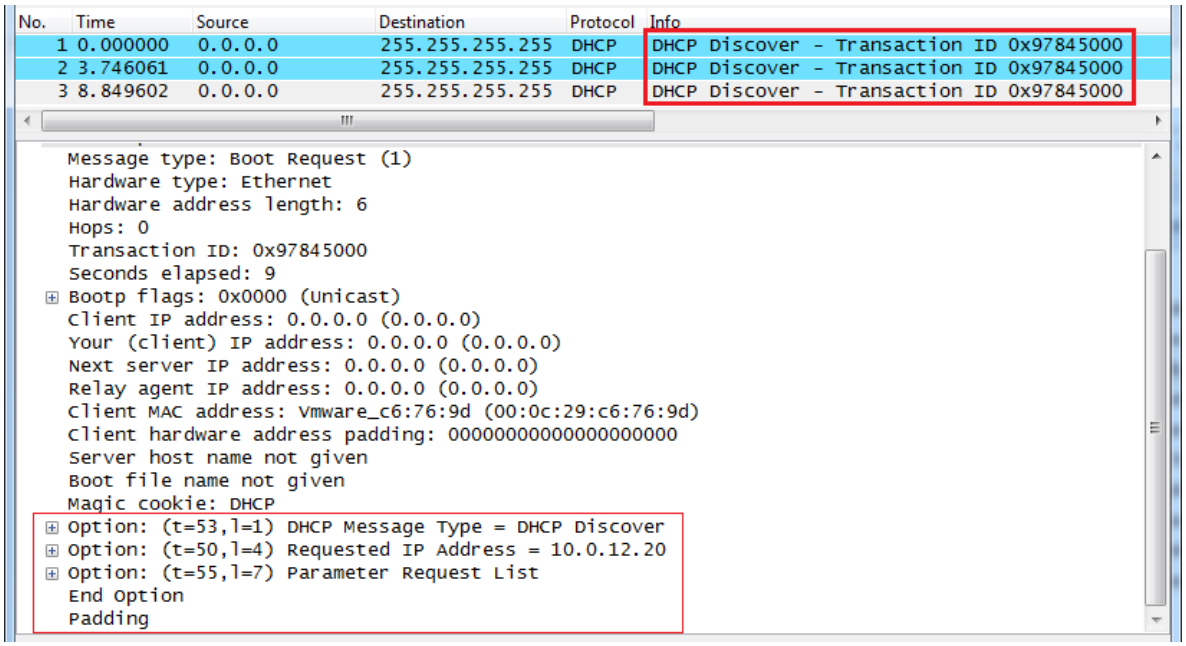

Figura 7.6 Ausencia de opción de autenticación.

7.1.2.3. Uso de clave o llave incorrecta

En este escenario para el caso de *Configuration Token*, el cliente y el servidor usaron un secreto distinto ("wrong_token" y "test_token" respectivamente), por lo que ambos descartaron cualquier mensaje que se enviaron entre sí de acuerdo a lo establecido en el RFC 3118. Para el caso de *Delayed Authentication*, si no se encontraba un identificador de llave o si la llave era distinta, el cálculo del hash o no se podía hacer o resultaba en un hash distinto, por lo que se descartaba el mensaje. La captura en la Figura 7.7 muestra el uso de un token distinto mientras que la captura en la Figura 7.8 muestra el uso de identificadores no aceptados en *Delayed Authentication* (el cliente sólo posee la llave con ID 1 y el servidor usa una llave con ID 2).

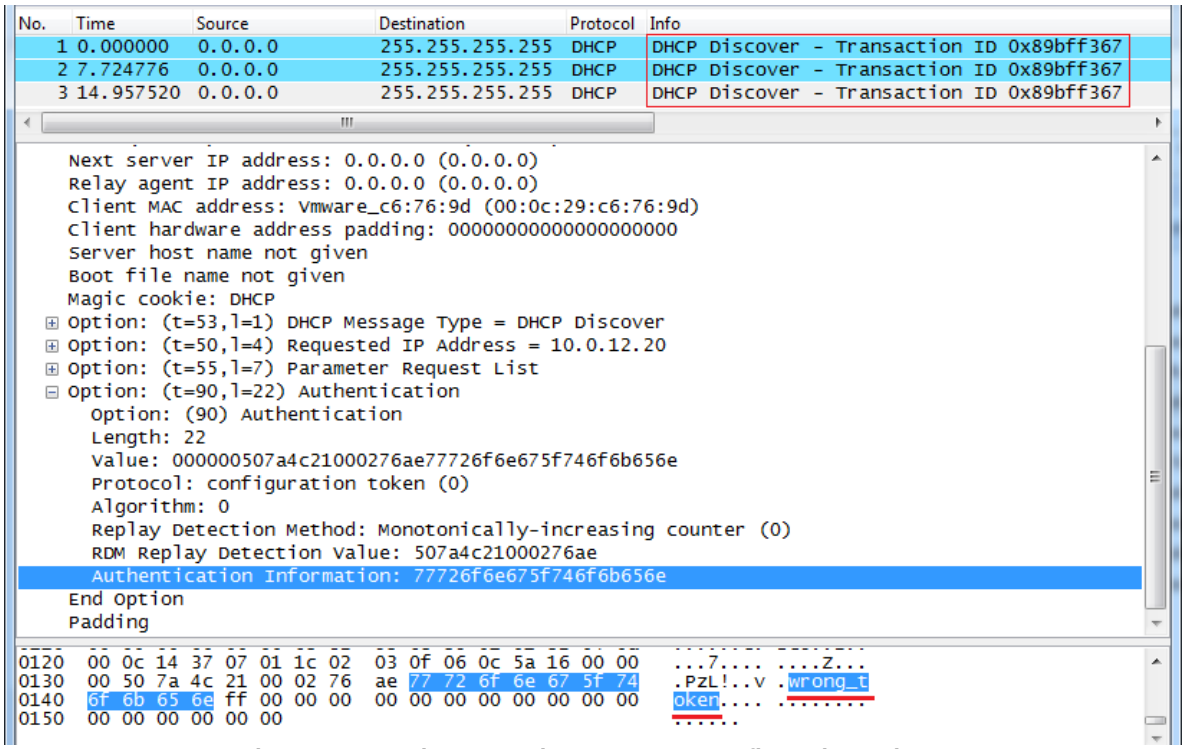

Figura 7.7 Uso de secreto incorrecto en *Configuration Token*.

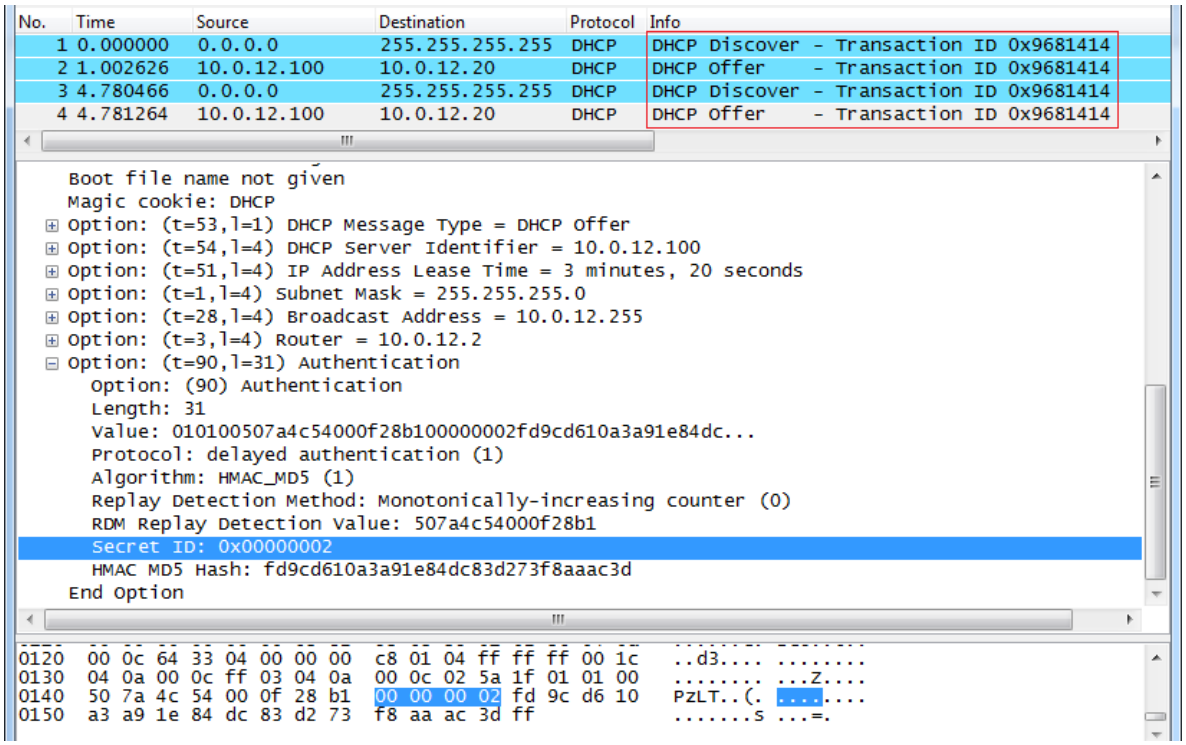

Figura 7.8 Llave incorrecta en *Delayed Authentication*.

7.1.2.4. Valor de detección de repetición inaceptable

En este escenario se modificó en el servidor el archivo que contiene los valores de detección de repeticiones y se cambió la marca de tiempo del cliente a una fecha en el futuro como se aprecia en la Figura 7.9. El servidor recibió un mensaje con un valor de detección inferior (no aceptable) por lo que el mensaje fue descartado. En la Figura 7.10 se muestra que el valor de detección de repetición enviado por el cliente es menor que el valor encontrado en el archivo del servidor.

Figura 7.9 Modificación en archivo de Replay Values del servidor.

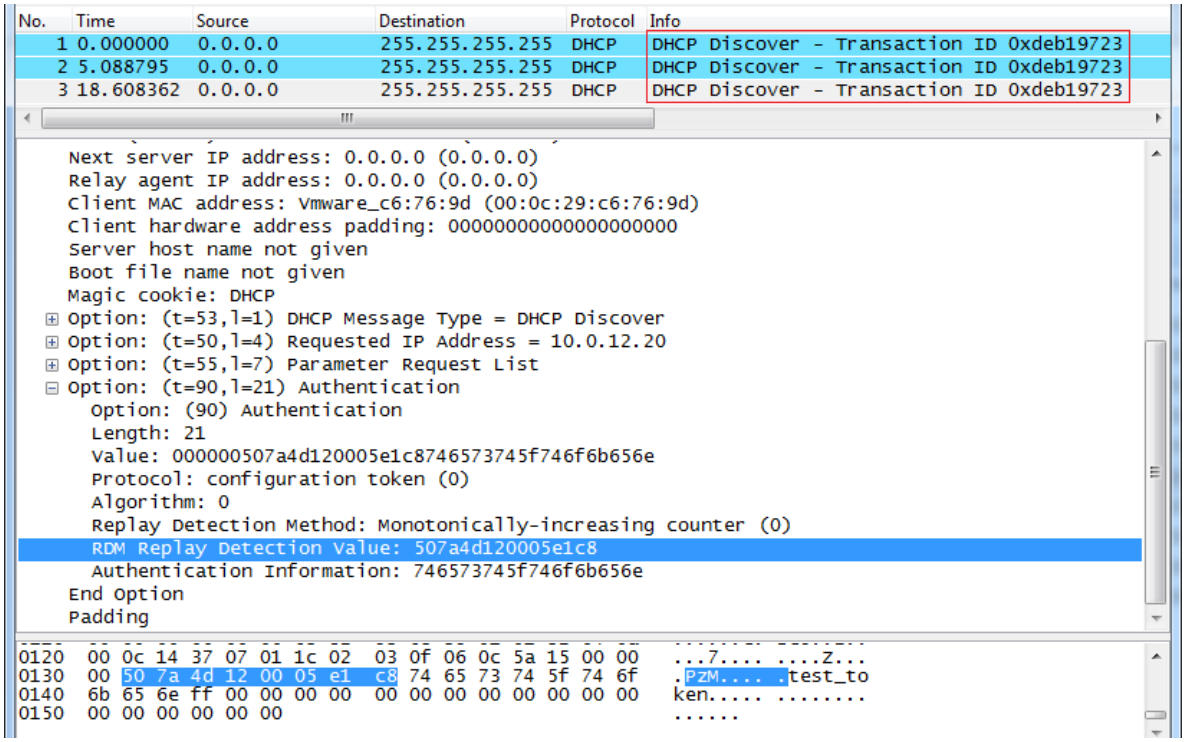

Figura 7.10 Valor no aceptable para el método de detección de repetición.

7.1.2.5. Valores no aceptados para los campos *protocol, algorithm* o *RDM*

En este escenario el cliente se modificó para enviar valores no aceptados o conocidos por el servidor para los campos mencionados (por simplicidad sólo se mostrará una de las pruebas, correspondiente al campo *RDM*). El servidor descartó el mensaje ya que no conocía el método de detección de ataques de repetición siendo usado por el cliente. La captura en la Figura 7.11 refleja este hecho y se resalta la palabra "Unknown", indicando que es un método no soportado actualmente.

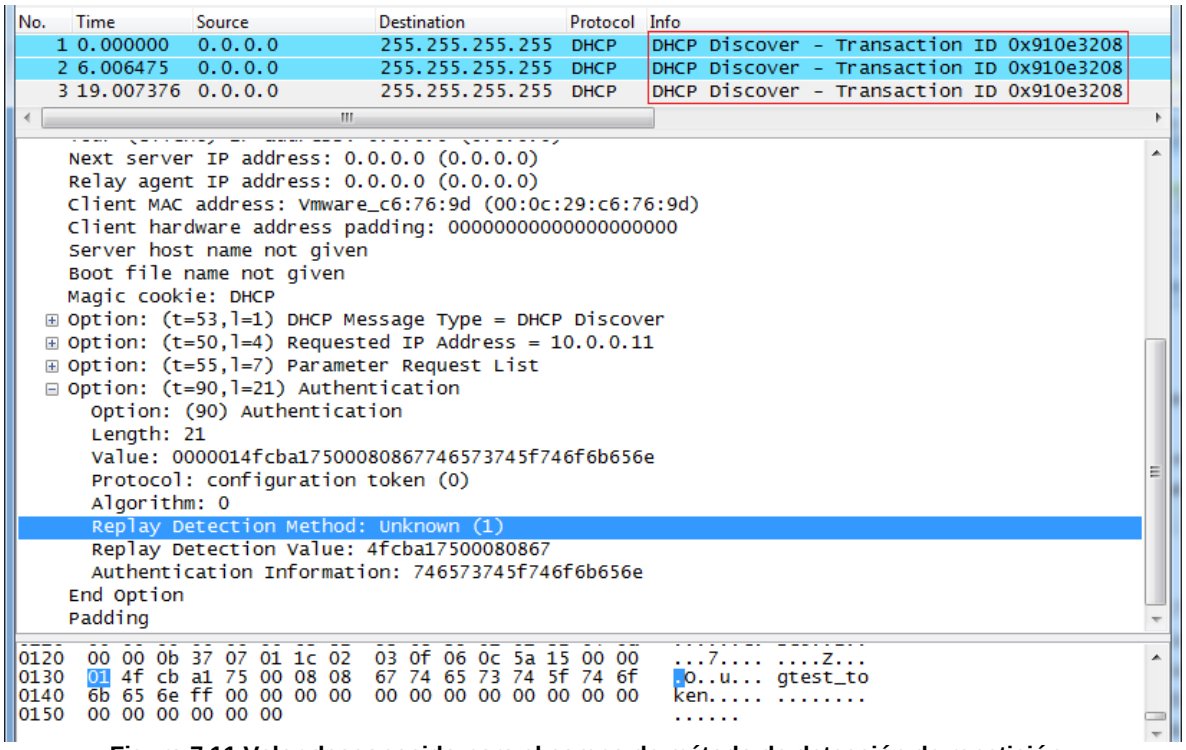

Figura 7.11 Valor desconocido para el campo de método de detección de repetición.

7.1.2.6. Dos o más opciones de autenticación presentes en el mensaje

En este escenario el cliente se modificó para enviar un mensaje DHCPDISCOVER que contenía dos opciones de autenticación idénticas. El servidor recibió el mensaje y encontró que había varias opciones de autenticación, por lo que descartó el mensaje. Las capturas que muestran el mensaje DHCP con dos opciones de autenticación se encuentran en la Figura 7.12 y Figura 7.13.

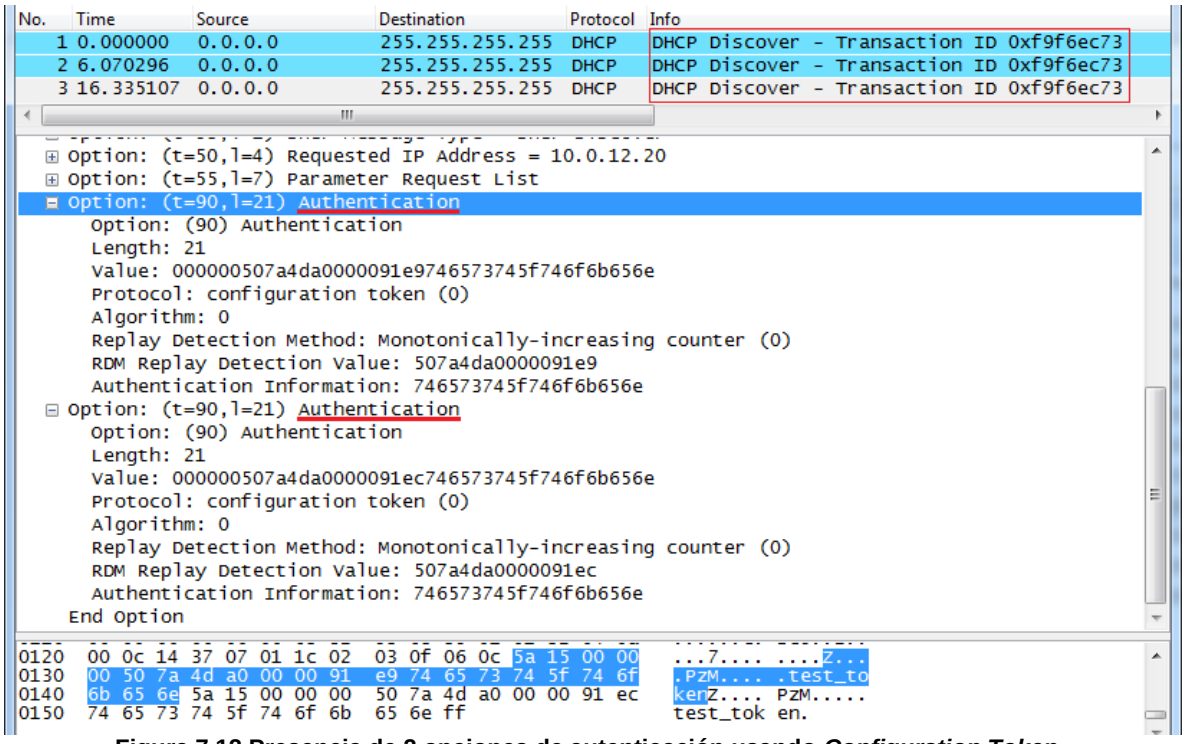

Figura 7.12 Presencia de 2 opciones de autenticación usando *Configuration Token*.

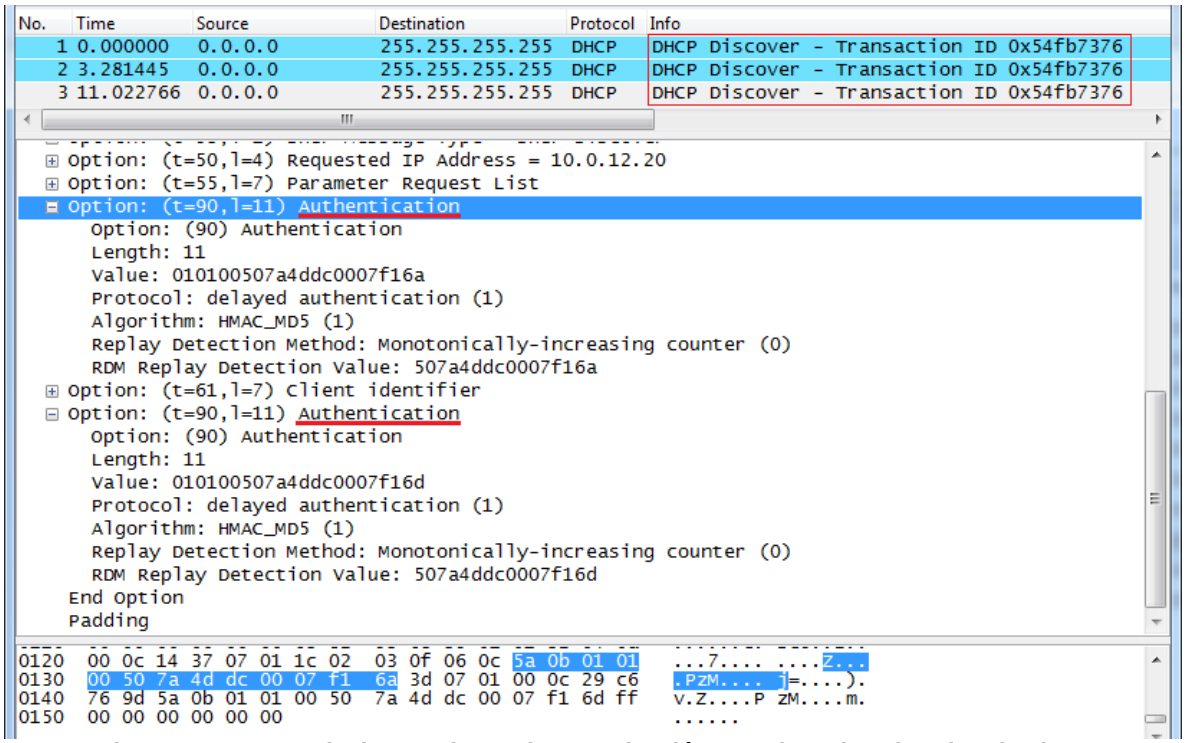

Figura 7.13 Presencia de 2 opciones de autenticación usando *Delayed Authentication*.

7.1.2.7. Agente de relevo entre un cliente y un servidor

En este escenario se ubicó al cliente y al servidor en dos sub redes distintas, pero una tercera máquina se configuró como agente de relevo ya que posee dos interfaces de red, cada una conectada a una de las subredes. El agente de relevo escuchaba y retransmitía peticiones DHCP hacia un servidor indicado por línea de comando. Se enviaron mensajes autenticados a través del agente de relevo y como resultado este modificó el valor del campo *giaddr* del mensaje DHCP, indicando que el mensaje pasó a través del agente de relevo señalado. Las capturas en la Figura 7.14 y la Figura 7.15 reflejan dicha acción.

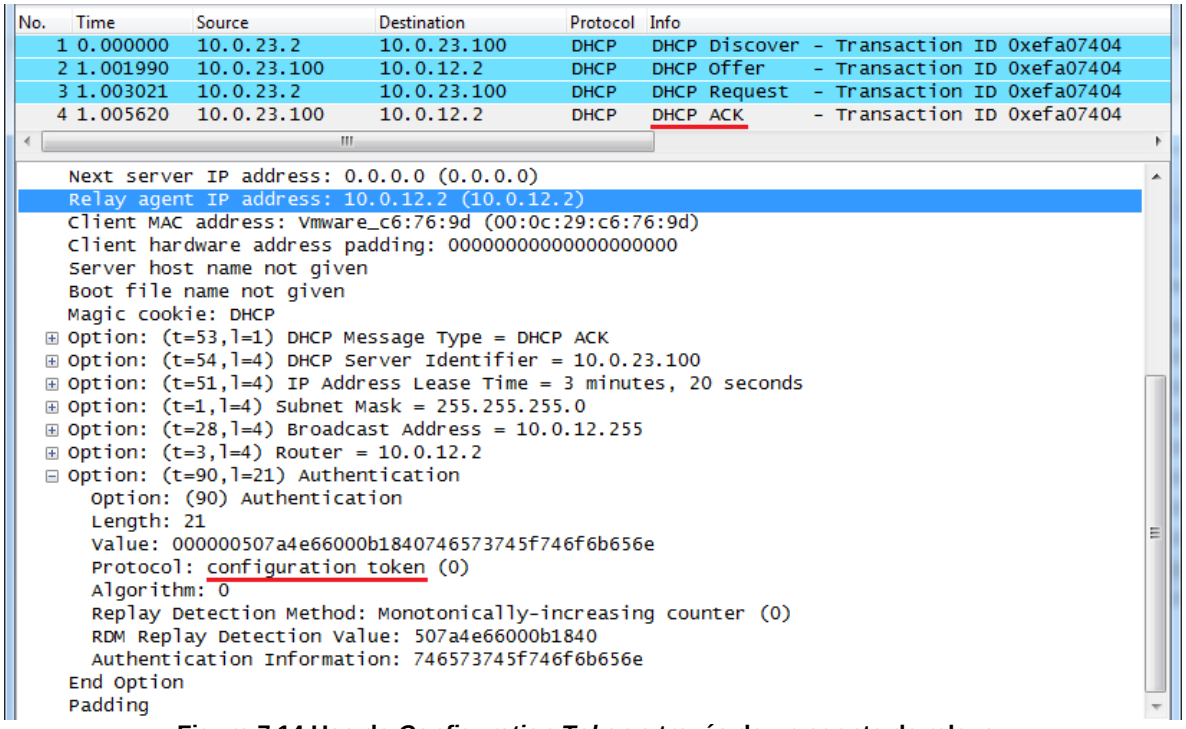

Figura 7.14 Uso de *Configuration Token* a través de un agente de relevo.

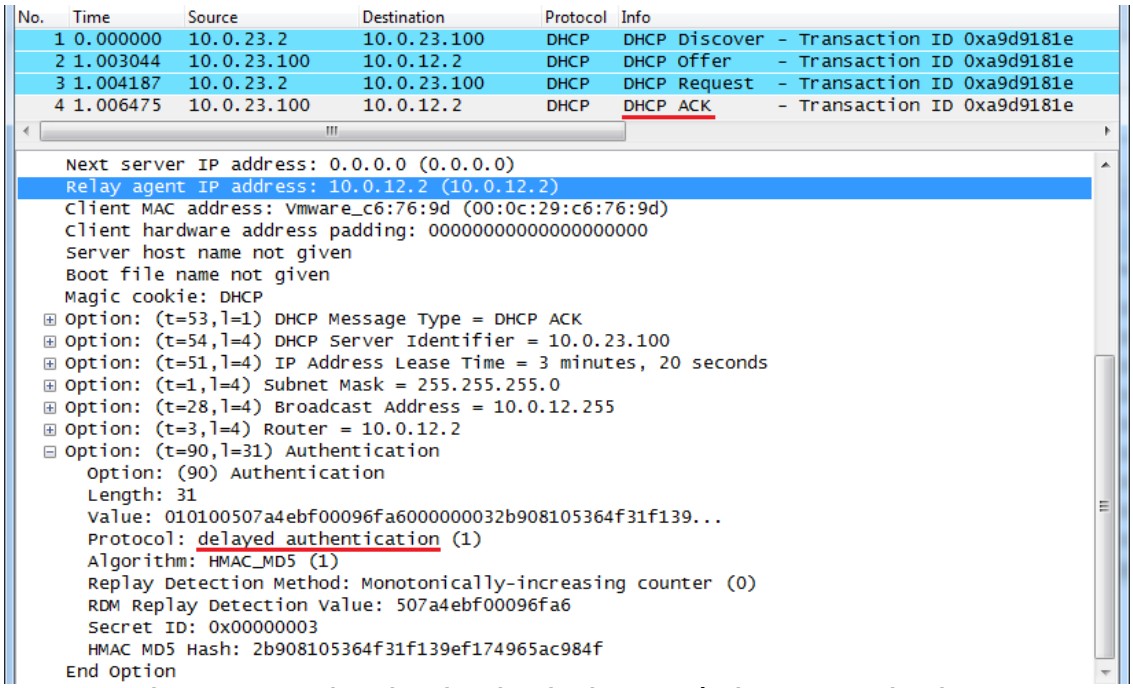

Figura 7.15 Uso de *Delayed Authentication* a través de un agente de relevo.

7.1.2.8. Ambiente variado

En este escenario se ubicó en una misma sub red a un gran número de clientes y servidores DHCP funcionando simultáneamente. Específicamente se contó con 10 servidores y 20 clientes cuyas configuraciones se pueden apreciar en la Tabla 7.1 y Tabla 7.2 respectivamente, para un total de 30 máquinas.

ID servidor	Autenticación	Dirección IP	Pool direcciones IP
Servidor 0	*Delayed Auth*, conjunto de llaves {K} y ID {A}	10.0.0.200	[10.0.0.1 - 10.0.0.9]
Servidor 1	*Delayed Auth*, conjunto de llaves {K} y ID {A}	10.0.0.201	[10.0.0.11 - 10.0.0.19]
Servidor 2	*Delayed Auth*, conjunto de llaves {K} y ID {A}	10.0.0.202	[10.0.0.21 - 10.0.0.29]
Servidor 3	*Delayed Auth*, conjunto de llaves {K} y ID {B}	10.0.0.203	[10.0.0.31 - 10.0.0.39]
Servidor 4	*Delayed Auth*, conjunto de llaves {K} y ID {B}	10.0.0.204	[10.0.0.41 - 10.0.0.49]
Servidor 5	*Delayed Auth*, conjunto de llaves {K} y ID {B}	10.0.0.205	[10.0.0.51 - 10.0.0.59]
Servidor 6	*Conf.Token*, Token "icarotoken"	10.0.0.206	[10.0.0.61 - 10.0.0.69]
Servidor 7	*Conf.Token*, Token "testtoken"	10.0.0.207	[10.0.0.71 - 10.0.0.79]
Servidor 8	No aplica	10.0.0.208	[10.0.0.81 - 10.0.0.89]
Servidor 9	No aplica	10.0.0.209	10.0.0.91 - 10.0.0.199]

Tabla 7.1 Configuración de servidores en ambiente variado.

ID Cliente	Autenticación
Cliente 0	*Delayed Authentication*, conjunto de llaves {K} con ID {A}
Cliente 1	*Delayed Authentication*, conjunto de llaves {K} con ID {A}
Cliente 2	*Delayed Authentication*, conjunto de llaves {K} con ID {A}
Cliente 3	*Delayed Authentication*, conjunto de llaves {K} con ID {A}
Cliente 4	*Delayed Authentication*, conjunto de llaves {K} con ID {A}
Cliente 5	*Delayed Authentication*, conjunto de llaves {K} con ID {A}
Cliente 6	*Delayed Authentication*, conjunto de llaves {K} con ID {B}
Cliente 7	*Delayed Authentication*, conjunto de llaves {K} con ID {B}
Cliente 8	*Delayed Authentication*, conjunto de llaves {K} con ID {B}
Cliente 9	*Delayed Authentication*, conjunto de llaves {K} con ID {B}
Cliente 10	*Delayed Authentication*, conjunto de llaves {K} con ID {B}
Cliente 11	*Delayed Authentication*, conjunto de llaves {K} con ID {B}
Cliente 12	*Configuration Token* con token "icarotoken"
Cliente 13	*Configuration Token* con token "icarotoken"
Cliente 14	*Configuration Token* con token "icarotoken"
Cliente 15	*Configuration Token* con token "testtoken"
Cliente 16	*Configuration Token* con token "testtoken"
Cliente 17	*Configuration Token* con token "testtoken"
Cliente 18	No aplica
Cliente 19	No aplica

Tabla 7.2 Configuración de clientes en ambiente variado.

Dependiendo de la configuración que poseían, un cliente sólo podía obtener los parámetros de configuración de red de aquellos servidores con los que se podía autenticar exitosamente y tanto el cliente como el servidor descartaban los mensajes que no estaban debidamente autenticados. Dicho comportamiento se puede ver reflejado en la Figura 7.16 y Figura 7.17 donde se pueden apreciar los logs de un cliente y de un servidor en particular.

Del lado del servidor se puede ver que las primeras dos líneas corresponden a mensajes que fueron descartados por no poseer una opción de autenticación. Las siguientes cuatro líneas corresponden a la autenticación y asignación de IP exitosa de un cliente. Las siguientes dos líneas corresponden al descarte de un mensaje por el uso de una llave desconocida y la última línea corresponde al descarte de un mensaje por usar un método de autenticación distinto al utilizado por el servidor.

Servidor DHCP

```
No Authentication option from 00:23:12:e7:37:89, packet discarded.
No Authentication option from a0:0b:ba:db:44:d3, packet discarded.
DHCPDISCOVER from 00:0c:29:4d:6c:1f via eth0
DHCPOFFER on 10.0.0.13 to 00:0c:29:4d:6c:1f via eth0
DHCPREQUEST for 10.0.0.13 (10.0.0.201) from 00:0c:29:4d:6c:1f via eth0
DHCPACK on 10.0.0.13 to 00:0c:29:4d:6c:1f via eth0
Key ID unknown.
HMAC not valid from 00:0c:29:cd:62:ee, packet discarded.
Wrong PROTOCOL in use from 00:0c:29:a9:9c:0f, packet discarded.
```

Figura 7.16 Extractos del log del servidor DHCP.

Del lado del cliente se puede ver que la primera línea corresponde al envío del mensaje DHCPDISCOVER en búsqueda de algún servidor DHCP. Las siguientes cuatro líneas corresponden a dos mensajes descartados porque los servidores usaron una llave desconocida. Finalmente las últimas cinco líneas corresponden a la recepción de una oferta por parte de un servidor autenticado, la petición y asignación de la dirección IP ofertada por el tiempo indicado.

Cliente DHCP

```
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 3
Key ID unknown.
HMAC not valid from 10.0.0.205, packet discarded.
Key ID unknown.
HMAC not valid from 10.0.0.203, packet discarded.
No Authentication option from 10.0.0.209, packet discarded.
DHCPOFFER from 10.0.0.201
DHCPREQUEST on eth0 to 255.255.255.255 port 67
DHCPACK from 10.0.0.201
bound to 10.0.0.13 -- renewal in 54 seconds.
```

Figura 7.17 Extractos del log del cliente DHCP.

Se puede observar que hay varios mensajes que son descartados en ambas partes por motivos que son indicados por pantalla y registrados en los logs de ejecución. Cabe destacar que el caso de éxito en la autenticación corresponde justamente a una petición realizada por el cliente seleccionado hacia el servidor seleccionado para este ejemplo.

7.1.3. Análisis de los Resultados

Luego de haber culminado las pruebas, se pudo corroborar que la solución desarrollada en este trabajo cumple con las funcionalidades necesarias para llevar a cabo correctamente la autenticación en mensajes DHCP tal como lo describe el RFC 3118, por lo tanto cumple con los objetivos planteados.

7.2. Pruebas de estrés

En estas pruebas se evaluó la capacidad que posee un servidor DHCP de atender correctamente las peticiones de varios clientes en un ambiente que constantemente tiene un alto número de transacciones por período de tiempo.

7.2.1. Preparación de las Pruebas

Para la realización de las pruebas de estrés se configuró un escenario donde un servidor DHCP estuvo sometido a responder al mayor número posible de peticiones en un corto período de tiempo. Como lo muestra la Figura 7.18, el escenario consistió en hacer que un servidor atienda las peticiones de 30 clientes simultáneamente. Además, para incrementar el número de transacciones se configuró en el servidor que la duración de las direcciones IP otorgadas tuviesen una duración de uno a dos segundos, lo que en promedio resultó en la renovación de 15 a 30 direcciones IP por segundo.

Ese número de transacciones se mantuvo más o menos constante por una duración de 80 minutos y a lo largo de la prueba se estuvo monitoreando el consumo de recursos en el lado del servidor, específicamente los niveles de uso del CPU y memoria RAM utilizada. Transcurrido dicho tiempo, se decidió detener la prueba al no observar variaciones. Esta prueba se realizó usando el código original del ISC DHCP, luego utilizando el método de autenticación *Configuration Token* y por último el método de *Delayed Authentication* para determinar si el uso de alguno de los métodos de autenticación reducía en forma significativa la capacidad del servidor de atender a un determinado número de clientes.

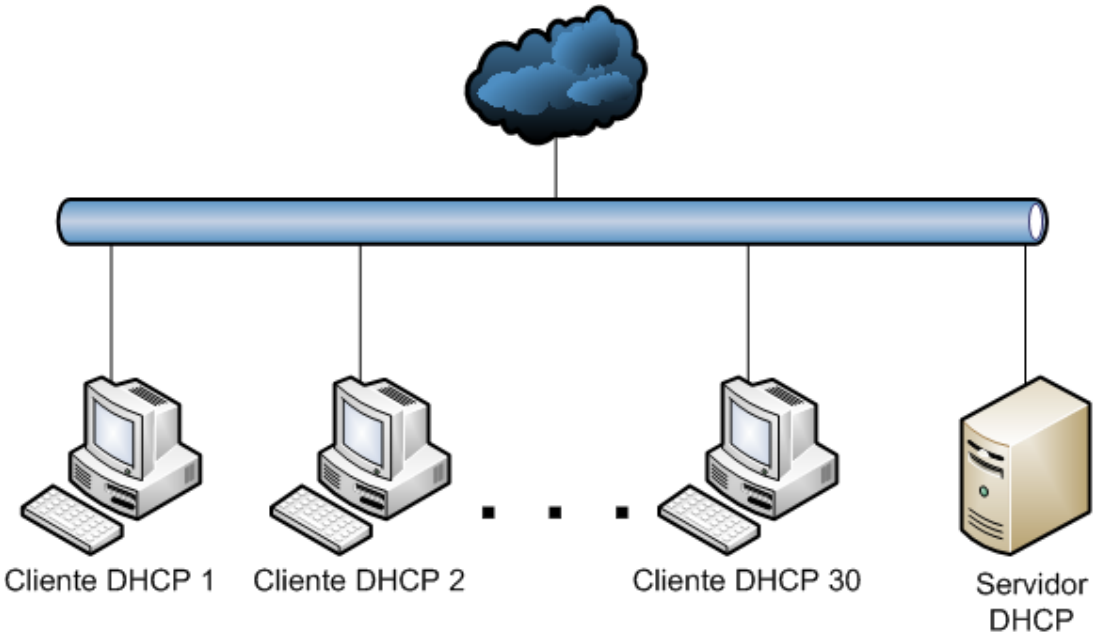

Figura 7.18 Topología de la red para pruebas de estrés.

7.2.2. Ejecución de las Pruebas

Las pruebas se realizaron bajo máquinas virtuales de Debian Squeeze 6.0.1 con entorno gráfico utilizando VMware Workstation 6. Las especificaciones de hardware de las máquinas huésped y de las máquinas virtuales son las siguientes:

Máquina huésped:

- CPU: Intel Core 2 Duo E6750 2.66 GHz.
- RAM: 2 GB DDR2 800 MHz Dual Channel.

Máquina virtual:

- CPU: Intel Core 2 Duo E6750 2.66 GHz (1 núcleo, 1 hilo).
- RAM: 256 MB DDR2 800 MHz Dual Channel

La medición del consumo de recursos fue realizado a través de la herramienta *top* [22] del sistema operativo Linux, observando únicamente al proceso *dhcpd*. Por cada intervalo de 10 minutos se registró el valor promedio del porcentaje de uso de CPU y memoria que consumía dicho proceso en el sistema. No se consideró analizar el impacto causado en la red debido a que en las pruebas realizadas se mantuvo relativamente constante la cantidad de data enviada. Aún agregando la opción de autenticación apenas se llegaba a sobrepasar el tamaño mínimo del mensaje DHCP (300 bytes).

Como se puede apreciar en la Tabla 7.3, Tabla 7.4 y Tabla 7.5, los valores recolectados corresponden a la ejecución del servidor DHCP sin método de autenticación, con el método *Configuration Token* y con el método *Delayed Authentication* respectivamente. A modo de resumen se puede observar los resultados obtenidos de los tres casos en la Figura 7.19.

Tiempo (m)	0	10	20	30	40	50	60	70	80
% mem (promedio)	1.7	1.7	1.7	1.7	1.7	1.7	1.7	1.7	1.7
% CPU (promedio)	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3

Tabla 7.3 Consumo de recursos utilizando ISC DHCP regular.

Tiempo (m)	0	10	20	30	40	50	60	70	80
% mem (promedio)	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8
% CPU (promedio)	1.7	1.7	1.7	1.7	1.7	1.7	1.7	1.7	1.7

Tabla 7.4 Consumo de recursos utilizando *Configuration Token*.

Tiempo (m)	0	10	20	30	40	50	60	70	80
% mem (promedio)	1.9	1.9	1.9	1.9	1.9	1.9	1.9	1.9	1.9
% CPU (promedio)	2	2	2	2	2	2	2	2	2

Tabla 7.5 Consumo de recursos utilizando *Delayed Authentication.*

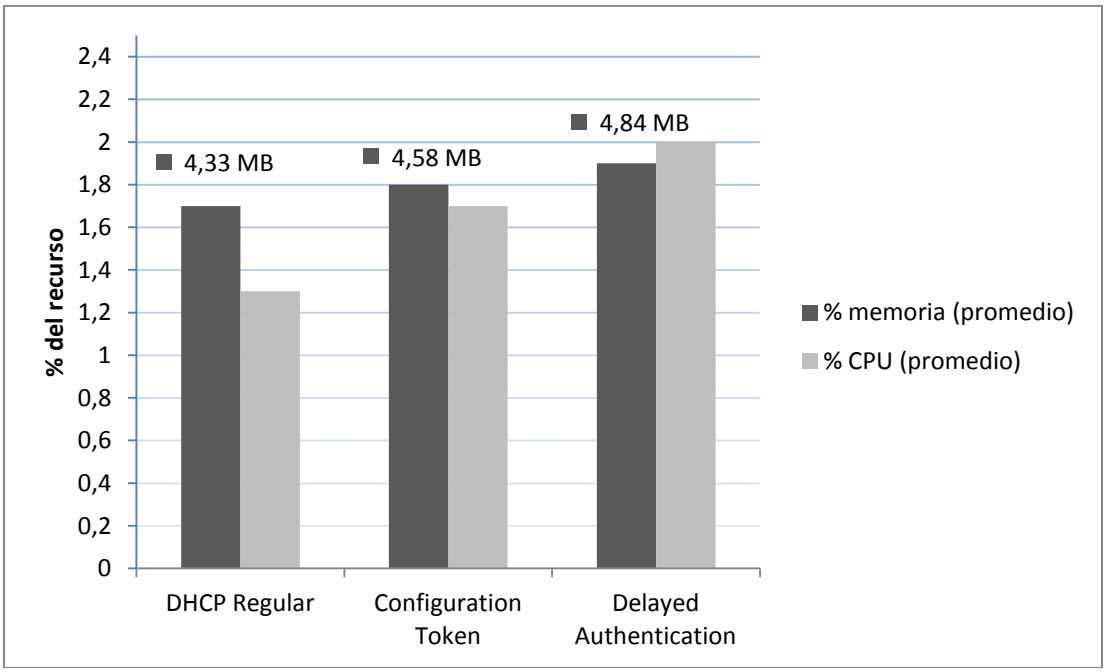

Figura 7.19 Porcentaje de consumo de recursos.

7.2.3. Análisis de los Resultados

Como se puede apreciar a partir de los datos obtenidos, el uso de cualquiera de los métodos de autenticación genera un aumento de la utilización del CPU con respecto a la ejecución regular, el cual es aproximadamente un 30% mayor al usar *Authentication Token* y un 50% mayor al usar *Delayed Authentication*. Aún así, el aumento de utilización del CPU es muy poco significativo, menos del 1%, por lo que se pudiese pensar que la utilización de alguno de los métodos de autenticación no representa mayor impedimento para que un sistema DHCP funcione correctamente bajo períodos de alta carga. También cabe destacar que las características de hardware que poseía el servidor eran algo modestos comparado con lo que puede obtenerse hoy en día, por lo que el impacto al usar la autenticación en otros equipos pudiese ser incluso menor.

De igual forma, sería interesante determinar con exactitud a partir de qué punto el servidor empezaría a presentar fallas motivado a excesivo tráfico o procesamiento con cada uno de los métodos de autenticación, sin embargo por limitaciones de infraestructura del lugar donde se realizaron las pruebas no se pudo lograr que eso ocurriese.

En lo que concierne a memoria consumida, se nota un ligero incremento que muy probablemente se deba a que toda información de las asociaciones entre llaves y clientes, valores de detección de repetición, etc., se cargan en memoria y se mantienen en una lista dinámica. Se espera que mientras mayor sea la cantidad de clientes atendidos por un servidor naturalmente incremente la cantidad de memoria consumida, pero este incremento no es lo suficientemente significativo para generar algún impacto en el equipo.

7.3. Pruebas de rendimiento

En estas pruebas se evaluó el costo adicional en tiempo que conlleva la utilización de los métodos de autenticación para la obtención de los parámetros de configuración en un cliente.

7.3.1. Preparación de las Pruebas

Para la realización de las pruebas de rendimiento se preparó un escenario sencillo que se ilustra en la Figura 7.20 donde un cliente y un servidor DHCP se encuentran en la misma sub red, con la finalidad de medir con la mayor certeza posible el tiempo que tarda un cliente en adquirir una dirección IP usando los varios métodos de autenticación para compararlos con el tiempo regular.

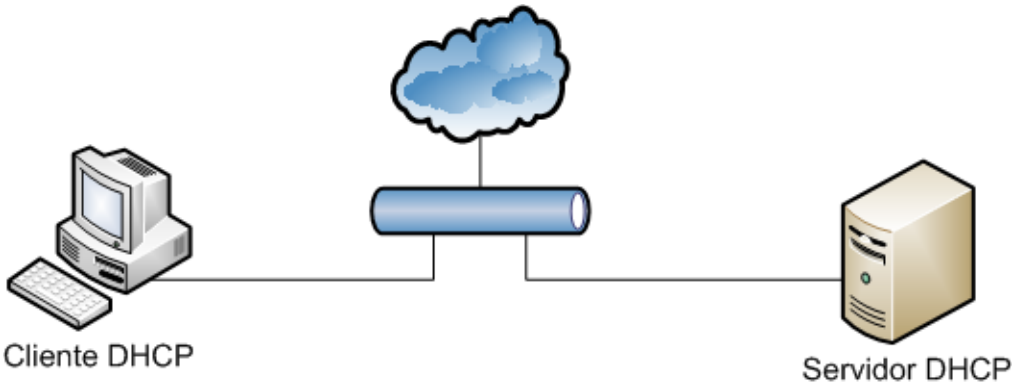

Figura 7.20 Topología de red para pruebas de rendimiento.

Para ello se tomó el tiempo transcurrido del lado del cliente desde que inició su demonio hasta que recibió un mensaje DHCPACK válido por parte del servidor. El tiempo se midió con ayuda de la función "gettimeofday()" para obtener la hora actual y se registró la hora a la que se inició el demonio, la hora a la que se obtuvo satisfactoriamente la dirección IP y la diferencia de estos dos tiempo dio como resultado el tiempo que fue necesario para obtener la dirección IP. Este experimento se repitió varias veces y se hicieron ciertos cálculos para obtener un valor promedio. Además, para estar seguros de que los resultados son lo más precisos posibles se realizaron las pruebas sobre una instalación nativa de Linux (Ubuntu 9.04) en lugar de una máquina virtual como fue mencionado en la realización de las pruebas anteriores.

7.3.2. Ejecución de las Pruebas

Para la realización de las pruebas se modificó únicamente el código fuente del cliente para obtener la hora actual en puntos claves, calcular la diferencia e ir almacenando dichos valores en un archivo separado por comas para su posterior análisis. Además se preparó un script para automatizar el proceso de la

realización de las pruebas de la siguiente forma: se inicia el demonio DHCP en el cliente con los parámetros necesarios, luego de obtener una dirección IP se esperan tres segundos, se procede a liberar la dirección IP asignada y se vuelve a esperar tres segundos. Todo el proceso mencionado anteriormente se repite doscientas cincuenta veces para cada uno de los casos, que son la ejecución regular del cliente DHCP, la ejecución usando *Configuration Token* y la ejecución usando *Delayed Authentication*.

Luego se procedió a calcular algunos valores estadísticos como el tiempo promedio, mínimo, máximo, entre otros, a modo de representar los valores en un histograma donde se viera reflejado el comportamiento de la aplicación durante las pruebas, las cuales corresponden a la Figura 7.21, Figura 7.22 y Figura 7.23. Finalmente la Figura 7.24 muestra una gráfica del estilo de cajas ("box plot"), indicando dentro de los rectángulos el rango de valores que con mayor probabilidad tome la duración del proceso de adquisición de una dirección IP para los escenarios evaluados.

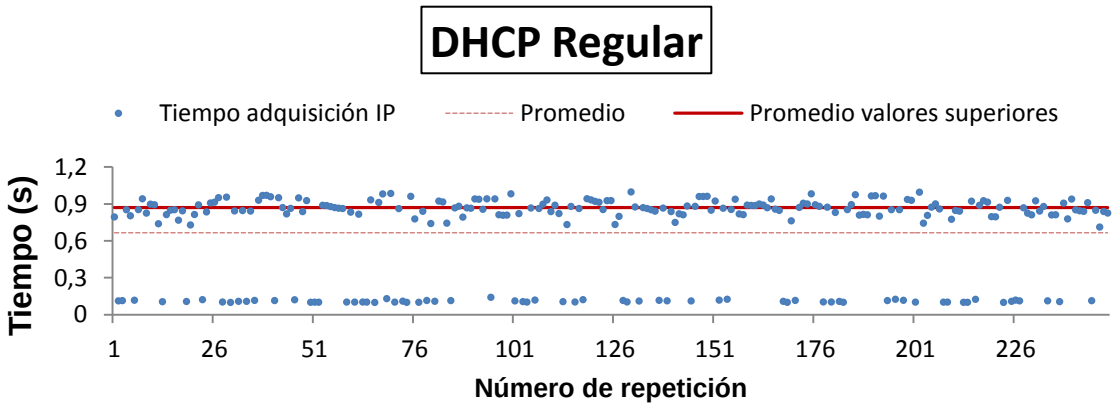

Figura 7.21 Duración del proceso para obtener una dirección IP (sin autenticación).

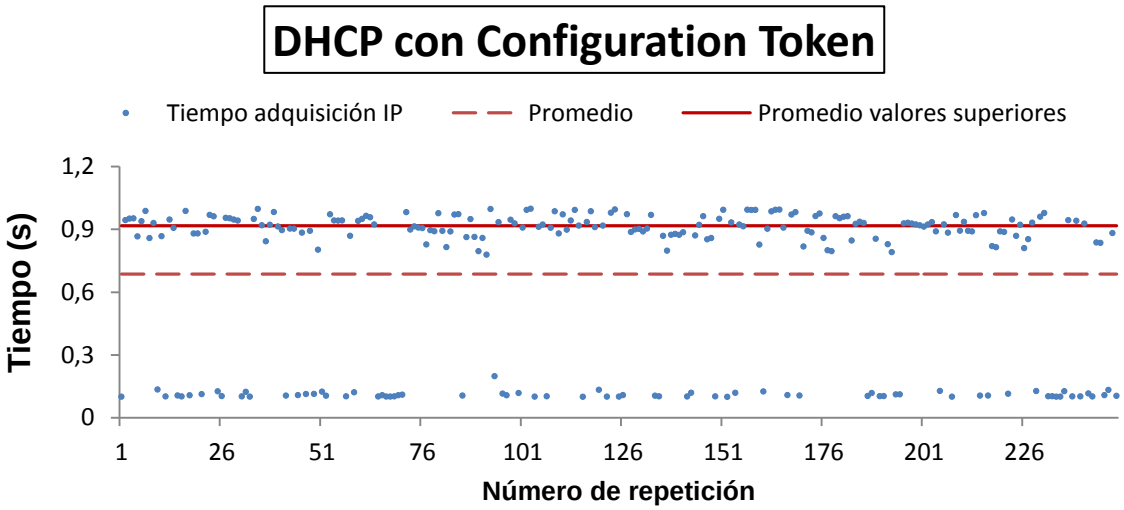

Figura 7.22 Duración del proceso para obtener una dirección IP (*Configuration Token*).

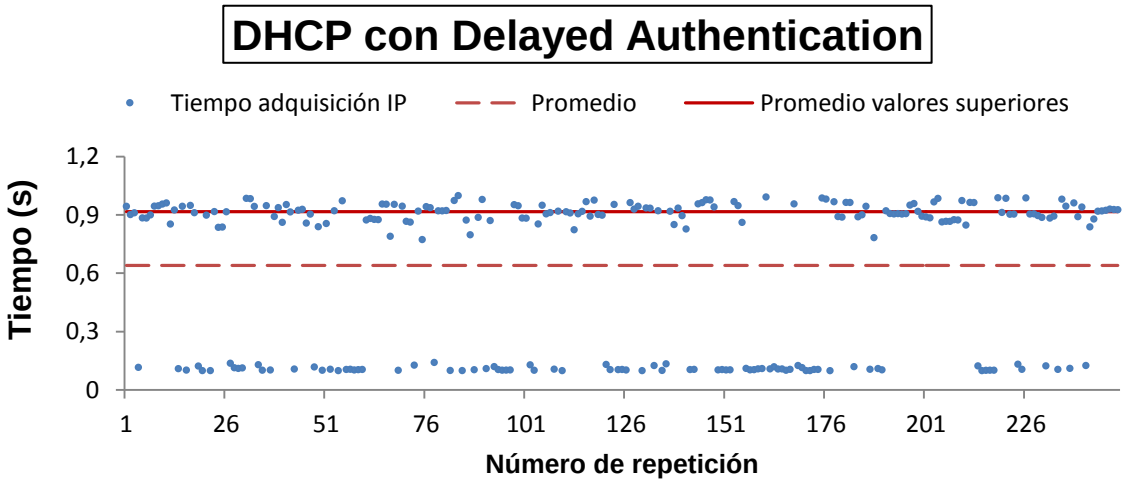

Figura 7.23 Duración del proceso para obtener una dirección IP (*Delayed Authentication*).

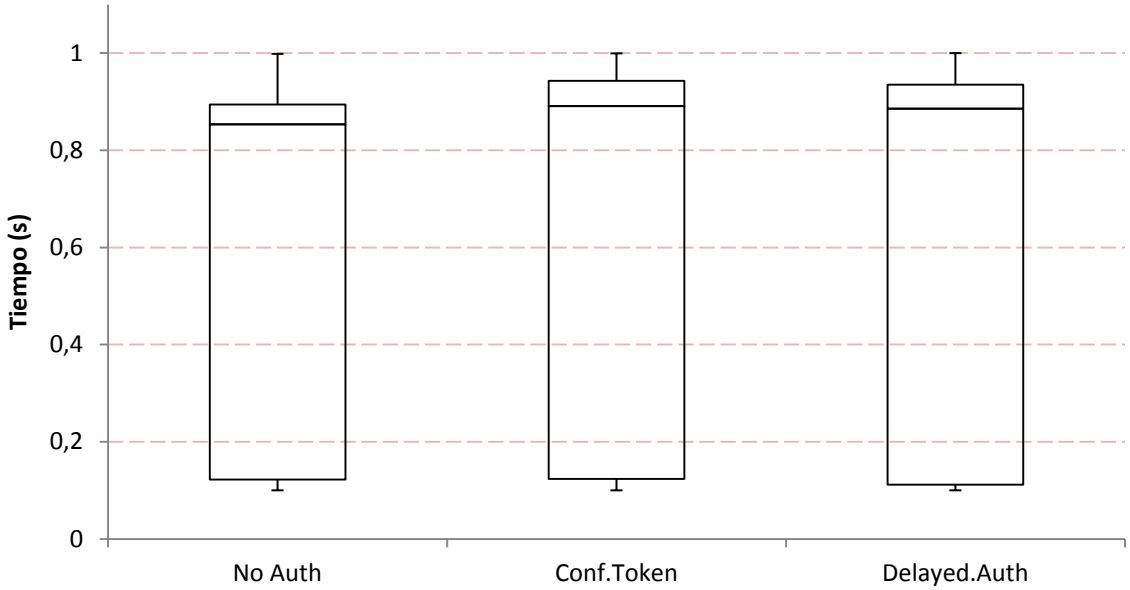

Figura 7.24 Diagrama de cajas de tiempo para obtener una dirección IP.

7.3.3. Análisis de los Resultados

Como se puede apreciar a partir de los resultados obtenidos, el proceso de adquisición de una dirección IP se puede llevar a cabo muy rápidamente (alrededor de 0.1 segundos) o en un tiempo un poco más razonable (alrededor de 0.9 segundos), aunque hay una mayor tendencia a caer en la segunda opción. Por ello se comparó los promedios de los tiempos correspondientes a la segunda opción y se observó que en los tres casos la cantidad de tiempo requerida para obtener una dirección IP es muy similar, con apenas un incremento de 0.045

segundos por usar cualquiera de los 2 métodos de autenticación para DHCP. Se presume que el motivo por el cual el servidor a veces asignaba muy rápidamente una dirección IP a un cliente era porque el servidor aún poseía alguna información remanente que relacionaba dicha dirección IP al cliente en cuestión, pero esto no se pudo determinar con exactitud.

Teniendo todo esto en cuenta, se puede afirmar que el uso de alguno de los métodos de autenticación genera una carga adicional en la ejecución regular de DHCP, pero la misma no es lo suficiente significativa como para afectar el rendimiento de DHCP.

8. Conclusiones

El protocolo DHCP es ampliamente utilizado para proveer una forma sencilla y automatizada de configurar apropiadamente una interfaz de red de algún dispositivo. Esto hace que sea una herramienta casi necesaria en muchos casos, ya sea para facilitar labores de administración en ambientes con gran cantidad de computadores o para cubrir la necesidad de usuarios con pocos conocimientos en el área de obtener una configuración que le permita acceder a Internet. Esta facilidad también se puede convertir en una desventaja cuando existen personas malintencionadas que quieran acceder a una red ya que DHCP no distingue usuarios, abriéndoles el paso para realizar cualquier cantidad de ataques a otros usuarios o sistemas. Por lo tanto, la implementación de un mecanismo que permita configurar sólo aquellos dispositivos autenticados es un método viable para mitigar dicho problema.

Este trabajo se enfocó en el análisis de una implementación existente de DHCP, la realización de modificaciones necesarias y pruebas para elaborar una solución final con soporte para mensajes de autenticación en DHCP, en este caso para dispositivos sobra funcionando bajo el sistema operativo Linux.

Los resultados obtenidos durante las pruebas demuestran que no sólo es una solución sencilla y funcional, sino que además no sobrecargan en mayor forma a las implementaciones ya existentes, por lo que se evidencia el logro de los objetivos que incentivaron el desarrollo de este trabajo.

En conclusión, la implementación de un mecanismo de seguridad que restrinja a quién se le otorga una dirección IP se puede llevar a cabo sin incluir elementos o equipos adicionales que pudiesen añadir complejidad a una red ya existente.

8.1. Contribuciones

Este trabajo tiene como principal aporte el proveer una solución real y factible para la implementación de mecanismos de autenticación para la obtención de una dirección IP en dispositivos funcionando bajo la plataforma Linux. También es un valioso aporte para la comunidad de administradores de sistemas ya que provee una alternativa segura y sencilla para la asignación de direcciones IP sin el problema de la adquisición de equipos especiales o la sobrecarga en la red que generan otros mecanismos existentes.

8.2. Limitaciones

Durante el desarrollo del presente trabajo se encontraron las siguientes limitaciones para la realización del mismo:

- Existe un número reducido de implementaciones código abierto de DHCP, especialmente en lo que respecta a los clientes, por lo que las alternativas para escoger algún lenguaje de programación o plataforma sobre la cual trabajar se ven afectadas. De hecho la decisión de trabajar sobre la implementación del ISC DHCP fue básicamente por no haber encontrado otra implementación que fuese apropiada para el desarrollo de la solución final.

- La etapa de análisis y comprensión del código fuente de la implementación escogida consumió una excesiva cantidad de tiempo debido a la gran complejidad del código correspondiente a los elementos que componen al ISC DHCP. Unido a esto también está el hecho que no se cuenta con mayor documentación que ayude a tener un mejor entendimiento sobre la estructura o funcionamiento para poder modificar más fácilmente el código. Además dependiendo de la naturaleza de los problemas encontrados, la comunidad de usuarios y la lista de correos pudiesen no ser de gran ayuda.

- La utilización del kit de firmas digitales (*Digital Signature Toolkit*) encontrado en el código fuente del ISC DHCP representó un problema, ya que dicho kit fue incluido para cumplir otros propósitos, por lo que el cliente y el servidor no tenían acceso a las funcionalidades de cifrado provistas en el kit. Para solventar este inconveniente se tuvo que recopilar todo el código fuente correspondiente al kit, renombrar varias de las funciones y estructuras que lo componían e incluirlo en los códigos del cliente y servidor ya que de lo contrario generaban conflictos al momento de compilación. Sería deseable que el grupo de trabajo del ISC hiciera las modificaciones pertinentes para habilitar el uso del kit y así evitar tener que hacer copias del código en los directorios pertinentes.

- La falta de bibliografía o trabajos relacionados al método de autenticación indicado en el RFC 3118 e implementado en este trabajo afectó en general varios aspectos del desarrollo de la solución. Ya que el tema se encontraba prácticamente en estado de abandono por alrededor de once años, no se contaba con aportes u opiniones de otros investigadores que pudiesen haber sido de ayuda para la elaboración de la solución. Además, no se tenía otra herramienta con la cual comparar la solución desarrollada para comprobar aspectos de interoperabilidad con distintas implementaciones.

8.3. Trabajos Futuros

Para realizar la extensión del presente trabajo o mejorar el mismo se puede trabajar en los siguientes aspectos:

- Añadir la utilización de opciones de autenticación para DHCPv6, ya que a pesar de formar parte del estándar descrito en el RFC 3315 en el año 2003 el soporte de los mismos todavía no ha sido incluido en el ISC DHCP.

- Se debe añadir en el cliente la posibilidad de aceptar mensajes de ofertas que no estén debidamente autenticados, a pesar de los riesgos que esto pueda traer.

- Añadir o diseñar otros métodos de autenticación adicionalmente a los que se encuentran descritos en este trabajo, ya que el método de *Configuration Token* es fácilmente violentado y *Delayed Authentication*, por la forma en la que está concebido, es susceptible a ciertos ataques de denegación de servicio, tanto para el cliente como para el servidor.

- Incluir otros algoritmos de hashing que sean más seguros que MD5, ya que hoy en día no se recomienda mucho su uso para funciones criptográficas. Cabe destacar que para asignar y utilizar un número de código a algún algoritmo (o para cualquier campo que contenga algún código identificador en general) se debe hacer una solicitud previamente al Internet Assigned Numbers Authority[3], que es el ente encargado de administrar la asignación de valores de códigos utilizados en muchos de los protocolos usados actualmente.

- Modificar el código fuente para realizar el manejo de los mensajes de autenticación a través de procedimientos más acorde a la forma en que se encuentra implementado el ISC DHCP, lo que pudiera resultar en un código más elegante y probablemente más eficiente. Sin embargo, dicha tarea dependerá del nivel de entendimiento que se tenga acerca del código fuente y de la ayuda que el ISC directamente pueda proveer para ello.

[3] http://www.iana.org/

9. Referencias

[1] R. Droms. Dynamic Host Configuration Protocol. RFC 2131. Marzo, 1997.

[2] D. Plummer. An Ethernet Address Resolution Protocol. RFC 826. Noviembre, 1982.

[3] J. Reynolds. BOOTP Vendor Information Extensions. RFC 1497. Agosto, 1993.

[4] S. Alexander, R. Droms. DHCP Options and BOOTP Vendor Extensions. RFC 2132. Marzo 1997.

[5] R. Droms, W. Arbaugh. Authentication for DHCP Messages. RFC 3118. Junio, 2001.

[6] H. Krawczyk, M. Bellare, R. Canetti. HMAC: Keyed-Hashing for Message Authentication. RFC 2104. Febrero, 1997.

[7] **Internet Systems Consortium.** "Software - (DHCP)". http://www.isc.org/software/dhcp. Octubre 2012.

[8] **GCC, the GNU Compiler Collection.** "GCC". http://gcc.gnu.org/. Octubre 2012.

[9] **GNU Project.** "GNU Make". http://www.gnu.org/software/make/. Octubre 2012.

[10] **su(1) - Linux man page.** "Su". http://linux.die.net/man/1/su. Octubre 2012.

[11] **VMware Workstation: Run Multiple OS, Linux, Windows 8 & More.** "VMware Workstation". www.vmware.com/products/workstation/. Octubre 2012.

[12] **Wireshark Foundation.** "Wireshark - Go Deep". http://www.wireshark.org/. Octubre 2012.

[13] **D. Rubel.** "Propuesta para el Diseño e Implementación de un Servidor DHCP con Autenticación de Mensajes". Universidad Central de Venezuela. Junio, 2011.

[14] R. Droms, J. Bound, B. Volz, T. Lemon, C. Perkins, M. Carney. Dynamic Host Configuration Protocol for IPv6 (DHCPv6). RFC 3315.Julio 2003.

[15] B. Croft, J. Gilmore, Bootstrap Protocol (BOOTP). RFC 951. Septiembre 1985.

[16] **Debian.** “Debian – The Universal Operating System”. http://www.debian.org/. Octubre 2012.

[17] **Ubuntu.** “Ubuntu for you”. http://www.ubuntu.com/ubuntu/. Octubre 2012.

[18] **BusyBox.** http://www.busybox.net/. Octubre 2012.

[19] **LoosyDHCP.** http://sourceforge.net/projects/loosydhcp/. Octubre 2012.

[20] **dhcp4java.** http://sourceforge.net/projects/dhcp4java/. Octubre 2012.

[21] **JDHCPD.** http://sourceforge.net/projects/jdhcpd/. Octubre 2012.

[22] **top(1) - Linux man page.** "top". http://linux.die.net/man/1/top/. Octubre 2012.

10. Anexos

Anexo Nº A Ejecución del DHCP con Autenticación

El proceso para ejecutar el demonio DHCP usando algún método de autenticación varía un poco dependiendo si se ejecutará el cliente o el servidor.

Inicio del demonio del cliente DHCP

Para poder ejecutar el cliente DHCP, la interfaz de red por la que se va a solicitar una dirección IP debe estar levantada pero sin aplicarle ninguna configuración. Esto se puede realizar colocando el texto **"iface ethN inet manual"** en el archivo **"/etc/network/interfaces"** (recomendado), o a través del comando **"ifconfig ethN up"**, pero habría que tener permisos de superusuario para ello y se tendría que ejecutar el comando cada vez que se encienda el computador. **"N"** corresponde al identificador de la interfaz de red a usar.

Además debe crearse el directorio y los siguientes archivos dentro del mismo, tal como se ilustra en la Figura 10.1. Para esto se debe poseer privilegios de superusuario.

```
# mkdir /var/db
# touch /var/db/client_replays.db

/* Solo en Delayed Authentication son necesarios estos archivos*/
# touch /var/db/client_keys.db
# touch /var/db/client_bindings.db
# touch /var/db/client_last_binding.db
```

Figura 10.1 Comandos para creación de directorios y archivos necesarios en el cliente.

Los archivos creados contendrán eventualmente la siguiente información:

- **client_replays.db:** Almacenará las direcciones IP de los servidores con los que se haya establecido un intercambio de mensajes y sus respectivos valores de detección de repetición. Comenzará como un archivo vacío.
- **client_keys.db:** Almacenará las llaves que fueron predistribuidas y sus respectivos identificadores.
- **client_bindings.db:** Almacenará el identificador de la llave que se asignó para intercambiar mensajes con un servidor en particular. Comenzará como un archivo vacío.
- **client_last_binding.db:** Almacenará la última asociación ID_Llave/Servidor, para ser usada en casos que no se pueda determinar fácilmente el servidor que le otorgó una dirección IP al cliente. Comenzará como un archivo vacío.

Usando Configuration Token en el cliente

Para iniciar al cliente usando el método de *Configuration Token*, se ejecuta el siguiente comando que se ilustra en la Figura 10.2.

```
# dhclient -ct Secreto ethN
```

Figura 10.2 Comando para iniciar el cliente usando *Configuration Token* con clave "Secreto".

- **Usando Delayed Authentication**

Para iniciar al cliente usando el método de *Delayed Authentication*, el archivo **"/var/db/client_keys.db"** debe tener previamente las llaves con sus identificadores respectivos. Los identificadores pueden tomar valores entre 0 y 4.294.967.295 (máximo valor representable con 32 bits) y las llaves consistirán en una cadena de texto con una longitud máxima de 64 caracteres sin espacios. Un archivo de ejemplo y su formato puede verse en la Figura 10.3. Finalmente se ejecuta el comando como se ilustra en la Figura 10.4. Por ahora solo se soporta el algoritmo "md5".

```
0 test1
765 llave
1 48421fe
4294967295 Una_Cadena_De_Maximo_64_Caracteres
```

Figura 10.3 Formato del archivo de identificadores y llaves para *Delayed Authentication*.

```
# dhclient -da md5 ethN
```

Figura 10.4 Comando para iniciar el cliente usando *Delayed Authentication* con algoritmo MD5.

Inicio del demonio del servidor DHCP

Para ejecutar el servidor DHCP, se debe tener un archivo de configuración (dhcpd.conf) válido en el directorio requerido por el servidor. Usualmente el directorio corresponde al directorio **"ETCDIR"** de acuerdo a la distribución de Linux que se esté usando. Para el caso de Debian o Ubuntu, el archivo de configuración estará ubicado en **"/etc/dhcpd.conf"**. Se puede tomar como referencia un archivo de ejemplo ubicado dentro del directorio del código fuente del ISC DHCP, bajo el directorio **"server"**.

Además debe crearse el directorio y los siguientes archivos dentro del mismo, tal como se ilustra en la Figura 10.5. Para esto se debe poseer privilegios de superusuario.

```
# mkdir /var/db
# touch dhcpd.leases
# touch /var/db/server_replays.db

/* Para Delayed Authentication son necesarios estos archivos*/
# touch /var/db/server_keys.db
# touch /var/db/server_bindings.db
```

Figura 10.5 Comandos para la creación de directorios y archivos necesarios en el servidor.

Los archivos creados contendrán eventualmente la siguiente información:

- **dhcpd.leases:** Contendrá la información de las concesiones de direcciones IP y toda la información de configuración otorgada a cada uno de los clientes atendidos. Es manipulado por el código original del ISC DHCP.
- **server_replays.db:** Almacenará las direcciones MAC de los clientes con los que se haya establecido un intercambio de mensajes y sus respectivos valores de detección de repetición. Comenzará como un archivo vacío.
- **server_keys.db:** Almacenará las llaves que fueron predistribuidas y sus respectivos identificadores.
- **server_bindings.db:** Almacenará el identificador de la llave que se asignó para intercambiar mensajes con un cliente en particular. Comenzará como un archivo vacío.

- **Usando Configuration Token**

Para iniciar al servidor usando el método de *Configuration Token*, se ejecuta el siguiente comando que se ilustra en la Figura 10.6.

```
# dhcpd -ct Secreto
```

Figura 10.6 Comando para iniciar el servidor usando *Configuration Token* con clave "Secreto".

- **Delayed Authentication**

Para iniciar al servidor usando el método de *Delayed Authentication*, el archivo **"/var/db/server_keys.db"** debe tener previamente las llaves con sus identificadores respectivos. Los identificadores pueden tomar valores entre 0 y 4.294.967.295 y las llaves consistirán en una cadena de texto con una longitud máxima de 64 caracteres.

Un archivo de ejemplo y su formato puede verse en la Figura 10.3 mostrada anteriormente para el cliente. Finalmente se ejecuta el comando como se ilustra en la Figura 10.7.

```
# dhcpd -da md5
```

Figura 10.7 Comando para iniciar el servidor usando *Delayed Authentication* con algoritmo MD5.