Centro de Investigación y de Estudios
Avanzados
del Instituto Politécnico Nacional

Departamento de Ingeniería Eléctrica

Sección de Computación

Análisis del Tráfico de una Red Local

Tesis que presenta

Jorge Enrique Morfín Galván

para obtener el Grado de

Maestro en Ciencias

en la Especialidad de

Ingeniería Eléctrica

Director de la Tesis

Dr. Arturo Díaz Pérez

México, D.F. **Marzo 2004**

Abstract

Try to predict a local network behavior is very hard, because there are many involved factors and the fast technologies changes. However, is very important to have a mechanism that allow us predict and analyze the network behaver.

This work describes the creation of a statistical model from the behavior of the traffic that goes in or out from the network. It also describes the construction a proposed system, to capture and analyze the network traffic, comparing it to the proposed statistical model. The system is composed by two modules: a capture module and an analysis module. The capture module capture all the packets that goes in or out from the local net, and transfer them to the analysis module. The analysis module makes the statistical analysis, the error percents when comparing the data to the model and graphics to present the results. It's also show the system obtained results: data captured and data storage and analysis results and total traffic by totals, protocols, address, ports (source and destiny).

This system is capable to detect some network attacks and the most frequent failures in a network.

Resumen

Es difícil tratar de predecir el comportamiento de una red, debido a que existen una gran cantidad de factores involucrados en dicho proceso y a los rápidos cambios de tecnología. Sin embargo, es de una gran importancia contar con un mecanismo que nos permita predecir y analizar el comportamiento de una red.

En este trabajo se describe la creación de un modelo estadístico del comportamiento del tráfico que entra o sale de la red. También se describe y construye un sistema propuesto para capturar y analizar el tráfico de una red local, comparándolo con el modelo estadístico propuesto. El sistema consiste de un módulo de captura y uno de análisis. El módulo de captura se encarga de capturar los paquetes que entran o salen de la red local y después los transfiere al módulo de análisis, el cual se encarga de realizar un análisis estadístico, porcentajes de error al comparar los datos con el modelo y creación las gráficas para presentar los resultados. Se muestran los resultados que se han obtenido con dicho sistema, lo cual incluye la captura de los datos que circulan por la red, el almacenamiento de los mismos y la presentación de los resultados del análisis y comparación por tráfico total, por protocolos, direcciones y puertos, tanto fuente como destino.

Con este sistema es posible detectar algunos ataques y fallas que se presentan con frecuencia en una red.

Agradecimientos

Agradezco el apoyo económico otorgado por el CONACyT a través de una beca que me permitió cursar la Maestría en el departamento de Ingeniería Eléctrica, Sección de Computación del CINVESTAV.

Agradezco al CINVESTAV, por brindarme la oportunidad y las facilidades para realizar esta investigación que culmina mis estudios de maestría.

A mi asesor, el Dr. Arturo Díaz Pérez, quien me guió durante el desarrollo de esta tesis.

A mis sinodales, los Dres. Luis Gerardo de la Fraga y Guillermo Morales Luna, por su colaboración en beneficio de la calidad del presente reporte.

A las secretarias de la Sección de Computación, porque sin ellas no realizaríamos trámite administrativo alguno, en especial a la Sra. Sofía Reza por su gran dedicación y empeño al realizar su trabajo.

A mi familia, por su apoyo.

A mi amada Elisa Guillaumín, quien me motivó y apoyó desde que vio mi inquietud de superación personal y académica y por su amor incondicional.

Índice general

Índice de figuras

Índice de cuadros

Capítulo 1

Introducción

El análisis del tráfico de una red es un problema complicado, debido a que son muchos los factores que están involucrados en éste proceso. Uno de los factores más importantes se debe a la variación de tráfico con respecto al tipo (entrada o salida), a la hora del día y al tipo de día (hábil o no hábil) que se analiza. El tráfico de entrada y salida de una red local no es completamente simétrico. Por ejemplo, en una red de clientes normalmente el tráfico de entrada es superior al de salida. La cantidad de tráfico en una red local, también varía de acuerdo a los hábitos de trabajo de la comunidad de usuarios. Por ejemplo, en una red de una organización típica el tráfico tiende a incrementarse a la mitad de la jornada laboral y se reduce substancialmente en un día no hábil.

Otro factor que afecta al comportamiento del tráfico está relacionado con los protocolos utilizados, el tipo de información transferida y la tecnología utilizada en la propia red. Por ejemplo, actualmente el tráfico que se observa sobre TCP es mucho mayor que el de UDP, pero la proporción se ha ido reduciendo conforme han surgido nuevas aplicaciones. Por otro lado, el tipo de información determina el comportamiento durante algunos períodos de tiempo. Cuando se transfieren datos se observan ráfagas de paquetes en períodos cortos. Por otra parte, cuando se transfieren audio y video la cantidad de paquetes transferidos se mantiene estable durante períodos de tiempo prolongados. Con respecto a la tecnología es claro que una red segmentada es mucho más eficiente que una red totalmente conectada y esto se observa

en el tráfico.

Se han realizado algunos intentos por tratar de modelar el tráfico que circula a través de una red [1][2], pero son modelos poco eficientes y precisos [3][4], o son modelos extremadamente complicados [5][6][7], por lo que resultan muy lentos para calcularse y poco eficientes para predecir el comportamiento de una red.

Estos trabajos han consistido en el intento de modelar matemáticamente el comportamiento de las conexiones entre los nodos que componen una red, proponiendo inicialmente una distribución de Poisson para representar el comportamiento de todas las comunicaciones, pero este modelo no representaba exactamente el comportamiento del tráfico [3], así que se comenzó a proponer diferentes modelos que se ajustan mejor a dicho comportamiento, inclusive llegando a proponer nuevos modelos para explicar mejor el comportamiento del tráfico [1]. Algunos modelos más complejos como por ejemplo de autosimilaridad y de lógica difusa son más precisos, pero muy complejos y de gran costo computacional, por lo que no son muy adecuados para intentar predecir el comportamiento de una red [6][7].

Es muy importante tener un modelo que se ajuste al comportamiento real de una red, ya que con este modelo se pueden analizar fallas en la red, ataques externos y la utilización de la capacidad de los canales de comunicación.

El saber si una red está en un modo de operación normal o no, nos puede ayudar a saber si está sufriendo algún tipo de ataque, como podría ser una inundación de tráfico o un ataque de denegación de servicio, etc. También se pueden detectar fallas en el funcionamiento de la red, como podría ser el mal funcionamiento de un servidor, ruteador o cortafuego, si algún segmento de la red está fuera de operación o sufrió de algún desperfecto en el cableado de la red. Mediante el tráfico observado en una red y teniendo una aproximación acerca del comportamiento de los usuarios se puede determinar la capacidad de comunicación disponible por usuario.

Es conveniente entonces, conocer al menos el comportamiento global de una red local entendido como el tráfico que entra y sale de dicha red. Para ello es suficiente desarrollar un modelo de comportamiento estadístico que

considere los factores mencionados inicialmente.

Un modelo estadístico es aquel que permite determinar de manera aproximada el valor de una respuesta observada, con base en un análisis estadístico previo del evento en cuestión [8].

A diferencia de los modelos analíticos, como por ejemplo los usados en lógica matemática, los modelos estadísticos están basados en una gran cantidad de eventos observados previamente. Un modelo estadístico no es muy preciso debido a su misma naturaleza, pero una vez que se ha obtenido, el comparar un valor obtenido mediante un nuevo experimento es, en general, muy rápido.

Los modelos estadísticos se pueden clasificar en *estáticos* o *dinámicos*, dependiendo de la importancia del factor tiempo. Otra forma de clasificarlos es en *continuos* o *discretos* dependiendo de el tipo de evento que se esté modelando.

El propósito del presente trabajo es diseñar un sistema que sea capaz de predecir el comportamiento de una red local. Para ello se propone usar un modelo estadístico que permita hacer predicciones con un pequeño margen de error. Así también, este sistema puede ser usado para detectar algunos de los ataques más comunes y de las fallas más frecuentes.

Adicionalmente, se desea realizar un estudio estadístico sobre el comportamiento de un red local, así como desarrollar y validar el modelo estadístico que se obtenga.

Para lograr un buen modelo estadístico es necesario recolectar primero una gran cantidad de datos, para que resulte ser una muestra representativa del tráfico que circula por la red, además se debe de tomar en cuenta el día, hora y minuto así como el tipo de día de que se trate (fin de semana, laborable, feriado, etc.).

Una vez que se cuenta con el modelo es posible ir recolectando los datos e ir realizando su análisis. La recolección de los datos plantea un problema, ya que este proceso puede interferir con el tráfico normal de la red y puede provocar un deterioro importante en la calidad de servicio. Es este trabajo se exploraron varias posibilidades y se decidió desarrollar un sistema basado

en software cuyas funciones de análisis actúan fuera de línea.

El sistema se compone de un módulo de captura y un módulo de análisis. El módulo de captura recolecta datos y los transfiere al módulo de análisis. El módulo de análisis procesa los datos, los almacena en una base de datos, realiza cálculo de errores y genera estadísticas, reportes y gráficas.

El documento de tesis está organizado de la siguiente manera: en el capítulo 2 se explica la importancia del análisis y modelado del tráfico de las redes, así como las dificultades para hacerlo y los trabajos previos con respecto al tema. En el capítulo 3 se explican las diferentes técnicas existentes para capturar el tráfico de las redes. Se indica la forma en que se calculó el modelo estadístico con base en los datos recolectados. En el capítulo 4 se explica la arquitectura del sistema propuesto para lograr el análisis de una red local. Finalmente en el capítulo 5 se muestran la interfaz gráfica y los resultados obtenidos con el presente trabajo. Las conclusiones de esta tesis aparecen en el capíulo 6, la bibliografía aparece al final de la misma.

Capítulo 2

Comportamiento del tráfico en redes locales

En este capítulo se explicará la importancia de realizar un modelo del tráfico que circula en una red local y las dificultades para lograrlo, así como los diferentes métodos y soluciones que se han propuesto para modelar el tráfico de una red, desde los primeros intentos hasta la actualidad.

Construir un modelo del tráfico en una red local es complicado debido a todos los factores que intervienen en el proceso de transmisión y recepción de datos entre dos nodos de una red. Los factores incluyen los protocolos utilizados, la tecnología con la que está construida la red, el diseño físico y lógico de la red, etc. Es prácticamente imposible realizar un análisis que incluya absolutamente a todos los factores involucrados, por lo cual es importante realizar una discriminación acerca de cuáles son los factores más importantes dentro del proceso y la influencia que tiene cada uno de éstos dentro del mismo.

2.1. Conceptos básicos

La comunicacción entre dos nodos conectados a una red local se realiza mediante protocolos, reglas que siguen los nodos para el intercambio de información. Las redes se organizan mediante jerarquías de protocolos organizados en capas o niveles. Cada capa anterior proporciona servicios a su

capa superior y la aisla de los detalles de los protocolos utilizados usados en las capas inferiores [9].

Las pilas de protocolos están generalmente basadas en el modelo OSI (Open Systems Interconection). El modelo OSI está basado en una propuesta desarrollada por ISO (International Standard Organization), como un primer paso internacional para la estandarización de los protocolos utilizados en las diferentes capas. El modelo lleva ese nombre debido a que trata de conexiones entre sistemas abiertos, esto es, sistemas que están abiertos para comunicaciones con otros sistemas.

El modelo OSI cuenta con siete capas, las cuales son:

Capa Física: Se encarga de transmitir los bits a través de un canal de comunicación.

Capa de Enlace de Datos: Es la encargada de tomar una transmisión con posibles interferencias y transformarla en una línea libre de errores de transmisión no detectados.

Capa de Red: Se encarga de controlar la operación de la subred.

Capa de Transporte: Su función es tomar los datos de la capa de sesión y si es necesario, dividirla en pedazos más pequeños, pasarlos a la capa de red y asegurarse de que los pedazos lleguen correctamente al otro extremo.

Capa de Sesión: Permite a los usuarios en diferentes computadoras establecer sesiones entre ellas.

Capa de Presentación: Realiza ciertas funciones que son requeridas suficientemente seguido, para garantizar encontrar una solución general para ellas, en vez de dejar que cada usuario resuelva los problemas.

Capa de Aplicación: Contiene una variedad de protocolos que son comúnmente utilizados.

El modelo TCP/IP se usaba en ARPANET, que fue la primera red de computadoras, y actualmente es utilizado en la Internet, por lo cual es muy importante. Este protocolo fue diseñado para que las conexiones entre computadoras permanecieran intactas mientras las computadoras origen y destino estuvieran funcionando, por lo que se requiere de una arquitectura flexible, que abarque desde la transferencia de archivos hasta la transmisión de discursos en tiempo real.

Este modelo cuenta con cuatro capas, las cuales son:

Capa del Nodo a la Red: El modelo solo indica que el nodo se ha de conectar a la red haciendo uso de algún protocolo que permita enviar paquetes IP.

Capa de Interred: Define un formato de paquete y protocolo oficial llamado IP. Esta capa se encarga de entregar paquetes a donde deban ir.

Capa de Transporte: Permite que las entidades pares en los nodos origen y destino lleven a cabo una conversación. Se definen dos protocolos de extremo a extremo, el TCP y el UDP.

Capa de Aplicación: Esta capa contiene todos los protocolos de alto nivel, como son FTP, HTTP, SNMP, SMTP, etc.

El estándar IEEE 802.3 define los protocolos de una red de transmisión basada en bus con control de operación descentralizado (CSMA/CD). Los nodos de una red así pueden transmitir cuando quieran, pero si dos o más paquetes chocan, cada computadora sólo espera un tiempo al azar y vuelve a intentar mandar el paquete. Ethernet es la tecnología que implementa casi en su totalidad dicho estándar.

Para establecer una comunicación entre dos nodos, las redes locales utilizan diversos equipos de comunicación. Cada uno de ellos ofrece servicios con capacidades diferentes lo cual los ubica en los niveles enlace de datos, de red y de transporte del modelo OSI. Entre los dispositivos más usados en una red local se encuentran los siguientes:

HUB: También llamado concentrador, es un aparato que sirve como medio de comunicación entre diferentes nodos de la red, en el que todos se encuentran escuchando y en el que pueden escribir. Cuando un nodo escribe en un canal, el mensaje es repetido en todos los canales, asegurandose así que el mensaje será escuchado por el destinatario.

Switch: Este aparato es similar al HUB, pero en éste caso, solo repite la señal recibida en el canal adecuado, en donde se encuentra el destinatario.

Gateway: Es la puerta de enlace de la red, sirve como salida para todos los paquetes de la red que se dirigen hacia otra red.

Router: También llamado ruteador, éste aparato trata de determinar la ruta más adecuada que debe de tomar un paquete y lo envía por el canal necesario para que cumpla con dicha ruta.

DNS: Es el servicio de dominio de nombres (Domain Name Service), el cual se encarga de traducir los nombres de las computadoras en direcciones IP.

2.2. Estudios sobre modelación del tráfico

Existen varios estudios que se han realizado con el propósito de predecir el comportamiento de los paquetes que circulan en una red, sin que se haya logrado todavía diseñar un modelo que incluya todos los factores involucrados en el proceso [1][3][5].

El principal problema para lograrlo es la rápida evolución de las redes, tanto en tecnología como en las aplicaciones y protocolos utilizados, es decir, la tecnología se desarrolla en un tiempo muy corto, menor al que tomaría una recolección de datos con su respectivo análisis para lograr un modelo que incluya todos los factores y que logre predecir de una manera exacta el comportamiento de las redes.

Hasta el momento se han logrado medir algunas propiedades de las redes, como son los trenes[1], las ráfagas[1][5], o la auto-similitud[6], así como también se han logrado diseñar algunos modelos que predicen su comportamiento [5][6][10][2][7].

Uno de los problemas con algunos modelos [1][3] es que no son muy precisos, ya que no manejan suficiente información o no es manejada de manera eficiente, es decir, no se dan prioridades a los diferentes factores involucrados en dicho proceso, por lo cual provocan demasiados errores al tratar de predecir el comportamiento de la red.

Otros modelos [5][7] presentan el problema de que son matemáticamente muy complejos, ya que incluyen casi toda la información de la red, la cual incluye una gran cantidad de información, por lo que son difíciles de manejar o es muy lento para poder realizar todos los cálculos necesarios, siendo éste tiempo algunas veces superior al tiempo en que se realiza el evento.

Los modelos pueden clasificarse, dependiendo de lo que se intente modelar, en modelos de tráfico punto a punto, y en modelos de comportamiento colectivo de una red. En los modelos punto a punto se intenta modelar las comunicaciones entre dos nodos de una red que se encuentran conectados entre sí, mientras que en los modelos de comportamiento colectivo se trata de analizar el comportamiento general de muchos nodos que se encuentran conectados con otros nodos, en otras redes, que están realizando nuevas conecciones y cerrando otras.

2.2.1. Modelo de tráfico punto a punto

Los primeros trabajos que se realizaron sobre el tráfico que circula a través de las redes, estudiaron el desempeño que tiene una red de área local, basados en el retardo que sufre un solo caracter para llegar desde su fuente hasta su destino. Se llegó a la conclusión de que el principal factor que afecta dicho retardo no es la conexión, ni la transmisión a través del cable, sino el paso del caracter a través de los diferentes intentos de transmisión a través de la red. Lo anterior se debe a los retrasos provocados por las colisiones, ya que el tiempo que ocupa el caracter para llegar desde un nodo de la red hasta otro

nodo cualquiera, es casi el doble de tiempo que en todos los procedimientos restantes (paso por cada capa del respectivo protocolo, paso por los buffers, etc.)[4].

Así también, inicialmente se llegó a la conclusión de que los arribos, el tamaño de los paquetes, y todas las demás variables involucradas en el proceso de comunicaciones eran totalmente aleatorias por lo que no era posible tratar de predecirlas con exactitud[1][5].

Las variables involucradas en el tráfico de red son muchas y van desde los protocolos y dispositivos de comunicación hasta el tipo de información que se transporta. Por ejemplo, para transmitir un mensaje electrónico se requiere de la transmisión de datos, usando algún programa de chat, que utiliza UDP, sobre TCP/IP, mientras que para transmitir una señal de audio es necesario garantizar una determinada calidad de servicio, para que el audio sea entendible. Por lo anterior, el comportamiento del tráfico varía entre otras cosas por la aplicación que se utilice.

El hardware también afecta al comportamiento del tráfico. En menor medida son los procesadores de las computadoras, la cantidad de memoria, y en mayor medida está la tecnología utilizada (ATM, Token Ring, Ethernet, etc.), los métodos de conexión de la red (HUB, SWITCH, replicadores, Wi-Fi, etc.) y la topología física de la red.

Los distintos protocolos utilizados también influyen en el comportamiento de una red (IPv4, IPv6, IPX, etc.), el tipo de paquetes que circulan en la red (TCP, UDP) y los protocolos utilizados, por ejemplo, FTP, HTML, SNMP, SMTP, etc..

A principios de los años ochenta se comenzó a modelar el tráfico de las redes mediante una distribución de Poisson $f(x, \lambda) = \frac{\lambda^x e^{-\lambda}}{x!}$ o alguna variación de la misma. En la figura 2.1 se muestra la función de distribución de Poisson, que por su simplicidad matemática, y facilidad de uso fue la más utilizada durante los primeros intentos de modelar el tráfico de una red. [11][3].

Paxon y Floyd [3] demostraron que el modelo para los paquetes que circulan en una red en un tiempo considerablemete grande no obedece a una distribución de Poisson, únicamente el proceso de arribos de los clientes a la

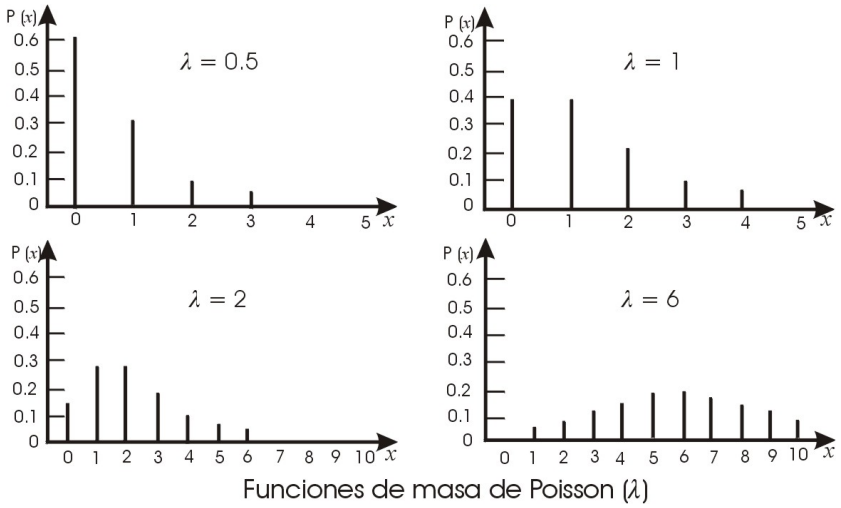


Figura 2.1: Distribuciones de Poisson para diferentes λ, donde λ es el parámetro de la distribución así como la media y la variancia. En el eje horizontal se muestra x y en el eje vertical se muestra la función de masa $p(x)$.

red sigue este tipo de distribución. Para diferentes tipos de paquetes TCP, como son el SMTP, WWW (HTTP) y otros, la distribución tiene un comportamiento por ráfagas (mostradas en la figura 2.2), es decir, son generados una gran cantidad de paquetes en un plazo de tiempo muy corto y pueden transcurrir grandes períodos de tiempo sin que se transmita ni un solo paquete. La cantidad de paquetes que se generan, la duración de la transmisión y el tiempo que transcurre entre las ráfagas es totalmente aleatorio.

Para los casos en que no es posible utilizar una distribución de Poisson, puede usarse otra distribución que ajuste mejor los datos obtenidos experimentalmente, como es una distribución de Pareto o alguna similar.

Jain y Routhier desarrollaron un método estadístico para poder evaluar qué tan bien se ajusta el modelo de Poisson al comportamiento real de la red[1]. Se describieron las propiedades de las ráfagas, las cuales son propiedades intrínsecas de los paquetes auto-generados, es decir, que son los generados no por los usuarios, sino por las mismas computadoras que componen la red, como podrían ser los paquetes de correo electrónico o ARP[5][6].

Después se analizaron los datos que fueron tomados en una red con

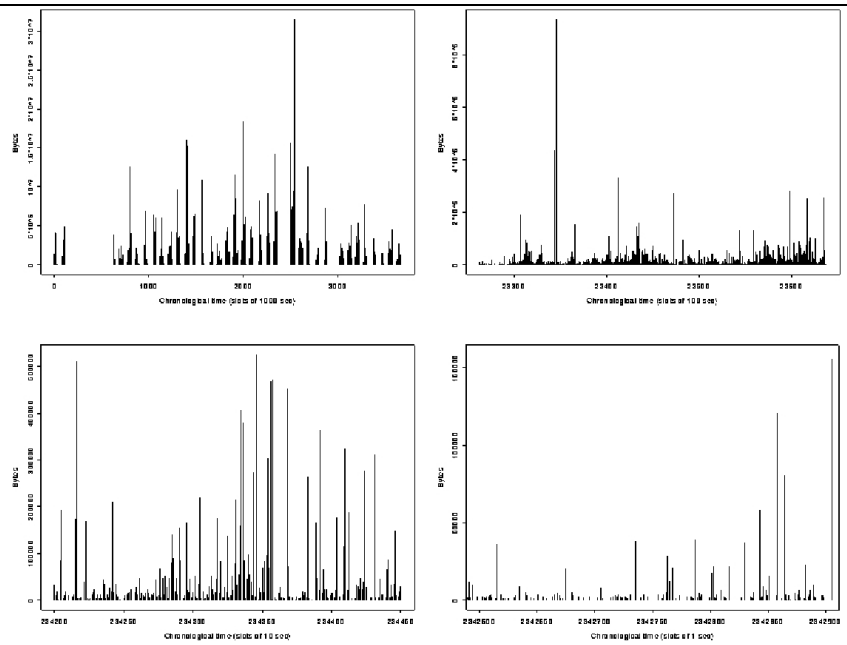


Figura 2.2: Ráfagas en la red, a diferentes escalas de tiempo. En el eje vertical se encuentra la cantidad de tráfico en un determinado instante, mientras que en el eje horizontal se grafica el tiempo.

topología de anillo y se trató de buscar los modelos que se adaptaran mejor al comportamiento de los paquetes. Primero se trataron de ajustar los datos al modelo de Poisson, tomando en cuenta que los eventos son independientes y que se encuentran exponencialmente distribuidos. Después se trató de ajustar los datos en algunas variaciones de Poisson. Al no encontrar modelos estadísticos que se ajustaran debidamente al comportamiento exhibido por los paquetes, se decidió proponer un nuevo modelo, el de trenes[1], el cual es mostrado en la figura 2.3

El modelo de trenes consiste en tomar cada nodo de la red como un punto de una gráfica totalmente conectada (como es el comportamiento de una red ethernet conectada mediante un concentrador), los trenes se desarrollan en un segmento entre dos puntos cualesquiera de la gráfica. Si un paquete va desde un punto A hasta el punto B, lo más probable es que los siguientes paquetes que pasen lleven su misma trayectoria y es posible calcular las probabilidades

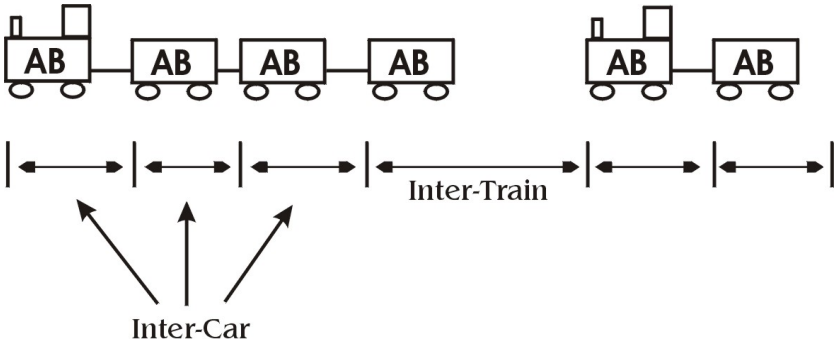


Figura 2.3: Modelo de trenes.

de que los paquetes tengan el mismo par fuente-destino.

En este modelo los tiempos que transcurren entre el paso de un tren y otro son totalmente aleatorios y muy grandes comparados con los tiempos que transcurren entre el paso de cada uno de los vagones que conforman el tren, que también son tiempos aleatorios.

El modelo de trenes es una generalización, del cual otros modelos son sólo casos especiales, es decir, si se ajustan los diferentes parámetros (tiempo entre los vagones, longitud de los vagones, tiempo entre los trenes, etc.) de esta distribución, se pueden obtener las distribuciones de Poisson o Poisson compuestas.

Se determinaron las características de este nuevo modelo como son el tiempo existente entre los paquetes, el tamaño de los paquetes, el tiempo entre los trenes, el tamaño de cada vagón, el tamaño de cada tren, etc. Este nuevo modelo se ajustó muy bien al comportamiento que presentaban los datos anteriormente recolectados. Aunque este modelo puede predecir adecuadamente el comportamiento entre un par de nodos, su utilidad para simular el comportamiento colectivo de toda una red local puede ser limitada ya que en ella existen muchos nodos comunicándose con otros nodos, y algunos de los nodos destino se encuentran fuera de la red.

2.2.2. Comportamiento colectivo de una red

Los modelos descritos en la sección anterior han sido planteados con el propósito de modelar comunicaciones punto a punto (entre un par de identidades que se comunican). Frecuentemente es necesario modelar el comportamiento colectivo de una red local, esto es, modelar el tráfico que entra y sale de una red local. En muchas situaciones el comportamiento colectivo es lo único prácticamente posible que se puede observar en una red local. Mediante la simulación masiva de comunicaciones punto a punto es posible modelar el comportamiento colectivo, sin embargo, dado que no todo el tráfico entra o sale de la red local, la simulación sería poco eficiente.

Willinger, Taqqu, Sherman, Wilson y Lelan demostraron que el tráfico que circula en las redes de área local no se ajusta a ningún modelo estadístico conocido, por lo que los anteriormente mencionados desarrollaron un modelo propio, llamado de auto-similitud, ya que presenta patrones fractales a diferentes niveles, dependiendo de la escala e intensidad del tráfico de la red[5][6].

La auto-similitud es una propiedad, que consiste en que el todo se parece a cada una de las partes y viceversa, con la pequeña diferencia de un factor de escala. Este concepto aplicado al tráfico que circula en una red significa que si se grafica la cantidad de tráfico que circula por la red en diferentes escalas de tiempo, todas las gráficas van a ser muy semejantes, como puede apreciarse en la figura 2.4, tomada de [5]. En esta figura se muestra la similitud de las gráficas del tráfico de una red en diferentes escalas de tiempo. El tráfico puede ser el tráfico total que circula por la red o el tráfico de alguno de los protocolos o el de las direcciones fuente y destino, es decir que puede ser el tráfico de cualquier parámetro de la red, y aún de esta forma se presenta la propiedad de auto-similitud.

En un trabajo posterior los mismos investigadores demostraron que un buen factor de medición de la propiedad de auto-similitud es el efecto Noah[6]. El efecto Noah consiste en tener una variancia infinita en la distribución de los pares fuente-destino de los paquetes de la red. Se presentaron varios análisis estadísticos de una gran cantidad de datos, enfocados a detectar la

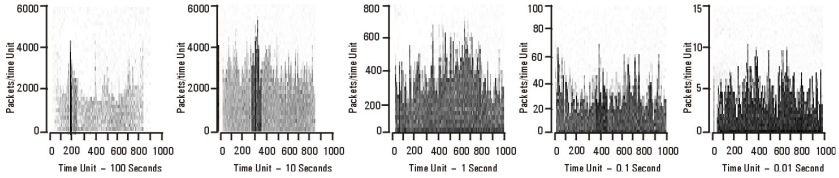


Figura 2.4: Auto-similitud. Las gráficas fueron tomadas a diferentes escalas de tiempo, de 100 seg. a 0,01 seg.

presencia de dicho efecto. Si el efecto Noah está presente en el grupo de datos, entonces se trata de un conjunto de datos que se comportan de acuerdo al modelo de auto-similitud, por lo que no es necesario realizar ninguna otra prueba estadística para demostrar este comportamiento.

Para verificar la presencia del efecto Noah se utiliza un estimador de Hill, el cual calcula la diferencia entre las observaciones que entran dentro del modelo que fue propuesto y las que no lo hacen, y entre más observaciones estén dentro de dicho modelo, menor será la variancia que presente la muestra.

Los estudios de auto-similitud se basan en una recolección previa de datos, que se realizó desde el mes de agosto de 1989 hasta el mes de febrero de 1992 en las redes locales de Bellcore Morris Research and Engineering Center (MRE). Estos datos cuentan con un gran volumen y calidad, ya que no se perdió ningún paquete que circuló por la red en estas mediciones[6].

Se desarrolló la definición matemática de auto-similitud (la variancia de una muestra se decrementa más lentamente que el recíproco del tamaño de la muestra, es decir, las autocorrelaciones decaen hiperbólicamente, implicando una función de autocorrelación no sumable). Se dieron algunos modelos estocásticos capaces de representar apropiadamente este comportamiento, así como algunos métodos estadísticos para poder analizar dicho tipo de datos y un método de prueba de la auto-similitud[6]. Este modelo no es muy práctico para utilizarlo en tiempo real debido a su gran costo computacional.

Recientemente se han desarrollado modelos matemáticos muy exactos, pero extremedamente complejos, utilizando lógica difusa. En dichos estudios se utiliza el modelo autoregresivo difuso para describir las características del tráfico en una red de alta velocidad. El modelo aproxima un proceso no lineal,

variante en el tiempo, con una combinación de varios procesos autoregresivos lineales locales, usando un método de agrupamiento difuso. Todo lo anterior requiere de muchos cálculos matemáticos, con un gran número de parámetros involucrados, por lo cual se requiere de demasiados recursos computacionales para realizar todos los cálculos necesarios.

En el modelo autoregresivo es necesario conocer el tráfico que circula por la red durante un intervalo Δt, para poder predecir el tráfico que circulará durante los siguientes n intervalos Δt. Primero se calcula el tráfico que circulará durante el primer intervalo Δt, y se compara con el tráfico que realmente circuló en dicho instante, se calcula el porcentaje de error ($e = \frac{v_r - v_e}{v_r} * 100$) entre el tráfico calculado y el real, y tomándolo como base se recalcula un parámetro de ajuste, y se repite el procedimiento hasta que el error sea menor a una cierta tolerancia ($e < \epsilon$), en ese momento se considera que el modelo se encuentra listo para predecir el tráfico futuro de los siguientes n intervalos Δt.

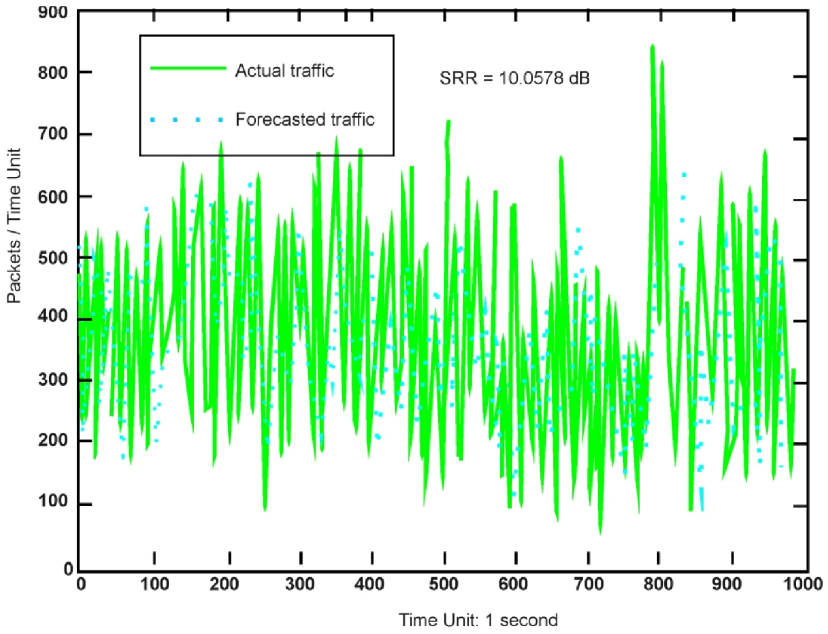


Figura 2.5: Comparación del modelo de lógica difusa contra el tráfico real.

En la figura 2.5 (tomada de [7]) se muestra una gráfica de comparación

entre el modelo autoregresivo creado con lógica difusa (puntos) y el tráfico real de una red (línea contínua). Como puede apreciarse, las diferencias entre el tráfico calculado y el tráfico real son muy pequeñas, de lo cual se deduce que el modelo es muy preciso.

El problema para la utilización de dichos modelos es que requieren de una gran cantidad de recursos para poder predecir el tráfico que circulará por la red, además de una gran cantidad de tiempo, por lo que actualmente no son muy recomendables para comparaciones en tiempo real del tráfico que se encuentra circulando en la red contra un determinado modelo de comportamiento.[7].

Cuando se modela el tráfico colectivo de toda una red local, es decir, cuando se toma a toda la red como un solo elemento, se presenta una reducción en las variables a considerar. Entre las variables más importantes que se deben de tomar en cuenta están las siguientes: tipo de red (red de clientes, de servidores o proveedor de servicios), tipo de tráfico (de entrada o de salida), tipo de día (hábil o no hábil), hora del día, etc.

El tráfico de salida con respecto del tráfico de entrada, normalmente es asimétrico, ya que en una red de servidores el tráfico de salida es superior al de entrada, mientras que en una red de clientes se invierte la proporción. En los días hábiles el tráfico es mucho más intenso que en un día no hábil, esto se debe principalmente al número de usuarios que se encuentran haciendo uso de la red en cada tipo de día, ya que en un día no hábil se encuentra una reducción bastante importante del número de usuarios, con respecto a un día laborable.

El tráfico de una red local no se mantiene constante durante el día. Las variaciones durante el día son bastante importantes, ya que generalmente en la noche y madrugada la cantidad de tráfico es muy pequeña, casi nula, pero durante la mañana y por la tarde el tráfico es mucho más intenso.

Otro de los factores que se deben de tomar en cuenta debido a su importancia es el protocolo que está siendo utilizado por el tráfico que circula a través de la red. En la gráfica 2.6 puede apreciarse la distribución del porcentaje del tráfico que es utilizado por cada protocolo. Los datos para esta

gráfica fueron tomados en la red del CINVESTAV.

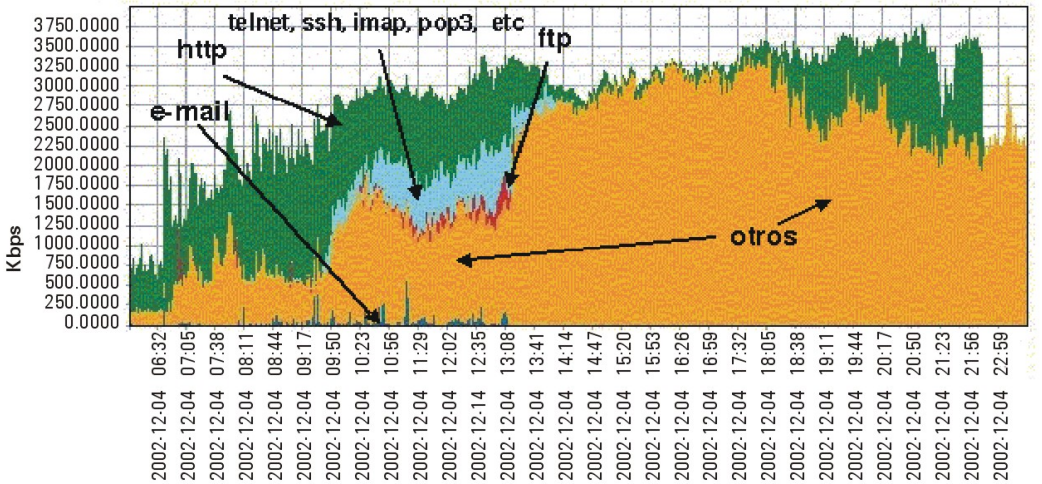


Figura 2.6: Cantidad de tráfico por protocolo. En el eje horizontal se representa el tiempo, mientras que en el eje vertical se representa la cantidad de tráfico en Kbps.

En esta misma gráfica se puede apreciar que se presentan diferentes distribuciones de los protocolos. Esto es debido a que en la primera parte de la gráfica, hasta las 13 hrs aproximadamente, estaba siendo aplicado un filtro para mantener la calidad de servicio. Como puede verse, al quitar el filtro, los otros protocolos (música, juegos, imagenes, etc.) utilizan todos los recursos, quitándole prioridad a los protocolos importantes (HTTP, SMTP, FTP, SSH, IMAP, POP3, TELNET).

En la gráfica 2.7 se presentan para mayor claridad los mismos datos, pero sin tomar en cuenta los otros protocolos.

De todo lo anterior se puede ver la importancia que tiene el verificar el uso que se está haciendo de la red, con el fin de mantener una calidad de servicio adecuada para el propósito de la red.

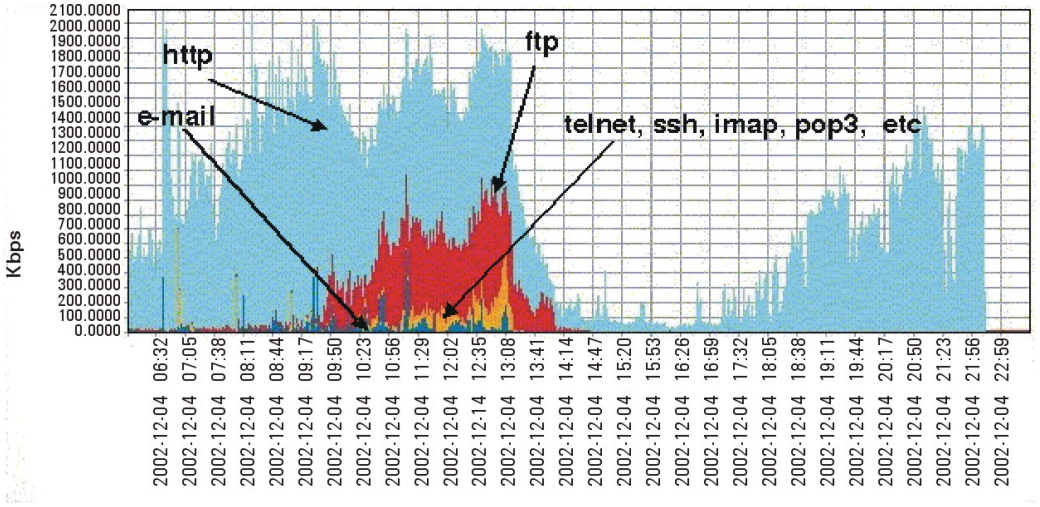


Figura 2.7: Cantidad de tráfico por protocolo, mostrándose únicamente los protocolos importantes. En el eje horizontal se representa el tiempo, mientras que en el eje vertical se representa la cantidad de tráfico en Kbps.

2.3. Utilización del modelo de tráfico

El comportamiento colectivo de una red es con frecuencia lo único que se puede observar de manera práctica en ella. Sin embargo, puede ser de utilidad para verificar la operación normal, para detectar fallas, para propósitos de seguridad y para medir el uso de los canales de comunicación:

1) Operación Normal. Sirve para poder decidir si la red se encuentra en operación normal, lo cual es dependiente del día y hora en la que esté operando la red. La cantidad de tráfico que se encuentra circulando normalmente en una red es dependiente de la hora y minuto del día, así como del día de que se trate, si el día es laborable, fin de semana o festivo. El tráfico a considerar es el entrante y saliente de la red, por lo que es esencial conocer las direcciones tanto fuente como destino, así como los puertos fuente y destino y el protocolo que está siendo utilizado. En la figura 2.8, se muestra el tráfico entrante como un color sólido y el saliente como una línea, de una red durante un día normal de operación.

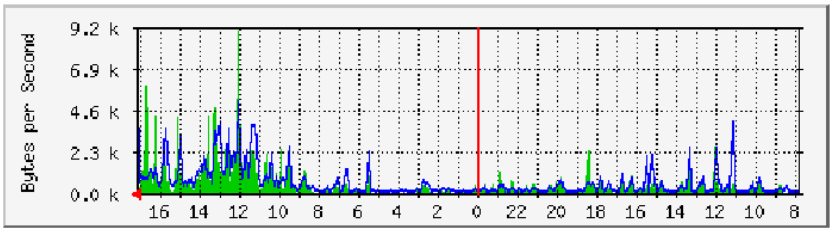


Figura 2.8: Tráfico normal de una red.

En la figura 2.8 pueden percibirse varios de los factores a considerar en el modelado de tráfico. Se ve claramente el comportamiento por ráfagas que se generan a partir de solicitudes individuales. Así también, el comportamiento colectivo hace que en un horario normal de trabajo, el tráfico tiende a incrementarse entre las 12 y 14 horas. Sube gradualmente y después desciende conforme la jornada de trabajo termina. Se observa también, que modelar cada uno de los factores puede ser útil pero para observar comportamientos generales no es tan importante como observar las tendencias del tráfico. Se puede observar los efectos del tráfico por ráfagas y percibir el comportamiento de auto-similitud, ambos ponderados por el tráfico colectivo durante la hora del día.

2) Detección de Fallas. El modelo de tráfico se puede usar también para saber si existe una falla en la red, o si está fallando alguno de los nodos, servidores, ruteadores, cortafuegos, etc. con tan sólo observar la gráfica del tráfico. En la figura 2.9 se presenta una falla en la red, la cual es visible por presentar la característica de tráfico constante, ya que normalmente el tráfico que circula por la red no es constante por períodos de tiempo muy prolongados. Cabe mencionar que cada tipo de falla, presenta un patrón de comportamiento diferente, pero bien definido para cada caso.

Así también, la falla a través del modelo de tráfico se puede detectar solo después de un período de tiempo prolongado. En la figura 2.9 se ve claramente que la falla se presenta durante cuatro horas de muestreo. Unas cuantas muestras no son suficientes para determinar la existencia

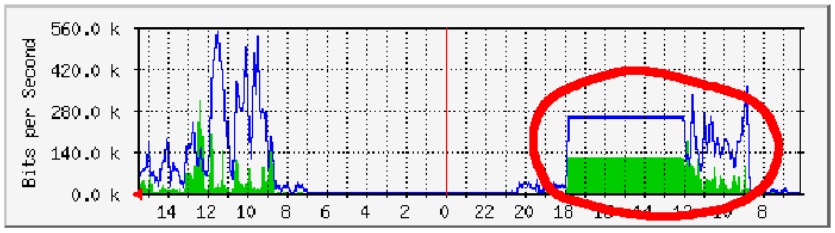


Figura 2.9: Tráfico de una red durante una falla.

de una falla.

3) Seguridad. Se pueden detectar los ataques que se realizan a la red o a servidores específicos así como escaneos de vulnerabilidades a las computadoras de la red, virus que circulen por la red, así como gusanos, etc. con sólo observar una gráfica del tráfico que existe en la red en un momento determinado. En la figura 2.10 se puede observar un ataque a la red, el cual se caracteriza por un tráfico elevado en un período de tiempo muy corto.

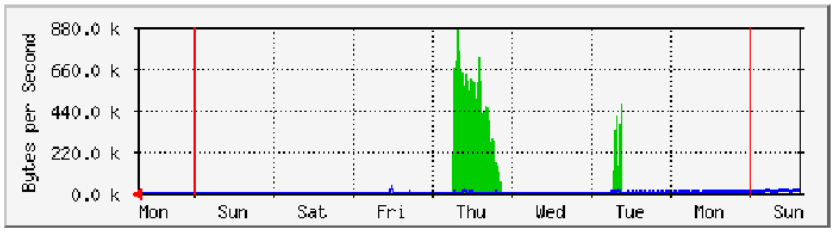


Figura 2.10: Tráfico de una red durante un ataque.

A diferencia de las fallas, cuando se presenta un ataque se nota un tráfico inusualmente alto, sostenido el cual no es necesariamente constante; el efecto de los trenes y ráfagas aún se puede percibir durante el ataque.

4) Uso de los Canales de Comunicación. Se puede observar cual es el uso que los usuarios le están dando a los canales de comunicación, y así es posible brindar una mayor calidad de servicio, ya que se puede reconfigurar la red para redistribuir el ancho de banda de acuerdo al

uso, servicios que se ofrecen, o necesidades específicas de la red. En la figura 2.11 se presenta un análisis sobre el ancho de banda disponible por usuario, dependiendo del número total de usuarios, así como de el ancho de banda total disponible por usuario y como se vería afectado al cambiar las variables de que depende dicho ancho de banda por usuario [12].

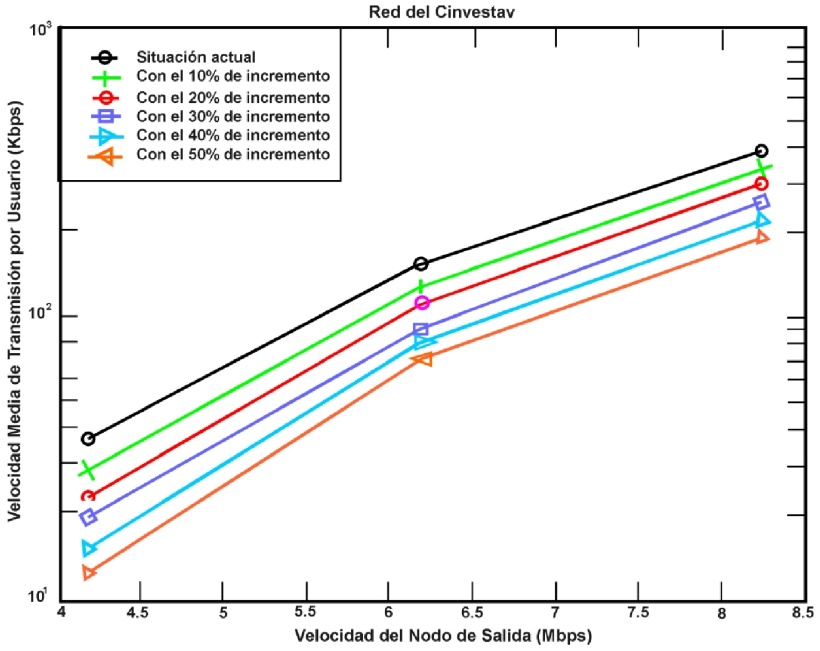


Figura 2.11: Velocidades de acceso por usuario para diferentes anchos de banda y diferente número de usuarios.

Como puede apreciarse en la figura 2.11, el ancho de banda por usuario va aumentando conforme se aumenta el ancho de banda total, pero disminuye conforme se registra un aumento en el número de usuarios.

Esta última figura fué calculada usando un modelo de simulación M/M/1 (una sola cola, un solo servidor y un solo tipo de usuario), tomando en cuenta las siguientes consideraciones:

- No se considera la arquitectura interna de la red, se consideran todas las computadoras conectadas directamente al enlace de salida.

- Se asigna la velocidad total de transmisión a cada usuario que arriba.
- Si un usuario al arribar, encuentra ocupado el enlace, se forma en una cola FIFO y espera hasta que se le asigne el enlace.
- El arribo de los usuarios es uniforme y se asignan mediante una distribución de Poisson, con $\lambda = 48{,}0$ sesiones por segundo.
- El número de paquetes en cada sesión usa una distribución geométrica con $\frac{1-p}{p} = 10$ paquetes.
- El tiempo entre los paquetes es una distribución de Pareto, con $\mu = 10$ segundos y $\alpha = 1{,}4$.
- El tamaño de los paquetes es una distribución lognormal con $e^{\mu+\frac{\sigma^2}{2}} = 4{,}1$ kbytes y una $e^{2\mu+\sigma^2}(e^{\sigma^2}-1) = 30$ kbytes.

Esta simulación se realizó para diferentes velocidades del enlace de salida, desde 4.096 Mbps (2 enlaces E1) hasta 8.192 Mbps (4 E1= E2), así como las características del tráfico web, la taza de arribo de sesiones ($\lambda = 36{,}0$ sesiones por segundo) y la velocidad media de transmisión por cada usuario (32 Kbps).

Estas mismas simulaciones (diferentes velocidades de transmisión) se repitieron para diferentes números de usuarios, los cuales se fueron incrementando desde un 10 %, hasta un 50 %.

La velocidad media de transmisión por usuario fué proporcionada por la Coordinación General de Serviciós de Computo Académico (CGSCA), mientras que los otros datos están basados en simulaciones anteriormente realizadas ([13][10]).

2.4. El modelo estadístico

La toma de datos para la creación del modelo estadístico se realizó en la red del CINVESTAV, la cual es una red clase B (148.247.x.x) y cuenta actualmente con 60 subredes, con aproximadamente 1800 nodos conectados.

Dicha red cuenta con un enlace a internet de 12 Mbps. En ésta red se realizan aproximadamente 1000000 de solicitudes HTTP diariamente (en un día hábil), así como también se reciben unos 5000 correos electróncos y se envían 3000 en el mismo período de tiempo.

La red del CINVESTAV presenta un cierto patrón de comportamiento, como puede apreciarse en las gráficas 2.12, 2.13 y 2.14.

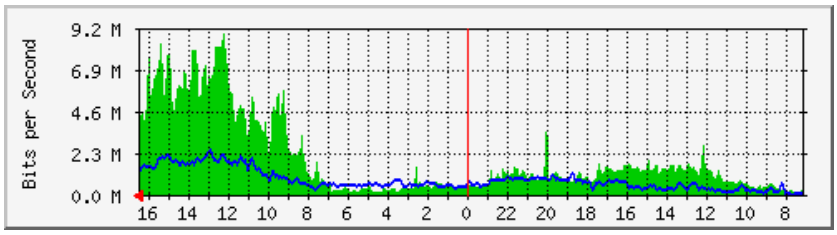


Figura 2.12: Tráfico de una subred durante un día.

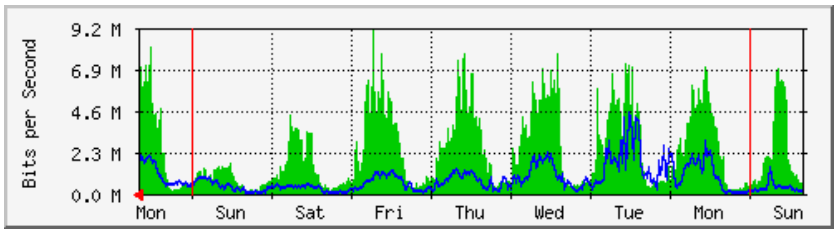


Figura 2.13: Tráfico de una subred durante una semana.

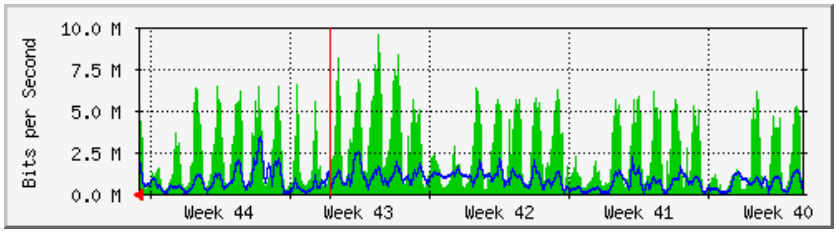


Figura 2.14: Tráfico de una subred durante un mes.

Lo primero que se puede ver en éstas gráficas es la asimetría del tráfico de entrada (con color sólido) y del tráfico de salida (una línea). Se observa

también que la cantidad de tráfico es totalmente dependiente de la hora del día, siendo menor durante las noches y aumentando en el transcurso del día. Por último también es posible observar que el tráfico depende del tipo de día, observándose una reducción del tráfico durante los fines de semana.

Por todo lo anterior, un problema interesante se refiere a la obtención de un mecanismo eficiente y adecuado para poder analizar el tráfico de todos los paquetes que circulan a través de una red local.

Para lograr lo anterior el sistema debe de responder solo unos minutos después de que el problema empiece a ocurrir. No es muy deseable enterarse de que se presentó un problema en la red una semana o unos días después, ya que el problema podría ser solucionado o minimizado si se conoce en el momento adecuado.

En una red grande, como la del CINVESTAV, muchos de estos problemas son detectados tiempo después de que ocurrieron, o pasan totalmente desapercibidos.

Por todo esto, un modelo adecuado para una red como la del CINVESTAV, sería un modelo estadístico, ya que no es necesario contar con una gran precisión, pero si es importante contar con un tiempo de respuesta corto, de solo unos minutos.

Debido a las características propias de la red, y a los requerimientos de uso del modelo, es suficiente tomar en cuenta el tipo de día, la hora del día y el tipo de tráfico (entrada y salida) para la construcción del modelo estadístico.

Como se presentan grandes diferencias en los tipos de tráfico, se procedió a la construcción de dos modelos, uno para el tráfico de entrada y otro para el tráfico de salida (figura 2.15).

En éstas dos gráficas de los modelos estadísticos, en el eje horizontal se encuentra representado el tiempo, el cual va desde las 0 hasta las 24 horas, y en el eje vertical está representado el número de paquetes que circulan por unidad de tiempo.

Ambos modelos fueron crados a partir de una gran cantidad de datos recolectados previamente de la subred 148.247.1.x/24 del CINVESTAV. Una

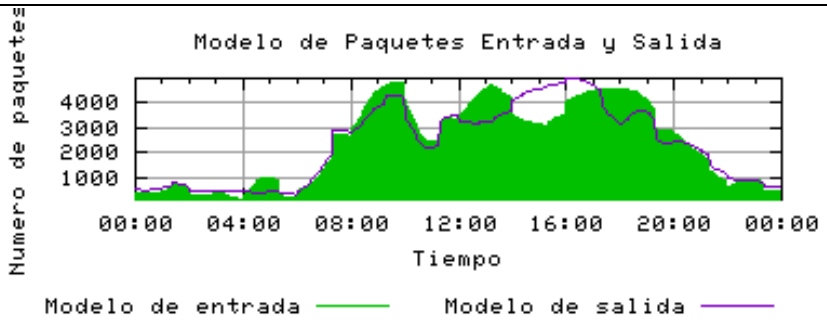


Figura 2.15: Modelo del tráfico de entrada (color sólido) y salida (una línea).

vez que se contó con los datos, se procedió a seleccionar las muestras representativas de cada día. Las muestras selecionadas fueron promediadas. Los datos promediados fueron sometidos a un proceso se suavización y se realizó un ajuste por el método de trazador cúbico.

A partir del comportamiento colectivo de la red modelada es posible agregar ráfagas de transmisión y el comportamiento generado por trenes de comunicación.

En el capítulo 2 se describe con detalle la construcción del modelo estadístico.

Capítulo 3

Captura y modelación del tráfico

En este capítulo se describirán los diferentes métodos existentes para capturar el tráfico que circula a través de las redes, el método utilizado para lograrlo y la creación del modelo estadístico con el tráfico capturado.

3.1. Captura del tráfico

Para poder obtener un modelo estadístico es necesario capturar una gran cantidad de paquetes que circulen por la red, para después proceder a su análisis estadístico.

Los métodos de medición de una red se pueden clasificar como métodos activos y métodos pasivos según su naturaleza, pero también se pueden clasificar basados en hardware o software, de acuerdo a como están construidos.

Los métodos activos reciben éste nombre debido a que el dispositivo que se está utilizando para llevar a cabo la medición genera, él mismo, algunos paquetes para poder probar y medir las características de una red.

Algunos ejemplos simples de métodos activos son: ping, traceroute y patchar.

ping: Esta herramienta nos ayuda a realizar una estimación sobre la latencia de la red, hacia una computadora destino en particular. Esta herramienta lanza paquetes ICMP (ECHO_REQUEST) hacia el nodo destino,

midiendo el tiempo que tarda en llegar la respuesta (ECHO_REPLY).

traceroute: Esta herramienta ayuda a determinar el camino que siguen los paquetes para llegar hacia una computadora destino, permitiendo de ésta manera revisar el ruteo de los paquetes en la red. Esta herramienta manda paquetes IP a el siguiente gateway con el campo TIME_TO_LIVE exacto para que el gateway responda con un paquete ICMP TIME_EXCEEDED, se utiliza el puerto 33434 UDP, el cual va incrementando en cada salto.

pathchar: Esta herramienta, desarrollada por Van Jacobson (al igual que traceroute) ayuda a estimar el ancho de banda, retraso y encolamiento de los paquetes, en cada salto a través de una determinada ruta. Esto lo hace mandando muchos paquetes de diferentes tamaños, varias veces, midiendo el tiempo que tardan en regresar, y realizando los cálculos necesarios para obtener los datos requeridos.

Los métodos pasivos de medición de una red monitorean todos los paquetes que circulan por ella, sin generar ningún tipo de tráfico adicional, sin embargo, la captura de paquetes puede llegar a degradar la calidad de servicio de la red debido a que se pueden introducir retrasos en el transporte de los paquetes.

La mayoría de las herramientas de medición que existen y que se utilizan, pertenecen a ésta categoría, un ejemplo de éstas herramientas son los *sniffers*. Un *sniffer* es una utilidad que captura, interpretan y almacenan los paquetes que circulan en una red, para su posterior análisis.

Con relación a los métodos basados en hardware o software, se tienen factores adicionales que pueden ser usados para analizar sus ventajas y desventajas, entre los cuales los factores más importantes son el costo, los retardos que se introducen en la red y si el análisis de los paquetes se realiza en línea o fuera de línea [14][15].

Cada una de éstas herramientas presentan ventajas y desventajas, como son el costo, tiempo de análisis, los retardos que introducen la red, etc. [14][15]

3.1.1. Sistemas de captura basados en hardware

La captura mediante hardware se realiza de una manera muy rápida y eficiente, pero para poder llevarse a cabo de una manera apropiada, se requiere contar con el equipo adecuado, como podría ser un equipo *NavTelIW96000*[14], mostrado en la figura 3.1.

Figura 3.1: Equipo NavTelIW96000, con sus aditamentos.

Este es un equipo especializado para capturar todo el tráfico que circule en una red, ya sea WAN o LAN y puede realizarlo independientemente de la tecnología utilizada, como puede ser Token Ring, ATM, Ethernet, Fast Etehernet, etc.

La desventaja de este equipo es que requiere de instalaciones específicas, como son temperatura, voltaje y corriente ininterrumpida, etc. Requiere también los módulos apropiados y las tarjetas necesarias de acuerdo a la tecnología que se esté utilizando en la red en ese momento.

Otro sistema basado en hardware es el *WinPharaoh*, el cual se muestra en la figura 3.2, este equipo tiene la capacidad de capturar el tráfico de las

redes locales y de área amplia, y analizar dicho tráfico de manera estadística, realiza detección de errores, en tiempo real.

Figura 3.2: Equipo WinPharaoh.

El equipo *WinPharaoh* también es capaz de aplicar filtros al tráfico de la red, para facilitar su análisis, y provee también un emulador de tráfico para realizar pruebas en la red y/o simulaciones de ciertas situaciones.

Este equipo es totalmente portátil, y cuenta con diferentes adaptadores de red integrados, como son ATM, Ethernet, etc. para utilizarse según las necesidades.

La desventaja de este tipo de equipos es la incompatibilidad con otros sistemas, ya que usa un formato propio no estándar, por lo que otros sistemas no pueden leer o interpretar los datos capturados por este equipo, además de su falta de capacidad para crecer, es decir, no es escalable, lo que es limitante, ya que no es posible agregar más tarjetas, procesadores, memoria, etc.

El equipo *Inter Watch*, que se muestra en la figura 3.3 cuenta con una arquitectura multiprocesador, módulos de análisis mediante hardware recon-

figurable, análisis de las siete capas del modelo OSI en tiempo real.

Figura 3.3: Equipo InterWatch Box.

El *Inter Watch* se puede obtener en versión rack o como equipo portátil, ambos con las mismas capacidades. Este hardware también cuenta con soporte gráfico para multiusuarios remotos, así como software compatible multiplataforma.

El equipo presenta una gran capacidad para ser escalado según las necesidades que se tengan en la red, ya que se puede aumentar desde el número de procesadores, adaptadores de red, capacidad de almacenamiento, hasta el software, como es el sistema operativo, programas, etc.

Tiene gran compatibilidad, ya que todas las gráficas, reportes y archivos que genera pueden ser guardados en diferentes formatos, para que puedan ser usados sin ningún problema en otros sistemas.

En la Tabla 3.1 se presenta un cuadro comparativo de los equipos de hardware anteriormente mencionados:

Ninguna de las compañías que venden este tipo de hardware publica los precios de este tipo de productos, sin embargo, éstos varían, dependiendo de sus capacidades y accesorios adicionales, desde los $75000 dólares hasta los $130000 dólares.

Tabla 3.1: Tabla de comparación de equipos de hardware de captura de datos.

	NavtelIW9600	**Winpharaoh**	**InterWatch**
Captura de datos	sí	sí	sí
Análisis de datos	sí	sí	sí
Análisis en tiempo real	sí	sí	sí
Multiusuarios	no	no	sí
Compatibilidad	media	no	alta
Generadores de tráfico	sí	sí	sí
Simuladores de tráfico	no	no	sí
Análisis de las 7 capas	no	no	sí
Reconfigurable	no	no	sí
Equipo Adicional	sí	no	opcional
Portátil	sí	sí	opcional
Procesador	sparc	celeron	powerpc
Escalabilidad	sí	no	sí

Por lo anterior, el costo de los sistemas de captura basados en hardware es su principal desventaja.

3.1.2. Sistemas de captura basados en software

La captura mediante software es mucho menos costosa, económicamente hablando, y puede realizarse desde alguna estación de trabajo, dotando a la interfaz de red de ciertas capacidades para capturar los paquetes mediante algún programa especializado, aún cuando la red se encuentre segmentada. Algunos de los programas que pueden usarse son *tcpdump* o *snort*[14], o cualquier otra herramienta de software desarrollada especialmente para captura de paquetes. La captura tiene lugar al momento en que los paquetes pasen por un ruteador o mecanismo equivalente, evitando que ésto se convierta en un cuello de botella.

Uno de los paquetes de captura más comunes es *tcpdump*, el cual está presente en la mayoría de los sistemas Unix, pero se limita únicamente a capturar los paquetes, ya sea en modo binario o ASCII y los presenta en la pantalla;

también tiene la opción de guardarlos en un archivo. Este paquete no presenta ninguna herramienta de análisis o de seguimiento de paquetes. Es muy fácil de usar, debido a sus reglas simples, pero su código no está optimizado por lo que puede degradar gravemente la calidad de servicio de la red, si es utilizado por largos periodos de tiempo. Desafortunadamente, tampoco cuenta con ningún tipo de conexión a bases de datos o visualizadores gráficos o de web. Esta herramienta solo es recomendable para tomar algunas muestras del tráfico de la red, por períodos cortos de tiempo.

Snort es un paquete que, además de capturar paquetes, también puede ser usado como detector de intrusos en tiempo real. Se puede colocar como demonio para ocupar menos recursos, tiene su propio lenguaje de reglas, las cuales son fácilmente modificables para adaptarlas a las necesidades de cada red. También cuenta con programas adicionales para poder realizar un análisis más eficiente, y con una interfaz web para poder consultar los resultados a través de la red. Una desventaja es que conforme crece el tamaño de la red y la complejidad de la topología de la misma, puede resultar muy complicado de configurarlo correctamente. Más aún puede requerir una gran cantidad de espacio en disco duro para guardar los paquetes y estadísticas en su base de datos. Es posible tener un monitor y varios sensores distribuidos a lo largo de una o varias redes, pero requiere de protocolos de comunicación que pueden comprometer la seguridad de la red.

Ethereal es un paquete de captura totalmente gráfico y presenta una muy buena herramienta de seguimiento y análisis estadístico de paquetes, pero no presenta un modo de captura binario, ni puede realizar el análisis ni el seguimiento en línea. Esta herramienta no es recomendable para realizar una recolección de datos ya que es muy lenta, pero en cambio es perfectamente útil para realizar el análisis "manual" de datos recolectados previamente. Por su lentitud solo es recomendable utilizar muestras pequeñas para su análisis.

Los tres paquetes anteriormente mencionado están basados en la biblioteca *libpcap*, la cual es distribuida mediante la licencia GPL. Esta biblioteca provee de una interface de alto nivel para sistemas de captura de paquetes.

En la Tabla 3.2 se presenta un cuadro comparativo entre los paquetes

anteriormente mencionados.

Tabla 3.2: Tabla de comparación de software de captura de datos.

	ethereal	**snort**	**tcpdump**
Velocidad de captura	lenta	rápida	aceptable
Archivos binarios	solo lectura	sí	sí
Eficiencia de captura	baja	alta	alta
Modificabilidad	difícil	fácil	media
Soporte de base de datos	no	sí	no
Interfaz gráfica de uso	sí	no gratuita	no
Interfaz gráfica de presentación	sí	independiente	no
Análisis de paquetes	sí	sí	no
Filtrado de paquetes	sí	sí	sí
Detección de intrusos	no	sí	no
Modificación de reglas	no	sí	no

La principal desventaja de los métodos basados en software es la velocidad, la cual no es comparable con la observada mediante sistemas de hardware y podría ser causa de un detrimento bastante significativo en la eficiencia de la red. Sus ventajas más importantes son el no requerir de equipos especializados, ni de instalaciones especiales, por lo cual no se tienen las restricciones económicas de la opción anterior.

Al realizarse la captura mediante software, lo más conveniente es realizar un análisis fuera de línea de los datos obtenidos, lo anterior con el fin de evitar que se degrade la calidad de servicio y operación de la red.

Como puede apreciarse en la Tabla 3.2, el paquete de software con mejores caracteristicas es *snort*, por lo que es el utilizado en el desarrollo del presente trabajo.

3.2. Recolección de datos

El tráfico que nos interesa capturar es el que entra o sale de una red, es decir, todos los paquetes que son creados en nuestra red y tienen un destino

fuera de la misma, o bien, aquellos paquetes que fueron creados por computadoras externas a nuestra red, pero que tienen como destino cualquier computadora de nuestra red.

Para poder realizar la captura de todos los paquetes que circulan por toda la red, depende de la tecnología con la que está construida. A continuación se describirán varios casos de una red ethernet (o fast-ethernet o gigabit-ethernet), por ser ésta la tecnología más comúnmente utilizada en las redes.

3.2.1. Red ethernet

En las redes ethernet es necesario colocar un dispositivo que realize dicha operación a la salida de la red, es decir en la puerta de enlace predeterminada de la red. Lo anterior debe de realizarse de forma lógica, y no necesariamente se refiere a la configuración física de la red.

Dependiendo de las conexiones de la red, la captura se puede realizar como se muestra en la figura 3.4, en la cual se observa una de las posibles configuraciones físicas para una red ethernet.

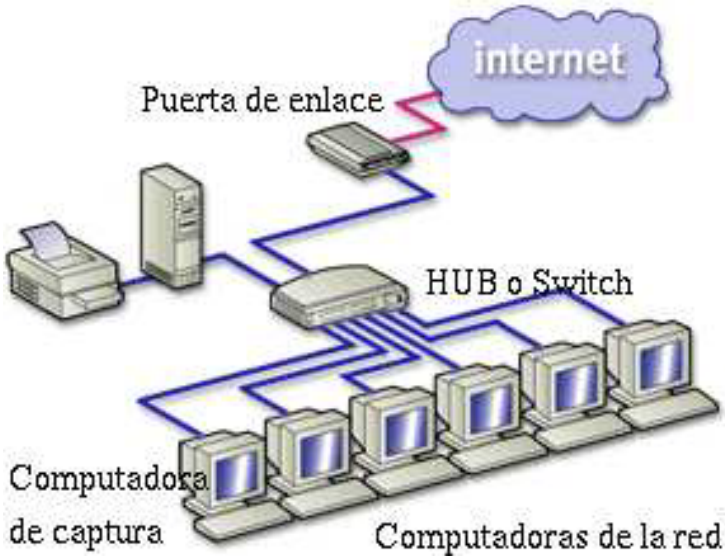


Figura 3.4: Configuración física de una red ethernet.

Totalmente conectada

Una red totalmente conectada es aquella en la que las computadoras se encuentran enlazadas a través de un concentrador o HUB. En este tipo de conexión, todos los integrantes de la red se encuentran escuchando todo lo que circula por la red, ya que comparten un solo canal lógico de comunicación, en el cual todos depositan sus mensajes y regresan a modo de escucha.

Este modelo es un poco lento, ya que si dos equipos intentan transmitir de manera simultánea, se presenta una colisión, y ambos equipos deberán volver a intentar retransmitir su mensaje.

En éste tipo de conexión, basta con poner la interfase de red de la computadora que realizará la captura, en modo promiscuo, ya que de ésta forma recolectará todos los paquetes que sean puestos en el canal de comunicación, aunque no sean dirigidos a ella. En la figura 3.5 se muestra la configuración lógica de una red totalmente conectada.

En éste tipo de red, es posible capturar todo el tráfico existente en la red, tanto el tráfico interno como el externo.

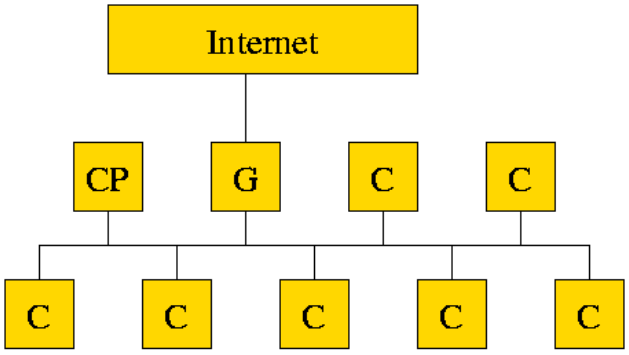


Figura 3.5: Configuración lógica de una red ethernet totalmente conectada. G=puerta de enlace, CP=captura de paquetes, C=computadora

Red segmentada

Si se tiene el caso de que la red ethernet se encuentre conectada a través de un switch, la red se encuentra segmentada. El problema se complica, ya

que no basta con poner la interfaz de red en modo promiscuo, pues no se lograrán capturar todos los paquetes que circulen por la red, solamente se capturarán los paquetes que se encuentren circulando en el mismo segmento al cual se encuentra conectado nuestro dispositivo de captura.

Si la red se encuentra segmentada, es necesario poner la computadora encargada de la captura entre la red y la puerta de enlace, obligando de ésta forma que todos los paquetes pasen a través de la computadora encargada de la captura de los paquetes. En éste caso solo es posible capturar el tráfico externo.

Lo anterior se puede lograr mediante una técnica llamada *hombre enmedio*, la cual consiste en mandar contestaciones ARP a todas las computadoras de la red, excepto a la puerta de enlace predeterminada. En el paquete de respuesta ARP, se anuncia la dirección MAC de la computadora de captura, como si fuera la dirección MAC de la puerta de enlace predeterminada. De ésta manera, todas las computadoras de la red mandan sus paquetes a la computadora de captura.

En la Tabla 3.3 se muestra la Tabla ARP de una computadora cliente que se encuentra en una red segmentada, en la cual se están realizando mediciones del tráfico que entra o sale de ella. En dicha tabla, la puerta de enlace real es la computadora 192.168.100.254 y la encargada de la captura de paquetes es la computadora 192.168.100.200. Todas las demás computadoras son clientes de la red.

Tabla 3.3: Tabla ARP de un cliente en una red donde se aplicó la técnica de *hombre enmedio*.

Address	HWtype	HWaddress	Flags	Mask	Iface
192.168.100.1	ether	00:0A:95:79:B8:54	C		eth0
192.168.100.2	ether	00:04:76:B9:79:57	C		eth0
192.168.100.3	ether	00:30:65:0A:DC:B7	C		eth0
192.168.100.4	ether	00:01:02:EB:A6:3F	C		eth0
192.168.100.200	ether	00:01:03:E9:45:42	C		eth0
192.168.100.254	ether	00:01:03:E9:45:42	C		eth0

En la Tabla 3.4 se muestra la tabla ARP de la puerta de enlace de la misma red segmentada.

Tabla 3.4: Tabla ARP de un cliente en una red donde se aplicó la técnica de *hombre enmedio*.

Address	HWtype	HWaddress	Flags	Mask	Iface
192.168.100.1	ether	00:01:03:E9:45:42	C		eth0
192.168.100.2	ether	00:01:03:E9:45:42	C		eth0
192.168.100.3	ether	00:01:03:E9:45:42	C		eth0
192.168.100.200	ether	00:01:03:E9:45:42	C		eth0

De ésta forma, la configuración lógica de la red cambia totalmente. En la figura 3.6 se muestra la configuración lógica de una red segmentada, despúes de aplicar la técnica *hombre enmedio*.

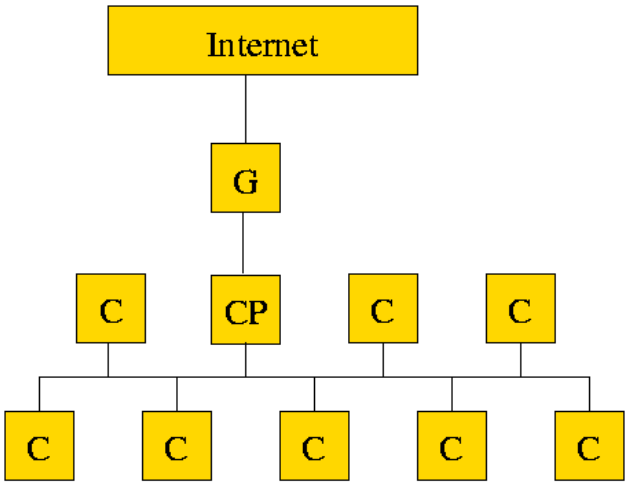


Figura 3.6: Configuración lógica de una red segmentada. G=puerta de enlace, CP=captura de paquetes, C=computadora

Todo lo anterior funciona para varios niveles de segmentación, como podría ser una red que cuente con varios switch o bridges en cascada. Esto es debido a que una o varias de las computadoras de la red, prodrían ser las puertas de enlace de algunas subredes internas. Si éste es el caso, todos los

paquetes enviados o recibidos por las computadoras de las subredes, simplemente serán capturados como si fueran creados por la computadora que sirve como puerta de enlace de esa subred.

Dentro de ésta configuración, es necesario que la computadora dedicada a la captura de datos, sea una computadora con mucha capacidad, sobre todo con tarjetas de red confiable y una gran cantidad de memoria, a fin de tratar de evitar que se convierta en un cuello de botella y provoque un degradamiento considerable en la calidad de servicio de la red en la que está realizando las mediciones.

No es necesaria una gran capacidad en disco duro, ya que los datos pueden ser borrados en cuanto son enviados a otra computadora que se encargue de realizar el análisis de los datos recolectados. Tampoco es necesario un procesador muy poderoso, ya que el análisis se puede llevar a cabo en otra computadora.

3.3. Creación del modelo estadístico

El modelo fué creado sobre la subred 148.247.1.0/24 del CINVESTAV, Esta subred cuenta con aproximadamente 10 servidores (HTTP, DNS, POP3, SMTP, etc.) y con aproximadamente 25 computadoras clientes, conectadas mediante un switch, es decir, la red se encuentra segmentada y en dicha red se tiene una latencia de 0.334924 ms aproximadamente. El muestreo de datos se realiza de manera discontínua, en períodos de 5 minutos de muestreo por 5 minutos de descanso. En estos muestreos se recoge un promedio de 7.5Mb de información, la información más grande recogida en ese período es de 150Mb aproximadamente, y la más pequeña de 4Kb. En un solo día, entran en promedio 19700 paquetes, y salen 19980 paquetes en promedio.

Para poder crear un modelo estadístico, es necesario contar con una gran cantidad de datos. Para lograr esto, se procedió a realizar una gran recolección de datos, basada en la configuración descrita anteriormente.

Los pasos para la creación de un modelo estadístico son: la recolección de datos, promediar los datos, suavización de los mismos, y finalmente un

ajuste.

La recolección de los datos fué realizada mediante muestreos durante las 24 horas, con una duración de cinco minutos cada uno de los muestreos, con cinco minutos de descanso entre cada uno de ellos. Esto es con el propósito de lograr una gran cantidad de datos representativos, pero sin degradar la calidad de servicio de la red en la que se están realizando los muestreos.

Una vez que se contó con todos los datos de varios meses, se procedió a descartar los días que no se consideran típicos, es decir, todos aquellos días en los que sucedieron algunas anormalidades en el tráfico que circuló por la red como podría ser un ataque, alguna falla en la red o un tráfico inusualmente alto o inusualmente bajo.

Una vez que fueron selecionados los datos, se tomaron los promedios de cada minuto de cada día, con la ecuación 3.1, haciendo distinción entre los días de trabajo y los fines de semana, ya que en estos últimos el tráfico es muy diferente al resto de la semana, como se mostró en el capítulo anterior.

$$\bar{x} = \sum_{i=1}^{n} \frac{x_i}{n} \tag{3.1}$$

donde n es el número de datos y x_i es el i-ésimo dato.

En la figura 3.7 se muestran los datos promediados que se obtuvieron mediante la ecuación 3.1, aplicada a todos los datos recolectados previamente durante meses. En ésta misma gráfica se puede apreciar el efecto de las ráfagas, a pesar de la promediación.

3.3.1. Suavización de los datos

La suavización debe de ser realizada ya que la curva promediada presenta grandes oscilaciones, y de ésta forma no es muy apta para la comparación con los datos que se reciben ni tampoco para la creación de un modelo estadístico en el que la velocidad es un factor muy importante. En realidad solo se necesita ver el comportamiento general del tráfico, no el comportamiento segundo a segundo del mismo.

Otra de las razones para realizar la suavización es que los datos presentan

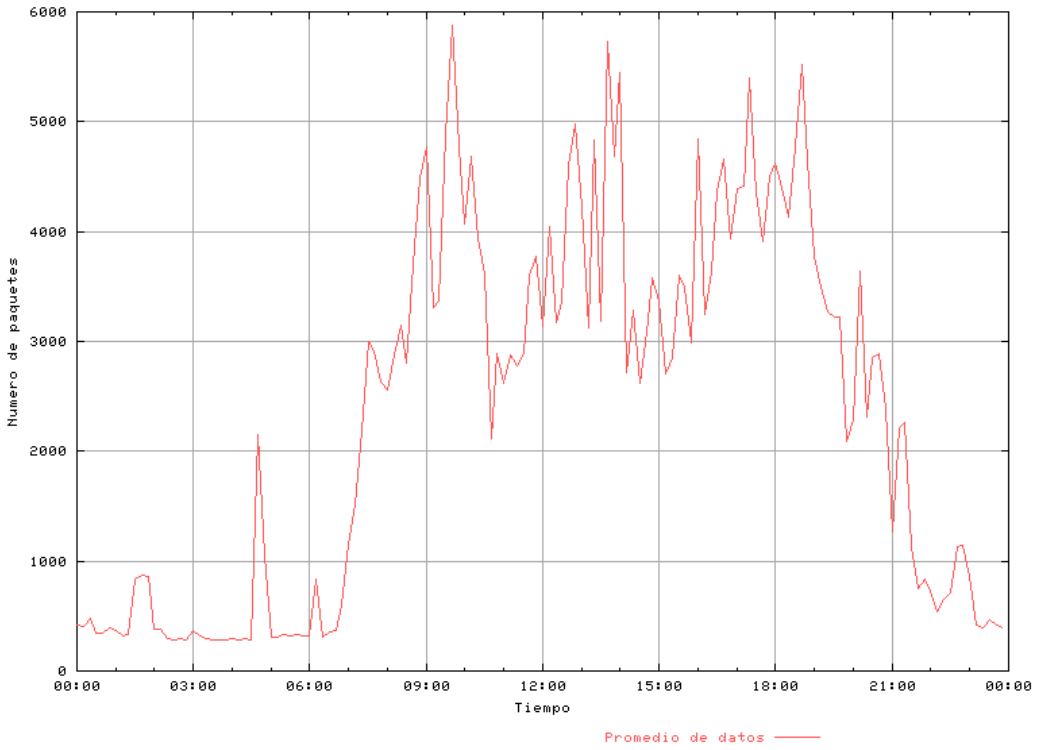


Figura 3.7: Datos promediados.

un comportamiento individual de ráfagas, el cual es semi-aleatorio, pero en el presente trabajo se ha puesto énfasis en el comportamiento colectivo de la red.

Ya contando con los datos promediados, se procedió a realizar una suavización de los mismos, mediante la función:

$$y_i = \begin{cases} y_i & \text{si} \quad i \leq d \\ \\ \frac{1}{d} \sum_{k=1}^{d} y_{i-k-1} & i > d \end{cases} \tag{3.2}$$

Esta función (3.2) es ampliamente utilizada en el procesamiento de señales, para realizar la suavización los datos que son recibidos, antes de ser procesados, eliminando el ruido de ésta forma. La función realiza un promedio de los útlimos d datos, y para el caso $d = 4$ se tratan de los útimos 15 minutos.

En la figura 3.8 se puede apreciar la suavización de los datos para di-

ferentes valores de d, en la cual se pueden apreciar las características anteriormente mencionadas sobre los resultados de las pruebas realizadas para diferentes valores del parametro d de la ecuación 3.2.

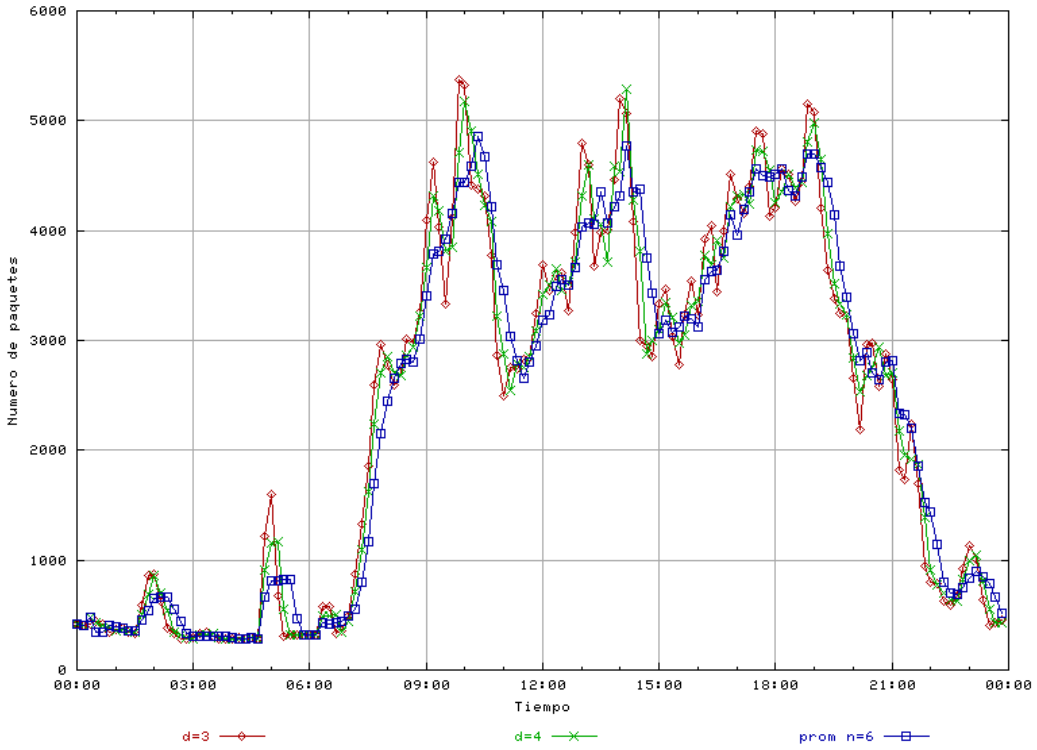


Figura 3.8: Suavización de los datos para diferentes valores de d.

De las pruebas se obtuvo que si $d < 4$, entonces la suavización es muy pobre, pero si $d > 4$ la suavización es tan grande, que se pierden las características esenciales de la curva, por lo que de ésta forma se determinó que el parámetro d adecuado es 4.

En la figura 3.9 se muestran los datos suavizados con $d = 4$.

3.3.2. Ajuste de los datos

El ajuste de la curva es realizado para poder realizar una interpolación de los datos, y de ésta forma obtener los datos minuto a minuto.

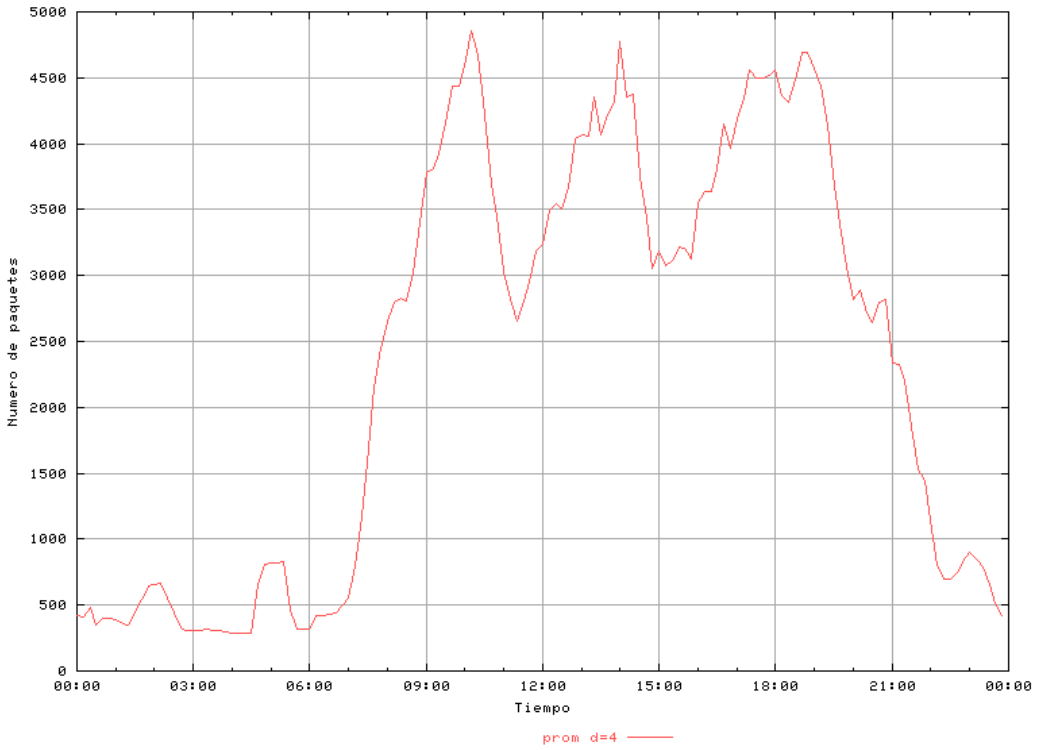


Figura 3.9: Datos suavizados para $d = 4$.

Una vez que se cuenta con todos los datos promediados y suavizados, se procedió a realizar un ajuste de la curva, mediante el método de la interpolación cúbica de trazador[16]. Dicho método fué utilizado debido a la naturaleza oscilatoria que presentan los datos promediados.

El método de trazador cúbico es el método de trazador, de tercer orden. Se eligió este método debido a que garantiza que la curva obtenida pasa por todos y cada uno de los puntos del conjunto original de datos.

El método consiste en tomar m datos (en nuestro caso $m = 4$, por ser cúbico), y ajustar los coeficientes de una ecuación de tercer grado, como la mostrada en la ecuación 3.3.

$$ax^3 + bx^2 + cx + d = 0 \tag{3.3}$$

Los coeficientes solo son válidos en el intervalo entre los dos primeros

datos ($[x_j, x_{j+1}]$) del grupo de m datos. Se descarta el primer elemento de los m datos y se incorpora el siguiente elemento del conjunto de n datos (se recorre el subintervalo $[x_j, x_{j+1}]$ para cada $j = 0, 1, \ldots, n-1$) y con ese nuevo grupo se ajustan nuevamente los coeficientes de la ecuación de tercer grado. Lo anterior se repite realizando un barrido de todo el conjunto de n datos.

El método arroja un conjunto de coeficientes ($\vec{x}$) para $n-1$ ecuaciones de tercer grado (S_j). Para encontrar todos los coeficientes solo se tiene que resolver la ecuación vectorial 3.4.

$$A\vec{x} = \vec{b} \tag{3.4}$$

Una vez solucionada esta ecuación vectorial, se tienen todos los coeficientes a_j, b_j, c_j, d_j, para $j = 0, 1, \ldots, n-1$. Así tenemos todos los polinomios cúbicos $S(x) = S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3$ para $x_j \leq x \leq x_{j+1}$.

Después se evaluan los o puntos intermedios en el subintervalo $[x_j, x_{j+1}]$ usando los coeficientes obtenidos para las ecuaciones cúbicas al aplicar el método de trazador. Este procedimiento se repite para todos los subintervalos donde $j = 0, 1, \ldots, n-1$.

De ésta forma se obtiene el conjunto de datos, con un intervalo de tiempo de un minuto, hasta formar las 24 horas de día, que conforman el modelo estadístico, para los días laborables.

Para una explicación matemática más amplia sobre dicho método consultar el apéndice A, en la página 89.

En la figura 3.10 se muestra el modelo del tráfico de entrada en una línea contínua, obtenido mediante el método arriba descrito. En la misma gráfica se muestra el promedio de los datos en una línea con triángulos y los datos suavizados en cuadros, para propósitos comparativos.

Este modelo, junto con el del tráfico de salida, son los utilizados para detectar cualquier problema en la red, como podría ser una falla, al compararlos con el tráfico actual.

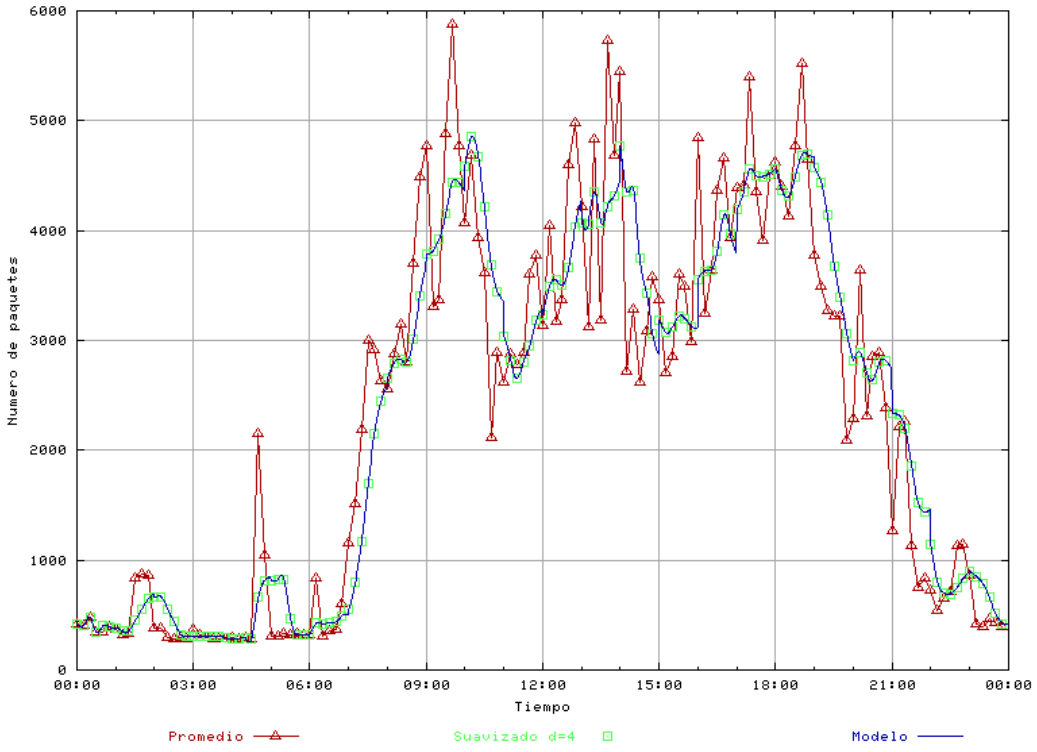


Figura 3.10: Datos promediados, suavizados con $d = 4$ y ajustados del tráfico de entrada.

Capítulo 4

Arquitectura del sistema

En este capítulo se describe la arquitectura del sistema que permite capturar y analizar todos los paquetes que componen el tráfico que se encuentra circulando a través de una red local y realiza estadísticas sobre los datos, hace las gráficas correspondientes, y muestra los resultados a través de una interfaz web. En caso de detectar alguna anormalidad en el comportamiento del tráfico, reporta al administrador de la red mediante un correo electrónico. El sistema también es capaz de determinar las computadoras que se encuentran activas dentro de la red.

Como se puede observarse en la figura 4.1, el sistema consta de dos computadoras, una se encuentra encargada de la captura de muestras de los paquetes que circulen en la red local y los manda a una segunda computadora, la cual los almacena en una base de datos y se encarga de realizar el análisis estadístico de los datos recolectados así como también será la encargada de ordenar dichos resultados, graficarlos y ponerlos a disposición del administrador de la red, mediante una interfaz web.

La conexión de las computadoras y puertos permitidos es controlada mediante el firewall *iptables*, que es interno del sistema.

4.1. Tipos de análisis

El análisis que se debe de realizar a los datos que están siendo obtenidos de la red puede ser en línea o fuera de ella.

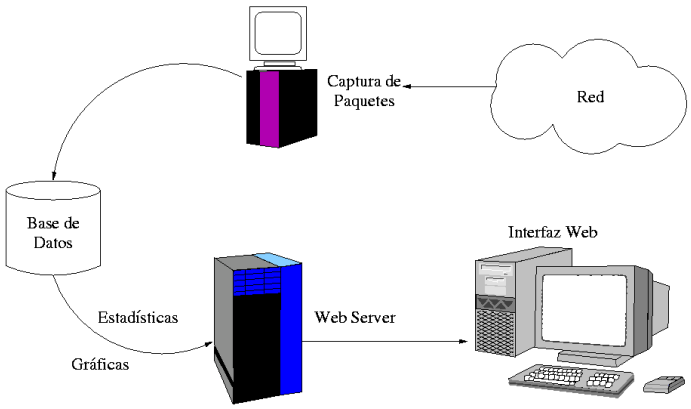


Figura 4.1: Esquema del sistema.

Análisis en línea

El análisis en línea es aquel que se realiza a los datos en cuanto éstos son obtenidos de la red, es un análisis en tiempo real. Este tipo de análisis tiene que realizarse a una gran velocidad, o de lo contrario se causaría un gran detrimento en la calidad de servicio que está proporcionando la red. Lo mejor para éste tipo de análisis, es que sea realizado por la misma computadora que esté realizando la captura de los paquetes, para que no exista un retraso al mandar los paquetes hacia otra computadora.

Análisis fuera de línea

El análisis fuera de línea es el que se realiza hasta que se dispone de toda la información necesaria para llevarlo a cabo. Este tipo de análisis puede ser realizado incluso por otra computadora, diferente a la que está realizando la captura, ya que primero se junta toda la información necesaria y después se procesa. En éste caso, no es necesario que el análisis se realice a una gran

velocidad, ya que la velocidad es irrelevante para la calidad de servicio de la red.

4.2. Como funciona el sistema

El sistema toma los datos de su base de datos, realiza las gráficas que serán mostradas en la interfaz web y realiza el análisis estadístico de dichos datos.

Los programas encargados del análisis, cálculo de errores, comparación del tráfico actual contra el esperado y de la realización de las gráficas están implementados en lenguaje *C*, utilizando el compilador *gcc*.

En el diagrama 4.2 se muestra el sistema, el cual consta de dos computadoras, a continuación se muestran las funciones que realiza cada una de ellas:

computadora 1

1. Desde $t = 0$ hasta $t = t + \Delta t$

   a) Recolectar datos

2. Transferir datos

3. Borrar datos

computadora 2

1. Recibir datos

2. Guardar paquetes en la base de datos

3. Realizar gráficas

4. Realizar estadísticas

5. Calcular errores

6. Si resultados= normal

a) Actualizar tablas

7. otro

   a) Realizar análisis detallado

   b) Toma de decisiones

   c) Realizar acciones necesarias

El sistema operativo de ambas computadoras es Linux. Todas las actividades se encuentran controladas por el demonio cron, para ser activadas automáticamente cada cinco minutos.

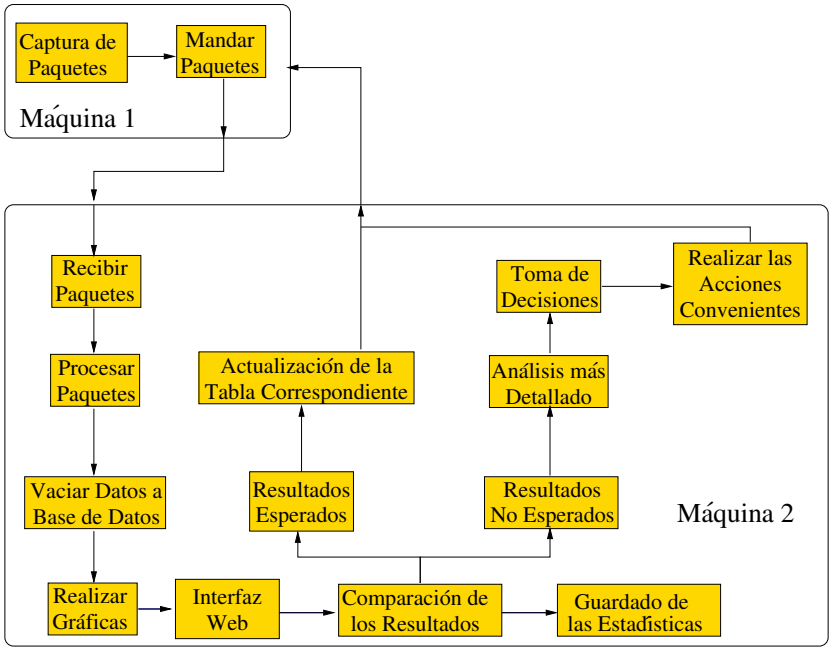


Figura 4.2: Diagrama del sistema.

4.2.1. Transferencia de datos

Para que la transferencia de los datos sea segura y no pueda ser leída por terceros, viaja encriptada por la red. Dicho procedimiento se realiza mediante openssh.

La autentificación es basada en host, utilizando el algoritmo de encriptación *DSA*. Primero se generan el par de llaves (pública y privada) utilizando el algoritmo mencionado anteriormente. Después la llave pública es copiada a la otra computadora y se coloca en el archivo de llaves autorizadas.

Para que el procedimiento anterior pueda funcionar, debe de verificarse que en el archivo de configuración del demonio de ssh se encuentre activada la opción de autentificación basada en host, la cual viene desactivada por omisión en la configuración predeterminada.

Al ser capturados los datos, son guardados en formato binario, para no causar un retraso en la captura, y es en este mismo formato que son transferidos. Los tamaños de estos archivos varían desde los 4kb hasta 150Mb aproximadamente por lo que su tiempo de transferencia varía desde unas centésimas de segundo hasta unos dos segundos.

4.3. Computadoras dentro de la red

Dentro de las funciones que tiene el sistema, está la de detectar las computadoras que se encuentran activas o inactivas dentro de la red, y cuánto tiempo llevan en dicho estado, ésto con el propósito de intentar detectar alguna posible falla en la red.

Para lograr dicha detección, el sistema construye paquetes de petición ARP para todas las posibles computadoras en la red y las manda por la interfaz de red correspondiente.

Un paquete de petición ARP pregunta “quien es X.X.X.X, decir a Y.Y.Y.Y”, y es mandado mediante un broadcast, asegurándose así que llegue a todos los nodos de la LAN, pero solo responde la computadora objetivo, las demás descartan inmediatamente el paquete. El paquete de respuesta ARP dice “X.X.X.X es hh:hh:hh:hh:hh:hh”, este paquete es unicast hacia la computa-

dora que mandó la petición, dado que el mensaje original incluye la dirección MAC de quien realizó la petición, no es necesario mandar otro mensaje para averiguarlo. A continuación se muestra el formato de un paquete ARP:

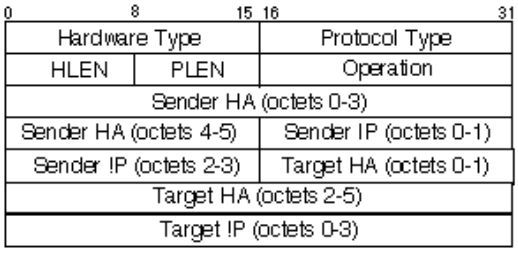


Figura 4.3: Formato de un mensaje ARP.

Después de mandar todas las peticiones ARP, el sistema cambia a modo de espera y comienza a recibir todas las respuestas de las computadoras que se encuentren activas dentro de la red.

El algoritmo 4.1 aquí utilizado es general para la detección de computadoras en una red. Aquí se presenta haciendo uso del protocolo ARP, pero pueden ser utilizados otros protocolos. El algoritmo presenta una velocidad bastante aceptable para redes clase C, pero decae grandemente para redes clase A y B.

Algoritmo 4.1: Detección de computadoras activas

ENTRADA Computadoras en la red (n); Tiempo de espera t_{max}.

SALIDA Computadoras activas en la red.

1. Abrir la tarjeta de red en modo escritura.
2. Para $i = 1$ hasta $i = n$ hacer
   - *a*) Crear paquete de petición ARP para la computadora i.
   - *b*) Escribir el paquete a la tarjeta de red.
3. Cerrar la tarjeta en modo de lectura.

4. Abrir la tarjeta en modo promiscuo de lectura.

5. Mientras $t < t_{max}$ hacer

   a) Recibir los paquetes de respuestas ARP.

6. Almacenar los datos recibidos en los paquetes en la base de datos.

7. Cerrar la tarjeta de red en modo de lectura promiscuo.

8. SALIDA Computadoras que se encuentran activas en el instante T.

9. PARAR

También es posible crear el procedimiento anterior, pero utilizando en protocolo *icmp* (ping), en lugar de ARP, pero de ésta manera no son detectadas todas las computadoras de la red, ya que podrían tener bloqueado dicho protocolo en su propio firewall, o totalmente desactivado.

El método del protocolo ARP también puede fallar, si las computadoras tienen desactivada las respuestas ARP, pero esto último es menos común que la desactivación del protocolo ICMP, por lo que este método se considera más confiable.

En cuanto se tiene la información de las computadoras que contestaron a las peticiones ARP, ésta es guardada en una tabla de la base de datos, como la mostrada en la Tabla 4.5.

Como puede apreciarse en la Tabla 4.5, los dos útlimos campos son la fecha y la hora, la cual es tomada del sistema y corresponden a la fecha y hora en la que las computadoras respondieron a la petición. Los tres primeros campos son la dirección IP, la dirección MAC y el nombre (si se encuentra registrada en el DNS) de la computadora que contesta la petición.

El procedimiento de detección de las computadoras activas en la red es activado por el demonio cron cada determinado período de tiempo Δt_2, el cual no es necesariamente igual al período de recolección de datos (Δt).

Los resultados de ésta detección pueden ser consultados en una de las páginas de la interfaz web, en la cual se muestra una tabla con las computadoras activas, marcadas con una esfera verde. Si las computadoras no han

respondido en menos de 24 horas, son marcadas con una esfera amarilla, pero si no han respondido en más de 24 horas, son presentadas con una esfera roja.

El código de colores anterior es para proporcionar una ayuda visual al administrador de la red, para que pueda localizar los posibles problemas más rapidamente.

4.4. La base de datos

El administrador de base de datos que se utilizó para la construcción del sistema es *mysql*. La base de datos solo puede ser accedida desde la misma computadora en la que se encuentra, además de requerir una autentificación del usuario, mediente password, para poder acceder. Las operaciones que puede realizar el usuario se encuentran restringidas para mayor seguridad.

La base de datos cuenta con 5 tablas, cuyo esquema se muestra en la figura 4.4.

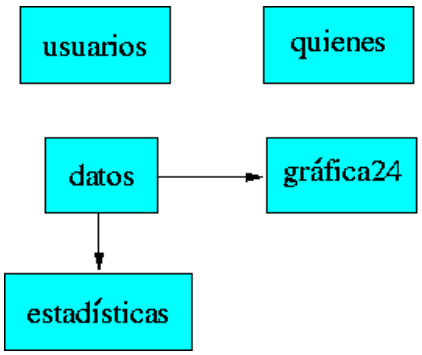


Figura 4.4: Tablas de la base de datos, y sus dependencias.

En ésta figura, las dependencias entre las tablas se muestran mediante flechas apuntando hacia las tablas dependientes, es decir, las tablas *usuarios* y *quienes* son totalmente independientes de todas las demás tablas, mientras que las tablas *grafica24* y *estadísticas* son dependientes de la tabla *datos*.

Las tablas mostradas en la figura 4.4 se describen a continuación con mayor detalle:

datos: Esta tabla es la que contiene toda la información que se recolecta sobre el tráfico de la red.

Los campos día y hora son llenados en base a la hora del sistema. Los demás campos son llenados usando los datos del encabezado del paquete, los cuales incluyen la dirección fuente y la destino, los puertos fuente y destino así como el protocolo del paquete, como puede ser TCP, ICMP, etc.

Tabla 4.1: Datos que se guardan los paquetes recibidos.

Field	Type	Null	Key	Default	Extra
dia	varchar(4)			0000	
hora	varchar(4)			0000	
proto	varchar(50)	YES		NULL	
fuente	int(10) unsigned			0	
destino	int(10) unsigned			0	
pfuente	smallint(5) unsigned	YES		NULL	
pdestino	smallint(5) unsigned	YES		NULL	

estadísticas: En ésta tabla es en donde se guardan las estadísticas calculadas sobre todo tráfico que ha sido capturado de la red. Los datos de ésta tabla son obtenidos mediante los programas de análisis de datos.

En ésta tabla se llenan los campos de día y hora nuevamente, así como el nombre de la computadora que recibe los paquetes, cuantos paquetes entran y salen del nodo, así como los totales por protocolo (tcp, udp, etc.).

grafica24: Es en ésta tabla en la que se guardan los datos necesarios para realizar la gráfica del tráfico de las últimas 24 horas. Esta tabla es llenada por los programas de análisis de datos, basados en los campos de la tabla *datos*.

Son llenados los totales de paquetes que entran y salen de la red, en el período Δt así como el día y la hora en que se registró dicho tráfico.

Tabla 4.2: Datos estadísticos del tráfico en la red.

Field	Type	Null	Key	Default	Extra
dia	varchar(4)			0000	
hora	varchar(4)			0000	
host	varchar(20)				
total	int(11)			0	
entran	int(11)			0	
salen	int(11)			0	
tcp	int(11)			0	
udp	int(11)			0	
icmp	int(11)			0	
otro	int(11)			0	

También es llenado el campo modificado, el cual es una bandera para que el programa de graficación pueda distinguir que datos debe utilizar. Esta tabla es necesaria para que en el programa encargado de realizar las gráficas no sea necesario realizar dichos cálculos de totales, lo que lo causaría un retraso al necesitar hacer más accesos a la base de datos.

Tabla 4.3: Datos del tráfico de entrada y salida de las últimas 24 horas.

Field	Type	Null	Key	Default	Extra
dia	varchar(4)			0000	
hora	varchar(4)			0000	
entran	int(11)			0	
salen	int(11)			0	
modificado	smallint(1)			0	

usuarios: Esta tabla contiene únicamente los usuarios que se encuentran registrados en el sistema, y es utilizada para realizar la autentificación de los mismos. Existen dos tipos de usuario, el administrador, con capacidad para crear más usuarios, borrar usuarios y cambiar contraseñas, y el usuario normal, que solamente puede acceder al sistema para consultas

Tabla 4.4: Datos de los usuarios del sistema.

Field	Type	Null	Key	Default	Extra
user	varchar(30)		PRI		
passwd	varchar(16)				
type	smallint(1) unsigned			2	

quienes: En ésta tabla se guardan los datos sobre todas las computadoras que se encuentran en la red, se encuentren activas o no en un determinado período de tiempo.

En la tabla se guardan la dirección IP, la dirección MAC, y el nombre de la computadora, en caso de estar registrada en el DNS, así como la última fecha y la hora en la que respondió.

Tabla 4.5: Datos del estado actual de las computadoras de la red.

Field	Type	Null	Key	Default	Extra
ip	char(16)		PRI	0	
mac	char(20)			0	
hostname	char(255)	YES		NULL	
fecha	date			0000-00-00	
hora	time			00:00:00	

4.5. Respaldos

Para no saturar su disco duro, la computadora encargada de realizar la captura, desecha todos los paquetes capturados en cuanto son transferidos a la segunda computadora.

La segunda computadora al recibir todos los paquetes capturados en un determinado período de tiempo, después de vaciarlos a la base de datos los mueve a un directorio especial de respaldo.

Cada 24 horas, el sistema, con la ayuda del demonio cron, realiza un respaldo de la base de datos y coloca en el directorio de respaldo también

son copiadas a ese directorio todas las gráficas realizadas por el sistema en las últimas 24 horas.

Inmediatamente que se termina de realizar el respaldo de la base de datos y las copias de los archivos, se procede a comprimir todo el contenido del directorio de respaldos, y colocar el nuevo archivo comprimido en una partición separada del sistema, la cual tiene un sistema de archivos reiser, para brindar mayor protección a los respaldos.

Se escogió el sistema de archivos reiser debido a sus características, entre las que destacan:

1. Es uno de los sistemas de archivos más veloces.

2. Es un sistema de archivos atómico, lo que previene que una operación sea interrumpida y los datos sean corrompidos.

3. Usa árboles bailantes (dancing trees), substituyendo a los árboles balanceados, lo que lo hace muy eficiente en cuestión de espacio.

4. Esta hecho basado en plugins, lo que permite hacer mejoras, actualizaciones o añadir capacidades sin necesidad de reformatear la partición donde se encuentre.

5. Esta diseñado para grados militares, lo que implica un código fácil de auditar.

De ésta manera, se respalda el sistema cada 24 horas, y cuando la partición de respaldo se encuentra llena a un 90 %, se manda un correo electrónico pidiendo al administrador del sistema que realize una copia de los datos en un cd o cinta de respaldo.

Si la partición llega a un 97 %, se le informa al administrador que es urgente la copia de los datos a otro dispositivo, ya que en el caso de llegar al 99 %, el sistema borrará todos los datos que contenga la partición. En un día normal, se capturan aproximadamente 100Mb de información.

El algoritmo 4.2 se utiliza para realizar los respaldos de datos, todas las gráficas, estadísticas de las últimas 24 horas y evitar la saturación del disco duro se presenta a continuación:

Algoritmo 4.2: Respaldo de los datos

ENTRADA Nombre de la partición en donde se encuentran los datos.

SALIDA Archivo de respaldo.

1. Vaciar la base de datos a un solo archivo de texto.
2. Copiar todos los archivos a respaldar (base de datos, gráficas, etc.) a un directorio de respaldo.
3. Crear un archivo comprimido con todo lo que contiene el directorio.
4. Mover el archivo comprimido a la partición de respaldo.
5. Borrar todos los archivos del directorio de respaldo.
6. Verificar espacio en la partición.
7. Si el espacio ocupado $\geq 99\,\%$ hacer
   a) borrar todos los datos de la partición.
8. Si el espacio ocupado $\geq 97\,\%$ hacer
   a) Enviar correo al administrador sobre urgencia de respaldar datos en otro medio.
9. Si el espacio ocupado $\geq 90\,\%$ hacer
   a) Enviar correo al administrador pidiendo respaldar datos en otro medio.
10. SALIDA Archivo comprimido con los respaldos.
11. PARAR

4.6. Descarga de los datos

Los datos son recibidos mediante el método de transferencia cifrada de datos. que ha sido descrito en secciones anteriores.

Los datos que se reciben, vienen en el formato binario utilizado por la biblioteca *pcap*, de unix, por lo que al ser recibidos por la segunda computadora, son leídos mediante un programa que entienda dicho formato y los descarga a una tabla de la base de datos, desde donde se encuentran disponibles para ser consultados por los programas de análisis, graficación, interfaz web, etc.

Los datos que se reciben son todos los paquetes (cabecera y contenido) que se capturaron durante un período de tiempo Δt, pero en la tabla de la base de datos solo se guarda la información relevante de cada paquete. En la Tabla 4.1 pueden verse los campos que contiene la tabla de la base de datos.

El contenido de los paquetes no es guardado, solo algunos datos de su encabezado, ya que de ésta forma se respeta la privacidad de los usuarios de la red y se ahorra espacio en el disco duro de la segunda computadora.

4.7. Análisis de los datos

Los programas de análisis de los datos son los encargados de realizar las gráficas del tráfico de la red, el análisis estadístico del tráfico, así como de detectar alguna anomalía y tomar las acciones correspondientes, como pueden ser un aviso al administrador del sistema. La computadora de análisis de datos debe de contar con mayores capacidades que la computadora de captura ya que también necesita una gran capacidad en disco duro, para almacenar todos los datos que se reciben de la red, los modelos estadísticos y los respaldos diarios de todos los datos.

Con base en los argumentos expuestos en el capítulo 2 y el inicio del presente capítulo, se concluye que para nuestros propósitos lo mejor es utilizar la captura de paquetes mediante software, y el análisis de dichos paquetes se debe de realizar fuera de línea para evitar el decaimiento de la calidad de servicio de la red.

El algoritmo 4.3 presenta el algoritmo general utilizado para realizar el

análisis de los datos conforme son recibidos y vaciados a la base de datos.

Algoritmo 4.3: Ciclo de análisis

ENTRADA Archivo de paquetes en formato binario.

SALIDA Estadísticas de los paquetes, porcentajes de error, gráficas comparativas, gráficas de error.

1. Vaciar el archivo recibido a la base de datos.
2. Calcular el total de paquetes recibidos.
3. Calcular el total de paquetes que entraron y salieron.
4. Calcular el total de paquetes por protocolo.
5. Calcular total de paquetes por cada nodo.
6. Realizar la gráfica del tráfico de las últimas 24 horas.
7. Realizar la suavización de los datos.
8. Realizar el ajuste de los datos.
9. Realizar la comparación de los datos contra el modelo.
   - *a*) Calcular el error y porcentaje de error.
   - *b*) Hacer las gráficas de comparación contra el modelo.
   - *c*) Hacer las gráficas de porcentaje de error.
10. Si existen errores grandes, realizar la acción adecuada.

4.7.1. Gráficas

Las gráficas que se presentan en la interfaz web son realizadas por los programas de análisis del tráfico.

La primera gráfica que se presenta es la del tráfico total de entrada y de salida de las últimas 24 horas. Esta gráfica se actualiza cada diez minutos. Los datos se toman de la base de datos, simplemente realizando la consulta sobre cuántos paquetes tienen como fuente o destino nuestra red a una determinada hora, dependiendo si deseamos saber el tráfico de entrada o salida.

Otra de las gráficas muestra los porcentajes de cada protocolo que han pasado en la red en un determinado período de tiempo. Para realizarla, simplemente se consulta el total de paquetes por cada protocolo, se suman para obtener el total y se calcula el porcentaje de cada uno de ellos.

También se presenta gráficamente la comparación entre el tráfico esperado contra el tráfico actual y el tráfico suavizado, tanto para el tráfico de entrada como para el tráfico de salida. Se grafica el modelo que ya se tiene calculado previamente, se usa el tráfico total de entrada (o salida) como en la primera gráfica, y se toman también los datos del tráfico después de pasarlo por un procedimiento de ajuste y suavización que fué descrito en el capítulo anterior.

Finalmente se muestra el error de la última hora, al comparar el tráfico que se esperaba de acuerdo con el modelo contra el tráfico que se ha recibido. También en este caso se realiza una gráfica para el tráfico de entrada y otra para el de salida. En éstas gráficas se muestra también el error acumulado.

4.7.2. Estadísticas

Cuando un usuario selecciona un intervalo de tiempo Δt, el sistema calcula el porcentaje de cada protocolo que ha sido utilizado durante dicho intervalo de tiempo.

Para realizar dicho cálculo, el sistema realiza una búsqueda en la base de datos, pidiendo el número de paquetes que se encuentren dentro del intervalo Δt, que correspondan a cada uno de los protocolos.

Una vez que se tienen los datos de todos los protocolos, el porcentaje se calcula mediante la formula:

$$\%\text{protocolo}_i = \frac{\#\text{paquetes\_protocolo}_i}{\sum_{i=0}^{n} \text{protocolo}_i} * 100 \tag{4.1}$$

donde n es el número total de los protocolos.

Después de realizar dichos cálculos, el sistema procede a crear una gráfica de barras con los resultados obtenidos de estas estadísicas, la cual es mostrada al usuario mediante la interfaz web del sistema.

Estas estadísticas son muy importantes, debido a que en ellas se pueden detectar algunas anormalidades que podrían estar ocurriendo en la red. En general el protocolo más utilizado es TCP, seguido del UDP, en una proporción aproximada del 80 %-20 %.

Esta aproximación de los porcentajes de los protocolos no es estática, va cambiando con el paso del tiempo, y el protocolo UDP es cada vez más utilizado dado que en los últimos años se ha registrado un incremento en las aplicaciones que utilizan el protocolo UDP (chat, messenger, etc.). Sin embargo, las aplicaciones que utilizan el protocolo TCP (servicios web, correo electrónico, ftp, etc) siguen siendo las más utilizadas en las redes.

El algoritmo 4.4 describe la secuencia de pasos para el cálculo de estadísticas.

Algoritmo 4.4: Cálculo de las estadísticas

ENTRADA Datos en la base.

SALIDA Estadísticas del tráfico de la red.

1. Calcular el total de paquetes recibidos.

   a) Realizar una consulta a la base de datos, sobre el total de paquetes en el intervalo de tiempo Δt.

2. Calcular el total de paquetes que entraron y salieron.

   a) Realizar una consulta a la base de datos, sobre el total de paquetes que entraron en el intervalo Δt.

b) Realizar una consulta a la base de datos, sobre el total de paquetes que salieron en el intervalo Δt.

3. Calcular el total de paquetes por protocolo.

   a) Realizar una consulta a la base de datos, sobre el total de paquetes del protocolo TCP en el intervalo Δt.

   b) Realizar una consulta a la base de datos, sobre el total de paquetes del protocolo UDP en el intervalo Δt.

   c) Realizar una consulta a la base de datos, sobre el total de paquetes del protocolo ICMP en el intervalo Δt.

   d) Realizar una consulta a la base de datos, sobre el total de paquetes de otros protocolos en el intervalo Δt.

4. Calcular total de paquetes por cada nodo.

   a) Realizar una consulta a la base de datos, sobre el total de paquetes enviados por cada nodo en el intervalo Δt.

   b) Realizar una consulta a la base de datos, sobre el total de paquetes recividos por cada nodo en el intervalo Δt.

4.7.3. Comparación de los datos con el modelo

Los datos no pueden ser comparados directamente con el modelo, ya que presentan una gran oscilación, similar a la propiedad de ráfagas que presenta el tráfico entre dos nodos. Es por ésta razón que primero son sometidos a un proceso de suavización y ajuste, similar al utilizado en la creación del modelo estadístico. Una vez realizado dicho proceso, se procede a la comparación con el modelo que ha sido previamente construido.

En la figura 4.5 se muestra la comparación del modelo estadístico con los datos originales, y con los datos después de ser sometidos al proceso de suavización.

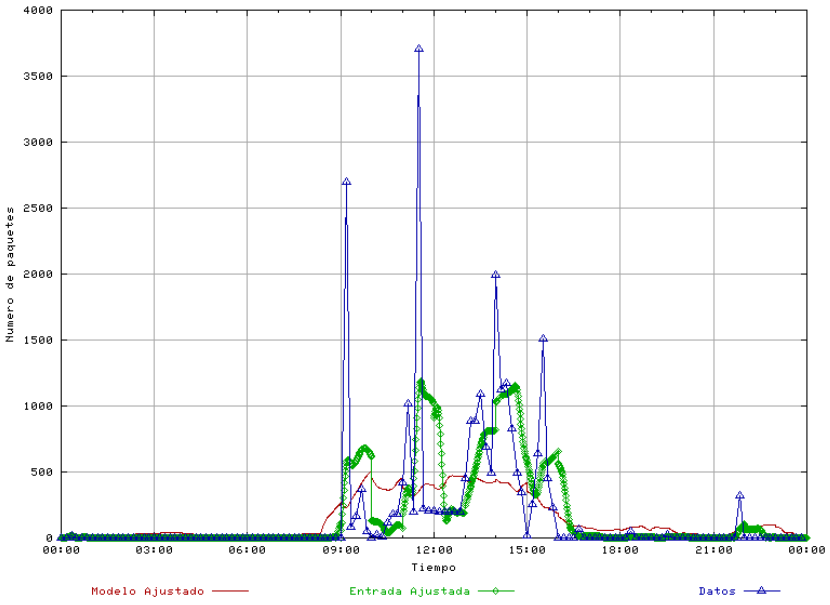


Figura 4.5: Comparación de los datos con el modelo.

Cálculo de error

Al momento de realizar la comparación, también se realiza el cálculo del error existente entre los datos que se obtienen en un intervalo Δt y los datos que son esperados, según el modelo estadístico con el que se cuenta para ese intervalo de tiempo.

Para realizar el cálculo de dicho error, se utiliza la ecuación 4.3, y para cualcular el porcentaje del error, es utilizada la ecuación 4.2.

$$\%\text{error} = \left| \frac{x_{\text{esperada}} - x_{\text{actual}}}{x_{\text{esperada}}} \right| * 100 \tag{4.2}$$

Con base en estos errores es posible realizar decisiones sobre la existencia o falta de alguna anomalía en el tráfico que se encuentra circulando por la red en el intervalo de tiempo Δt.

En caso de existir, la anomalía puede ser considerada de diferentes maneras, dependiendo de la magnitud del error (que tanto se aleje del modelo estadístico del tráfico), marcando así zonas de prevención, de alerta o de falla

en la red.

En la figura 4.6, se muestra el error obtenido al comparar los datos suavizados y ajustados de un determinado intervalo Δt, contra el modelo estadístico para ese mismo intervalo de tiempo.

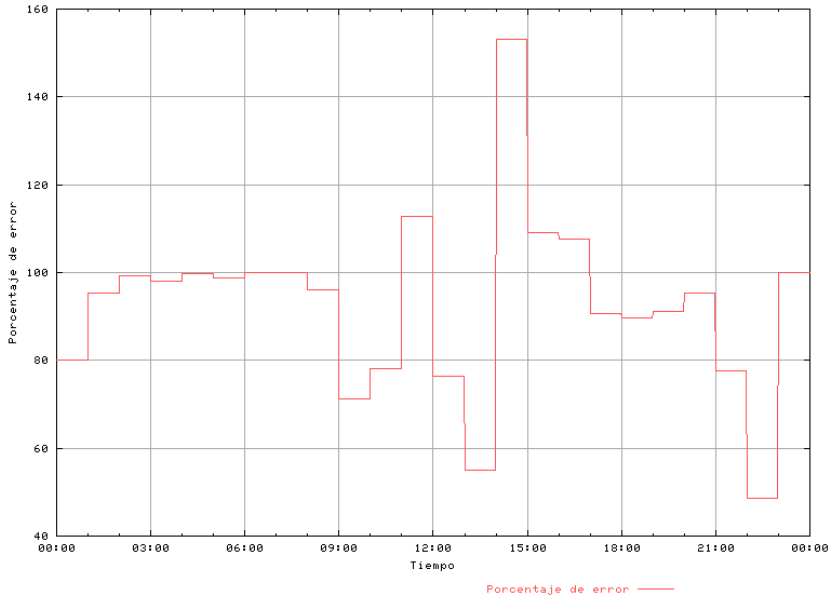


Figura 4.6: Errores obtenidos en la comparación.

4.7.4. Toma de decisiones

Los programas que realizan el análisis del tráfico, también se encargan de calcular el error entre el tráfico esperado contra el tráfico recibido usando la ecuación 4.3, en base a esto, es posible determinar si esta ocurriendo alguna anomalía dentro de la red o no.

$$\text{error} = \left| \frac{x_{\text{esperada}} - x_{\text{actual}}}{x_{\text{esperada}}} \right| \tag{4.3}$$

En el caso de que todo se encuentre dentro de los límites normales, es decir, que el error sea menor a una tolerancia ϵ, el sistema se limita a actualizar todos los datos dentro de la tabla de la base de datos. Cuando esté ocurriendo

alguna anomalía en la red ($error > \epsilon$), entonces el sistema tiene que decidir sobre las acciones que deberá llevar a cabo.

Las posibles acciones pueden ir desde un simple aviso al administrador del sistema hasta tomar la decisión de cerrar totalmente una determinada dirección IP, un puerto o un protocolo.

Las acciones a tomar se pueden clasificar en dos tipos diferentes:

Alerta 1: Es cuando el error no es demasiado grande. En este caso el sistema se limita a mandar un correo electrónico al administrador de la red, informando sobre una posible anomalía o error en la red.

Alerta 2: Es cuando el error es muy grande, entonces el sistema intenta averiguar que está sucediendo, si el problema es un exceso de tráfico, tratará de averiguar si procede de alguna dirección, protocolo o puerto. Una vez realizada la detección, mandará un correo electrónico al administrador de la red, informando de lo que está ocurriendo .

Si el problema es que no se ha recibido ningún paquete, el sistema intentará mandar una notificación al administrador, avisándole que posiblemente exista una falla en la red o en una determinada computadora de la red, pidiéndole que realize una revisión del equipo o computadora a la brevedad posible.

A continuación se presenta el algoritmo utilizado en la toma de decisiones del sistema:

Algoritmo 4.5: Toma de decisiones

ENTRADA n (número de paquetes recibidos); Valor de la tolerancia ϵ_1 para la alerta 1; Valor de la tolerancia ϵ_2 para la alerta 2; Error.

SALIDA Acción dependiendo de los valores del error.

1. Si $n = 0$ hacer

   a) Mandar correo al administrador sobre falla en la red.

2. Otro, si $error < \epsilon_1$ hacer

   a) Hacer nada

3. Otro si $\epsilon_1 < error$ y $error < \epsilon_2$ hacer

   a) Enviar correo de alerta al administrador, sobre posible anomalía en la red.

4. Otro si $error > \epsilon_2$ hacer

   a) Consultar sobre el (los) nodo(s) con mayor tráfico.

   b) Consultar los puertos de mayor tráfico en el (los) nodo(s) de mayor tráfico.

   c) Consultar los protocolos de mayor tráfco en el (los) nodo(s) de mayor tráfico.

   d) Bloquear los puertos de mayor tŕafico de el (los) nodos(s) de mayor tráfico.

   e) Bloquear los protocolos de mayor tŕafico de el (los) nodos(s) de mayor tráfico.

   f) Mandar un correo al administrador de la red sobre los problemas encontrados y las acciones tomadas.

Capítulo 5

Interfaz web

Como ya se comentó en capítulos anteriores, todos los resultados del sistema de análisis de tráfico son presentados a los usuarios mediante una interfaz web.

Para poder hacer uso de la interfaz web, primero se debe de instalar el sistema de análisis del tráfico en dos computadoras, una encargada de la captura de los datos y otra encargada de realizar el análisis de los mismos.

5.1. Instalación del sistema

Las dos computadoras cuentan con el sistema operativo Linux, que es la plataforma de desarrollo del sistema.

5.1.1. Computadora de captura

1. Instalar versión modificada de snort.
2. Agregar scripts de control de inicio y fin de captura.
3. Modificar el archivo de fin de captura para indicar la dirección de la computadora a la cual se deben de mandar los datos.
4. Crear las llaves para realizar la transferencia a la otra computadora.
5. Intercambiar llaves

6. Modificar el archivo /etc/crontab para el intervalo de tiempo de captura y descanso deseados.

5.1.2. Computadora de análisis

Para instalar el sistema de análisis se deben seguir los pasos siguientes:

1. Instalar PHP.
2. Instalar MYSQL.
3. Crear la base de datos en MYSQL.
4. Instalar el servidor web APACHE.
5. Configurar APACHE para trabajar con SSL, PHP y MYSQL.
6. Instalar los archivos de la interfaz web (PHP).
7. Instalar versión modificada de snort.
8. Instalar los archivos binarios del sistema.
9. Instalar los archivos de configuración del sistema.
10. Configurar los parámetros del sistema (tipo de gráficas, porcentajes de error para la toma de decisiones, etc.).
11. Configurar el demonio SSH para permitir autentificaciones basadas en host.
12. Crear llaves de autentificación.
13. Intercambiar llaves.
14. Agregar el script de control de los programas.
15. Modificar el archivo /etc/crontab para ejecutar los programas automáticamente y a quien se deben de mandar los correos electrónicos de avisos.

5.2. La interfaz web

La interfaz debe de poder ser consultada únicamente por personal autorizado para dicho propósito, por lo que cuenta con un sistema de autentificación mediante nombre de usuario y contraseña. Todos los usuarios se encuentran almacenados en la base de datos del sistema, y los passwords se encuentran cifrados mediante el algoritmo DES. Más aún, la interfaz gráfica solo recibe peticiones de las computadoras autorizadas para realizar las consultas. Esto es controlado mediante un firewall creado con *iptables*, el cual viene integrado en el núcleo del sistema operativo.

Debido a que la información que se muestra mediante la interfaz web puede comprometer la seguridad de la red, no se puede correr el riesgo de que sea interceptada, por lo que viaja encriptada a través de la red. Para lograr lo anterior se utiliza un certificado de autentificación y una encriptación de 128 bits mediante la capa SSL, por lo que se utiliza el protocolo HTTPS.

Para la construcción de la interfaz web, se utiliza el servidor web *APACHE*, y para la creación del certificado de autentificación se utilizó *Open SSL*, así como también para la encriptación de los datos. Toda la interfaz está construida haciendo uso del lenguaje *PHP*, el cuál fué creado explícitamente para construir páginas web dinámicas.

La interfaz del sistema de análisis de tráfico cuenta con dos vistas, una de usuario, y la otra de administrador:

El administrador solo es capaz de dar de alta nuevos usuarios en el sistema, de borrar usuarios ya existentes, o de cambiar la contraseña de los usuarios. Por otra parte, la vista de usuario permite ver las gráficas del tráfico total de entrada y salida, las comparaciones del tráfico actual contra los valores esperados, tanto de entrada como de salida. También puede ver las gráficas de error y error acumulado de entrada y salida y puede ver las computadoras que se encuentran presentes en la red.

Para acceder al sistema, se debe ingresar el nombre de usuario y la contraseña, el sistema toma los dos datos y encripta el password, después compara si existe ese usuario en la base de datos del sistema, si el usuario existe, entonces compara el password encriptado contra el que se encuentra en la

base de datos, si coinciden, entonces el usuario tiene acceso al sistema y se abre una sesión para dicho usuario.

En todas y cada una de las páginas de la interfaz web, se realiza una verificación de usuario válido en la sesión, para evitar de ésta manera que alguien pueda acceder directamente a alguna página del sistema sin autentificarse.

Dentro de la interfaz también existe la posibilidad de realizar una consulta específica a la base de datos, para lo que se accede a una página en la cual se puede llenar una forma electrónica y al ser procesada, es convertida en una consulta SQL, ésta es ejecutada en la base de datos y se despliega el resultado en otra página de la interfaz.

Las búsquedas son muy útiles si se requieren detalles específicos sobre lo que sucede en la red, además de ser fácil de utilizar. En la forma simplemente se tiene que decidir el intervalo de tiempo en el que se desea realizar la búsqueda, se puede seleccionar una dirección IP específica, un puerto, un protocolo, tanto de fuentes como destinos. Es posible también obtener el total de los resultados obtenidos.

La red sobre la que se encuentra funcionando el sistema de análisis es la subred 148.247.1.0/24, la cual cuenta con aproximadamente diez servidores (web, ftp, correo electrónico, etc.) y alrededor de 25 computadoras cliente. En esta red se capturan un promedio de 10Mb por cada período de captura. El modelo estadístico fué desarrollado para esta red, siguiendo los pasos descritos en la Sección 3.3.

5.3. Ingreso

Al ser accedida la interfaz web, despliega una pantalla como la mostrada en el figura 5.1, en la cual se pide autentificación mediante un login y un password.

Para poder hacer uso de la interfaz web es necesario contar con un navegador web que soporte una “encriptacion fuerte”, es decir que soporte una llave criptográfica igual o mayor a 128 bits, ya que de lo contrario no se podrá acceder. Algunos navegadores de este tipo son Mozilla, Safari, Konqueror,

Figura 5.1: Pantalla de autentificación para acceder a los resultados.

Netscape, etc. en sus versiones más recientes.

5.3.1. Tráfico en la red

Una vez autentificados ante la interfaz web, se permite acceder a los datos y se muestra una gráfica con el tráfico total de entrada y salida de las últimas 24 horas, como la que se muestra a continuación en la figura 5.2. Como se puede apreciar, el tráfico de entrada se muestra en una línea con triángulos, y el tráfico de salida se muestra en una línea con cuadros. Esta es una forma más fácil de darse cuenta cual tráfico es mayor y detectar si existe alguna irregularidad en el tráfico de la red. Esto es de gran importancia, ya que el tráfico de entrada debe de ser mayor que el tráfico de salida, en una red de computadoras cliente, mientras que si se trata de una red de servidores, el tráfico se invierte.

En el caso particular de la figura 5.2, el tráfico de salida y el de entrada son similares, ya que en nuestro caso de estudio, la red se encuentra formada

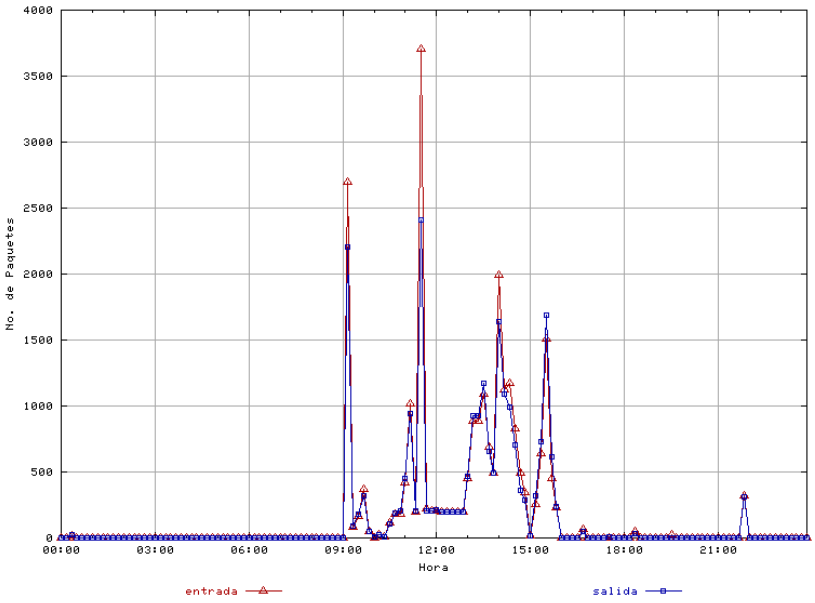


Figura 5.2: Tráfico de las últimas 24 horas.

tanto de servidores, como de computadoras clientes.

En la gráfica es posible seleccionar un determinado período de tiempo, que sea de interés, y poder observar los detalles del tráfico de la red durante éste.

5.3.2. Estadísticas

Una vez seleccionado el intervalo de tiempo deseado, la interfaz web presenta una gráfica como la mostrada en la figura 5.3, en la cual se muestra el porcentaje de cada protocolo utilizado durante el período de tiempo previamente seleccionado. Como se puede apreciar en el ejemplo de la figura 5.3, el protocolo más utilizado es TCP (80 % aproximadamente), seguido de UDP (15 %), de ICMP y de otros.

En la misma página web, también se muestra una tabla sobre las computadoras que han tenido mayor tráfico durante ese mismo intervalo de tiempo, como la mostrada en la figura 5.4. En esta tabla se muestran en una columna la cantidad de paquetes que han sido enviados o recibidos por cada una de

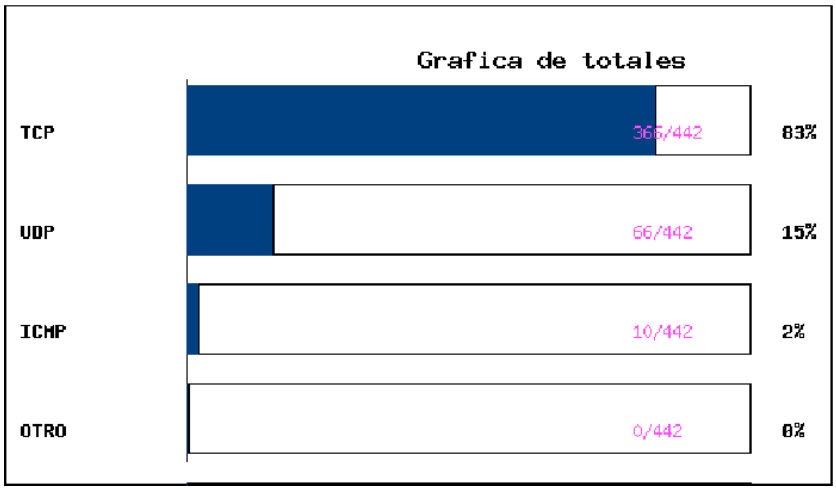


Figura 5.3: Porcentajes del tráfico en un determinado período de tiempo.

ellas y en la otra columna la dirección IP de la computadora que generó dicho tráfico.

Mas Usados

Total	Host
1219	148.247.30.252
5	148.247.30.160

Figura 5.4: Host más usados durante un período de tiempo. Esta es una captura de pantalla de la aplicación.

Los datos de la figura 5.4 se encuentran ordenados de mayor a menor de acuerdo a la cantidad total de paquetes que entraron o salieron por nodo en el intervalo de tiempo previamente seleccionado. En la tabla se muestran únicamente los nodos que han registrado tráfico en tal intervalo de tiempo. Esto sirve para poder detectar de una manera más rápida y fácil las computadoras que están generando mayor tráfico dentro de la red en el intervalo

de tiempo que ha sido seleccionado previamente.

5.3.3. Detalles de un nodo

A través de ésta tabla es posible pedir detalles de alguna determinada computadora que sea de interés. Para ello se selecciona el nodo, presionando el botón izquierdo del ratón, lo cual llevará a otra pantalla, en donde se mostrarán los detalles del nodo, como se muestra en la figura 5.5. En ella se puede apreciar un desglose del número total de paquetes tanto de entrada como de salida y el número total de paquetes utilizados por cada protocolo.

Detalles sobre el nodo: 148.247.30.252

Total	Host	Entrante	Saliente	TCP	UDP	ICMP	OTRO
1219	148.247.30.252	643	576	1204	14	1	0

Figura 5.5: Detalles de un determinado host. Esta es una captura de pantalla del sistema realizado.

En ésta figura se puede ver el tipo de comunicaciones que está llevando a cabo un determinado nodo de nuestra red, tanto de entrada como de salida.

Así es posible detectar si un nodo está recibiendo o generando algún tipo de tráfico diferente al que debería, o simplemente existe alguna pequeña irregularidad en el tráfico, de acuerdo con los porcentajes esperados de cada uno de los protocolos.

5.3.4. Búsqueda

También se cuenta con un mecanismo de búsqueda, como el mostrado en la figura 5.6, gracias al cual es posible localizar las computadoras que mandaron paquetes a cada una de las computadoras de la red o las computadoras a las que les fueron mandados paquetes desde cualquier computadora de la red local.

Como se puede apreciar en la figura 5.6, la forma electrónica para hacer la consulta es simple. También se debe seleccionar un nuevo intervalo de tiempo en el que se desea realizar la consulta. Esto es muy útil para el

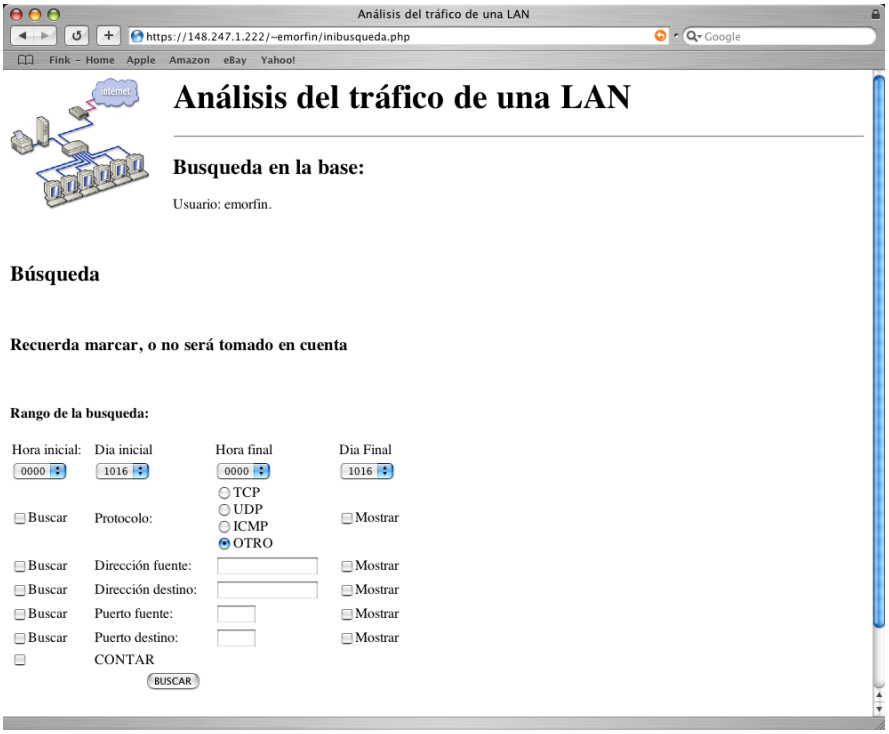


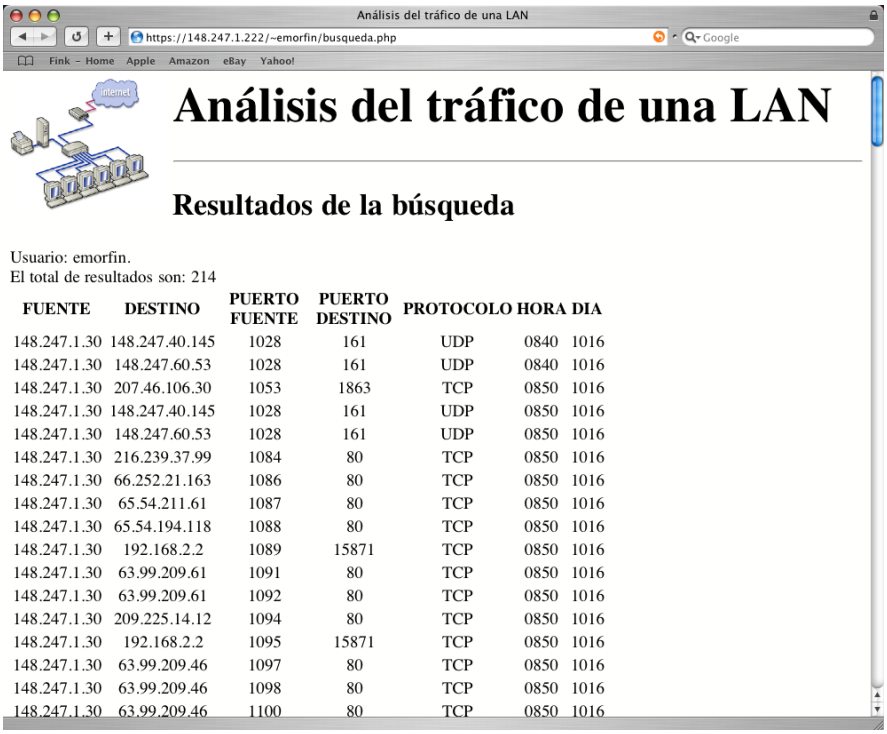
Figura 5.6: Forma de búsqueda de detalles del tráfico de la red.

caso de que se desee consultar todas las comunicaciones que ha realizado un nodo específico, por un determinado período de tiempo. Este nuevo intervalo de tiempo a seleccionar, puede ser aún más amplio que el anterior, ya que puede llegar a abarcar varios días hacia atrás, o puede ser un intervalo muy pequeño, ya que se pueden seleccionar solamente unos cuantos minutos de un día determinado.

Una vez indicado el criterio de búsqueda, ahora simplemente tenemos que seleccionar los datos que deseamos que nos muestre del resultado encontrado, como podría ser el número total de paquetes, el protocolo utilizado en la comunicación, la dirección fuente y/o destino de la comunicación y/o los puertos fuente y/o destino de la conexión.

Después de llenar la forma, es necesario mandarla al sistema, presionando el botón de búsqueda. En cuanto la forma se recibe por el servidor, ésta es convertida en una consulta de lenguaje SQL para poder mandarla al servidor

de la base de datos.



FUENTE	DESTINO	PUERTO FUENTE	PUERTO DESTINO	PROTOCOLO	HORA	DIA
148.247.1.30	148.247.40.145	1028	161	UDP	0840	1016
148.247.1.30	148.247.60.53	1028	161	UDP	0840	1016
148.247.1.30	207.46.106.30	1053	1863	TCP	0850	1016
148.247.1.30	148.247.40.145	1028	161	UDP	0850	1016
148.247.1.30	148.247.60.53	1028	161	UDP	0850	1016
148.247.1.30	216.239.37.99	1084	80	TCP	0850	1016
148.247.1.30	66.252.21.163	1086	80	TCP	0850	1016
148.247.1.30	65.54.211.61	1087	80	TCP	0850	1016
148.247.1.30	65.54.194.118	1088	80	TCP	0850	1016
148.247.1.30	192.168.2.2	1089	15871	TCP	0850	1016
148.247.1.30	63.99.209.61	1091	80	TCP	0850	1016
148.247.1.30	63.99.209.61	1092	80	TCP	0850	1016
148.247.1.30	209.225.14.12	1094	80	TCP	0850	1016
148.247.1.30	192.168.2.2	1095	15871	TCP	0850	1016
148.247.1.30	63.99.209.46	1097	80	TCP	0850	1016
148.247.1.30	63.99.209.46	1098	80	TCP	0850	1016
148.247.1.30	63.99.209.46	1100	80	TCP	0850	1016

Figura 5.7: Consultas sobre las comunicaciones de la red en un determinado período de tiempo.

El resultado de la consulta SQL es convertido en una tabla de HTML, mediante PHP, y mostrada a través de la interfaz web. En la figura 5.7 se muestra un ejemplo del resultado de una consulta realizada mediante el método anteriormente descrito.

Los resultados de la búsqueda muestran únicamente los campos no repetidos en la base de datos, es decir, si en la base de datos existen dos o más registros iguales, la interfaz web mostrará únicamente un solo resultado. Esto evita llenar la tabla de la página web con resultados duplicados. Esto nos sirve para evitar una cuenta errónea del número total de comunicaciones que existen en la red en un intervalo de tiempo.

5.3.5. Nodos activos

Otra de las tablas que pueden ser consultadas en la interfaz web es la de las computadoras que se encuentran dentro de la red, la cual se muestra en la tabla 5.1.

Tabla 5.1: Estado de las computadoras de la red. El estado se representa con un círculo de color determinado (detalles en el texto).

Estado de la red.				
IP	**Nombre**	**Fecha**	**Hora**	**Estado**
192.168.1.1	Maq1.org	28/IV/03	13:00	
192.168.1.2	No reg.	27/IV/03	12:00	
192.168.1.3	No reg.	28/IV/03	05:00	
192.168.1.4	No reg.	28/IV/03	13:00	
192.168.1.5	No reg.			

En ésta tabla se muestra el número IP de cada computadora, su nombre en el caso de que se encuentren registradas en el DNS, así como la última fecha y hora en la que se encontraban activas y una columna sobre el estado del nodo, el cual es un pequeño círculo de un color determinado:

Azul: Cuando la computadora está registrada en el DNS y está activa.

Verde: Si la computadora se encuentra activa, pero no se encuentra registrada en el DNS.

Amarilla: Si la computadora se encuentra inactiva por más de 1 hora, pero menos de 24.

Roja: Si la computadora se encuentra inactiva más de 24 horas.

Morada: Para las computadoras que nunca han estado activas en la red (nunca han contestado una petición ARP).

Un sistema similar está realizado en Hawk [17].

5.3.6. Gráficas

Como ya se había mencionado en los capítulos anteriores, en la interfaz web también es posible observar las gráficas creadas por el sistema.

Estas gráficas son la comparacón entre el modelo estadístico, el tráfico como se va recibiendo, y el tráfico después de ser sometido a un proceso de suavización, similar al utilizado en la creación del modelo estadístico, tal como se muestra en la figura 5.8.

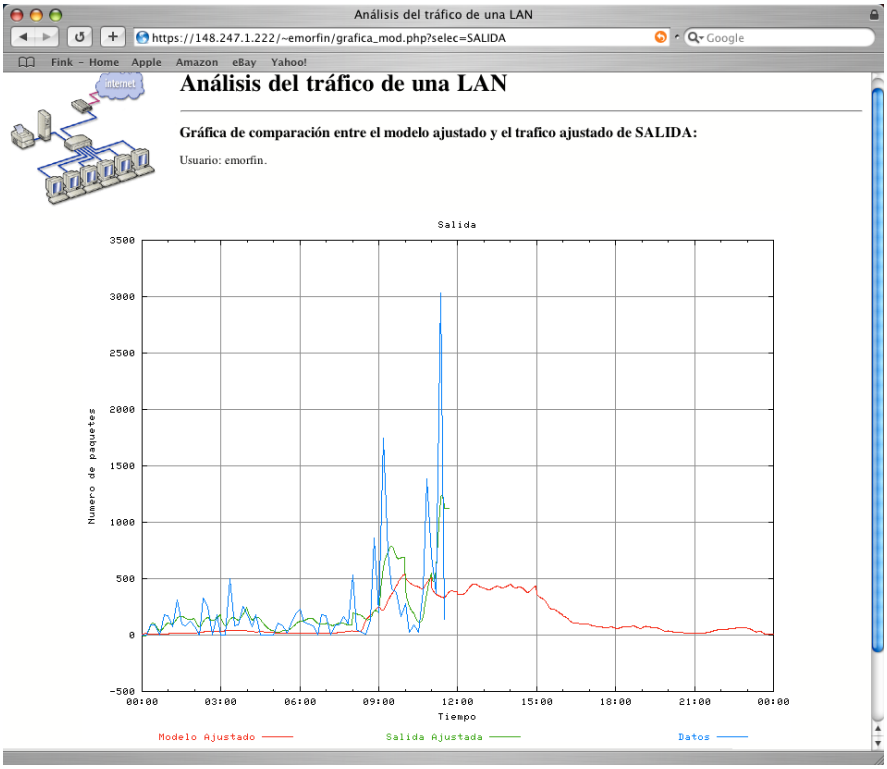


Figura 5.8: Comparación del modelo con el tráfico en bruto y el tráfico suavizado.

Otra de las gráficas que es posible observar en la interfaz web, son las gráficas de errores, como la mostrada en la figura 5.9, gracias a las cuales es posible observar qué tan grande es la diferencia entre el modelo estadístico con el tráfico que se ha capturado en ese mismo intervalo de tiempo.

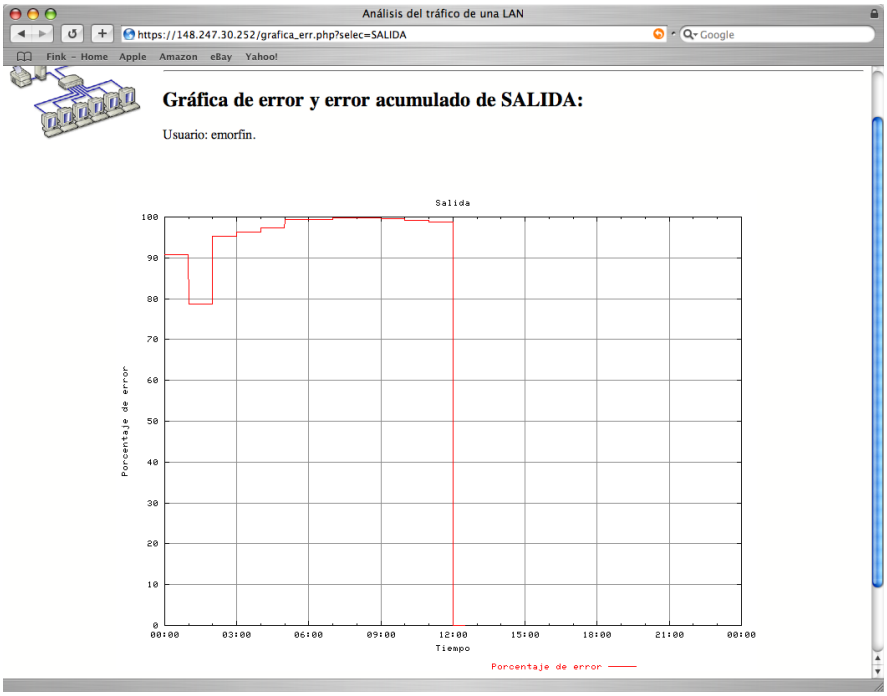


Figura 5.9: Gráfica de error.

5.4. Resultados

Gracias a todos los programas de análisis y comparación de datos, así como de la interfaz web es posible detectar varios problemas que ocurren en la red, como son ataques, correo spam, virus, escaneo de computadoras, gusanos, etc.

En la figura 5.10 se muestra una de las gráficas creadas por los programas de análisis del tráfico y mostradas en la interfaz web, en ésta figura se puede apreciar la comparación entre el modelo estadístico de entrada y el tráfico que está entrando a la red, en la cual se encuentra expandiéndose un gusano a través de las computadoras que componen ésta red. Esta gráfica ayudó a detectar un problema para poder solucionarlo antes de que se propagara y tuviera efectos mayores.

Como puede apreciarse en ésta figura, el tráfico que genera éste gusano es más alto de lo que sería considerado como normal, manteniéndose casi

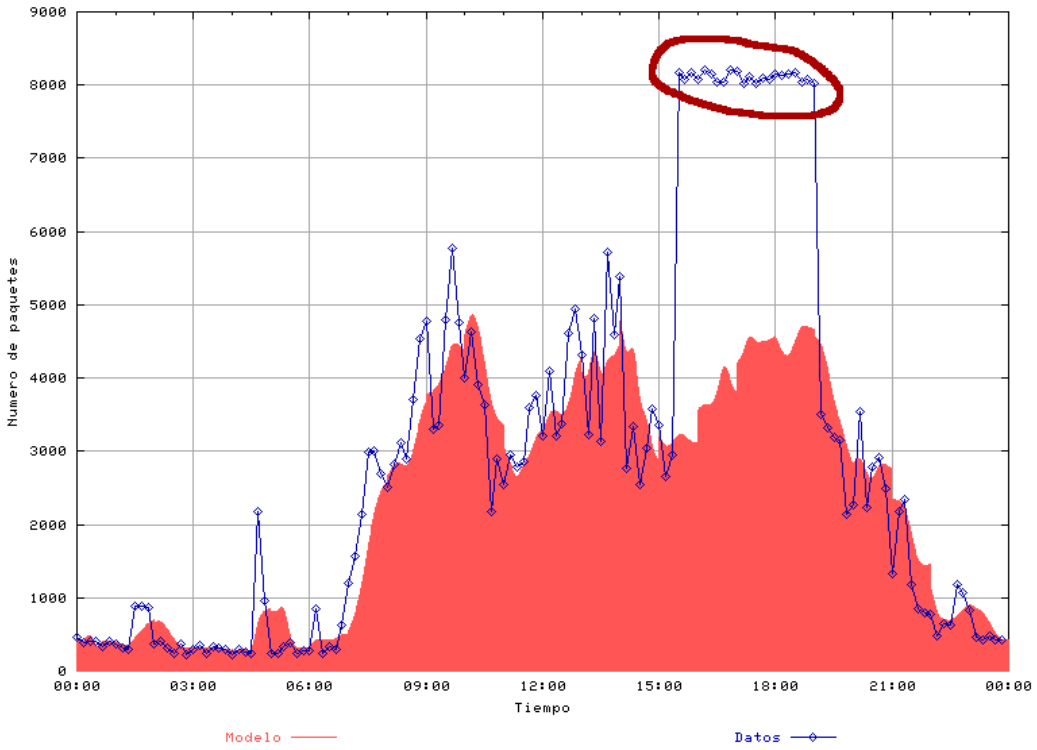


Figura 5.10: Comparación del modelo estadístico contra el tráfico de la red durante la expansión de un gusano.

constante durante un intervalo considerable de tiempo, que es el intervalo que tardó en controlarse y solucionarse éste problema.

En la figura 5.11 se puede apreciar el error porcentual y el error porcentual acumulado calculados al comparar el modelo estadístico con el tráfico capturado en la red (suavizado y ajustado) durante el intervalo de la expansión de un gusano. Como se puede observar, el porcentaje de error se mantiene bajo durante el período de funcionamiento normal de la red, pero es considerablemente elevado en el intervalo de tiempo que el gusano estuvo expandiéndose. El error se reduce cuando el problema es solucionado. Mediante una consulta al sistema es posible detectar la IP de la computadora que esté generando problemas en la red.

Es posible también realizar la detección de algunas fallas en la red, ya sea

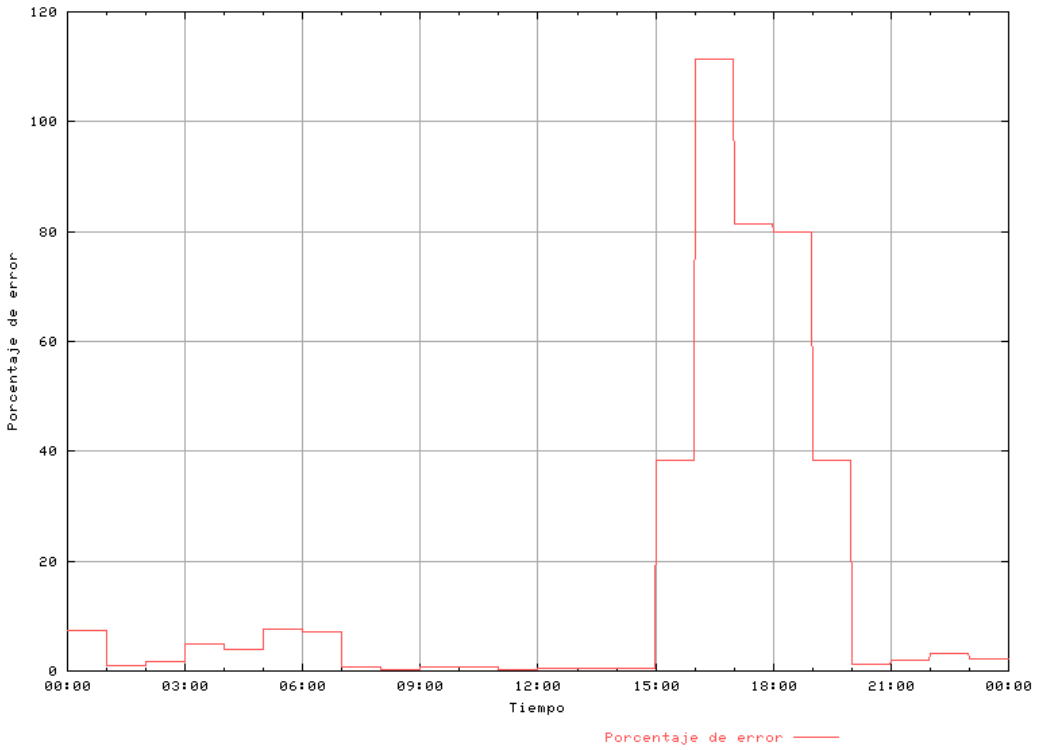


Figura 5.11: Error y error acumulado durante la expansión de un gusano en la red.

fallas en las comunicaciones con una determinada computadora, fallas en los servidores, o la falla de un concentrador o switch.

En la figura 5.12, se puede apreciar otra de las gáficas creadas mediante los programas de análisis y mostradas en la interfaz web, en la cual se muestra una falla.

La falla se debe a problemas que se presentaron en la puerta de enlace predeterminada de la red. Debido a estos problemas, el tráfico es mucho más bajo de lo esperado, y en algunos casos el tráfico es cero, es decir inexistente. Cuando el tráfico es cero, se debe a que durante algunos períodos de tiempo, la puerta de enlace no se encontraba activa. El tráfico que aparece en el intervalo de la falla es debido a la inestabilidad de la puerta de enlace (gateway).

En la figura 5.13 se puede apreciar el error porcentual calculado al realizar la comparación del tráfico capturado durante la falla de la puerta de enlace

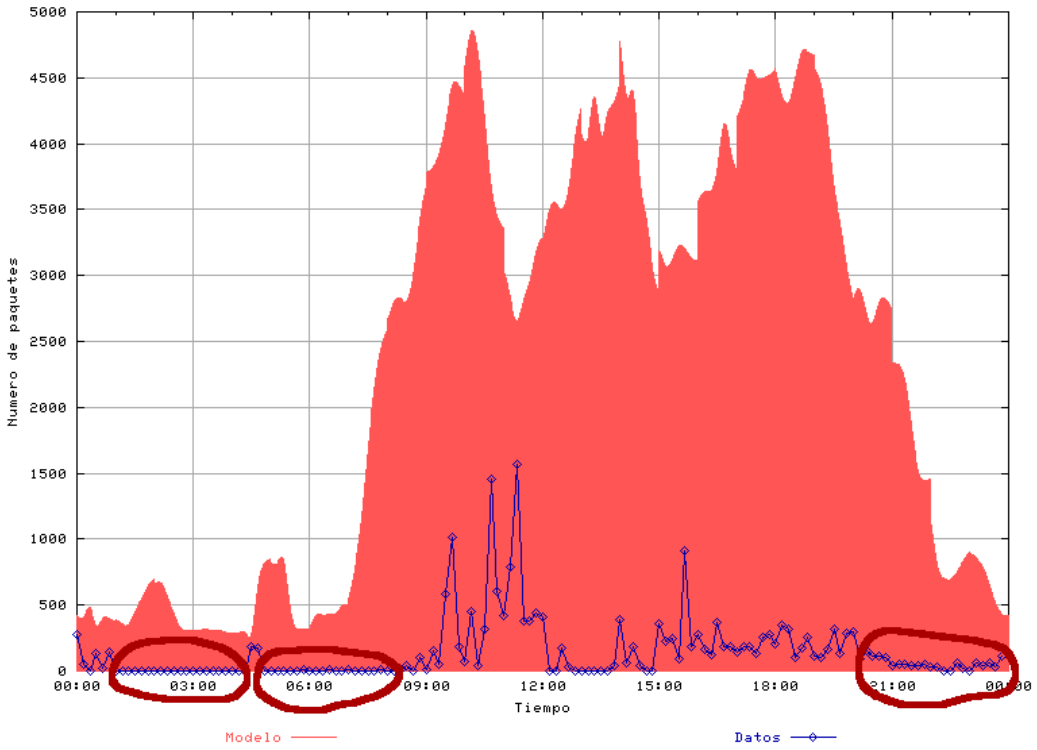


Figura 5.12: Comparación del modelo estadístico contra el tráfico de la red durante una falla.

predeterminada de la red (suavizado y ajustado) contra el modelo estadístico.

En ésta figura se puede apreciar los altos porcentajes de error, los cuales se encuentran cerca del 100 % en cada intervalo de una hora, esto es debido a la gravedad de la falla. El porcentaje de error se mantuvo alto durante todo el día debido a que por su naturaleza fué difícil solucionar el problema.

En el momento en que el sistema detecta alguna irregularidad en la red, manda un correo electrónico al administrador, avisando del problema que detectó y solicitando que consulte la interfaz web, para poder obtener más detalles sobre lo que está ocurriendo.

Las decisiones y los correos electrónicos que manda el sistema están basadas en los porcentajes de error, ya que si el error es menor a un 20 % se considera normal, si el error es mayor a dicho porcentaje, pero menor a 80 %

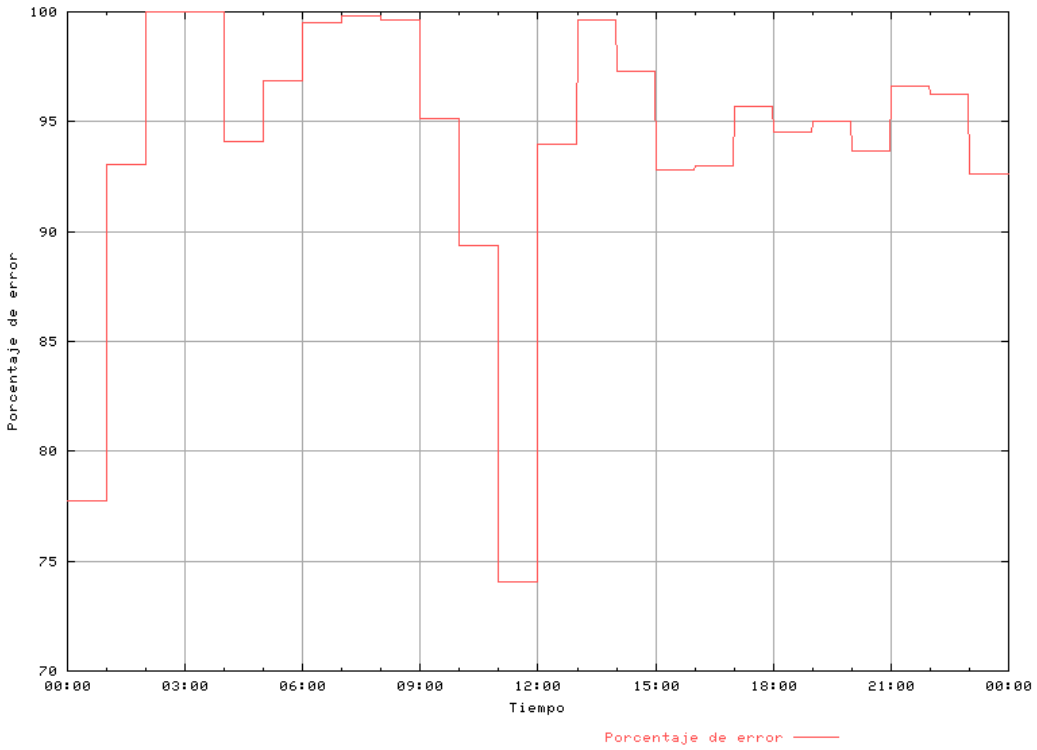


Figura 5.13: Error y error acumulado durante la falla de la puerta de enlace predeterminada de la red.

el mensaje es solo de alerta, pero si el error es mayor al 80 %, el mensaje es de alarma. Estos parámetros están fijos en los programas de cálculo de error y toma de decisiones.

El sistema también ha reportado algunas pequeñas anormalidades en la red, debidas a un incremento momentáneo en el tráfico en un determinado instante, pero que son provocados aleatoriamente ya sea por un aumento espontáneo en el número de usuarios de la red, o debido a que un solo usuario tiene alguna necesidad específica momentánea, como podría ser la descarga de archivos muy grandes desde la red (por ejemplo una imagen ISO), ya sea mediante el uso del protocolo FTP, HTTP, AFS, SAMBA, SSH o alguna otra aplicación particular.

De igual manera ha sido reportado por el sistema en algunas ocasiones el

caso contrario, es decir, bajas en la cantidad de tráfico en la red, también de forma aleatoria, debido a una disminución temporal de usuarios de la red.

Se han presentado también falsos positivos debido a que el modelo utilizado es estático.

Capítulo 6

Conclusiones

El análisis y modelación del tráfico de las redes es un problema complejo, que se encuentra aún en estudio. Ya se ha logrado modelar las conexiones punto a punto, pero no las conexiones de toda una red, aunque ya existen algunos modelos basados en la autosimilaridad y lógica difusa.

El presente trabajo contribuye de alguna manera en el avance para la solución de la modelación del tráfico de una LAN ya que se presenta un método sencillo para la creación de un modelo estadístico estático de una LAN, el cual puede ser utilizado para predecir el tráfico de la red.

Actualmente se cuenta con un mecanismo eficiente que nos permite capturar el tráfico que circula por una LAN y realizar su correspondiente análisis.

En el presente trabajo se presenta un modelo estadístico del tráfico que circula por una LAN, que si bien no es extremadamente preciso, es adecuado para los propósito de detección de algunos ataques a la red y de fallas de los diferentes componentes que integran la red.

El hecho de que el modelo no sea muy preciso, le da la ventaja de ser lo bastante rápido como para poder realizar todo el análisis y las detecciones ya sea de fallas o ataques solo unos cuantos minutos después de que empezaron a ocurrir.

La detección en tiempo real no es posible con nuestro sistema, ya que para ello creemos que sería necesario contar con muchos más recursos, como hardware especializado; o extender el modelo a un mayor margen de error, lo que lo haría no aceptable, ya que se perdería demasiada resolución al

momento de detectar alguna falla o ataque a la red.

Por medio de éste modelo es posible realizar una predicción del tráfico que circulará por la red en los siguientes instantes de tiempo, y con base en esta predicción realizar una detección de irregularidades en la red.

El modelo es estático, es decir, se realizó un modelo del tráfico de la red en base al tráfico que circuló por la red durante un determinado período de tiempo en el que se realizó un muestreo de los paquetes que circularon por la red.

El sistema no es capaz de detectar todos los ataques que se presenten en una red, debido a que el modelo estadístico es estático. En el caso de presentarse una falla, no siempre será capaz de mandar el correo de aviso al administrador de la red.

Se desarrolló una interfaz gráfica que permite realizar consultas de una manera sencilla a la base de datos; pero no es capaz de realizar consultas más complejas o específicas a la misma.

El tráfico que circula en las redes es dinámico, es decir, constantemente está cambiando de acuerdo a el número de usuarios o computadoras que se encuentran conectados, el tipo de servicios que presta o requiere la red, etc. por lo que el modelo debe ser dinámico.

Un modelo dinámico debe adaptarse conforme la red va cambiando, tanto en sus requerimientos como en su composición y servicios que brinda.

Como trabajo a futuro, se propondría un analizador dinámico. Para lograrlo es necesario que el tráfico capturado sea evaluado y en caso necesario incorporarlo a los datos con los que ya se cuenta, con el propósito de crear un nuevo modelo en unos minutos. Este nuevo modelo sería más preciso que el anterior.

También se debe contar con un mecanismo que su función sea la de evaluar los datos que ya no son adecuados al nuevo comportamiento de la red y con base en ello eliminar los conjuntos de datos para mantener un modelo adecuado al comportamiento actual de la red.

Apéndice A

Interpolación cúbica de trazador

Las aproximaciones polinómicas son un método apropiado en muchas circunstancias, pero la naturaleza oscilatoria de los polinomios de mayoir grado y la propiedad de que una fluctuación sobre una porción pequeña de un intervalo puede incidir en fluctuaciones muy grandes sobre el rango entero, restringen su uso cuando se aproximan muchas de las funciones que surgen en situaciones físicas reales.

Un enfoque alternativo que puede usarse para obtener funciones interpolantes consiste en dividir el intervalo en una colección de subintervalos y construir un polinomio aproximadamente diferente en cada subintervalo. La aproximación con funciones de este tipo se llama **aproximación polinómica segmentaria**.

El tipo más simple de aproximación polinómica segmentaria es la interpolación lineal segmentaria que consiste en unir un conjunto de datos de puntos

$$[x_0, f(x_0)], [x_1, f(x_1)], \ldots, [x_n, f(x_n)])$$

con una serie de líneas rectas.

La desventaja de enfocar un problema de aproximación usando funciones de éste tipo es que en cada uno de los extremos de los subintervalos, no hay ninguna seguridad de diferenciabilidad, lo cual, en un contexto geométrico,

significa que la función interpolante no es "lisa" en esos puntos. Frecuentemente por las condiciones físicas es claro que se requiere ésta condición y en estos casos la función aproximante debe ser continuamente diferenciable.

Posiblemente el tipo más simple de función polinómica segmentaria diferenciable en un intervalo entero $[x_0, x_n]$ es la función que se obtiene ajustando un polinomio cuadrático entre cada par de nodos sucesivos. Esto se hace construyendo una cuadrática en $[x_0, x_1]$ que coincida con la función en x_0 y x_1, otra cuadrática en $[x_1, x_2]$ que coincida con la función en x_1 y x_2, y así sucesivamente. Como un polinomio cuadrático general tiene tres constantes arbitrarias y sólo se requieren dos condiciones para ajustar los datos en los extremos de cada subintervalo, existe la flexibilidad suficiente para permitir que las cuadráticas se escojan además de tal manera que, la interpolante tenga derivada contínua en $[x_0, x_n]$.

La interpolación polinómica segmentaria más común usando polinomios cúbicos entre parejas sucesivas de nodos se llama **interpolación cúbica de trazador**. Un polinomio cúbico general involucra cuatro constantes; así que hay suficiente flexibilidad en el procedimiento del trazador cúbico para garantizar no solamente que la interpolante sea continuamente diferenciable en el intervalo, sino que tenga también una segunda derivada continua. La construcción del trazador cúbico, sin embargo, no supone que las derivadas de la interpolante coinciden con las de la función ni siquiera en los nodos.

Dada una función f definida en $[a, b]$ y un conjunto de números, llamados los nodos, $a = x_0 < x_1 < \ldots < x_n = b$, un interpolante cúbico de trazador, S, para f es una función que satisface las siguiente condiciones:

1. S es un polinomio cúbico, denotado S_j, en el subintervalo $[x_j, x_{j+1}]$ para cada $j = 0, 1, \ldots, n-1$
2. $S(x_j) = f(x_j)$ para cada $j = 0, 1, \ldots, n$
3. $S_{j+1}(x_{j+1}) = S_j(x_{j+1})$ para cada $j = 0, 1, \ldots, n-2$
4. $S'_{j+1}(x_{j+1}) = S'_j(x_{j+1})$ para cada $j = 0, 1, \ldots, n-2$
5. $S''_{j+1}(x_{j+1}) = S''_j(x_{j+1})$ para cada $j = 0, 1, \ldots, n-2$

6. se satisface una del siguiente conjunto de condiciones de frontera:

$$S''(x_0) = S''(x_n) = 0 \qquad \text{(fontera libre)} \tag{A.1}$$

$$S'(x_0) = f'(x_0) \quad \text{y} \quad S'(x_n) = f'(x_n) \quad \text{(frontera sujeta)} \tag{A.2}$$

Cuando se satisfacen las condiciones de frontera libre, el trazador se llama **trazador natural** y su gráfica se aproxima a la forma que tomaría una varilla larga flexible si se le forzara a pasar por cada uno de los puntos de los datos $(x_0, f(x_0)), (x_1, f(x_1)), \ldots, (x_n, f(x_n))$.

En general, las condiciones de frontera sujeta nos llevarán a aproximaciones más exactas ya que incluyen más información acerca de la función; sin embargo, para que este tipo de condición a la frontera se implemente, es necesario tener, ya sea los valores de la derivada en los extremos, o una aproximación precisa de estos valores.

Para construir el interpolante cúbico de trazador para una función f dada, se pueden aplicar las condiciones de la definición a los polinomios cúbicos.

$$S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3$$

para cada $j = 0, 1, \ldots, n - 1$.

Claramente,

$$S_j(x_j) = a_j = f(x_j)$$

y se aplica la condición (3),

$$a_{j+1} = S_{j+1}(x_{j+1}) = S_j(x_{j+1}) = a_j + b_j(x_{j+1} - x_j) + c_j(x_{j+1} - x_j)^2 + d_j(x_{j+1} - x_j)^3$$

para cada $j = 0, 1, \ldots, n - 2$.

Como el término $(x_{j+1} - x_j)$ se usará repetidamente en el presente desarrollo, es conveniente introducir una notación más simple,

$$h_j = x_{j+1} - x_j$$

para cada $j = 0, 1, \ldots, n-1$. Si, además definimos $a_n = f(x_n)$, se puede ver que esto implica la ecuación

$$a_{j+1} = a_j + b_j h_j + c_j h_j^2 + d_j h_j^3 \tag{A.3}$$

se satisface para cada $j = 0, 1, \ldots, n-1$.

De una manera similar, definimos $b_n = S'(x_n)$ y observamos que

$$S_j'(x) = b_j + 2c_j(x - x_j) + 3d_j(x - x_j)^2$$

implica que $S_j'(x_j) = b_j$ para cada $j = 0, 1, \ldots, n-1$. Aplicando la condición (4),

$$b_{j+1} = b_j + 2c_j h_j + 3d_j h_j^2 \tag{A.4}$$

para cada $j = 0, 1, \ldots, n-1$.

Otra relación entre los coeficientes de S_j se puede obtener definiendo $c_n = \frac{S''(x_n)}{2}$ y aplicando la condición (5). En este caso

$$c_{j+1} = c_j + 3d_j h_j \tag{A.5}$$

para cada $j = 0, 1, \ldots, n-1$.

Despejando d_j de la ecuación (A.5) y sustituyendo este valor en las ecuaciones (A.3) y (A.4) da las nuevas ecuaciones

$$a_{j+1} = a_j + b_j h_j + \frac{h_j^2}{3}(2c_j + c_{j+1}) \tag{A.6}$$

y

$$b_{j+1} = b_j + h_j(c_j + c_{j+1}) \tag{A.7}$$

para cada $j = 0, 1, \ldots, n-1$.

La relación final que involucra a los coeficientes se puede obtener resolviendo la ecuación apropiada en la forma de la ecuación (A.6), primero para b_j,

$$b_j = \frac{1}{h_j}(a_{j+1} - a_j) - \frac{h_j}{3}(2c_j + c_{j+1}) \tag{A.8}$$

y luego, con una reducción del índice, para b_{j-1},

$$b_{j-1} = \frac{1}{h_{j-1}}(a_j - a_{j-1}) - \frac{h_{j-1}}{3}(2c_j + c_j).$$

Sustituyendo estos valores en la ecuación derivada de la ecuación (A.7), cuando el índice se reduce en uno, se obtiene el sistema lineal de ecuaciones

$$h_{j-1}c_{j_1} + 2(h_{j-1} + h_j)c_j + h_j c_{j+1} = \frac{3}{h_j}(a_{j+1} - a_j) - \frac{3}{h_{j-1}}(a_j - a_{j-1}) \tag{A.9}$$

para cada $j = 1, 2, \ldots, n-1$. Este sistema tiene como incógnitas sólo a $\{c_j\}_{j=0}^{n}$ ya que los valores de $\{h_j\}_{j=0}^{n-1}$ y $\{a_j\}_{j=0}^{n}$ están dados por el espaciamiento entre los nodos $\{x_j\}_{j=0}^{n}$ y los valores de f en los nodos.

Nótese que una vez que se conocen los valores de $\{c_j\}_{j=0}^{n}$ encontrar las constantes restantes $\{b_j\}_{j=0}^{n-1}$ de la ecuación (A.8) y $\{d_j\}_{j=0}^{n-1}$ de la ecuación (A.5) y construir los polinomios cúbicos $\{S_j\}_{j=0}^{n-1}$ es una cuestión sencilla.

La mayor pregunta que surge en conexión con ésta construcción es si los valores de $\{c_j\}_{j=0}^{n}$ pueden ser encontrados usando el sistema de ecuaciones dado en (A.9) y si es así, si estos valores son únicos. Los siguientes teoremas indican que, cuando cualquiera de las condiciones de frontera dadas en la parte (6) de la definición se imponen, la respuesta a ambas preguntas es definitiva.

Teorema 1 *Si f es una función definida en $[a, b]$, entonces f tiene un único interpolante de trazador natural, o sea un único interpolante de trazador que satisface las condiciones de frontera libre $S''(a) = S''(b) = 0$.*

Demostración 1 *Con la notación usual, $a = x_0 < x_1 < \ldots < x_n = b$, las condiciones de frontera en este caso implican que $c_n = \frac{S''(x_n)}{2} = 0$ y que*

$$0 = S''(x_0) = 2c_0 + 6d_0(x_0 - x_0)$$

así que $c_0 = 0$.

Las dos ecuaciones $c_0 = 0$ *y* $c_n = 0$ *junto con las ecuaciones en (A.9) producen un sistema lineal descrito por la ecuación vectorial* $A\vec{x} = \vec{b}$, *donde* A *es la matriz de* $(n+1)$ *por* $(n+1)$

$$A = \begin{bmatrix} 1 & 0 & 0 & \dots & \dots & 0 \\ h_0 & 2(h_0+h_1) & h_1 & \ddots & & \vdots \\ 0 & h_1 & 2(h_1+h_2) & h_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & h_{n-2} & 2(h_{n-2}+h_{n-1}) & h_{n-1} \\ 0 & \dots & \dots & 0 & 0 & 1 \end{bmatrix}$$

y $\vec{b}$ *y* $\vec{x}$ *son los vectores*

$$\vec{b} = \begin{bmatrix} 0 \\ \frac{3}{h_1}(a_2 - a_1) - \frac{3}{h_0}(a_1 - a_0) \\ \vdots \\ \frac{3}{h_{n-1}}(a_n - a_{n-1}) - \frac{3}{h_{n-2}}(a_{n-1} - a_{n-2}) \\ 0 \end{bmatrix} \quad y \quad \vec{x} = \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{bmatrix}$$

La matriz A *es esctrictamente dominante diagonalmente, por lo tanto, es sistema lineal tiene una solución única para* $c_0, c_1, \dots, c_n$.

La solución al problema del trazado cúbico con las condiciones de frontera $S''(x_0) = S''(x_n) = 0$ se obtener aplicando el algoritmo de sección A.1.

A.1. Algoritmo de trazador cúbico natural

Para construir el interpolante cúbico de trazador S para la función f, definida en los números $x_0 < x_1 < \dots < x_n$, y que satisface $S''(x_0) = S''(x_n) = 0$:

ENTRADA n; x_i y $a_i = f(x_i)$ para $i = 0, 1, \dots, n$.

SALIDA a_j, b_j, c_j, d_j para $j = 0, 1, \ldots, n-1$.

(Nota: $S(x) = S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3$ para $x_j \leq x \leq x_{j+1}$)

1. Para $i = 0, 1, \ldots, n-1$ tomar $h_i = x_{i+1} - x - i$.

2. Para $i = 1, 2, \ldots, n-1$ tomar

$$\alpha_i = \frac{3[a_{i+1}h_{i-1} - a_i(x_{i+1} - x_{i-1}) + a_{i-1}h_i]}{h_{i-1}h_i}$$

3. Tomar $l_0 = 1$;

   $\mu_0 = 0$;

   $z_0 = 0$.

4. Para $i = 1, 2, \ldots, n-1$ tomar $l_i = 2(x_{i+1} - x_{i-1}) - h_{i-1}\mu_{i-1}$;

$$\mu_i = \frac{h_i}{l_i};$$
$$z_i = \frac{(\alpha_i - h_{i-1}z_{i-1})}{l_i}.$$

5. Tomar $l_n = 1$;

   $z_n = 0$;

   $c_n = 0$.

6. Para $j = n-1, n-2, \ldots, 0$ tomar $c_j = z_j - \mu_j c_{j+1}$;

$$b_j = \frac{(a_{j+1} - a_j)}{h_j} - \frac{h_j(c_{j+1} + 2x_j)}{3};$$
$$d_j = \frac{(c_{j+1} - c_j)}{(3h_j)}.$$

7. SALIDA (a_j, b_j, c_j, d_j para $j = 0, 1, \ldots, n-1$).

8. PARAR.

Bibliografía

[1] R. Jain and S. A. Routhier. Packet trains - measurements and a new model for computer network traffic. *IEEE Journal on Selected Areas in Communications*, Sac. 4 No. 6:986–995, 1986.

[2] M. Molina, P. Castelli, and G. Foddis. Web traffic modeling, exploting tcp connections. *IEEE Network*, May-Jun:46–55, 2000.

[3] V. Paxson and S. Floyd. Wide-area traffic: The failure of poisson modeling. *ACM/SIGCOM '94*, pages 257–268, 1994.

[4] D.N. Murray and P. H. Enslow Jr. An experimental study of the performance of a local area network. *IEEE Communications Magazine*, 22(11):48–53, 1984.

[5] W. Willinger, M. Taqqu, R. Sherman, and D. V. Wilson. Self-similarity through high variability: Statistical analysis of ethernet lan traffic at the source level. *IEEE/ACM Transactions on Networking*, 5(1):71–86, 1997.

[6] W. E. Leland, M. S. Taqqu, W. Willenger, and D. V. Wilson. On the self-similar nature of ethernet traffic (extended version). *IEEE/ACM Transactions on Networking*, 2(1):1–15, 1994.

[7] C. Bor-Sen, P. Sen-Chueh, and W. Ku-Chen. Traffic modeling, prediction, and congestion control for high-speed networks: A fuzzy ar aproach. *IEEE Transactions on Fuzzy Systems*, 8(5):491–508, 2000.

[8] Averill M. Law and W. David Kelton. *Simulation Modeling and Analysis*. McGraw-Hill, 2000.

[9] Andrew S. Tanenbaum. *Computer Networks*. Prentice Hall, third edition, 1996.

[10] M. Rivero and D. Lara. Análisis del retardo de la transmisión de paquetes en edge sobre sistemas celulares de tercera generación. Sexta Conferencia de Ingeniería Eléctrica CINVESTAV, Septiembre 2000.

[11] V. Frost and B. Melamed. Traffic modeling for telecomunications networks. *IEEE Communications Magazine*, 32(3):70–80, 1994.

[12] M. Rivero and D. Lara. Desempeño y dimensionamiento del nodo de salida de la red de datos del cinvestav. Reporte de la Sección de Comunicaciones, Mayo 2002.

[13] M. Rivero. Transmisión de datos por paquetes en sistemas celulares de tercera generación: La norma edge. Master's thesis, Sección de Comunicaciones del CINVESTAV, Octubre del 2000.

[14] C. Williamson. Internet traffic measurement. *IEEE Internet Computing*, Nov-Dic:70–74, 2001.

[15] K. C. Claffy. Measuring the internet. *IEEE Internet Computing*, Jan-Feb:73–75, 2000.

[16] R. Burden and J. Douglas. *Análisis Numérico*. Grupo Editorial Iberoamérica, tercera edition, 1985.

[17] G. Heim. Maintaining dns sanity with hawk. *The Journal for Unix and Linux systems administrators*, (12), 2002.