TRABAJO ESPECIAL DE GRADO

ESTUDIO DEL COMPORTAMIENTO DEL PROTOCOLO DTLS PARA VOIP, BASADA EN SEÑALIZACIÓN SIP SOBRE UDP, CONSIDERANDO EL TIEMPO DE RESPUESTA COMO VARIABLE DE INTERÉS.

Presentado ante la Ilustre
Universidad Central de Venezuela
por el Br. Peña C, Josua D.
para optar al Título de
Ingeniero Electricista

Caracas, 2011

TRABAJO ESPECIAL DE GRADO

ESTUDIO DEL COMPORTAMIENTO DEL PROTOCOLO DTLS PARA VOIP, BASADA EN SEÑALIZACIÓN SIP SOBRE UDP, CONSIDERANDO EL TIEMPO DE RESPUESTA COMO VARIABLE DE INTERÉS.

TUTOR ACADÉMICO: PhD. Carlos Moreno

Presentado ante la Ilustre
Universidad Central de Venezuela
por el Br. Peña C, Josua D.
para optar al Título de
Ingeniero Electricista

Caracas, 2011

CONSTANCIA DE APROBACIÓN

DEDICATORIA

A mi madre, Ana María Carreño y a mi hermano, Jonathan Peña por ser mi motivación para seguir adelante.

AGRADECIMIENTOS

Agradezco a todas las personas que contribuyeron al desarrollo de esta investigación, especialmente a:

Anderson Nascimento por iniciarme en el mundo de la seguridad de redes.

Carlos Moreno por guiarme en todo el proceso investigativo.

Ebert Brea por su apoyo en la recolección de datos estadísticos.

Franklin Da Costa Silva por devolverme el amor por la carrera.

Pedro Pinto por creer en mí y apoyarme en el proceso de intercambio.

Kenneth Ramírez y Julio Miquilena por ayudarme a colectar los datos.

Carmen Peláez por estar siempre dispuesta a ayudar en todo.

José Páez tío, maestro y amigo por su ayuda incondicional.

María Auxiliadora Rojas por ser una gran amiga y colaboradora siempre.

Gladys Bruzual por ser la primera profesora en tenderme una mano cuando tuve dificultades.

Zaldívar Bruzual, Caruis Parra, Glauce Martins, Diego de Tássio, Talita Ribeiro, el Sr. Nonato y la Sra. María por toda su ayuda y colaboración durante el intercambio.

Krýstina Sedláková, Yuver Maceda, André Vieira, Rafael Moser, Fernanda Canchola, Paola Coronado, Uirá Godoi, Luciano Grossi, Joabe D'Souza, Anie Gomez, Guilherme Santos, Laila Wanic, Linniker Corado, Nayara De Souza, Estevão Fonseca, Adriell Santana, Renielton Santos, Reyna Lari, Jorge Aranda, Ramiro Anastasi, Andy Chunco y otras personas, por ser mi familia durante el intercambio.

Eugenia Cubillán, Sinamaica Henríquez, Johanel Da Silva, María Eugenia Rodríguez, Margarita Flores, Carla Adam, Santiago Gómez, Yelini Arcila, Ismael Carneiro, por ser parte de la familia no sanguínea que me dio Dios. Y finalmente a mí padre Johnny Rey Peña por darme la vida.

Peña C., Josua D.

ESTUDIO DEL COMPORTAMIENTO DEL PROTOCOLO DTLS PARA VOIP, BASADA EN SEÑALIZACIÓN SIP SOBRE UDP, CONSIDERANDO EL TIEMPO DE RESPUESTA COMO VARIABLE DE INTERÉS.

Tutor académico: PhD Carlos Moreno. Tesis. Caracas. U.C.V. Facultad de Ingeniería. Escuela de Ingeniería Eléctrica. Ingeniero Electricista. Opción: Comunicaciones. Institución: UCV 2011. 83 h. + anexos.

Palabras Clave: DTLS-Datagram Transport Layer Security, SIP-Session Iniciation Protocol, UDP-User Datagram Protocol, Tiempo de respuesta.

Resumen. Se estudia el comportamiento de un sistema de voz sobre IP tanto cuando se protege la señalización SIP usando el protocolo DTLS como cuando sólo se usa SIP directamente sobre UDP. Para realizar esto, se hizo una documentación teórica de los protocolos, se hicieron mediciones reales para el sistema sin seguridad, se escogió un simulador para realizar el estudio de comportamiento y se construyó una función de comparación de desempeño de ambos sistemas que relaciona el nivel de seguridad con el tiempo de establecimiento de la llamada. De los resultados se infiere que un sistema provisto de seguridad demora más tiempo en establecer la conexión mas el desempeño del mismo es superior. Finalmente se sugieren algunas actividades a realizar para futuros proyectos de investigación.

ÍNDICE GENERAL

LISTA DE TABLAS

LISTA DE FIGURAS

ACRÓNIMOS

SIP : Session Initiation Protocol
DTLS : Datagram Transport Layer Security
TLS : Transport Layer Security
UDP : User Datagram Protocol
VoIP : Voice over IP
ICMP : Internet-Control Message Protocol
MAC : Media Access Control
TCP : Transmission Control Protocol
IP : Internet Protocol
DCCP: Datagram Congestion Control Protocol
SCTP : Stream Control Transmission Protocol

INTRODUCCIÓN

Con el uso extendido de los sistemas de comunicación y el crecimiento de los delitos informáticos, se hace de vital importancia, proteger las comunicaciones entre sistemas. Particularmente, las llamadas realizadas a través de la red de redes, requieren especial trato con respecto a la seguridad. Son vulnerables al uso indebido de la información transmitida y se podría modificar los mensajes de señalización para escuchar lo que se esté transmitiendo o disfrazar la identidad de alguna de las partes involucradas.

Pensando en eso, la presente investigación busca determinar la relación entre el tiempo de establecimiento de una llamada IP y el nivel de seguridad de un sistema, cuando se usa el protocolo de señalización SIP y el protocolo de datagramas UDP. Se estudiarán dos escenarios, uno sin seguridad (estructura SIP-UDP) y otro con seguridad provista por el protocolo de seguridad de la capa de transporte orientado a datagramas (DTLS en inglés). Esto se realizará mediante la recolección de datos reales de la estructura sin seguridad y mediante la simulación de ambas estructuras usando un software de diseño.

En los siguientes capítulos se describirán los diferentes protocolos, se documentará las razones por las cuales se escogió un simulador en particular, se describirán los experimentos realizados tanto para obtener los datos reales como para obtener los datos de la simulación y se analizarán dichos datos en aras de finalmente, exponer las conclusiones a las que se llegó.

Finalmente, harán recomendaciones para trabajos futuros y que complementen la investigación, así como otros aspectos que aunque están relacionados, se escapan de los alcances de esta investigación.

CAPÍTULO I

1.1. Planteamiento del Problema

La utilización del Transfer Layer Security (TLS) para el cifrado de un sistema de telecomunicaciones implica lidiar con retrasos en la transmisión, retrasos que se ven aumentados si se usa el protocolo de capa de transporte Transmission Control Protocol (TCP). Para las aplicaciones de transferencia de voz sobre IP (VoIP) es necesario tener un retraso mínimo debido a que mayores demoras empeoran la calidad de servicio ofrecido al cliente.

Otra implicación de usar el protocolo TCP para VoIP es que es necesaria la retransmisión de paquetes perdidos o de los cuales no se haya acusado el recibo, haciendo que puedan aparecer trozos de la conversación que ya fueron dichos, interfiriendo con el buen desarrollo de la llamada.

Estos problemas pueden ser corregidos mediante la utilización de otros protocolos de seguridad y de transferencia de datos. Evidencia de dicha corrección se encuentra en la investigación desarrollada por (McGrew & Rescorla, 2010) [4]. En su trabajo, desarrollaron una extensión del protocolo Datagram Transport Layer Security (DTLS) para usarla como protocolo de seguridad de un sistema basado en Secure Real-time Transport Protocol (SRTP), protocolo que a su vez usa el User Datagram Protocol (UDP) como protocolo de capa de transporte.

Basándose en estos hechos, surgen varias interrogantes acerca de la confiabilidad de un sistema y su velocidad de funcionamiento. ¿Se entorpece el funcionamiento del sistema si se incorpora otro protocolo para ofrecer seguridad? ¿Se podría garantizar la seguridad de la señalización al usar el protocolo DTLS? ¿Cuál será el vínculo, si lo hay, entre la seguridad de un sistema y su velocidad de

respuesta? ¿Será que al variar las configuraciones de conexión (escenarios) la seguridad se afecta? ¿Y la velocidad respuesta?

1.2. Antecedentes

Entre los trabajos en los cuales se menciona el uso de los protocolos SIP y DTLS se encuentran:

Modadugu, N., y Rescorla, E. en 2004 desarrollaron "*The Design and Implementation of Datagram TLS"*, documento en el cual se describe el protocolo DTLS y la manera de implementarlo.

Posteriormente, en 2006, Brito y Moura de Sousa realizaron su trabajo de grado llamado "Análise de protocolos para prover confidencialidade e integridade em sessões de Telefonia IP", donde expusieron lo siguiente:

> *"Uma desvantagem de utilizar TLS para a proteção da sinalização é o delay agregado com o uso do TCP como protocolo da camada de transporte. Um trabalho futuro seria utilizar o DTLS como alternativa ao TLS, tendo em vista que o Datagram Transport Layer Security utiliza o protocolo UDP" pág. 100* [1]

Ellos después de analizar diversos protocolos para proveer confidencialidad e integridad en sesiones de telefonía IP, propusieron como un trabajo futuro, la utilización del protocolo *Datagram Transport Layer Security (DTLS)* como alternativa al TLS en vista que el protocolo DTLS usa el protocolo UDP.

En 2007 Modadugu y Jennings publicaron ante la IETF un documento titulado: "Session Iniciation Protocol (SIP) over Datagram Transport Layer Security (DTLS)", en él se explica de manera simplista, que el protocolo SIP puede ser usado sobre DTLS, mas no expone datos ni resultados de experimentos realizados. De aquí

se puede asumir que ellos están trabajando sobre el sistema mas hasta la fecha, no se han presentado avances sobre dicha investigación.

En 2010 McGrew y Rescorla publicaron, también ante la IETF, un documento titulado: *"Datagram Transport Layer Security (DTLS) Extension to Establish Keys"* donde decriben una extensión al protocolo DTLS para el intercambio de llaves previo a la transferencia segura de voz en tiempo real y el flujo del protocolo de control *Secure RTP*.

Luego, teniendo en cuenta los datos recogidos en estas investigaciones se pueden tomar como fundamento para la presente investigación.

1.3. Justificación

El avance de la tecnología ha obligado la actualización de los sistemas de comunicación, desde la aparición de los celulares hasta el desarrollo de súper computadoras personales. Esto aumentó las posibilidades de establecer conexiones comunicacionales a varios niveles haciendo que los costos de operación se redujeran. Es así que se llega al desarrollo de programas de computadoras que permiten realizar llamadas telefónicas entre computadores y hacia teléfonos convencionales. Luego, a estos programas se les agrega la capacidad de proveer llamadas con video y posteriormente se incluyeron las videoconferencias.

Con la llegada de las aplicaciones descritas arriba, también llegó la necesidad de proteger la información que transita entre los puntos que se estén comunicando. Y es así que surgen los protocolos de cifrado, para evitar el mal uso de la información intercambiada por dos o más usuarios.

Sin embargo, debido a la sensibilidad intrínseca de los sistemas que manejan el tráfico de voz y video en tiempo real, es importante establecer protocolos de seguridad rápidos y eficientes. Luego, dependiendo de los protocolos usados para el

cifrado, de los protocolos de señalización y de transporte, se originan retrasos en la transmisión que pueden generar problemas a la comunicación. Por tanto, se hace de vital importancia establecer una correlación entre los retrasos ocurridos en el sistema y el nivel de seguridad aportado al mismo.

Es importante destacar que debido a lo novedoso de los protocolos estudiados en esta investigación, la mayoría de las informaciones se encuentran en lenguas extranjeras por lo que se harán citas en el idioma original y luego se comentará sobre su interpretación en español.

1.4. Objetivos Generales

Estudiar el comportamiento del protocolo DTLS para VoIP, basada en señalización SIP sobre UDP, considerando el tiempo de respuesta como variable de interés.

1.5. Objetivos Específicos

- Señalar las ventajas de la utilización del protocolo DTLS sobre el protocolo TLS
- Enumerar los productos, tanto software como hardware, que utilizan DTLS
- Revisar en la literatura la clasificación de niveles de seguridad para sistemas seguros
- Determinar un criterio de medición del tiempo de respuesta para el establecimiento de una conexión SIP sobre UDP con cifrado DTLS
- Definir escenarios de transmisión de voz para la medición del tiempo de respuesta con DTLS
- Simular y medir los tiempos de respuesta para cada escenario
- Estudiar la relación entre el tiempo de respuesta y el nivel de seguridad de cada escenario

CAPÍTULO II. MARCO TEÓRICO

2.1. Generalidades sobre criptografía

Antes de iniciar este estudio, es importante conocer algunos conceptos básicos sobre criptografía, el "arte de escribir con clave secreta o de un modo enigmático" (Real Academia Española, 2001) [8].

El proceso de ocultar la esencia de un mensaje es denominado **cifrado**, y el proceso inverso es llamado **descifrado**. Ambos procesos son llevados a cabo usando funciones matemáticas especiales y son llamadas **algoritmos criptográficos**.

El proceso de cifrado implica el uso de una **llave de cifrado**, que usualmente es un grupo de bits generados según una regla definida por el tipo de algoritmo escogido, que usada en conjunto con el mensaje oculta la información contenida en el mensaje obteniéndose entonces el ***hash*** de éste.

Luego, para recuperar la información oculta, el receptor usa una **llave de descifrado.** Ambas llaves pueden ser iguales (**criptografía simétrica**) o diferentes (**criptografía de llave pública**).

En la criptografía de llave simétrica, la llave de descifrado puede ser calculada a partir de la llave de cifrado y viceversa. En cambio, en la criptografía de llave pública hay dos llaves, una de cifrado (**llave pública**) y una de descifrado (**llave privada**). Por tanto, alguien puede cifrar un mensaje con la llave pública de otro usuario pero sólo éste podrá descifrarlo.

2.2. Seguridad de sistemas informáticos

Para ofrecer un sistema informático seguro, es necesario proveer tres prerrogativas básicas: autenticidad, integridad y confiabilidad.

Autenticidad: es necesario garantizar que las partes involucradas en la comunicación tengan la certeza de que la persona (o terminal) que esté al otro lado, sea realmente a quien están contactando. Esto se logra a través de protocolos de tipo:

marca de tiempo (se acepta el mensaje si y sólo si, tiene una marca de tiempo reciente); desafío-respuesta (se cifra un número difícil de adivinar con la llave pública de la otra parte, ésta lo devuelve descifrado al requeridor) o a través de un centro de distribución de llaves (KDC en inglés, se encarga de mediar la autenticidad de los afiliados y debe ser reconocido como la autoridad confirmadora de identidad).

Integridad: también es necesario garantizar que la información que se transmita, no sea modificada, interceptada o borrada por terceros durante la comunicación. Esto se logra mediante la utilización de protocolos que ofrezcan acuse de recibo y pidan la retrasmisión de los paquetes que no hayan llegado (TCP por ejemplo). Y para la seguridad de la información, es necesario usar un protocolo que cifre los paquetes de tal manera que con uno o varios paquetes, no sea posible recuperar la información oculta (EAX, CWB o IAPM).

Confiabilidad: para muchas aplicaciones, es necesario tener la seguridad de que la otra parte haya recibido los mensajes. Es por ello que los protocolos seguros deben proveer esta capacidad. Entre los protocolos que la proveen están: TCP, SCTP y TLS.

2.3. Modelo OSI

El modelo de Interconexión de Sistemas Abiertos (OSI en inglés), es un sistema desarrollado por el *International Organization for Standarization* en 1977 para permitir la interconexión entre los sistemas diseñados por desarrolladores diferentes. Está estructurado en siete capas y dos grupos, el primer grupo engloba los servicios de transporte (capas del 1 al 4) y el segundo grupo engloba los servicios soporte al usuario (capas 5, 6 y 7).

2.1.1 Capas del modelo OSI

1. Capa física: agrupa los medios físicos y materiales utilizados para el envío de la información. Esta capa se ocupa de la transmisión de bits a través de un canal de comunicación, esto es logrado al usar niveles de tensión, variaciones de frecuencias o intervalos de tiempo determinados. También debe garantizar que los bits enviados sean los bits recibidos. Ejemplo: par trenzado, cable coaxial, fibra óptica, ondas de radio, entre otros.

2. Capa de enlace: realiza la adaptación del medio físico común a un medio sin errores para la capa de red. Esto se realiza mediante la segmentación de los datos en tramas de asentimiento y agregando un patrón de bits entre tramas para sincronía. También se encarga de solucionar problemas de reenvío o de duplicación de mensajes.

Las tramas podrían contener informaciones sobre la cantidad de caracteres y los bits de inicio y fin.

Entre los servicios ofrecidos a la capa de red están:

Servicio sin acuses de recibo: se envían tramas de datos directamente desde la fuente. Esto es apropiado cuando el tráfico es en tiempo real (ej. VoIP) o si la frecuencia de errores es baja.

Servicio con acuses de recibo: se envían mensajes de confirmación de recepción por cada trama recibida.

Control de flujo: mediante el uso de protocolos de limitación de datos, se controla la cantidad de tramas que pueden ser recibidas bajo la permisión implícita o explícita del receptor. Por ejemplo, el remitente puede mandar un número cualquiera de tramas pero luego tiene que esperar permiso para enviar el resto de la información.

Detección y corrección de errores: un ejemplo de un protocolo de capa de enlace usado para la detección de errores es el HDLC (High-level Data Link Control). Este protocolo está orientado a bit, sus especificaciones cubren las informaciones que llevan cada bit de la trama.

3. Capa de Red: controla la operación de la subred. La operación principal es eliminar los embotellamientos resultantes de la saturación de la red, debido al envío de paquetes. Ofrece pues, mecanismos de encaminamiento de dichos paquetes a sus destinatarios. También, provee el conteo de paquetes enviados y funge como intérprete entre protocolos de diferentes naturalezas o de direcciones desiguales.

En resumen, esta capa debe manejar la topología de la subred, evitar congestiones y administrar saltos cuando la fuente y el destino están en redes distintas.

4. Capa de transporte: adapta los datos recibidos de las capas superiores y los divide en unidades menores para pasarlos a la capa de red, certificando que todos los segmentos sean recibidos correctamente, independientemente del hardware que se esté usando.

Para optimizar la capacidad de transporte, también es posible implementar multicanalización de varias conexiones de la misma red. Este proceso es transparente a la capa de sesión.

Ofrece prestaciones a la capa de sesión para garantizar un servicio eficiente y confiable a los clientes de la misma, que usualmente son procesos de la capa de aplicación.

En este nivel, el hardware y el software son llamados entidad de transporte y pueden situarse en sistemas operativos, en programas, en tarjetas, entre otros.

Aunque las operaciones de esta capa y la capa de red son muy parecidas, la principal diferencia es que la de red es parte de la subred por lo que los usuarios no

tienen control sobre ella; caso contrario ocurre con la capa de trasporte que permite una interacción con el usuario para mejorar el servicio.

5. Capa de Sesión: permite el inicio de sesión entre usuarios para compartir sistemas a distancia o transferir archivos entre máquinas diferentes.

Esta capa provee servicios de gerencia de turnos en el tráfico de información, administración de tareas para diversos protocolos y sincronización de operaciones con el tiempo de caída del sistema.

6. Capa de Presentación: se encarga de los aspectos de sintaxis y semántica de la información transmitida, como la codificación de la misma según un código preestablecido.

Esto, debido a que cada máquina maneja las informaciones en formatos diferentes. Además, provee compresión de la información para menor número de bits.

7. Capa de Aplicación: es la capa más cercana al usuario compuesta por el programa o el conjunto de programas que generan información para ser transmitidas en una red.

Mediante el uso de terminales virtuales de orden abstracto, administra los protocolos usados con más frecuencia.

Ejerce la función de traductor entre distintas máquinas de transferencia de archivos. Un ejemplo de esto se evidencia en el sistema de correo electrónico, que puede ir a cualquier parte del mundo y ser leído en cualquier tipo de terminal fijo o móvil.

Esta capa, al definir diversos protocolos, garantiza un conjunto de normas para las aplicaciones de red.

2.4. *User Datagram Protocol*

El *User Datagram Protocol* (UDP, propuesto en el RFC 768 del 28 de agosto de 1980 por J. Postel) es un protocolo de capa de transporte que permite el envío de mensajes a través de la conmutación por paquetes con el mínimo número de pasos. Está orientado a las transacciones y no provee garantía sobre la entrega o la duplicación de paquetes. Tampoco garantiza la reorganización de los paquetes en el receptor.

La figura 1 muestra el formato de los paquetes UDP:

bits	0 – 15	16 – 31
0	Source Port Number	Destination Port Number
32	Length	Checksum
64	Data	

Figura 1 Formato de los paquetes UDP[1]

2.4.1. Campos del paquete UDP

***Source port*:** es un campo opcional y cuando es usado indica la dirección del puerto de salida del mensaje y puede ser usado para direccionar paquetes de respuesta.

Destination port: en él se indica la dirección de destino del paquete

Length: indica la longitud del datagrama, en octetos, incluyendo este campo y la carga.

Checksum: indica una suma de comprobación de 16 bits que abarca una pseudocabecera IP (con las direcciones IP de origen y de destino, el protocolo y la longitud del paquete UDP), la cabecera UDP, los datos y ceros hasta completar un

[1] Imagen de (Internet Engineering Task Force, 1980) [4]

múltiplo de 16. Este campo también es opcional en IPv4, aunque es usado con frecuencia (en IPv6 su uso es obligatorio).

La figura 2 muestra los campos para el cálculo de la suma de comprobación cuando se usa IPv4. Los campos marcados en rosa son los que conforman la pseudocabecera IP.

bits	0 – 7	8 – 15	16 – 23	24 – 31
0	Source address			
32	Destination address			
64	Zeros	Protocol	UDP length	
96	Source Port		Destination Port	
128	Length		Checksum	
160	Data			

Figura 2 Campos de la pseudocabecera IP[2]

2.5. *Datagram Transport Layer Security* (DTLS)

Es un protocolo de capa de transporte (definido en el RFC4347 de abril de 2006 por E. Rescorla y N. Modadugu) que provee seguridad a protocolos que usan datagramas. La información es cargada en registros y estos componen la capa de registro o *Record layer*.

2.5.1. *Record layer*

Para procesar la información en los registros, los mismos deben estar completos para evitar retrasos por fragmentación. Esto obliga a que el DTLS ajuste el tamaño del registro a un solo datagrama.

El uso de tamaños de registro definidos y de registros en cada datagrama, asegura que no haya almacenamiento parcial de información optimizando el uso de la

[2] Imagen de (Internet Engineering Task Force, 1980) [4]

memoria y evitar ataques de negación de servicio. También evita que se pierdan fragmentos de mensajes que inutilizarían la información. Luego, debido a la incertidumbre sobre el tamaño de los paquetes fragmentados y a la capacidad limitada de almacenamiento, no es posible saber cuánto tiempo es necesario antes de descartar los fragmentos.

El formato de un registro DTLS se muestra a continuación:

+	Byte +0	Byte +1	Byte +2	Byte +3
Byte 0	Content type			
Bytes 1..4	Version	Epoch	Sequence_number	Payload Length
Bytes 5..(*m*-1)	Payload			
Bytes m..(p-*1*)				
Bytes p..(q-*1*)				

Figura 3 Formato de registro DTLS[3]

A continuación, se comentan los campos agregados para modificar la estructura básica del TLS:

2.5.1.1. *Epoch*

Este campo está compuesto por números usados en ambos extremos del sistema, para identificar el tipo de cifrado que fue usado en la protección de la carga útil. También son especialmente útiles para eliminar cualquier ambigüedad presentada a la hora de la negociación de reinicio de sesión; esto se logra manteniendo el mismo número *epoch* en mensajes que hayan sido cifrados con la misma llave e incrementándolo al cambiar la llave.

[3] Imagen de (Internet Engineering Task Force, 2006) [3]

2.5.1.2. *Sequence number*

El número de secuencia agregado en cada registro, sirve para mantener el orden de los paquetes recibidos aun cuando haya pérdida de alguno. Esto es vital a la hora de usar VoIP debido a la necesidad de mantener el flujo de paquetes aunque se hayan perdido algunos durante la transmisión.

2.5.1.3. *Payload length*

Este campo indica el tamaño total del registro y usualmente es del PMTU (Tamaño Máximo de Camino de Transmisión). El mismo depende del protocolo que se esté implementando, por ejemplo, al usar Ethernet, el PMTU es de 1518 bytes.

2.5.2. Modo de cifrado

Este protocolo usa el modo CBC (Criptografía de Bloques Encadenados) para el cifrado. Este consiste en:

- Fragmentar el mensaje en claro (*plaintext*)
- Escoger un vector de inicialización (VI) aleatorio
- Aplicar XOR al segmento con el VI
- Cifrar el primer bloque junto con el VI
- Este bloque cifrado (*ciphertext*), es usado para cifrar el siguiente bloque y así sucesivamente.

En la siguiente figura se puede apreciar la idea general para el cifrado:

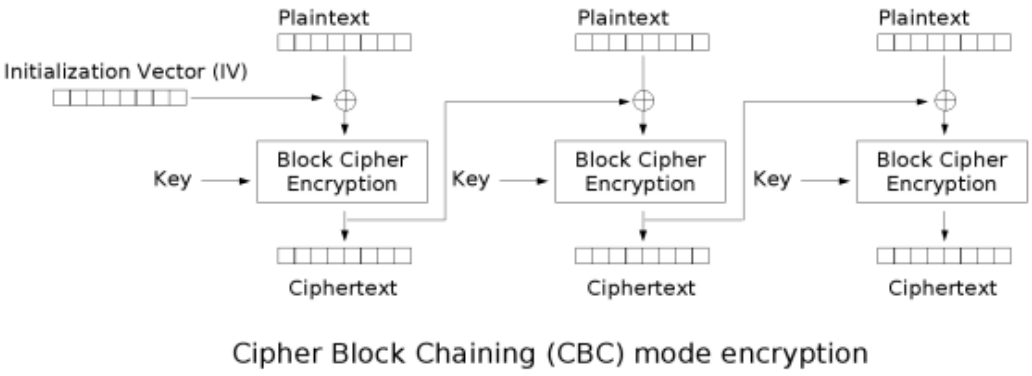


Figura 4 Modo de cifrado CBC (Cifrado) [4]

Este método de cifrado simplifica el descifrado de los mensajes, ya que aunque haya habido una pérdida de paquetes, los diferentes fragmentos pueden ser descifrados por separado. La figura 5 muestra dicho proceso.

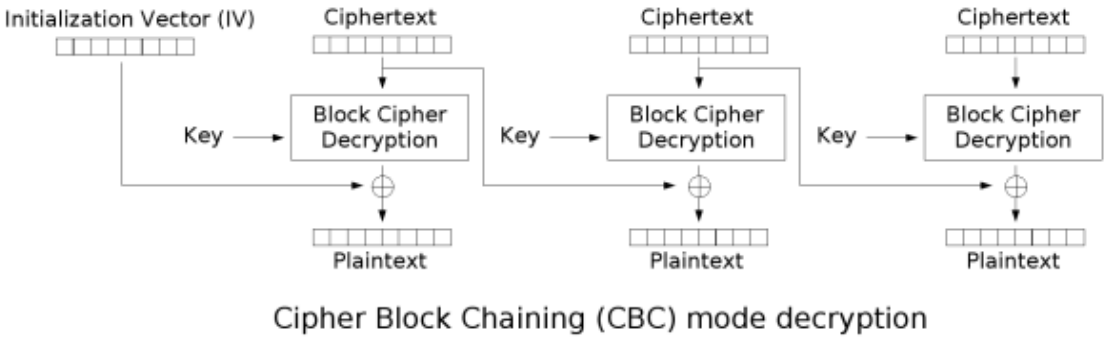


Figura 5 Modo de cifrado CBC (Descifrado)[5]

[4] Imagen de (Modadugu & Rescorla, 2004) [6]

[5] Imagen de (Modadugu & Rescorla, 2004) [6]

2.5.3. Protocolo de Handshake

La siguiente figura muestra el proceso de establecimiento de la conexión de este protocolo:

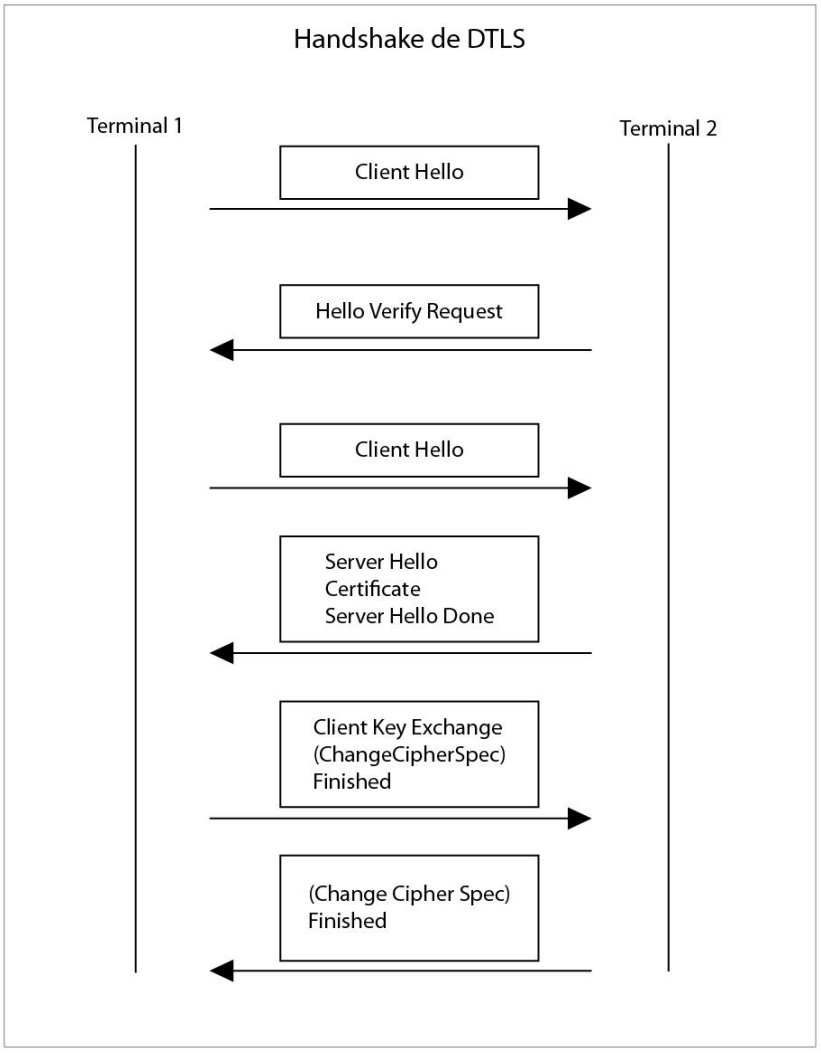


Figura 6 *Handshake* de DTLS[6]

[6] Imagen adaptada de (Internet Engineering Task Force, 2006) [3]

El primer mensaje de *ClientHello* contiene la versión del protocolo que soporta, un número aleatorio, la identificación de la sesión, un campo vacío para el almacenamiento de un *cookie* que será suministrado por el servidor, sugerencias sobre el cifrado que pueden usar y el método de compresión que soporta. Luego, el servidor envía un mensaje de *HelloVerifyRequest* escogiendo la versión del protocolo y un *cookie,* un número que debe ser generado de tal manera que no haya consumo de memoria en almacenamiento, como por ejemplo usar la dirección IP del cliente y generar un *hash* de ésta con una llave global. En seguida el cliente verifica el *cookie* recibido y reenvía el mensaje de inicio de sesión con el *cookie* verificado. El servidor certifica al cliente y envía tres mensajes al cliente *ServerHello, Certificate* y ServerHelloDone. El primer mensaje da respuesta a los requerimientos de versión del protocolo DTLS, el tipo de llave a usar, y el método de compresión. El segundo, que depende del protocolo de cifrado escogido, se puede utilizar para verificar la autenticidad del servidor. Finalmente, en el tercero se indica que terminó la etapa de negociación del *handshake.* Posteriormente, el cliente puede enviar un mensaje de certificación de su identidad para luego enviar un mensaje de *ClientKeyExchange* para negociar la llave de la sesión, cifra esta información con la llave pública del servidor. También envía de *ChangeCipherSpec* indicando que la información que de allí en adelante se envíe, va a estar cifrada y autenticada, y envía un mensaje de *Finished* que contiene los *hash* y los MAC de todos los mensajes enviados previamente. Por su parte el servidor también envía sus mensajes de *ChangeCipherSpec* y de *Finished* para así finalizar el *handshake* y comenzar la transmisión. Ambas partes verifican los *hash* y los MAC, si alguno falla, se descarta el *handshake* y se comienza el proceso.

2.5.4. Tiempo de desconexión y mecanismos de retransmisión

La probabilidad de pérdida de paquetes durante el *handshake* siempre está presente y es por eso que se necesitan mecanismos de retransmisión de paquetes. El DTLS usa un temporizador simple en cada extremo del sistema y permanece enviando el último mensaje hasta recibir una respuesta. La siguiente imagen muestra la máquina de estados que describe este proceso:

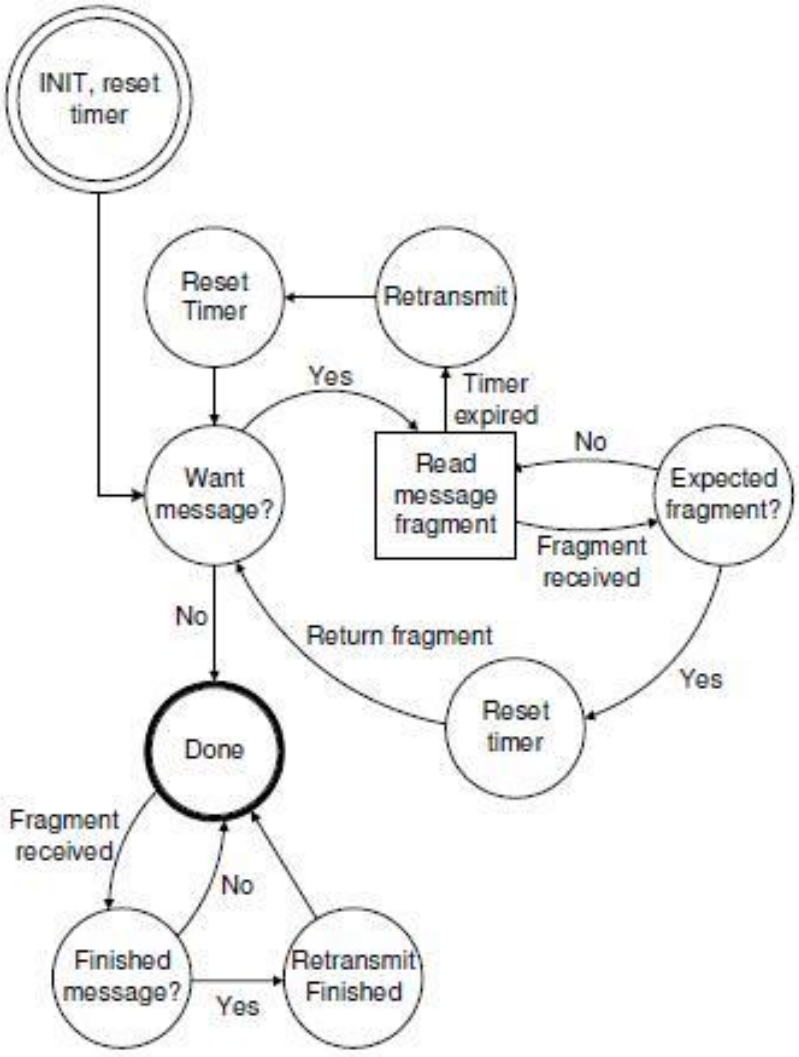


Figura 7 Máquina de estados para retransmisión DTLS[7]

[7] Imagen de (Modadugu & Rescorla, 2004) [6]

El bloque que controla la recepción de mensajes es el bloque de *Read message fragment*. La máquina cambia de este estado si se recibe un fragmento o si el contador expira. En el primer caso, compara si el fragmento recibido es el deseado, si lo es, reinicia el contador, sino, descarta el fragmento y solicita la lectura de más fragmentos. En el segundo caso, se pide la retransmisión del último grupo de mensajes y se reinicia el contador.

2.5.5. Mensajes de Handshake y fragmentación

El tamaño de los paquetes de *handshake* puede ser más grande que el tamaño de los mensajes de registro DTLS, esto obliga que haya modificaciones en los mensajes de *handshake* para que quepan en los registros. La modificación es llevada a cabo con la inclusión de los campos "message_seq", "frag_offset" y "frag_length".

Estos campos son usados para mantener la secuencia de los mensajes, para saber la compensación hecha a cada fragmento y su longitud, respectivamente.
En la siguiente imagen se muestra la estructura modificada

<table>
<tr><th>+</th><th>Byte +0</th><th>Byte +1</th><th>Byte +2</th><th>Byte +3</th></tr>
<tr><td>Byte 0</td><td colspan="2">22</td><td colspan="2"></td></tr>
<tr><td>Bytes 1..4</td><td colspan="2">Version</td><td colspan="2">Length</td></tr>
<tr><td>Bytes 5..8</td><td rowspan="2">Message type</td><td colspan="3">Handshake message data length</td></tr>
<tr><td></td><td>Message_Seq</td><td>frag_offset</td><td>frag_length</td></tr>
<tr><td>Bytes 9..(n-1)</td><td colspan="4" rowspan="3">Handshake message data</td></tr>
<tr><td>Bytes n..(n+3)</td></tr>
<tr><td>Bytes (n+4)..</td></tr>
</table>

Figura 8 Estructura modificada de los mensajes de *handshake*[8]

[8] Imagen adaptada de (Internet Engineering Task Force, 2006) [3]

Durante el *handshake*, se usan diversos campos y mensajes, entre ellos:

Message Length: este campo contiene la longitud de todo el mensaje para simplificar la selección del espacio de almacenamiento en el receptor.

Message Sequence Number: es un número de secuencia adicionado a los mensajes de handshake y de ChangeCipherSpec, que son propios e independientes de los números de secuencia de registros (RSN). Esto hace que el receptor pueda recibir registros y sus duplicados (debido a retransmisión) con diferentes RSN.

Fragment Offset and Length: como ya se mencionó, el *offset* (compensación) y la longitud total del mensaje fragmentado, deben conocerse de antemano para que la transmisión sea llevada a cabo. Esta información es incluida en la estructura del *handshake* y se muestra en la figura 8.

Finished Message: este mensaje es enviado para la confirmación de la correcta negociación de llaves y de algoritmos. Para ello el DTLS calcula el *hash* de cada mensaje de *handshake* como si fuera un fragmento aislado. Este *hash* es comparado con los *hash* de los fragmentos en el remitente.

Mensajes de alerta: son mensajes generados en el caso que haya ocurrido algún error. La siguiente tabla muestra los mensajes de alerta que pueden ser generados durante la sesión.

Tabla 1 Mensajes de alerta DTLS

Código	Descripción		Nivel de gravedad
0	Close notify	Notificación de cierre	advertencia/fatal
10	Unexpected message	Mensaje inesperado	fatal
20	Bad record MAC	Error de registro MAC	fatal
21	Decryption failed	Fallo al descifrar	fatal
22	Record overflow	Desborde de registro	fatal
30	Decompression failure	Falla de descompresión	fatal
40	Handshake failure	Error de *handshake*	fatal
41	No certificate	Ningún certificado	advertencia/fatal
42	Bad certificate	Certificado errado	advertencia/fatal
43	Unsupported certificate	Certificado no soportado	advertencia/fatal
44	Certificate revoked	Certificado revocado	advertencia/fatal
45	Certificate expired	Certificado expirado	advertencia/fatal
46	Certificate unknown	Certificado desconocido	advertencia/fatal
47	Illegal parameter	Parámetro ilegal	fatal
48	Unknown CA (Certificate authority)	Unidad certificadora desconocida	fatal
49	Access denied	Acceso denegado	fatal
50	Decode error	Error de decodificación	fatal
51	Decrypt error	Error de descifrado	advertencia/fatal
60	Export restriction	Restricción de exportación	fatal
70	Protocol version	Versión de protocolo	fatal
71	Insufficient security	Seguridad insuficiente	fatal
80	Internal error	Error Interno	fatal
90	User cancelled	Usuario cancelado	fatal
100	No renegotiation	Renegociación no válida	advertencia
110	Unsupported extension	Extensión no soportada	advertencia
111	Certificate unobtainable	No se puede obtener el certificado	advertencia
112	Unrecognized name	Nombre no reconocible	advertencia
113	Bad certificate status response	Respuesta errada de estado de certificado	fatal
114	Bad certificate hash value	Valor errado de *hash* de certificado	fatal

2.6. Aplicaciones del protocolo DTLS

Para comentar sobre los diferentes softwares existentes que usan este protocolo, es imperativo comentar que dicho protocolo a pesar de ser una alternativa de código abierto al IPSec (de uso extendido en la actualidad), no ha gozado de gran recepción debido a la falta de difusión.

Dentro de la gran gama de informaciones disponibles en internet, se pueden encontrar varios softwares que usan el DTLS. Especial énfasis se le puede dar a aquellos proyectos que lo ofrecen como materia prima de otros sistemas, entre ellos están:

yaSSL: *yet another Secure Socket Layer* que en español quiere decir "una nueva capa de conexión segura". Es una empresa desarrolladora de software de seguridad en la capa de transporte. De acuerdo con su página web, ellos ofrecen líneas de código de DTLS para ser usadas con distribuciones realizadas por los desarrolladores, tanto particulares como empresariales. Una de las aplicaciones que ofrecen, permite la integración de estos protocolos de seguridad a aplicaciones escritas en el lenguaje de programación C; este producto es llamado CyaSSL.

GnuTLS: es un proyecto que tiene como objetivo desarrollar una base de datos para que los desarrolladores puedan ofrecer servicios más seguros y fiables en las capas de transporte. Esto lo hace a través de la implementación de estándares del grupo de trabajo TLS de la IETF y de entre los protocolos propuestos por ese ente se encuentra el DTLS.

En el ámbito de los equipamientos, hasta la fecha de culminación de esta investigación, no se ha implementado ningún sistema. Esto debido a que el protocolo DTLS es de capa de transporte y no de capa física, lo que dificulta la posibilidad de implementar un hardware que use el protocolo a nivel físico.

2.7. Transport Layer Security y *Datagram Transport Layer Security*

El protocolo de seguridad en la capa de transporte (*Transport Layer Security* en inglés) fue presentado por primera vez en el RFC 2246 titulado "The TLS Protocol Version 1.0". Allí lo describen como un protocolo de capa de transporte usado para garantizar la seguridad de la conexión a nivel de esa capa; que usa un sistema confiable de transporte (TCP, DCCP, entre otros); que almacena las informaciones cifradas en registros (que pueden ser fragmentados) y que está orientado a la conexión. Debido a que el uso del protocolo TCP implica acuses de recibos de todos los paquetes enviados, las transmisiones en sensibles al retraso como VoIP, Video Streaming, entre otros, se ven afectadas y se entorpece el servicio prestado. Esta fue la principal razón por la cual se modificó el protocolo TLS para crear el DTLS.

Al ser creado basándose en un protocolo anterior, el DTLS tiene grandes similitudes con el TLS, y se establece como una de las principales diferencias el uso de datagramas para el transporte. También se modificó la segmentación de los registros para mejorar la velocidad de recuperación de los mensajes. Esto resulta sumamente beneficioso a la hora de ofrecer servicios de VoIP.

2.8. Session Initiation Protocol (SIP)

El protocolo de inicio de sesión (SIP) es un protocolo de capa de aplicación (definido por primera vez en el RFC 2543 de marzo de 1999) usado para el establecimiento, la modificación y el cierre de sesiones multimedia y llamadas IP. Éste, acepta mapeo de nombres y servicios de redirección, habilitando al usuario a iniciar y recibir servicios de comunicación desde cualquier ubicación.

Para poder describir mejor este protocolo, es necesario definir algunos conceptos relacionados con el mismo, como por ejemplo: *call, proxy, session, client, invitation,* entre otros.

Call: (llamada) está compuesta por todos los participantes de una conferencia, invitados por un servidor y es marcada unívocamente con un identificador

independiente. Es decir, una llamada IP punto a punto será mapeada con un solo identificador SIP.

Cliente: es un programa de capa de aplicación que envía requisiciones SIP. Los agentes de usuario y *proxies* contienen clientes y servidores.

Respuesta final: es el mensaje de fin de transacción SIP, todos los mensajes que tiene la estructura 2xx, 3xx, 4xx, 5xx y 6xx son mensajes de respuesta final.

Caller: parte que requisita un servicio de conferencia. Es importante destacar que el *caller* no necesariamente es quien crea dicha conferencia.

*C**alled party o Callee***: parte a la cual se está invitando a formar parte de la conferencia.

Invitación: es la requisición enviada a un usuario o servicio para iniciar una sesión. Esta invitación consiste en dos transacciones, la requisición *INVITE* seguida por otra requisición de *ACK* (estas transacciones serán explicadas más ampliamente adelante en este documento).

Servidor *proxy*: es un programa intermediario que actúa como un servidor y un cliente con el fin de hacer peticiones en nombre de otros clientes. Estas solicitudes, son atendidas a nivel interno o pasándolas a otros servidores modificando el mensaje para adaptarlo si fuera necesario.

Registrar: es un servidor que acepta las requisiciones de registro (mensajes *REGISTER*) y está localizado, usualmente, en el mismo nivel de los servidores *proxies*

Servidor: es un programa de capa de aplicación que acepta las requisiciones de los clientes y envía respuestas para las mismas.

Sesión: una sesión multimedia está constituida por remitentes, receptores y el flujo de datos entre ellos (de remitente a receptor). De acuerdo con esta definición, un cliente puede iniciar varias llamadas dentro de la misma sesión.

Transacción SIP: ocurre entre un cliente y un servidor, comprende todos los mensajes intercambiados desde el mensaje de requisición enviada por el cliente hasta el mensaje de respuesta final por parte del servidor. Se identifica a través del número de secuencia de un segmento sencillo de la llamada.

Conociendo estos conceptos, es posible describir el proceso llevado a cabo para la realización de una llamada SIP. La siguiente figura muestra los mensajes que son enviados y posteriormente se explicará la estructura de cada uno de ellos.

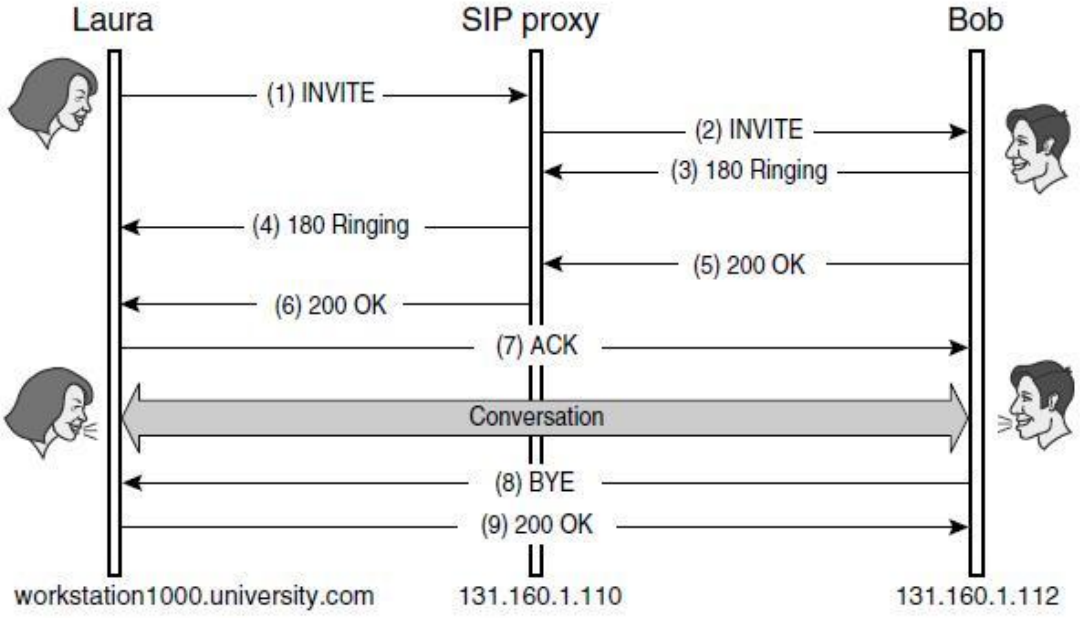


Figura 9 Mensajes intercambiados en una llamada SIP[9]

[9]Imagen de (McGraw-Hill Companies, 2004) [5]

El proceso es el siguiente, el remitente pide al servidor proxy, a través de un mensaje de INVITE, que lo conecte con Bob. El servidor busca la información de contacto del receptor en sus registros y envía un mensaje de INVITE directo al destinatario; el UAS destinatario responde con un mensaje de *RINGING.* Al el destinatario contestar la llamada, es enviado un mensaje de OK al remitente y este responde con un mensaje de ACK. Justo después de este mensaje la conversación puede iniciarse y su fin puede ser solicitado, por cualquier parte, al usar el mensaje de *BYE,* al cual es respondido con un mensaje de OK.

Al entender el proceso de establecimiento de una llamada SIP, es necesario entender la estructura de los mensajes enviados. En el siguiente apartado se explica el contenido de cada mensaje.

2.8.1. Estructura de los mensajes SIP

Los mensajes SIP pueden clasificarse en dos grandes grupos: Mensajes de Requisición y Mensajes de Respuesta. Dentro de estos grupos hay diversos mensajes que cumplen con siguiente estructura: cabecera + cuerpo. Las cabeceras son las que marcan las diferencias entre unos y otros. En los siguientes apartados se explican los campos presentes en ambos grupos y las cabeceras que pueden ser agregadas.

2.8.1.1. Mensajes de requisición

Estos mensajes están clasificados en seis categorías de acuerdo con (McGraw-Hill Companies, 2004)[5]

- INVITE
- ACK
- OPTIONS
- BYE
- CANCEL
- REGISTER

Los cuales contienen: una línea de requisición, varias cabeceras, una línea en blanco y el cuerpo del mensaje.

La línea de requisición comprende tres elementos: método (*method*), *Request-URI* (dirección del destinatario) y la versión del protocolo a usar.

Ejemplo: **INVITE sip:Bob.Johnson@company.com SIP/2.0**

En el campo de método es dónde se diferencian los tipos de mensajes, a continuación se describirán los seis tipos de mensajes.

INVITE: invita a un usuario unirse a una sesión SIP.

ACK: confirma la recepción de un mensaje INVITE

OPTIONS: solicita informaciones sobre las capacidades y prestaciones de un servidor.

BYE: solicita el fin de la sesión SIP.

CANCEL: solicita la anulación de transacciones pendientes. Por ejemplo, si un servidor que ha recibido un mensaje de INVITE pero aún no ha confirmado la recepción, va a anular la sesión al recibir el mensaje de CANCEL.

REGISTER: informa al servidor (REGISTRAR) sobre la localización actual del solicitante.

2.8.1.2. Mensajes de respuesta

Consiste en una línea de estado (*status line*), varias cabeceras, una línea en blanco y el cuerpo del mensaje. No todos los mensajes llevan cuerpo de mensaje.

La línea de estado está compuesta de tres elementos: versión del protocolo, código de estado (*status code*, reporta el estado de la transacción) y una *reason phrase* (una frase que no tiene función para el sistema pero sí para la persona que lo usa).

Ejemplo: **SIP/2.0 180 Ringing**

2.8.1.3. Cabeceras SIP

La siguiente tabla muestra las diferentes cabeceras que pueden ser adicionadas a los mensajes SIP.

Tabla 2 Cabeceras SIP[10]

Accept	Content-encoding	Max-forwards	Route
Accept-encoding	Content-language	Mime-version	Server
Accept-language	Content-length	Organization	Subject
Alert-info	Content-type	Priority	Supported
Allow	Cseq	Proxy-authenticate	Timestamp
Also	Date	Proxy-authorization	To
Authorization	Encryption	Proxy-require	Unsupported
Call-ID	Error-info	Record-route	User-agent
Call-info	Expires	Require	Via
Contact	From	Response-key	Warning
Content-disposition	In-reply-to	Retry-after	WWW-authenticate

Las cabeceras más resaltantes para esta investigación son:

Call-ID: representa la relación de señalización SIP entre dos o más usuarios. Identifica invitaciones particulares y todas las transacciones relacionadas a esa invitación. En el mensaje SIP asociado aparece como sigue:

***Call-ID*: ges456fcdw21lkfgte12ax@workstation1234.university.com**

***Cseq* (*command sequences*)**: este encabezado muestra la secuencia del mensaje SIP. Está compuesto por dos campos: un número entero usado para diferenciar requisiciones en la misma sesión y una frase donde se indica el tipo de método usado.

Cseq= 1 INVITE

Encryption: esta cabecera indica si el contenido del mensaje está cifrado o no. Tiene tres campos: tipo de cifrado, el esquema de cifrado y los parámetros de cifrado. La

[10] Tabla de (McGraw-Hill Companies, 2004) [3]

siguiente figura muestra la cabecera de un mensaje cifrado con PGP[11] blindado que usa el código ASCII y el mensaje cifrado.

```
Encryption: PGP version=2.6.2,encoding=ascii

hQEMAxkp5GPd+j5xAQf/ZDIfGD/PDOM1wayvwdQAKgGgjmZWe+MTy9NEX8O25Red
h0/pyrd/+DV5C2BYs7yzSOSXaj1C/tTK/4do6rtjhP8QA3vbDdVdaFciwEVAcuXs
ODxlNAVqyDi1RqFC28BJIvQ5KfEkPuACKTK7WlRSBc7vNPEA3nyqZGBTwhxRSbIR
RuFEsHSVojdCam4htcqxGnFwD9sksqs6LIyCFaiTAhWtwcCaN437G7mUYzy2KLcA
zPVGq1VQg83b99zPzIxRdlZ+K7+bAnu8Rtu+ohOCMLV3TPXbyp+err1YiThCZHIu
X9dOVj3CMjCP66RSHa/ea0wYTRRNYA/G+kdP8DSUcqYAAAE/hZPX6nFIqk7AVnf6
IpWHUPTelNUJpzUp5Ou+q/5P7ZAsn+cSAuF2YWtVjCf+SQmBR13p2EYYWHoxlA2/
GgKADYe4M3JSwOtqwU8zUJF3FIfk7vsxmSqtUQrRQaiIhqNyG7KxJt4YjWnEjF5E
WUIPhvyGFMJaeQXIyGRYZAYvKKklyAJcm29zLACxU5alX4M25lHQd9FR9Zmq6Jed
wbWvia6cAIfsvlZ9JGocmQYF7pcuz5pnczqP+/yvRqFJtDGD/v3s++G2R+ViVYJO
z/lxGUZaM4IWBCf+4DUjNanZM0oxAE28NjaIZ0rrldDQmO8V9FtPKdHxkqA5iJP+
6vGOFti1Ak4kmEz0vM/Nsv7kkubTFhRl05OiJIGr9S1UhenlZv9l6RuXsOY/EwH2
z8X9N4MhMyXEVuC9rt8/AUhmVQ==
=bOW+
```

Figura 10 Cabecera y mensaje SIP cifrado[12]

From: contiene la identificación del remitente y su dirección SIP (URL)

From: Bob Johnson <sip:Bob.Johnson@company.com>

To: contiene la dirección pública del destinatario, debe mantenerse igual durante toda la sesión y los servidores *proxies* no pueden modificarlo.

To: SIP:Bob@university.com

Via: esta cabecera almacena la dirección de todos los servidores *proxies* por los cuales pasan los mensajes SIP entre el remitente y el destinatario simplificando el manejo de las respuestas. Normalmente luce como sigue:

Via: SIP/2.0/UDP workstation1234.company.com

---

[11] PGP: *pretty good privacy*, protocolo de cifrado creado por Philip Zimmermann en 1991

[12] Imagen de (McGraw-Hill Companies, 2004) [5]

La siguiente figura muestra un mensaje SIP con estos campos:

Figura 11 Ejemplo de mensaje SIP[13]

[13] Imagen de (McGraw-Hill Companies, 2004) [5]

CAPÍTULO III. MARCO METODOLÓGICO

3.1 Selección del simulador

En los preliminares a esta investigación, se propuso la utilización del simulador de software libre NS3. Este programa es de distribución gratuita, de libre modificación por parte de sus colaboradores y se encuentra disponible en internet para su descarga.

Luego de haber realizado el proceso de adecuación de la computadora (instalación de Linux Ubuntu y la actualización del sistema) y de la instalación del programa, se reparó en la complejidad de uso del mismo. No existen manuales que expliquen bien los procedimientos para realizar las simulaciones, la interfaz del programa no es amigable y para modificar los comandos, es necesario familiarizarse enteramente con el mismo, ocasionando una inversión temporal más allá del tiempo estipulado para la investigación. Es por estas razones que se decidió estudiar la posibilidad de utilizar otro software, y se llegó a la necesidad de comparar dos softwares disponibles, el OPNET Modeler® y el AweSim!

OPNET Modeler®: es un software desarrollado y distribuido por la empresa estadounidense OPNET. Según su página de internet, el Modeler, es un programa que puede ser usado para el análisis y el diseño de redes de comunicación, dispositivos, protocolos y aplicaciones. Los usuarios pueden analizar las redes simuladas para comparar el impacto de los diseños de diferentes tecnologías de punto a punto. Este programa, está disponible gratuitamente en internet, pero la persona interesada, debe solicitar la licencia a la compañía comprobando su estatus de investigador o estudiante universitario. Entre los protocolos disponibles para la simulación, se encuentran el SIP y el UDP, mas no el DTLS, y por ser un software propietario, no permite la modificación del código fuente para adaptarlo a la investigación. Es por tanto, incompatible con la investigación y fue descartado.

AweSim!: es un simulador desarrollado por el investigador Alan Pristker y un grupo de colaboradores, basándose en los principios básicos de la simulación discreta y continua combinadas, potenciadas con la interacción de diferentes procesos. Es una distribución que tiene licencia gratuita disponible para estudiantes e investigadores, presenta una interfaz gráfica bastante intuitiva y fácil de manejar. El menú de ayuda es bastante completo y simplifica un poco el uso del simulador. También presenta una gran versatilidad de opciones de simulación, ofreciendo simulación para una gran gama de situaciones y procesos. El Instituto Superior Técnico de la Universidad Técnica de Lisboa colocó a disposición el siguiente link para descargar la versión 3.0 "http://courses.tagus.ist.utl.pt/deg/si0/web/software/" (la información está disponible sólo en portugués).

Gracias a estas cualidades, se escogió este software para llevar a cabo las simulaciones para la presente investigación.

3.2 Descripción del modelo

3.2.1 Preliminares

Para la investigación, se decidió modelar el proceso de establecimiento de una llamada SIP, protegido con DTLS y transportada en datagramas UDP así como también la medición del tiempo de establecimiento de la llamada. Para ello se tomaron las siguientes suposiciones:

- La distribución de llaves para el cifrado, fue realizada previamente de manera segura.
- Las partes involucradas ya han sido registradas ante el proxy SIP.
- Sólo se estudiará el establecimiento efectivo de la comunicación.
- Se estudiarán sólo dos escenarios, con seguridad DTLS y sin seguridad.
- El tiempo de respuesta a medir es el tiempo comprendido entre el envío del primer paquete INVITE y la recepción del mensaje ACK, ya que este mensaje marca el inicio de la transmisión de datos.

La siguiente figura muestra los mensajes intercambiados durante una llamada SIP:

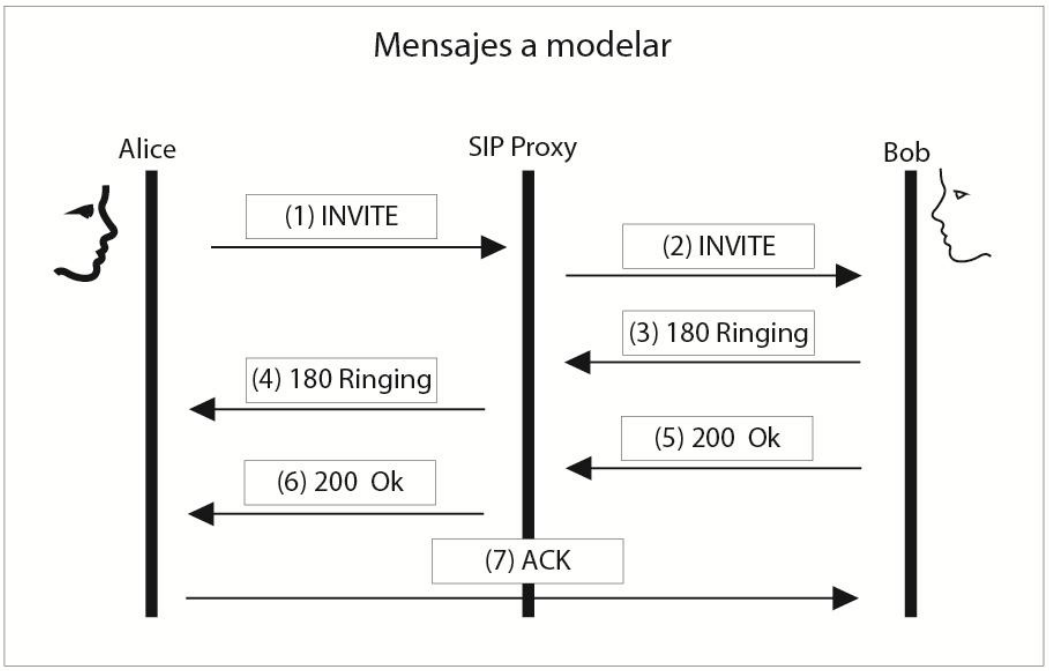


Figura 12 Mensajes a modelar[14]

Se pueden apreciar dos partes interesadas en comunicarse (Alice y Bob). A través de sus Agentes de Usuario, inician una comunicación con la ayuda de un servidor proxy al cual han sido registrados previamente. Los mensajes de señalización que interesan para esta investigación son los cuatro primeros mostrados en la figura: INVITE, RINGING, OK y ACK. Los tres primeros mensajes son procesados por el servidor proxy para establecer un canal de comunicación, ya el mensaje de ACK es enrutado directamente sin procesar debido a que ya se ha establecido el canal.

[14] Imagen adaptada de (McGraw-Hill Companies, 2004) [5]

Cada uno de los mensajes descritos, deben ser procesados por el remitente, el servidor y el receptor. Los procesos son muy similares tanto sin seguridad como con seguridad. Las siguientes figuras muestran los procesos realizados en ambos casos:

Sin seguridad:

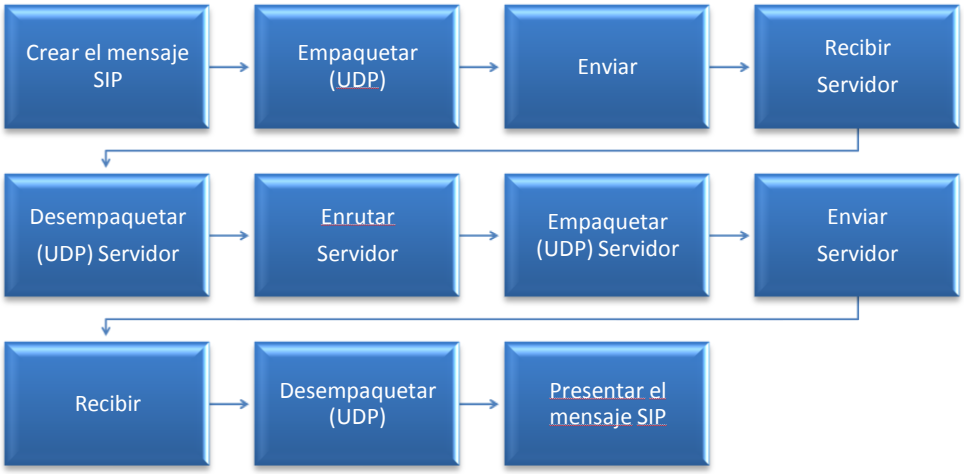


Figura 13 Procesos a modelar sin seguridad

El proceso comienza con la generación del mensaje SIP, que es empaquetado en la trama UDP para luego ser enviado a un servidor Proxy a través de un canal dado. Este servidor proxy toma la trama UDP, desempaqueta el mensaje, verifica la ruta a seguir, lo empaqueta de nuevo en la trama UDP y lo reenvía al destinatario. Finalmente el AU del destinatario toma el mensaje SIP de la trama UDP y lo presenta.

Un proceso similar se realiza al enviar un mensaje cifrado con el protocolo DTLS, se genera el mensaje SIP, se cifra con el DTLS, luego se empaqueta en la trama UDP para luego ser enviado a un servidor Proxy a través de un canal dado. Este servidor proxy toma la trama UDP, desempaqueta el mensaje, descifra el mensaje con el protocolo DTLS, verifica la ruta a seguir, lo cifra nuevamente con DTLS, lo empaqueta en la trama UDP y lo reenvía al destinatario. Finalmente el AU del destinatario toma el mensaje SIP de la trama UDP, lo descifra y lo presenta. La siguiente figura muestra el proceso de manera esquemática.

Con seguridad:

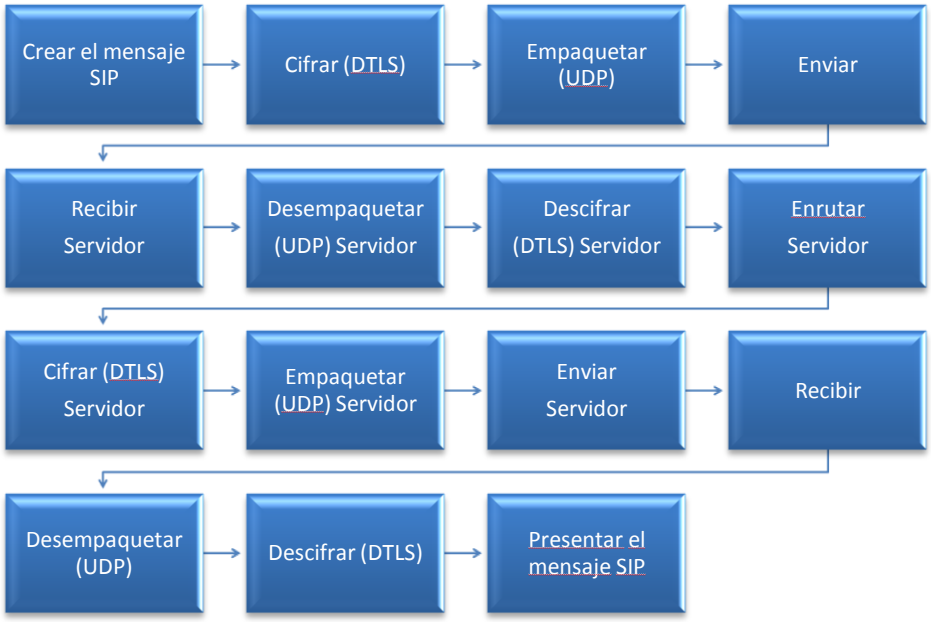


Figura 14 Procesos a modelar con seguridad

De acuerdo con los esquemas anteriores y para realizar un modelo de dichos procesos, se puede modelar el cifrado y descifrado como un retardo agregado a cada actividad.

Según los resultados obtenidos por (Modadugu & Rescorla, 2004) [6] al desarrollar uno de los experimentos más importantes con el protocolo DTLS, se determinó que la latencia del protocolo es de 42,9ms. Al no indicar la manera con la

que se calculó esa latencia, se asumió que la misma es *one-way* y se le adicionó al tiempo de procesamiento de cada mensaje.

3.2.1.1 Estudio previo

Antes de poder modelar los procesos descritos en el apartado anterior, fue necesario determinar el tiempo de establecimiento real del modelo sin seguridad y utilizar este dato como semilla para la simulación de cada proceso. Esto se realizó con la finalidad de validar los datos arrojados por el simulador. Para obtener este resultado, se desarrolló un experimento en donde se realizaron 38 llamadas SIP entre dos softphone y se observaron los paquetes enviados a través del analizador de protocolos Wireshark™.

El experimento implicó la creación de dos direcciones SIP registradas en un servidor SIP, la instalación y configuración del softphone en dos computadores conectados a internet a través de redes distintas. El servidor usado fue el ekiga.net[15] y el softphone, el Express Talk[16]. A través del analizador de protocolos, se midió el tiempo invertido en el establecimiento de la conexión y mediante la utilización de un programa de manejo de hojas de cálculo, se construyeron las siguientes tablas de datos y figuras:

[15] Disponible en www.ekiga.net (visitado el 1 de mayo de 2011)

[16] Disponible en www.nch.com.au/talk/index.html (visitado el 1 de mayo de 2011)

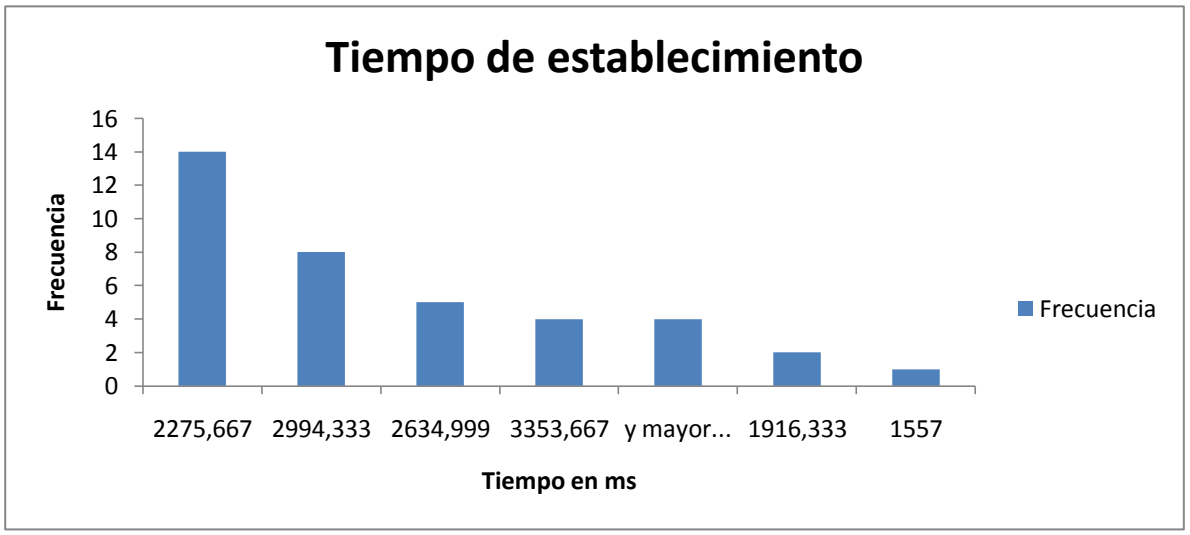


Figura 15 Histograma de tiempo de establecimiento llamada SIP sin seguridad

Del histograma se puede concluir que el tiempo más reportado está cerca de los 2275,667 ms. Los datos estadísticos se muestran en la siguiente tabla:

Tabla 3 Resumen estadístico para llamada SIP sin seguridad

	Tiempo (milisegundos)
Promedio	2.509,184
Desviación estándar	583,175
Varianza	340.093,080
Mínimo	1.917
Máximo	3.713

De los resultados estadísticos se destacan el promedio de 2509,184 ms y la desviación estándar de 583,175 ms.

Surgió, inmediatamente, la necesidad de caracterizar los datos recogidos a través de una función de distribución estadística para tener un modelo más coherente con los resultados experimentales. Para tanto, se usó el modelo sin seguridad y

diversas distribuciones aleatorias para determinar la más acertada en lo que concierne a la descripción del tiempo de establecimiento.

Se usaron distribuciones aleatorias a lo largo del modelo, de manera tal que los valores estadísticos del tiempo de establecimiento se acercaran a los datos de la tabla 3. Luego se realizaron 38 corridas para verificar la similitud en los campos de promedio y desviación estándar.

Las distribuciones usadas son: exponencial, normal, de Poisson y triangular. Es importante recalcar que debido a que la distribución normal puede tomar valores negativos (y no existen tiempos negativos), se tomó el valor absoluto de la misma, lo que arroja unos valores aceptables aún cuando no representa una distribución normal propiamente dicha.

La siguiente tabla recoge los datos colectados en dichas simulaciones:

Tabla 4 Distribuciones estadísticas para el modelo de simulación

	Real	Exponencial	Error (%)	Normal	Error (%)	Poisson	Error (%)	Triangular	Error (%)
Promedio	2509,184	2473,837	1,409	2627,928	4,732	2509,719	0,021	2640,006	5,214
Desviación Estándar	583,175	614,863	5,434	342,555	41,260	51,179	91,224	96,997	83,367

De los datos presentados, se puede verificar que la distribución que guarda una menor diferencia porcentual, es la distribución exponencial. Por tanto, se escogió la misma como la distribución más apropiada para la representación del tiempo de establecimiento en el simulador.

Asimismo se puede inferir, que como dicha función describe de una manera aceptable los procesos del sistema sin seguridad, también describirá el sistema protegido con DTLS.

3.2.2 Modelo adaptado al simulador

El simulador procesa ciertos elementos llamados entidades de tal manera que hacen las veces de informaciones o mensajes en tránsito. Un investigador podría modelar casi cualquier proceso usando una entidad y modificando sus atributos para hacer ver que un proceso ha sido realizado.

Dichas entidades son generadas por la función "CREATE", ésta solicita la frecuencia con que se generarán las entidades y la cantidad a generar. La cantidad escogida para esta investigación fue de 7520 entidades, que representan 80 mensajes de 94 segmentos. Se eligió éste número bajo las premisas de que el protocolo DTLS cifra bloques de 128 bits y el tamaño del paquete viene dado por la MTU del sistema que en este caso fue escogido de 1500 bytes. Por tanto, 1500 bytes=12mil bits implicando 94 bloques de 128 bits, en otras palabras:

$$1500\ bytes * 8 = \frac{12mil\ bits}{128\ bits} = 94\ bloques/mensaje$$

Las entidades "transitan" de un nodo a otro mediante una "ACTIVITY", las actividades pueden representar simples conectores o generadores de retraso dependiendo de las opciones que se seleccionen.

Puede ser necesario modelar la segmentación y/o el agrupamiento de algún elemento, esto se logra a través de los nodos "UNBATCH" y "BATCH" respectivamente.

Para modificar los atributos de una entidad y modelar un cambio realizado por algún proceso, se requiere el uso del nodo "ASSIGN"; éste almacena un valor dado en la variable escogida

También puede ser necesario modelar un encolamiento, esto se logra con el nodo "QUEUE", al cual se le puede asignar un valor de elementos en cola al comienzo de la simulación, el número de elementos que puede encolar y el archivo donde se guardará la información instantánea de encolamiento.

Ya para colectar los datos procesados, se utiliza el nodo “COLCT”, éste nodo puede guardar información de las variables del sistema. Si es necesario hacer referencia a otro proceso o punto en el simulador sin que ello implique la realización de una actividad, se puede usar el nodo “GOON”. Finalmente, para indicarle el término de la simulación al simulador, se usa el nodo “TERMIN”.

Para conocer el funcionamiento de otros nodos ver el Manual de Referencia Rápida (Pritsker Corporation, 1997) [7] o consulte la ayuda del simulador.

La siguiente tabla resume los nodos usados para la simulación:

Tabla 5 Nodos usados en la simulación (Fuente: Autor)

Nodo	Uso en la simulación
Activity 	Para transiciones y retrasos
Unbatch 	Para simular fragmentación y desempaquetado
Colct 	Para extraer los datos de la variable de interés
Create 	Para la generación de entidades
Goon 	Como nodo ficticio
Queue 	Para modelar la espera previa a un proceso
Batch 	Para simular el desfragmentado y el empaquetado
Termin 	Para marcar el final de la simulación

Para las simulaciones, se usaron distribuciones retrasos pseudoaleatorios (usando la función DRAND()) para modelar el procesamiento interno de los computadores involucrados, del orden de los microsegundos, de acuerdo con (Tanenbaum, 2003)[9]. Ya para marcar el tiempo de establecimiento, se usó la distribución exponencial ofrecida por el simulador, como bien se comentó anteriormente. La siguiente tabla muestra las distribuciones usadas y una breve descripción de cada una:

Tabla 6 Distribuciones aleatorias utilizadas

Distribución	Descripción
DRAND(IS)	Describe una distribución aleatoria entre cero y uno, usando como semilla el término "IS" generado por la función generadora de números aleatorios.
EXPON(p,x)	Describe una distribución exponencial con los valores: promedio (p), usando como semilla el término "x" generado por la función generadora de números aleatorios

3.2.2.1 Modelo sin seguridad

El modelo propuesto es el mostrado en la siguiente figura:

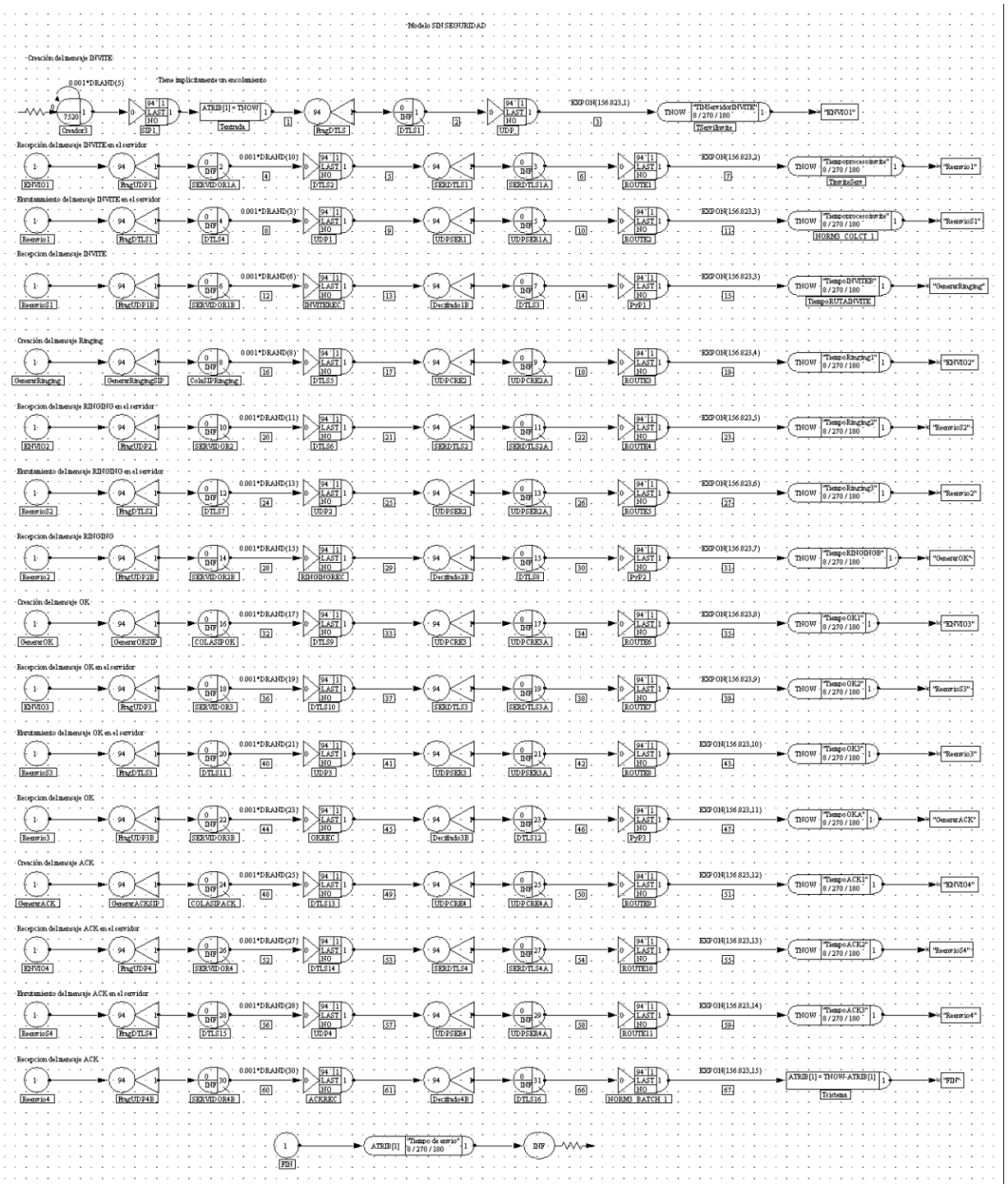


Figura 16 Modelo sin seguridad[17]

[17] Para ver una imagen mayor, ver anexo A

Este modelo cumple con la estructura de la figura 13 para la transmisión de cada mensaje, a continuación se realiza una explicación detallada del mismo.

Se segmentarán los modelos en grupos de cuatro líneas de figuras, donde la primera línea representa el proceso de uno de los terminales, la segunda representa el proceso de envío al servidor y el proceso del servidor, la tercera línea representa el reenvío al receptor, por último, la cuarta línea se representa el procesamiento de recepción del otro terminal.

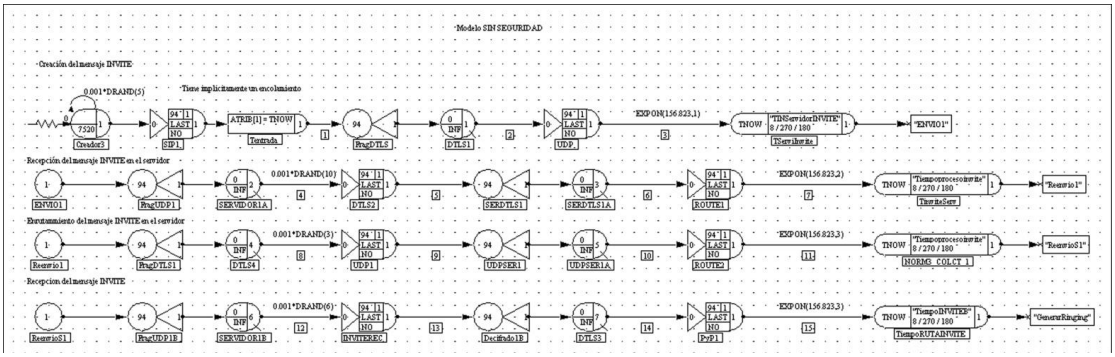


Figura 17 Procesamiento del mensaje INVITE (Sin seguridad)[18]

Se modeló la creación del mensaje INVITE, luego el empaquetado UDP, el procesamiento del servidor proxy, el empaquetado UDP post procesamiento, el desempaquetado y la presentación del mensaje. Para simular el retraso total por esta etapa, se colocó un retraso de acuerdo a la distribución exponencial y a la tabla 4.

Para las siguientes etapas, se repite el mismo proceso y se asumieron los datos para alimentar la distribución aleatoria exponencial que representa el retraso por cada segmento.

[18] Para ver una imagen mayor, ver anexo A

La siguiente figura, muestra el proceso de generación, enrutamiento y recepción del mensaje RINGING.

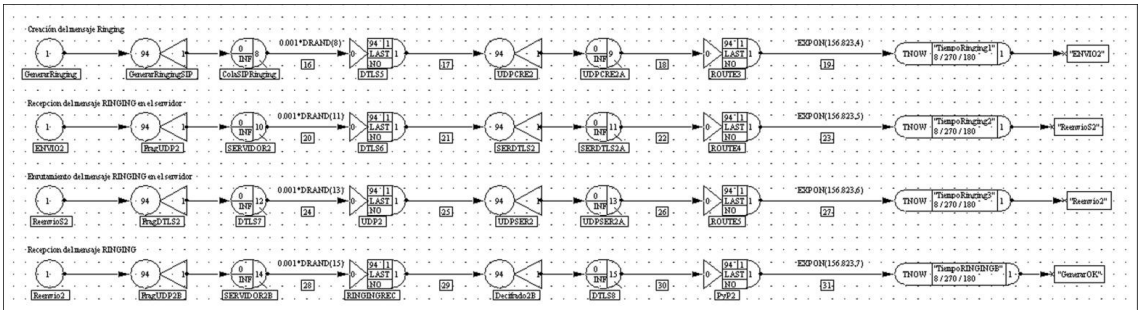


Figura 18 Procesamiento del mensaje RINGING (Sin seguridad)[19]

Como ya se indicó, el proceso es muy similar al ya descrito y para simular la creación del mensaje RINGING, se insertó una distribución aleatoria con promedio y desviación estándar por mensaje descritos anteriormente en la tabla 4.

Luego se repite el proceso para la creación, el enrutamiento y la recepción del mensaje OK. Se asumió que la contestación de la llamada es automática e inmediata.

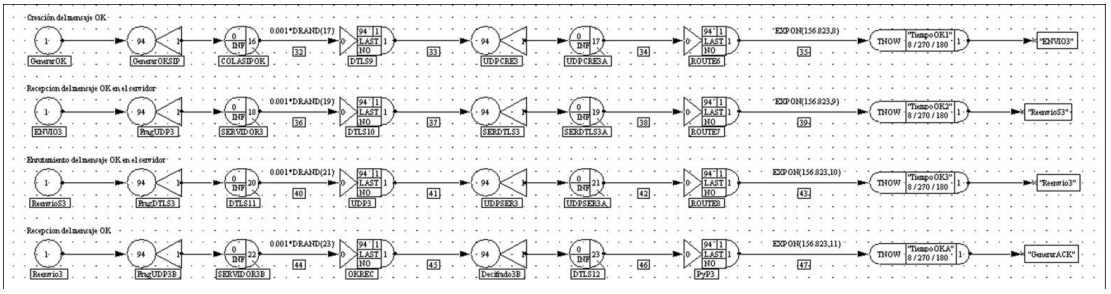


Figura 19 Procesamiento del mensaje OK (Sin seguridad)[20]

[19] Para ver una imagen mayor, ver anexo A

[20] Para ver una imagen mayor, ver anexo A

Finalmente, se modela el proceso de Procesamiento del mensaje ACK. Se resume el tiempo de retraso insertado por todos estos procesos, mediante el uso de la misma distribución aleatoria exponencial descrita anteriormente. Inmediatamente después de la simulación de la recepción del mensaje ACK, se procede a medir el tiempo y es este el utilizado como tiempo de establecimiento del sistema. Por último se termina la simulación usando el nodo TERMIN.

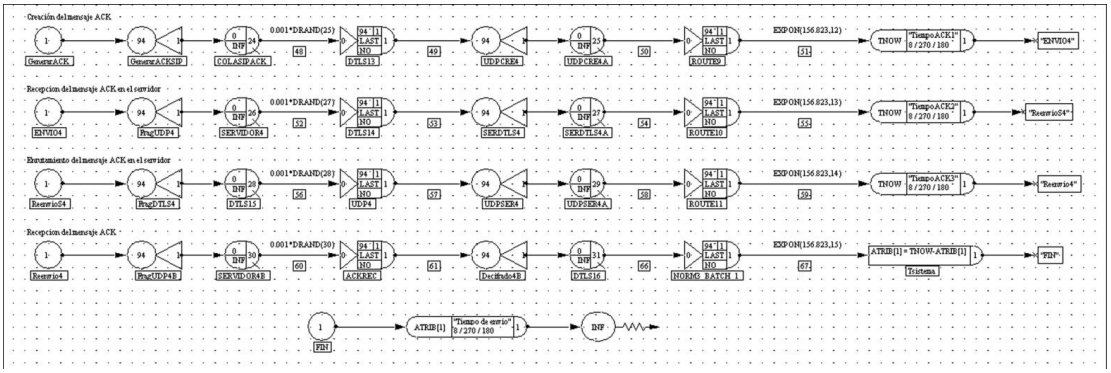


Figura 20 Procesamiento del mensaje ACK (Sin seguridad)[21]

Este proceso se repitió 38 veces y se obtuvieron los datos estadísticos recogidos en las tablas 8 y 9. (El resumen generado por el simulador se incluye en los anexos).

[21] Para ver una imagen mayor, ver anexo A

3.2.2.2 Modelo con seguridad

Asumiendo que el sistema se comporta de manera similar (sin seguridad y con ella) debido a que la diferencia principal entre ambos escenarios, es la presencia de un retraso mayor, se tomó como base el modelo sin seguridad. Al igual que en el dicho modelo, la distribución aleatoria usada para describir el retraso fue la exponencial, con un tiempo de establecimiento promedio de acuerdo con la tabla 4. Y también se cumple la estructura planteada en la figura 14

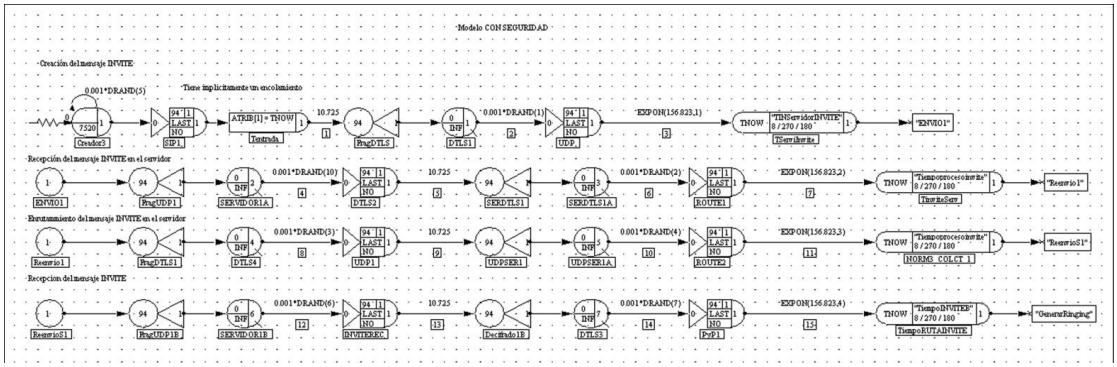


Figura 21 Procesamiento del mensaje INVITE (Con seguridad)[22]

Al igual que en el escenario sin seguridad, se modeló el retraso por creación, cifrado DTLS, empaquetado UDP, envío al servidor, desempaquetado UDP, descifrado DTLS, enrutamiento, cifrado DTLS, empaquetado UDP, reenvío, desempaquetado UDP, descifrado DTLS y recepción de cada mensaje como un retraso insertado al final del último proceso (de acuerdo con la figura 14). Este retraso se insertó de acuerdo con la tabla 4, estableciéndose el tiempo de retraso promedio por mensaje en 670,196ms y modelando una distribución estándar de 145,794ms para mantener la proporcionalidad.

[22] Para ver una imagen mayor, ver anexo A

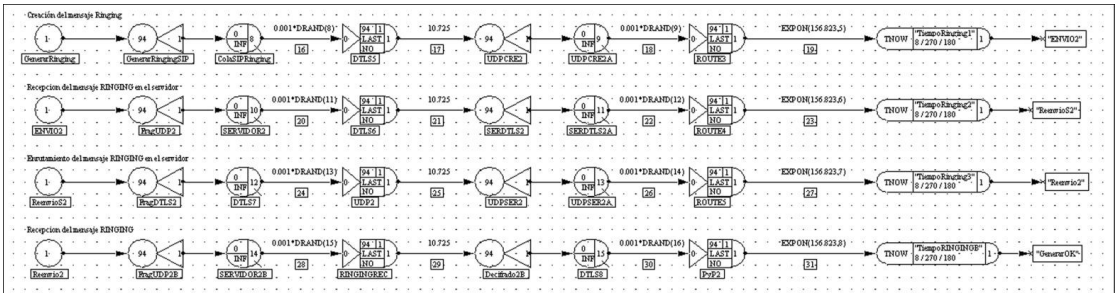


Figura 22 Procesamiento del mensaje RINGING (Con seguridad)[23]

Se repite el proceso para la creación del mensaje RINGING.

Posteriormente, se crea, se procesa y se envía el mensaje OK para indicar que la llamada fue contestada. La siguiente figura, muestra dicho proceso:

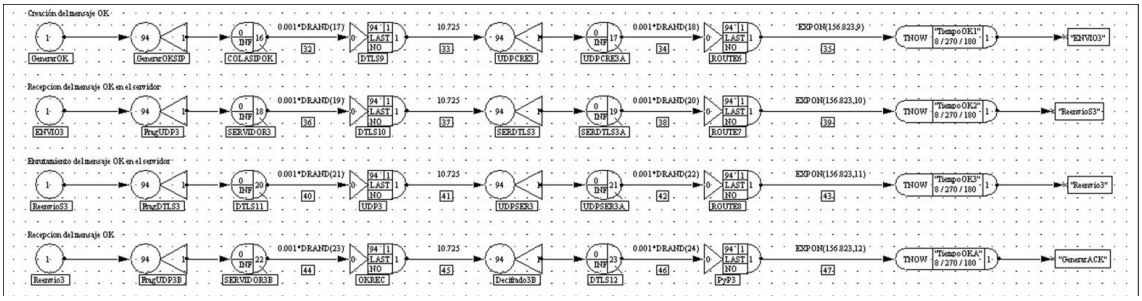


Figura 23 Procesamiento del mensaje OK (Con seguridad)[24]

[23] Para ver una imagen mayor, ver anexo A

[24] Para ver una imagen mayor, ver anexo A

Finalmente, para cerrar el proceso y medir el tiempo de establecimiento del sistema, se modeló el proceso de la figura 14 para el mensaje de ACK como se muestra en la figura 24:

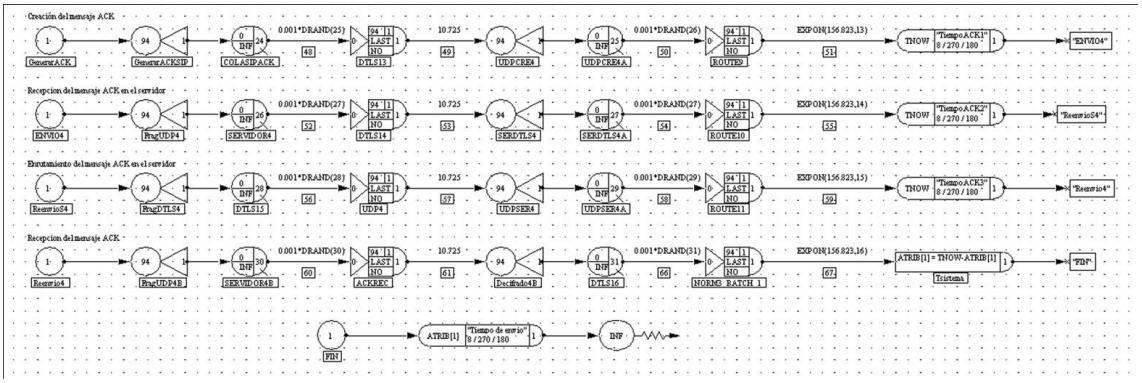


Figura 24 Procesamiento del mensaje ACK (Con seguridad)[25]

Luego, se repitió este proceso 38 veces y se obtuvieron los datos estadísticos para construir las tablas 8 y 9. (El resumen generado por el simulador se incluye en los anexos)

[25] Para ver una imagen mayor, ver anexo A

El modelo general para el sistema con seguridad es el siguiente:

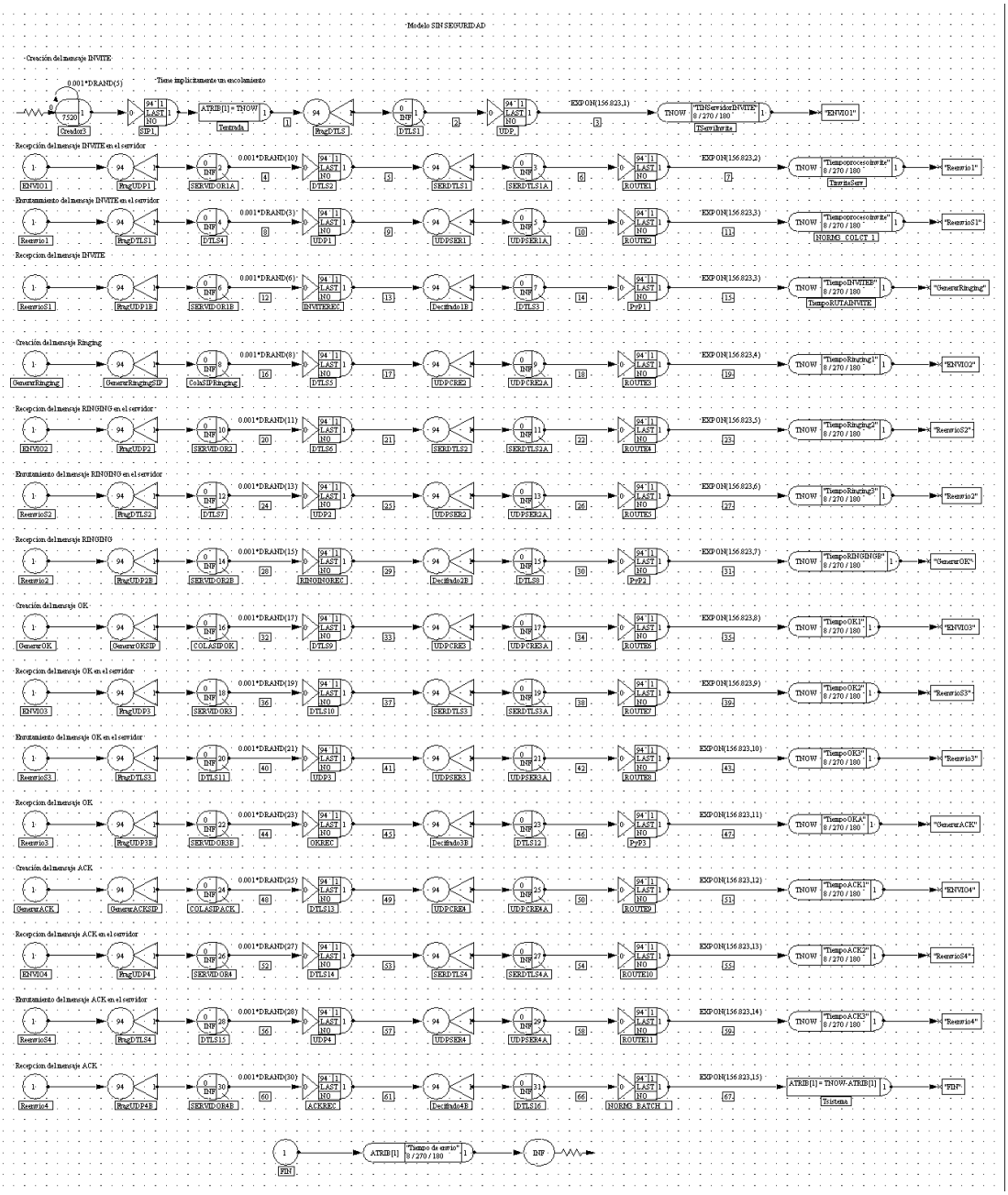


Figura 25 Modelo con seguridad[26]

[26] Para ver una imagen mayor, ver anexo A

Haciendo un análisis de los datos reales colectados y haciendo una estimativa del tiempo de retraso agregado por el cifrado DTLS, se tiene la siguiente tabla:

Tabla 7 Tiempo de transmisión

		Tiempo de transmisión por mensaje (ms)	
	Tiempo de establecimiento (ms)	**Sin seguridad**	**Con seguridad (Estimado)**
Máximo	3713,000	928,250	-
Mínimo	1917,000	479,250	-
Promedio	2509,184	627,296	670,196*
Desviación estándar	583,175	145,794	-

*Para estimar el retraso del proceso con seguridad, se agregó el tiempo de 42,9ms al promedio por mensaje. Esta adición se realizó de manera proporcional en cada etapa del proceso y se tomó como un retraso determinístico debido a la característica de cifrado por bloques del DTLS. Con esta información se pudo alimentar las distribuciones aleatorias del simulador para estudiar un modelo sin seguridad más ajustado a la realidad.

3.2.3 Configuración del archivo de control

Para realizar las simulaciones, fue necesario configurar los datos de control del simulador a través de la aplicación *Control Builder* ofrecida por AweSim! La siguiente figura muestra un ejemplo de dicha configuración.

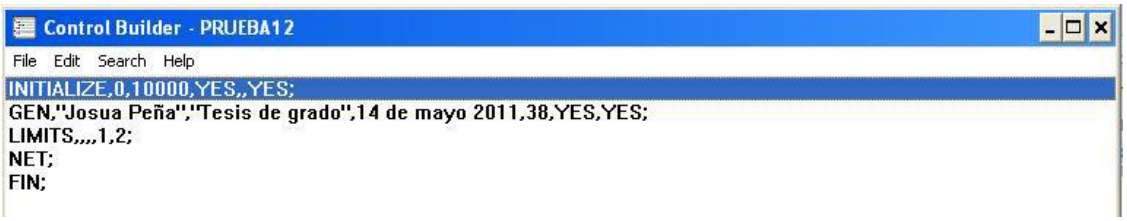


Figura 26 Archivo de control

El campo INITIALIZE utilizado para indicar el tiempo máximo de simulación por corrida. Este tiempo es definido por el investigador y si los procesos simulados terminan antes de finalizar este tiempo, la simulación acaba automáticamente.

El campo GEN es usado para colocar los datos del investigador, los datos del proyecto, la fecha en que fue realizado y el número de simulaciones a realizar. Se muestra el número de 38 simulaciones efectuadas por cada escenario.

El campo LIMITS fue usado para indicar el número de elementos presentes en los vectores globales del sistema. Se usó la variable global Atrib[] en su formato de vector fila con dos elementos para almacenar datos de modificación de atributos y medir el tiempo de establecimiento. Al comenzar la simulación, se guardó el tiempo de generación del primer paquete y al final del proceso, se restó del tiempo actual del sistema al recibir el mensaje de ACK (TNOW para el simulador). En este campo, se puede indicar el número máximo de entidades circulando al mismo tiempo en el sistema. Como se utilizó la versión estudiantil del AweSim!, dicho número, aunque modificable, está limitado a 300 entidades.

El campo NET se puede usar para leer la descripción de la red concatenada al escenario actual, para fines de la investigación, se usaron los valores por defecto puesto que se usó el mismo programa de control para cada escenario.

El campo FIN se utilizó para marcar la culminación de la simulación.

Para ambos escenarios se usó el mismo programa de control puesto que las informaciones dispuestas en él, no variaban entre los escenarios.

CAPÍTULO IV ANÁLISIS DE RESULTADOS

De acuerdo con el teorema de la Probabilidad Aproximada de Selección Correcta (APCS, en inglés), demostrado por (Chun-Hung Chen, Yücesan, & Chick, 2000) [2] que reza:

"Theorem 1 Given a total number of simulation samples T to be allocated to k competing designs whose performance is depicted by random variables with means $J(\theta_1), J(\theta_2), \dots, J(\theta_k)$, and finites variances $\sigma_1^2, \sigma_2^2, \dots, \sigma_k^2$respectively, as $T \to \infty$, the Approximate Probability of Correct Selection (APCS) can be asymptotically maximized when

$$(1)\frac{N_i}{N_j} = \left(\frac{\frac{\sigma_i}{\delta_{b,i}}}{\frac{\sigma_j}{\delta_{b,j}}}\right)^2, i, j \in \{1,2, \dots, k\}, \text{and } i \neq j \neq b$$

$$(2)N_b = \sigma_b \sqrt{\sum_{i=1, i\neq b}^{k} \frac{N_i^2}{\sigma_i^2}},$$

where N_i is the number of samples allocated to design i, $\delta_{b,i} = \bar{J}_b - \bar{J}_i$, and $\bar{J}_b \leq \min_i \bar{J}_i$."

Figura 27 Teorema de la Probabilidad Aproximada de Selección Correcta (APCS)

El teorema indica que hay una relación entre el número de simulaciones y la varianza calculada a partir de los datos recogidos en dichas simulaciones.

Dependiendo de la cantidad de simulaciones y de escenarios diferentes, se puede determinar el número óptimo de simulaciones al aplicar las ecuaciones 1 y 2.

Para el caso particular en que k=2 y b=1, el teorema queda $\frac{N_1}{N_2} = \frac{\sigma_1}{\sigma_2}$. Donde se puede observar que hay una relación de proporcionalidad directa entre el cociente del número de simulaciones y el cociente de las desviaciones estándar de cada grupo de simulaciones.

Ésta fue la regla usada para verificar la relación entre el número de muestras tomadas en cada escenario.

Tabla 8 Escogencia del número de simulaciones

	Desviación Estándar	Número de simulaciones
Escenario 1 (Sin seguridad)	614,863	38
Escenario 2 (Con seguridad)	610,295	38

Luego de acuerdo con el teorema APCS:

$$\frac{N_1}{N_2} = \frac{\sigma_1}{\sigma_2} \rightarrow \frac{614{,}863}{610{,}295} = 1{,}007 \cong \frac{38}{38}$$

Los resultados obtenidos por escenarios, son los siguientes:

Tabla 9 Resultados estadísticos por escenario

Tiempo de establecimiento (en ms)	**Escenario**		
	Sin seguridad	Con seguridad	Diferencia
Promedio	2473,837	2651,355	177,518
Máximo	3515,794	4329,147	813,353
Mínimo	1232,417	1431,531	199,114
Desviación Estándar	614,863	610,295	-

Como se podía esperar al incluir más procesamiento sobre los mensajes transmitidos, el tiempo promedio del escenario con seguridad es mayor al escenario sin seguridad. Aunque en esencia ambos escenarios representan lo mismo, una transmisión de mensajes, operacionalmente diferentes, esto imposibilita hacer conclusiones sobre la diferencia temporal entre ellos. Es por ello que se hace

necesario utilizar un instrumento que vincule la presencia de mecanismos de seguridad y el tiempo de establecimiento.

En este sentido, se tomaron las siguientes premisas para construir una función matemática para el estudio del desempeño de un sistema:

- A mayor tiempo de establecimiento, menor el desempeño del sistema
- A medida en que se adicionen características de seguridad[27], el desempeño es mayor
- Cada característica de seguridad tiene el mismo peso
- A pesar de que el desempeño del sistema desmejore al aumentar el tiempo de establecimiento, mientras más mecanismos de seguridad estén presentes, el desempeño es mayor.

Por tanto, la función D de desempeño es la siguiente:

$$D(t, A, C, I) = A + C + I - \log(t)$$

donde $A, C\ e\ I \in \{0,1\}$ y t>0

A: marca la presencia de autenticidad

C: marca la presencia de confiabilidad

I: marca la presencia de integridad

t: es el tiempo de establecimiento expresado en segundos

Es importante resaltar la escogencia de la función logaritmo, para mostrar una diferencia apreciable entre las diferentes curvas, se necesitaba una función que decreciera monótonamente. Se estudió el uso de una función inversamente proporcional (f(x)=1/x) pero esta función decrece muy rápidamente y no facilita la

[27] En este estudio se tomaron como características de seguridad sólo la Autenticidad, la Confiabilidad y la Integridad.

comparación entre las diferentes curvas. Luego la función f(x)= -log(x) ofrece un decrecimiento más lento y la adición de una constante (f(x)= M – log(x), con M constante) simplifica la observación de varias curvas en la misma gráfica. Es por ello que se realizó la selección de la mencionada función logarítmica.

La representación de dicha función para la presencia de uno, dos y tres parámetros de seguridad es la siguiente:

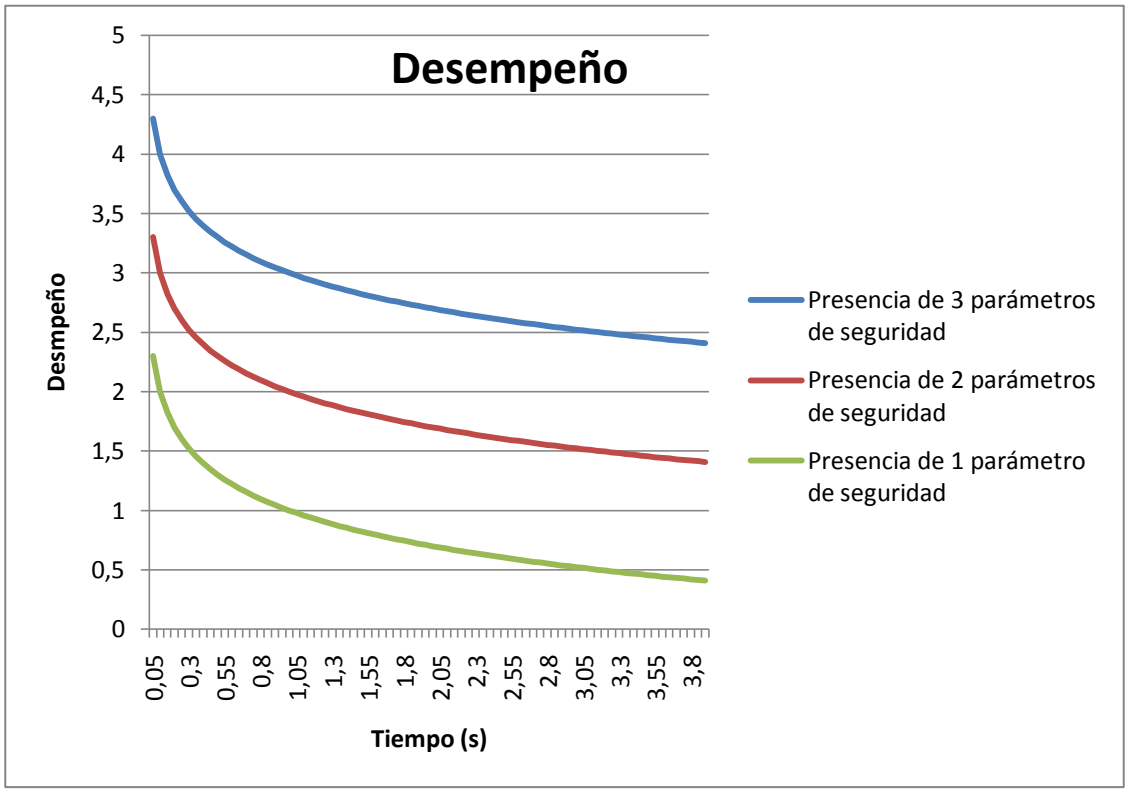


Figura 28 Función de desempeño (Fuente: Autor)

Es importante recalcar las siguientes afirmaciones:

- El sistema sin seguridad está provisto de autenticidad y confiabilidad para el establecimiento de la conexión, mediante la autenticación realizada por el protocolo SIP al registrarse ante el servidor proxy y los mecanismos propios de retransmisión.

- El sistema seguro está provisto de autenticidad, confiabilidad e integridad durante el establecimiento de la comunicación, ya que la autenticidad la provee el protocolo SIP, la confiabilidad y la integridad las provee el DTLS.
- Por la naturaleza del proceso estudiado, comunicación en tiempo real, usualmente no se provee confiabilidad para el tráfico de voz. Esto debido a que se perdería tiempo pidiendo la retransmisión de los paquetes faltantes y, durante la llamada, se podría insertar un segmento de conversación anterior entorpeciendo la comunicación.

Al buscar en la figura 28 el tiempo promedio de establecimiento para ambos escenarios, el escenario sin seguridad, posee dos parámetros de seguridad por lo que su desempeño se observa en la curva 2 (roja). Ya el escenario con seguridad, posee tres parámetros y se mide en la curva 3 (azul).

Por tanto, se puede inferir que un sistema que utilice la estructura SIP-DTLS-UDP, aunque demore más tiempo en establecer la conexión, tendrá un mejor desempeño que otro con estructura SIP-UDP.

CONCLUSIONES

Durante el proceso investigativo, se realizó una recolección de información relacionada a la estructura de tres protocolos de comunicación, SIP, DTLS y UDP.

SIP es un protocolo de capa aplicativa usado para la señalización previa al establecimiento de una llamada VoIP. Es un protocolo que provee autenticidad y confiabilidad más no provee integridad.

DTLS es un protocolo de seguridad en la capa de transporte orientado a datagramas y se basa en la modificación del protocolo Transport Layer Security para simplificar su adaptación a sistemas que previamente usaban TLS. Es un protocolo que provee integridad, autenticidad y puede proveer confiabilidad a través de un sistema sencillo de reloj para retransmisión de paquetes de señalización.

UDP es el protocolo de transporte de datagramas consagrado por el uso en sistemas de comunicación. Es un protocolo que empaqueta los mensajes en datagramas que son enviados de un segmento a otro sin confirmación de su recepción.

Teniendo en cuenta estas informaciones, se estudió una estructura que combinara SIP-UDP y otra que combinara SIP-DTLS-UDP.

Con la primera estructura, se recolectaron datos reales a través del desarrollo de un experimento en donde se midió el tiempo real promedio para el establecimiento de una llamada VoIP. Con los datos recolectados, se procedió a desarrollar un modelo de simulación usando un software propio para ello. Se estudió la posibilidad de usar el software libre NS3, el propietario OPNET Modeler® y el propietario pero de distribución gratuita para estudiantes, AweSim! El NS3 resultó ser un programa difícil de usar y de programar, con poca literatura disponible y que requería mucho tiempo de estudio para ser explotado.

El OPNET Modeler® resultó ser más fácil de usar pero difícil de obtener e imposible de modificar para que cumpliera con las necesidades de la investigación. Es necesario pagar una licencia para su utilización, también hay una versión gratuita para estudiantes pero al no permitir la modificación del kernel, resultó inútil para el presente trabajo.

Finalmente se optó por el AweSim! debido a que permite modelar, virtualmente, todo tipo de sistema. A pesar de sus características más agradables y de su disponibilidad, el software requiere una buena familiarización para su uso. Tampoco existe una literatura de fácil acceso ni en español, lo que implica una búsqueda amplia y tardía, apartando el hecho de tener buenas habilidades en lengua inglesa.

Luego de haber seleccionado el simulador y de tener los datos estadísticos reales, se desarrolló un modelo de simulación acorde con los datos recolectados. Se establecimiento la distribución exponencial como la distribución que describe, de manera sencilla y con muy buena aproximación, el proceso de una llamada SIP real.

Después de tener la estructura entera, se procedió a proponer las condiciones con las que se estudiaría el segundo sistema. Se utilizó la estructura SIP-UDP como base y se infirió que el tiempo invertido para el cifrado de los mensajes, se podría modelar como un retraso más en el proceso de establecimiento de la llamada.

Posteriormente, para poder comparar los resultados obtenidos, se construyó una función matemática que relacionara el tiempo de establecimiento con el nivel de seguridad de un sistema.

Debido a que el modelo propuesto para el sistema sin seguridad se pudo validar con los datos reales y se usó como base para el segundo modelo, se puede

inferir que la estructura segura modelada, también está validada y, la estructura real, cumplirá con los resultados aquí expuestos.

Finalmente, a través de la mencionada función de desempeño, se pudo inferir que un sistema que utilice la estructura SIP-DTLS-UDP, aunque demore más tiempo en establecer la conexión, tendrá un mejor desempeño que otro con estructura SIP-UDP.

RECOMENDACIONES

A continuación una serie de aspectos que deberían ser considerados y que complementan la investigación:

Debido a la poca información, relacionada al uso del AweSim!, disponible en la lengua castellana, se recomienda la traducción de textos y publicaciones, para simplificar la búsqueda y las venideras investigaciones.

Se recomienda la experimentación para complementar la validación de los datos recogidos en esta investigación a través de la implementación de la estructura SIP-DTLS-UDP.

Para explotar la capacidad del simulador AweSim! se recomienda la simulación de la estructura SIP-DTLS-UDP para la comunicación Multicast.

Se sugiere realizar un análisis exhaustivo y riguroso de los parámetros y distribuciones probabilísticas presentadas en el modelo.

Realizar pruebas ante situaciones de alto volumen de tráfico, para lo cual se necesitaría un simulador más robusto.

Desarrollar un análisis de sensibilidad del sistema con relación a sus parámetros característicos.

En lo que respecta a los diferentes modos de cifrado del DTLS, se debe estudiar el uso de alternativas al CBC y verificar el tiempo de establecimiento para llamadas IP.

Asimismo se debe estudiar el uso del protocolo DTLS para el cifrado del transporte de voz en llamadas IP

Debido a la falta de un curso de estadística en el pensum de estudio de la carrera, se recomienda incluir alguna materia electiva o algún curso extracurricular, para los estudiantes de la escuela, para que se simplifique el proceso investigativo en el desarrollo de trabajos de grado.

También, en este sentido, se recomienda adaptar un programa de iniciación científica como el establecido en la Universidad de Brasilia, en el cual estudiantes de los primeros semestres de la carrera, trabajan en un proyecto de la línea de investigación de los profesores de la escuela. Esto ayuda a ganar experiencia desarrollando proyectos similares al trabajo de grado.

.

REFERENCIAS BIBLIOGRÁFICAS

[1] BRITO, D. J., & MOURA DE SOUSA, G. (2006). *Análise de protocolos para prover confidencialidade e integridade em sessões de Telefonia IP*. Brasilia: Universidade de Brasília.

[2] Chun-Hung Chen, J. L., Yücesan, E., & Chick, S. E. (2000). Simulation Budget Allocation for further enhacing the efficiency of ordinal optimization. *Discrete Evente Dynamic Systems: Theory and Applications, 10* , 251-270.

[3] Internet Engineering Task Force. (Abril de 2006). *Datagram Transport Layer Security*. Recuperado el 10 de Enero de 2011, de RFC EDITOR: http://www.rfc-editor.org/rfc/rfc4347.txt

[4] Internet Engineering Task Force. (28 de Agosto de 1980). *User Datagram Protocol.* Recuperado el 15 de Octubre de 2010, de RFC EDITOR: http://tools.ietf.org/html/rfc768

[5] McGraw-Hill Companies. (2004). The Session Initiation Protocol: SIP. En *SIP Demystified* (págs. 89-157). McGraw-Hill Companies.

[6] Modadugu, N., & Rescorla, E. (2004). *The Design and Implementation of Datagram TLS*. Recuperado el 19 de Octubre de 2010, de Sitio web de la Internet Society:
http://www.isoc.org/isoc/conferences/ndss/04/proceedings/Papers/Modadugu.pdf

[7] Pritsker Corporation. (1997). *Visual SLAM Quick Reference Manual.* Indianápolis, IN. USA.

[8] Real Academia Española. (2001). *Diccionario de la lengua española - Vigésima segunda edición*. Recuperado el 01 de Abril de 2011, de buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=criptograf%EDa

[9] Tanenbaum, A. (2003). Redes de computadoras. En A. Tanenbaum, *Redes de computadoras* (pág. 277). México DF: Pearson, Prentice Hall.

BIBLIOGRAFÍA

(s.f.). Recuperado el 28 de Abril de 2011, de Protocolos de autenticación y gestión de claves: http://www.inf.utfsm.cl/~rmonge/seguridad/cripto-06-bn.pdf

(s.f.). Recuperado el 28 de Abril de 2011, de Seguridad (2/4): Protocolo de autenticación: http://informatica.uv.es/it3guia/ARS/transparencias_1c/seguridad2-santi.ppt

Bellare, M., Rogaway, P., & Wagner, D. (18 de Enero de 2004). Recuperado el 23 de Abril de 2011, de The EAX Mode of Operation: http://www.cs.ucdavis.edu/~rogaway/papers/eax.pdf

BRITO, D. J., & MOURA DE SOUSA, G. (2006). *Análise de protocolos para prover confidencialidade e integridade em sessões de Telefonia IP*. Brasilia: Universidade de Brasília.

Bruzual, Z. (2008). *Instructivo y Modelo para realizar el anteproyecto de trabajo de grado.* Caracas: Escuela de Ingeniería Eléctrica, Universidad Central de Venezuela.

Chun-Hung Chen, J. L., Yücesan, E., & Chick, S. E. (2000). Simulation Budget Allocation for further enhacing the efficiency of ordinal optimization. *Discrete Evente Dynamic Systems: Theory and Applications, 10* , 251-270.

Cortés Peña, J., Pérez Noguerol, G., & Sarmiento Losada, A. (s.f.). Recuperado el 2 de Marzo de 2011, de SIP: Session Iniciation Protocol: http://trajano.us.es/~fornes/RSR/2005/SIP/PresentacionSIP.ppt

GnuTLS. (14 de Abril de 2011). *The GNU Transport Layer Security Library*. Recuperado el 22 de Abril de 2011, de Overview: http://www.gnu.org/software/gnutls/

Herrera, E. (2002). *Introducción a las telecomunicaciones modernas.* Limusa.

Internet Engineering Task Force. (Abril de 2006). *Datagram Transport Layer Security*. Recuperado el 10 de Enero de 2011, de RFC EDITOR: http://www.rfc-editor.org/rfc/rfc4347.txt

Internet Engineering Task Force. (Junio de 2002). *SIP: Session Initiation Protocol*. Recuperado el 10 de Enero de 2011, de RFC EDITOR: http://tools.ietf.org/pdf/rfc3261.pdf

Internet Engineering Task Force. (28 de Agosto de 1980). *User Datagram Protocol.* Recuperado el 15 de Octubre de 2010, de RFC EDITOR: http://tools.ietf.org/html/rfc768

Lerch, D. (27 de Julio de 2007). *El blog de Daniel Lerch*. Recuperado el 18 de Marzo de 2011, de http://dlerch.blogspot.com/2007/07/modos-de-cifrado-ecb-cbc-ctr-ofb-y-cfb.html

McGraw-Hill Companies. (2004). SIP: Protocol Operation. En *SIP Demystified* (pág. 119). McGraw-Hill.

McGraw-Hill Companies. (2004). The Session Initiation Protocol: SIP. En *SIP Demystified* (págs. 89-157). McGraw-Hill Companies.

McGrew, D., & Rescorla, E. (Mayo de 2010). *Datagram Transport Layer Security (DTLS) Extension to Establish Keys*. Recuperado el 19 de Octubre de 2010, de Sitio web de la Internet Engineering Task Force: http://tools.ietf.org/pdf/rfc5764.pdf

Modadugu, N., & Jennings, C. (10 de Octubre de 2007). *Internet Engineering Task Force.* Recuperado el 30 de Noviembre de 2010, de Session Iniciation Protocol (SIP) over Datagram Transport Layer Security (DTLS): tools.ietf.org/html/draft-jennings-sip-dtls-05

Modadugu, N., & Rescorla, E. (2004). *The Design and Implementation of Datagram TLS*. Recuperado el 19 de Octubre de 2010, de Sitio web de la Internet Society: http://www.isoc.org/isoc/conferences/ndss/04/proceedings/Papers/Modadugu.pdf

OPNET. (s.f.). Recuperado el 8 de Enero de 2011, de OPNET Modeler: http://www.opnet.com/solutions/network_rd/modeler.html

Pritsker Corporation. (1997). *Visual SLAM Quick Reference Manual.* Indianápolis, IN. USA.

Real Academia Española. (2001). *Diccionario de la lengua española - Vigésima segunda edición*. Recuperado el 01 de Abril de 2011, de buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=criptograf%EDa

Sanchez Ramos, J. (s.f.). Recuperado el 4 de Febrero de 2011, de Nodos empleados en Visual Slam (Awesim): http://www.material_simulacion.ucv.cl/en%20PDF/CompleSlam1.pdf

Stevenson, F. A. (NOVIEMBRE de 1999). *Cryptanalysis of Contents Scrambling System,.* Recuperado el 20 de Febrero de 2011, de http://www.cs.cmu.edu/~dst/DeCSS/FrankStevenson/analysis.html

T. Dierks, C. A. (Enero de 1999). *The TLS Protocol Version 1.0*. Recuperado el 10 de Diciembre de 2010, de http://tools.ietf.org/html/rfc2246

Tanenbaum, A. (2003). Redes de computadoras. En A. Tanenbaum, *Redes de computadoras* (pág. 277). México DF: Pearson, Prentice Hall.

yaSSL. (2011). Recuperado el 20 de Abril de 2011, de Overview of yaSSL and CyaSSL: http://yassl.com/yaSSL/Docs_Overview_of_yaSSL_and_CyaSSL.html

GLOSARIO

Modo de operación EAX: modo de operación para protocolos de criptografía de bloques, diseñado para proveer tanto autenticación como integridad de los mensajes en un esquema de dos ciclos, uno para garantizar la integridad y otro para la autenticidad de cada bloque.

Modo de operación CWC: modo de operación que provee tanto cifrado como integridad de los mensajes. Fue diseñado por Tadayoshi Kohno, John Viega and Doug Whiting.

Modo de operación IAPM: Integrity Aware Parallelizable Mode, modo de operación que permite el proceso de mensajes en paralelo y provee integridad a los mensajes cifrados.

Stream Control Transmission Protocol: protocolo de la capa de transporte que funciona de manera similar a los protocolos TCP y UDP. Provee algunas de las prestaciones de cada protocolo: es orientado a mensajes como el UDP y provee confiabilidad a través de un control de congestión como el TCP.

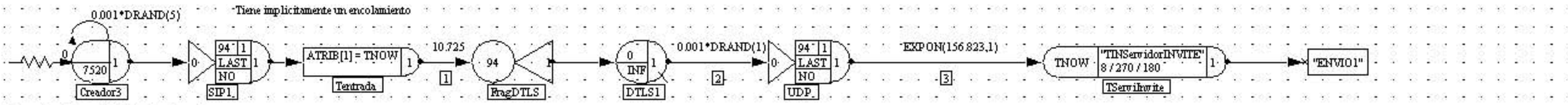
Transport Layer Security: es un protocolo de capa de transporte que provee seguridad de las comunicaciones a través del internet usando criptografía de llave simétrica y también confiabilidad a través de código de autenticación de mensajes (MAC en inglés)

MAC Código de autenticación de mensajes: es un segmento de información agregado a la cabecera de los mensajes, para confirmar la autenticidad de los mismos.

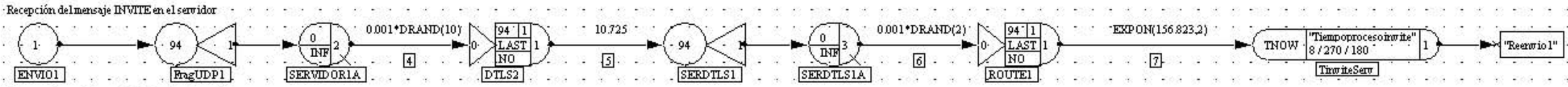
ANEXOS

Modelo CONSEGURIDAD

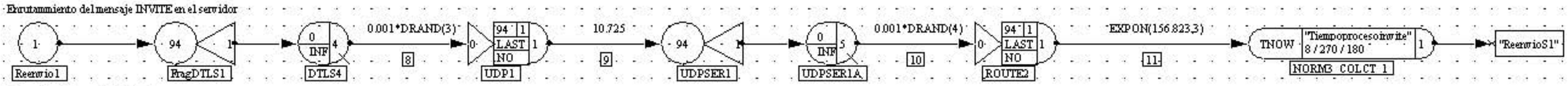
Creación del mensaje INVITE



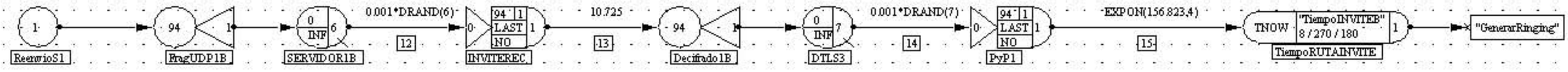
Recepción del mensaje INVITE en el servidor



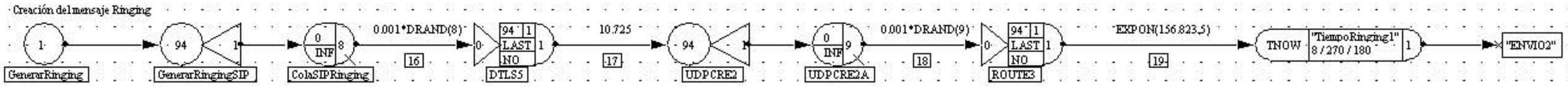
Enrutammiento del mensaje INVITE en el servidor



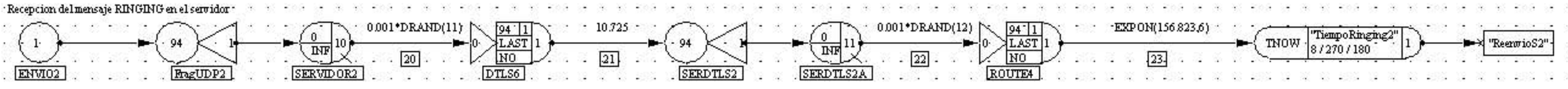
Recepcion del mensaje INVITE



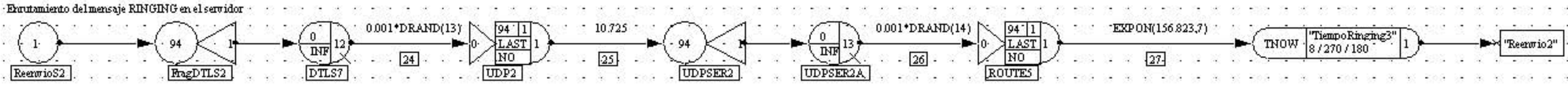
Creación del mensaje Ringing



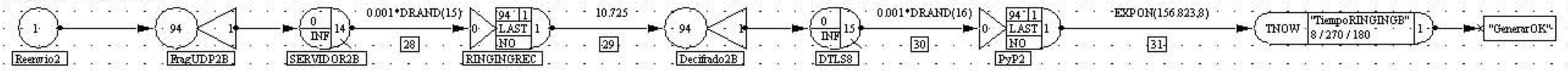
Recepcion del mensaje RINGING en el servidor



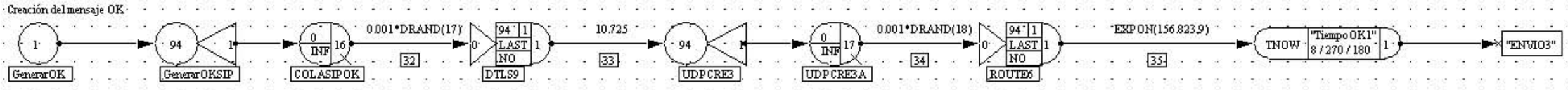
Enrutamiento del mensaje RINGING en el servidor



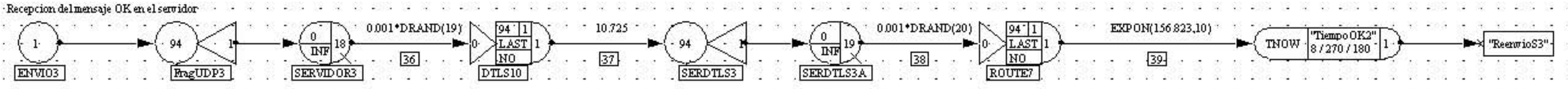
Recepcion del mensaje RINGING



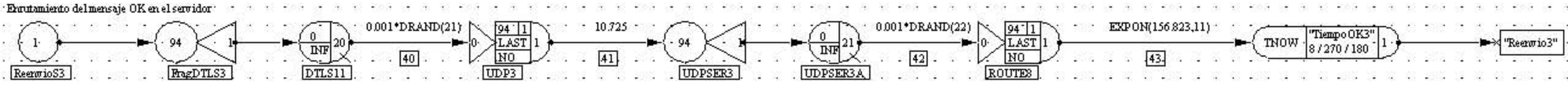
Creación del mensaje OK



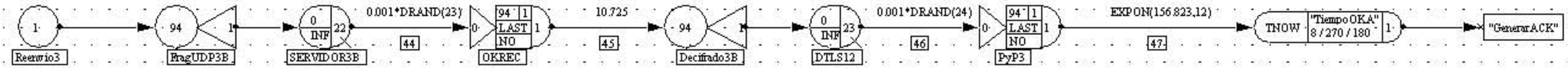
Recepcion del mensaje OK en el servidor



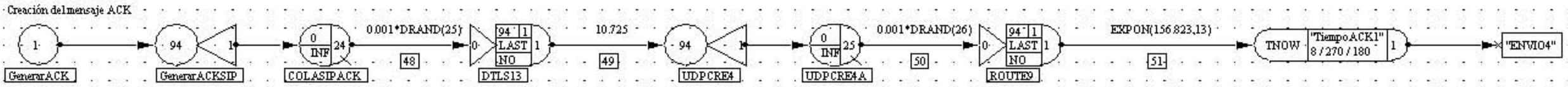
Enrutamiento del mensaje OK en el servidor



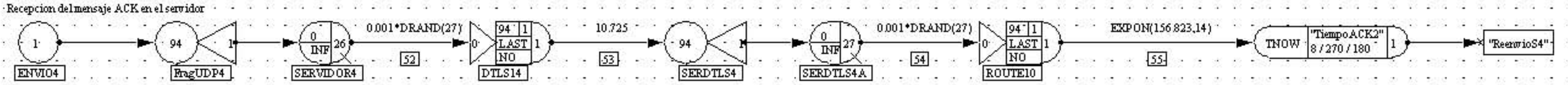
Recepcion del mensaje OK



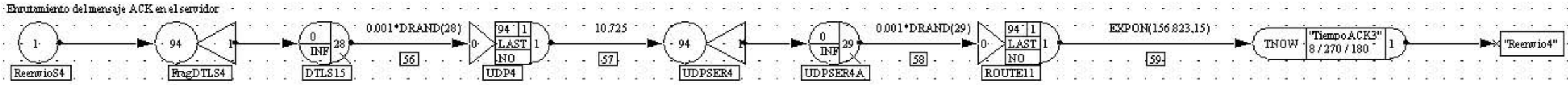
Creación del mensaje ACK



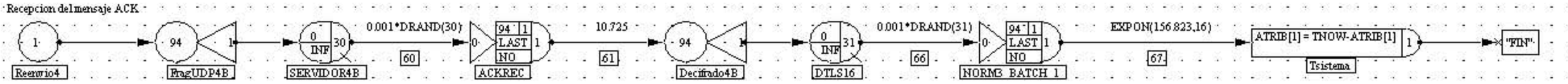
Recepcion del mensaje ACK en el servidor

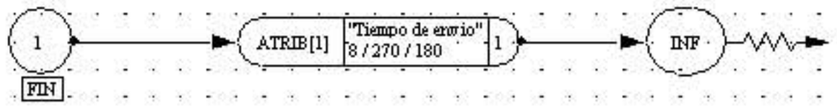


Enrutamiento del mensaje ACK en el servidor

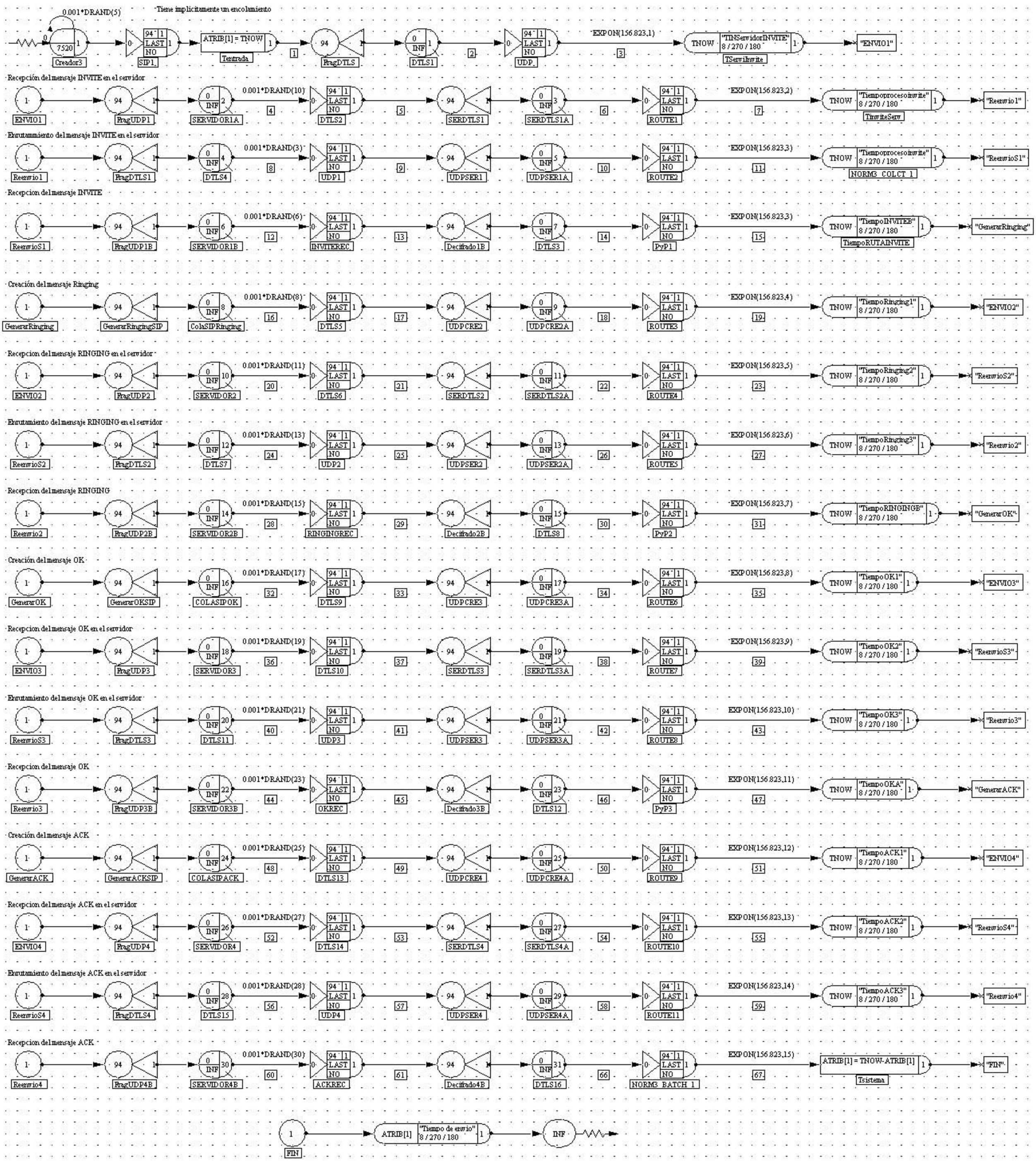


Recepcion del mensaje ACK





Modelo SIN SEGURIDAD



Exponencial				
Muestra	Promedio	Desviación est	Mínimo	Máximo
1	2561,157	590,248	1317,052	4016,791
2	2513,230	661,451	918,265	4209,416
3	2530,737	662,157	1034,035	4234,991
4	2355,743	616,987	1276,274	4664,900
5	2431,374	575,858	1346,886	3754,157
6	2491,940	632,724	987,898	4435,576
7	2567,634	626,102	1289,446	4193,645
8	2376,657	552,515	1519,160	4075,356
9	2470,756	656,422	1401,815	5295,705
10	2508,484	608,538	1127,091	3907,702
11	2398,821	603,912	1089,622	4576,549
12	2362,129	532,638	1096,219	3616,796
13	2418,080	558,934	1427,133	3702,815
14	2369,455	534,306	1225,453	3542,436
15	2597,309	677,116	1468,381	4288,019
16	2426,889	582,002	1002,950	3848,314
17	2513,940	619,217	1306,681	4567,124
18	2519,058	582,991	1421,893	4310,522
19	2403,076	576,844	1451,145	4373,455
20	2459,270	637,442	1094,470	4406,803
21	2475,809	611,373	1302,322	4319,797
22	2508,417	682,266	1160,699	4873,919
23	2453,985	664,303	1075,520	4158,603
24	2529,458	685,715	1241,991	4195,213
25	2427,026	598,732	1200,799	3987,207
26	2544,738	593,274	1099,286	4020,819
27	2394,090	607,871	1258,266	4547,992
28	2494,396	638,309	1372,852	4073,177
29	2517,025	650,224	1301,396	4484,382
30	2376,900	542,672	1247,083	3880,337
31	2522,995	600,521	1360,654	4051,855
32	2519,246	627,635	1351,330	4669,574
33	2515,302	763,277	866,049	5309,131
34	2554,596	629,768	1482,642	4005,757
35	2390,709	518,676	1368,078	4397,906
36	2572,067	614,998	1208,491	4013,516
37	2477,011	642,060	950,997	4879,175
38	2456,314	604,732	1181,504	3702,800
Promedios	2473,837	614,863	1232,417	4252,427

Triangular				
Muestra	Promedio	Desviación es	Mínimo	Máximo
1	2633,858	99,823	2433,409	2894,265
2	2642,335	86,063	2499,686	2887,228
3	2633,122	91,060	2433,371	2917,310
4	2630,130	101,046	2354,065	2929,615
5	2647,676	88,423	2397,647	2842,209
6	2652,529	98,357	2480,719	2890,096
7	2614,087	98,920	2441,667	2883,605
8	2638,743	111,043	2330,085	2853,668
9	2632,935	92,295	2339,531	2807,916
10	2637,171	93,033	2415,661	2857,510
11	2669,481	100,905	2435,363	2983,725
12	2644,112	100,799	2444,506	2872,163
13	2646,720	92,079	2394,015	2910,304
14	2640,430	112,721	2391,321	3041,045
15	2617,274	91,969	2330,400	2856,896
16	2648,125	99,325	2461,020	2875,438
17	2640,627	93,134	2380,667	2902,934
18	2622,265	98,217	2379,121	2857,984
19	2667,743	94,741	2444,024	2884,235
20	2656,770	109,339	2448,100	2896,520
21	2631,076	96,381	2433,167	2897,985
22	2634,131	90,497	2424,200	2807,304
23	2652,429	107,750	2435,656	2971,771
24	2623,068	90,127	2433,961	2904,978
25	2645,700	104,927	2375,213	2865,380
26	2636,774	80,991	2449,221	2843,705
27	2642,528	100,529	2428,799	2884,378
28	2629,847	91,221	2404,089	2838,768
29	2646,826	102,111	2393,985	2879,477
30	2647,874	97,412	2456,570	2890,456
31	2635,307	93,051	2419,369	2900,832
32	2637,416	91,864	2423,536	2886,053
33	2635,450	91,109	2423,949	2841,032
34	2631,977	97,509	2438,325	2906,729
35	2648,434	99,702	2361,982	2892,625
36	2651,707	104,905	2328,774	2874,398
37	2637,267	99,831	2366,607	2876,513
38	2636,298	92,693	2427,704	2861,456
Promedios	2640,00637	96,9974211	2412,09171	2886,01332

Poisson				
Muestra	**Promedio**	**Desviación es**	**Mínimo**	**Máximo**
1	2509,061	52,067	2379,720	2626,696
2	2510,140	47,762	2399,700	2643,737
3	2507,186	53,472	2353,697	2608,706
4	2511,318	55,575	2396,723	2609,719
5	2513,307	49,817	2392,705	2647,692
6	2497,195	53,817	2373,718	2614,733
7	2517,390	55,557	2385,735	2633,695
8	2518,910	49,555	2381,735	2620,733
9	2509,555	46,960	2384,700	2612,745
10	2510,764	54,617	2354,724	2627,724
11	2507,768	56,887	2370,692	2628,719
12	2513,754	49,904	2390,740	2620,693
13	2514,863	54,007	2406,710	2677,715
14	2515,594	45,645	2370,695	2607,772
15	2514,689	41,351	2407,719	2612,709
16	2504,861	48,929	2347,714	2608,718
17	2503,130	49,537	2410,705	2599,715
18	2506,168	49,426	2376,705	2624,701
19	2499,026	46,959	2417,708	2641,715
20	2515,743	51,771	2407,711	2650,713
21	2506,957	49,124	2387,692	2625,726
22	2522,331	52,066	2417,701	2675,700
23	2508,566	48,124	2398,711	2638,694
24	2508,053	50,787	2360,687	2609,751
25	2509,269	48,469	2401,713	2645,695
26	2508,102	57,077	2366,711	2645,694
27	2510,044	50,929	2382,722	2613,709
28	2503,711	51,684	2364,717	2614,696
29	2503,765	51,344	2382,708	2667,698
30	2504,753	51,736	2368,705	2659,731
31	2509,676	49,346	2386,703	2609,716
32	2507,229	54,099	2400,735	2638,697
33	2507,363	59,441	2381,740	2656,700
34	2517,967	51,128	2419,701	2644,708
35	2513,426	49,169	2385,713	2610,744
36	2507,601	52,213	2399,698	2629,705
37	2511,464	49,805	2384,695	2634,701
38	2508,637	54,638	2367,698	2636,720
Promedios	**2509,719**	**51,179**	**2386,026**	**2631,004**

ABS(NORMAL)				
Muestra	**Promedio**	**Desviación es**	**Mínimo**	**Máximo**
1	2678,801	320,392	1939,815	3519,643
2	2776,352	374,781	1754,665	3689,455
3	2616,020	322,725	1883,222	3350,657
4	2506,276	304,727	1765,232	3339,780
5	2606,599	390,223	1881,751	3634,078
6	2601,651	331,038	1905,901	3387,115
7	2665,763	319,963	1958,489	3578,707
8	2612,705	366,462	1837,423	3615,063
9	2607,691	366,799	1647,070	3598,796
10	2618,798	388,687	1742,001	3740,102
11	2592,505	341,327	1967,380	3381,183
12	2617,351	360,868	1986,217	3482,352
13	2608,044	328,187	1777,110	3532,161
14	2669,030	361,227	1964,680	3523,498
15	2633,971	372,635	1796,203	3649,723
16	2650,490	297,812	1984,384	3258,286
17	2678,064	355,602	1848,207	3440,041
18	2598,797	319,957	1804,203	3471,379
19	2661,703	348,565	1729,111	3383,407
20	2580,510	339,514	1913,294	3436,093
21	2662,636	315,733	1916,095	3475,630
22	2683,401	322,843	1969,536	3436,527
23	2595,252	340,907	1757,734	3246,568
24	2669,595	387,509	1818,729	3803,998
25	2637,465	298,562	1870,705	3577,776
26	2655,180	323,211	1987,825	3412,877
27	2631,883	332,191	1900,383	3591,351
28	2584,813	407,843	1688,781	3512,544
29	2592,172	309,842	1935,641	3343,721
30	2587,595	345,817	1826,484	3516,016
31	2635,721	352,512	1870,826	3420,265
32	2615,572	355,658	1782,870	3691,348
33	2647,012	340,586	1984,791	3735,737
34	2657,285	312,628	2084,718	3389,170
35	2622,324	371,791	1840,822	3827,185
36	2615,175	329,058	1747,664	3571,869
37	2591,948	307,895	1901,754	3419,761
38	2595,095	351,004	1730,554	3460,562
Promedios	**2627,928**	**342,555**	**1860,586**	**3511,695**