SnS

Serial Network Sniffer

Proyecto de Fin de Carrera

Gabriel Botti Naccarato
Francisco Pérez Pequeño
Paola Sciarra Gatti

Tutor: Ing. Julio Pérez

Universidad de la República
Facultad de Ingeniería

Contenido

Capitulo 1

Capitulo 2

Capitulo 3

Capitulo 4

Capitulo 5

Capitulo 6

Capitulo 7

Capitulo 8

Conclusiones

ANEXO I

ANEXO II

ANEXO III

ANEXO IV

ANEXO V

Contenido de Figuras

Contenido de Tablas

Capitulo 1

INTRODUCCIÓN GENERAL DEL PROYECTO

1.1 Objetivos y descripción del proyecto.

El proyecto de grado SnS (Serial Network Sniffer) consistió en el diseño e implementación de un analizador de protocolos de redes de datos, cuya interfaz es la V.35.

El objetivo del proyecto fue lograr una herramienta, para un primer diagnóstico del problema, que sustituyera los analizadores Hewlett Packard utilizados hoy en día por los técnicos de ANTEL, los cuales tienen un costo aproximado de 40.000 USD. En su lugar se presenta el analizador SnS, específico para interfaz V.35, a un costo menor a los 2.000 USD. El analizador SnS cuenta con menos prestaciones que los analizadores HP mencionados pero el mismo es adecuado según los requerimientos planteados por el cliente.

Al presentarse un problema de comunicación entre dos equipos con interfaz V.35, es necesario acceder al lugar con un equipo portátil y económico.

SnS es un equipo, que dadas sus dimensiones y peso lo hacen sencillo de transportar. Analiza los datos que están comunicando los equipos, identificando la causa de los problemas de red, a partir de la capa de enlace, a un costo sensiblemente menor que el de los analizadores disponibles en el mercado.

1.2 Metodología de trabajo.

El proyecto se dividió en cuatro etapas principales:

1. Estudio teórico del problema
2. Diseño y selección del hardware analizador
3. Desarrollo del software analizador de protocolos
4. Procesamiento de datos obtenidos y conclusiones

Para el estudio teórico del problema, en primer lugar, se tomaron en cuenta los requerimientos del cliente y luego se investigó acerca de las funcionalidades que presentaba el equipo analizador que ANTEL utiliza actualmente.

En la segunda etapa, luego de definidas las funcionalidades con las cuales cuenta el equipo, se investigó acerca del hardware que se encontraba disponible en el mercado para lograr el mejor diseño acorde tanto a los requerimientos técnicos como económicos.

Una vez definido el hardware se comenzó a desarrollar el software de captura, incluyendo una interfaz gráfica amigable para el usuario del equipo analizador.

La última etapa se basó en la realización de pruebas y procesamiento de los datos adquiridos para su posterior análisis. Se compararon los datos obtenidos en las mediciones realizadas con el analizador que utiliza actualmente ANTEL y el equipo SnS.

1.3 Organización del documento.

En el primer y segundo capítulo capitulo se explican las características y procedimientos generales del proyecto. Se introduce la estructura básica y se definen las principales etapas en las que se desarrolló este trabajo. Se presentan los componentes físicos utilizados en el analizador SnS y se introduce el software de captura diseñado. Además se explican las motivaciones y objetivos del proyecto así como la metodología de trabajo y la organización del documento.

En los capítulos 3 y 4 se estudian detalladamente los fundamentos, tanto teóricos como prácticos, en los que se sustentó o se basó la elección del hardware y software utilizado en el equipo SnS.

En los capítulos 5 y 6 se detalla el diseño del analizador desarrollado. Se describe el diseño del hardware del equipo y su principal componente, la tarjeta adquisidora de datos Sangoma S5141. Se detalla el cable de interconexión T utilizado y el software de captura desarrollado.

En el capítulo 7 se presenta un análisis comparativo de las capturas entre el analizador SnS y el Advisor de HP utilizado actualmente por ANTEL. Se explican las pruebas realizadas y se analiza la precisión con que el SnS inserta las marcas de tiempo.

El análisis de los costos relacionados con el proyecto se desarrolla en el capítulo 8.

Se dedica un capítulo final correspondiente a las conclusiones del proyecto.

Se adjuntan un manual de usuario y un manual de instalación a la documentación. En el manual de usuario se describen detalladamente las diferentes alternativas para acceder y controlar el analizador SnS.

Capitulo 2

DESCRIPCIÓN DEL ANALIZADOR SNS

El proyecto de grado SnS surgió con el interés por parte del Ing. Álvaro García de ANTELDATA, de obtener un analizador de protocolos para la interfaz ITU V.35 que cumpliera con ciertos requisitos básicos como el bajo costo, comparado con los analizadores Hewlett Packard que dispone en este momento y la portabilidad del mismo por parte de los técnicos de ANTEL.

Por ser un proyecto de grado, la duración total del mismo insumió los tres semestres destinados a ello. En cuanto a los requerimientos económicos, en una primera instancia el Ing. Álvaro García planteó un monto máximo de 1000 USD para la compra de los componentes del analizador, debiendo luego ampliar dicho monto debido a los costos que insumió la importación del hardware. Finalmente ANTEL puso a disposición del proyecto un monto total de 1150 USD para la compra de la tarjeta adquisidora utilizada. El costo del resto del hardware corrió por parte nuestra.

A estos requisitos se sumaron los requisitos técnicos inherentes al analizador mismo. A continuación se listan los requerimientos técnicos solicitados por la contraparte de ANTEL.

- Lectura de dos canales o puertos.
- Velocidad máxima de lectura de al menos 2.048Mbps por canal sincrónico.
- Interfaz ITU V.35
- Cada trama deberá tener una marca de tiempo, insertada con una precisión mayor a 1ms.
- Formato de datos HDLC.
- Tamaño y peso que permitan la portabilidad del equipo analizador.
- Conexión vía Ethernet a una PC portátil.
- Almacenamiento de datos adquiridos en un archivo.
- Análisis de datos con un programa que permita ser accedido por una interfaz gráfica.

El equipo SnS es capaz de adquirir datos en forma continua, leyendo de los dos canales simultáneamente a una velocidad máxima de 2.048Mbps por canal sincrónico. Se insertan marcas de tiempo a cada trama. La conexión a una PC portátil se realiza mediante la interfaz Ethernet. La interconexión entre el analizador y los equipos DTE y DCE es a través de un cable tipo T, el cuál no afecta la transmisión existente.

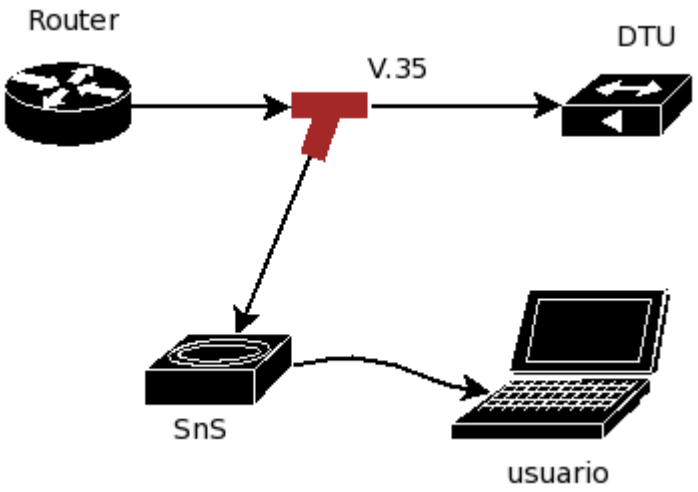


Figura 1: Conexión del analizador SnS

Inicialmente el formato de datos capturados debía contener el formato HDLC (*High-Level Data Link Control*), la versión final del analizador SnS amplía el formato de los datos a CHDLC (*Cisco HDLC*), Frame Relay y PPP (*Point to Point Protocol*). Los datos adquiridos son almacenados en un archivo para luego ser analizados con el software Ethereal. El formato del mencionado archivo es tal que el software Ethereal lo reconoce y por tanto es capaz de analizar. Por las características de los datos y la información a desplegar, estos son guardados con el formato *Network Associates Sniffer*.

En su primera versión final, el equipo SnS se basa en una plataforma de PC comercial con una placa compacta y moderna, siendo sus dimensiones 300mm x 280mm x 56mm y un peso aproximado de 2, 5 Kg, incluyendo el cable T de interconexión.

El componente principal de hardware del analizador SnS es la tarjeta adquisidora de datos Sangoma S5141, capaz de adquirir datos de los puertos primario y secundario simultáneamente a una velocidad máxima de 2.048Mbps. La tarjeta fue configurada con el formato de trama HDLC, a partir del cual se realizaron las capturas de formatos de tramas CHDLC, PPP y Frame Relay.

El sistema operativo (SO) elegido para desarrollar el software del analizador es LINUX en la distribución Mandrake 10.1. Sobre el mencionado SO se desarrolló el software de captura,

utilizando para esto el lenguaje de programación C y las librerías (APIs) del fabricante Sangoma.

El software desarrollado realiza la captura de datos de ambos puertos en forma simultánea e independiente, a una velocidad máxima de 2.048Mbps. Una vez iniciada la captura, el programa principal es capaz de capturar los datos de ambos puertos y según la condición seleccionada de fin de captura atenderá la interrupción. Los datos son guardados en archivos independientes, con los cuales se realiza el *merge* de los datos según las marcas de tiempo insertadas. Una vez efectuado y guardado el archivo final con los datos, el usuario podrá analizar la captura con el software analizador Ethereal.

Como se mencionó anteriormente, para el análisis de los datos capturados se utiliza el programa Ethereal, el cual es un analizador reconocido tanto en el ámbito académico como profesional por su gran versatilidad y por su calidad de software libre.

En el analizador SnS se encuentran instalados un servidor *vnc* y un servidor *ssh*. Estos servidores brindan al usuario la posibilidad de realizar transferencias de archivos a una PC portátil mediante una conexión de red, lo cual implica la ventaja de trabajar en forma remota si el usuario así lo prefiere. Una opción puede ser la de conexión remota vía Internet, siendo ésta opción sumamente ventajosa hoy en día por ser las conexiones a Internet de tan fácil acceso.

El analizador, al estar basado en una plataforma PC, brinda además la posibilidad de conexión de monitor y teclado y por lo tanto de ser controlado directamente.

Para inicializar el programa SnS, el cual da inicio a la captura, se debe invocar al mismo pasándole diferentes parámetros. Dentro de los parámetros que deberá seleccionar el usuario, se encuentra el formato de trama, la condición de fin para la captura y el nombre del archivo en el cual será guardada la captura para su posterior análisis.

El analizador SnS puede ser controlado remotamente desde una PC portátil con sistema operativo Windows conectado al analizador vía Ethernet. Para esto se brinda una interfaz grafica instalada en la PC con sistema operativo Windows. Por medio de ésta interfaz se lanzan los comandos al analizador utilizando el protocolo *ssh*. Los comandos ejecutados remotamente dan inicio a la captura, setean la condición de fin de captura y copian el archivo final desde el analizador SnS hacia la PC portátil, para su posterior análisis con el Ethereal.

Si el usuario lo desea, podrá abrir una consola del analizador desde el PC portátil utilizando el protocolo *ssh*. Desde esta consola el usuario puede iniciar la captura, pasando los parámetros necesarios al programa SnS. Uno de los parámetros consiste en la opción de trabajar en modo interactivo, en el cual se le pregunta al usuario el formato de trama, condición de fin y nombre del archivo.

Entre las opciones que presenta esta versión del analizador, con respecto al formato de trama a seleccionar, se encuentran disponibles los protocolos CHDLC, PPP y Frame Relay. En cuanto a las opciones de finalización de captura es posible elegir la finalización manual, o sea en el

instante que el usuario así lo decida, la finalización por tiempo así como también se presenta la opción de seleccionar la cantidad de paquetes a capturar. El usuario debe elegir el nombre del archivo en el cual será guardada la captura para su posterior análisis, sin que sea necesario agregar una extensión particular de archivo.

Capitulo 3

ELECCIÓN DE HARDWARE

3.1 Elección del Hardware Adquisidor de Datos

En el presente capítulo intentaremos describir brevemente el proceso de selección del hardware del analizador SnS. Partimos de una búsqueda general contemplando la especificación funcional del proyecto. Durante el proceso de selección se presentaron varias opciones, algunas de las cuales implicaban cambios leves en la especificación funcional del proyecto. Luego de evaluar las distintas opciones, optamos por una tarjeta de bus PCI con dos puertos seriales V.35.

Como se mencionó anteriormente, el hardware seleccionado debía ser tal que nos permitiera cumplir con las especificaciones funcionales del proyecto. A continuación la citamos nuevamente.

- Lectura de dos canales o puertos.
- Velocidad máxima de lectura de 2.048Mbps por canal sincrónico.
- Interfaz ITU V.35
- Cada trama deberá tener una marca de tiempo, insertada con una precisión mayor a 1ms.
- Formato de datos HDLC.
- Tamaño y peso que permitan la portabilidad del equipo analizador.

- Conexión vía Ethernet a una PC portátil.
- Almacenamiento de datos adquiridos en un archivo.
- Análisis de datos con un programa que permita ser accedido por una interfaz gráfica.

Durante el transcurso del proyecto, por solicitud del Ing. Álvaro García por la contraparte de ANTEL, se trabajó además de con las tramas HDLC iniciales, con tramas CHDLC (Cisco HDLC), PPP y Frame Relay.

Visualizamos entonces dos grandes bloques en la implementación del analizador, el hardware adquisidor de datos provenientes de la interfaz V.35 de los equipos a analizar y el análisis de los datos adquiridos. En este apartado nos ocuparemos de la elección del hardware, no obstante haremos una pequeña mención al software, ya que el mismo tuvo implicancias en la elección del hardware. Luego nos enfocaremos en la elección del hardware adquisidor propiamente dicho y terminaremos con una descripción de cómo se implementó el analizador.

3.1.1 Ventajas de una plataforma basada en una PC estándar

Para el análisis de los datos adquiridos, se utiliza un software libre ya implementado, probado y disponible en el medio. El software puede ser instalado en la PC del usuario, o en el caso que el hardware adquisidor incluya un computador el mismo puede instalarse en este último, como es el caso del analizador SnS. De este modo se evita la necesidad de instalar previamente el software analizador en la PC que el usuario pueda disponer para manejar el analizador, lo cual consideramos representa una gran ventaja.

3.1.2 Características físicas y eléctricas de la interfaz de conexión

La señal eléctrica correspondiente a la transmisión de datos y reloj de la interfaz V.35 está definida en la norma V.11 de la ITU-T. Las señales analizadas corresponden a transmisiones de como máximo una velocidad de 2,048Mbps. Por el motivo mencionado previamente, las señales eléctricas pueden llegar a tener una componente principal en frecuencia de 2,048MHz. A dicha frecuencia debe tenerse en cuenta la interferencia electromagnética, por ejemplo en el diseño de los cables si los mismos son largos.

3.2 Ventajas del Diseño Basado en Hardware Estándar

En facultad se nos aconsejó que para el proyecto tal cual como estaba planteado y en el tiempo que debíamos desarrollarlo utilizáramos hardware estándar. Si finalmente se decidía incluir en el proyecto el diseño y creación del hardware, el proyecto tendría que tener otros objetivos, dado que el tiempo que esto insumiría no permitiría el desarrollo de un producto final. Tal como fue planteado el proyecto por parte del Ing. Álvaro García, el objetivo consistía en un

producto terminado. Por este motivo fue descartada la posibilidad de implementar el hardware como parte de éste proyecto.

Una opción intermedia podría haber sido realizar el prediseño del hardware y solicitar a una empresa que lo implemente. Por este motivo se contactó a la empresa nacional *Controles*, la cual nos planteó que no es económicamente viable realizar una única unidad, aclarándonos que sin lugar a dudas superaría el costo de los 1.500 USD.

Ambas opciones presentan los riesgos intrínsecos a un nuevo producto, como por ejemplo la posibilidad de '*bugs*', desconocimiento del tiempo medio entre fallas, incertidumbre acerca de la garantía y posibilidad de no llegar en el tiempo supuesto en el desarrollo del producto.

3.3 Hardware del Analizador SnS

Entre la variedad existente en el mercado de opciones para el hardware del analizador, buscamos dispositivos capaces de adquirir los datos de la interfaz V.35 y comunicarlos por medio de una interfaz estándar del tipo bus USB, PCMCIA, PCI o Ethernet.

La opción elegida para desarrollar el analizador fue una tarjeta PCI con interfaz V.35. Aquí encontramos productos probados, para los cuales los fabricantes ofrecen garantía hasta por 5 años. Existen tarjetas de 1, 2, 4 o más puertos. Finalmente optamos por este tipo de hardware para implementar el analizador.

Una vez tomada la decisión de utilizar una tarjeta PCI con interfaz V.35, surgieron dos opciones para la conexión: una opción consistía en conectar la tarjeta a una PC y luego la PC al *laptop*. La otra opción consistía en conectarla a un *laptop* a través de un adaptador PCI a PCMCIA.

Los adaptadores PCI a PCMCIA, como el que sugiere el fabricante MAGMA[1], no garantizan el ancho de banda suficiente para trasmitir la captura de datos hacia el *laptop*, debido a que el ancho de banda depende de la arquitectura y del sistema operativo. El precio del adaptador en origen es de 979USD, si a este precio se le suman los gastos de importación y el costo de la tarjeta PCI se superaría ampliamente el monto proporcionado por ANTEL para el proyecto.

Se planteó entonces cómo comunicar una tarjeta PCI con una PC portátil. La solución encontrada fue la de utilizar una PC portátil con un *slot* PCI. Esta opción presenta múltiples ventajas que detallaremos en la implementación del analizador SnS.

[1] http://www.mobl.com/expansion/

El hardware del analizador SnS es una PC portátil (mini computadora) con una tarjeta PCI que proporciona una interfaz V.35. Como se mencionó anteriormente la tarjeta utilizada captura los datos de ambos puertos a 2.048Mbps.

El chasis final del analizador SnS, incluye una tarjeta madre de bajo consumo con *slot* PCI, una tarjeta adquisidora de datos con interfaz V.35, un disco duro para instalar el sistema operativo y el software de captura.

La tarjeta madre trae incorporada una interfaz Ethernet, por medio de la cual es posible el manejo remoto del analizador desde otra computadora, mediante un servicio de tipo *vnc* y/o *ssh*.

Al ser una PC con características estándar se tienen dos alternativas con respecto a la ubicación del software analizador. Una opción consiste en instalar el software analizador en la mini PC, como es el caso del SnS, lo cual tiene la ventaja de que los datos capturados son analizados directamente, sin tener que transferirlos por la red. La segunda alternativa es instalar el software analizador en una PC portátil, con lo cual los datos capturados son transferidos a la PC del usuario. La ventaja de esta opción reside en que los requerimientos de procesador y disco de la mini PC son menores.

NOTA: Los detalles de manejo del analizador se adjuntan en el Manual de Usuario del equipo SnS.

3.4 Alternativas Descartadas

3.4.1 Módulos pequeños Stand Alone.

Aquí podemos ubicar módulos con microcontrolador e interfaz Ethernet, como por ejemplo los Rabbits[2]. Tienen interfaz Ethernet y son capaces de manejar HDLC. El precio del kit de desarrollo al mes de agosto de 2004 se encontraba en el entorno de los 350 USD. Su tamaño es de aproximadamente 7cm x 5cm x 3cm. Presentan el inconveniente que si bien son capaces de manejar datos a 2.048Mbps, no pueden enviarlos por la interfaz Ethernet, debido al tiempo insumido en la ejecución del código necesario para realizar esa tarea. Esta opción quedó entonces descartada.

3.4.2 Kits de desarrollo de FPGA o DSP

[2] Por mas datos dirigirse a la página http://www.rabbitsemiconductor.com/

Existen kits de desarrollo con capacidad para manejar los datos de una interfaz V.35 y enviarlos a través de un bus USB o de la interfaz Ethernet. A pesar de que un FPGA o DSP sería económicamente viable, el problema se presenta al integrarlo a una tarjeta. Otra vez los problemas de diseño e implementación son las limitantes, debiendo utilizar entonces los kits de desarrollo. Estos kits tienen un precio que rondaban los 2500 USD[3] a la fecha de la búsqueda, debiéndose incluir también en algunos casos licencias de software. Por supuesto, en este caso también estaríamos ante un diseño nuevo, corriendo los riesgos inherentes al nuevo producto. El tiempo de familiarización con estos dispositivos y la técnica de programación, así como también el tiempo requerido para el desarrollo propiamente dicho, sería demasiado extenso para permitirnos terminar el proyecto en el plazo estipulado. Teniendo en cuenta también que este proyecto tiene más aspectos que el de *'bridgear'* los datos de la interfaz V.35 a otra interfaz, esta opción quedó también descartada.

3.4.3 Tarjetas PCMCIA con interfaz V.35

En el momento de iniciar el proyecto no se encontró tarjetas de esta clase que garantizaran alcanzar los 2.048Mbps requeridos. Son fuertemente dependientes del hardware y sistema operativo en el que residen. Esta opción quedó por tanto también descartada. Un ejemplo de las tarjetas que se encuentran disponibles en el mercado es del fabricante FarSite[4].

3.4.4 Tarjetas PCI adquisidoras de datos.

Estas tarjetas no proveen ninguna funcionalidad específica acerca de la interfaz V.35 o de algún protocolo de capa de enlace. El precio no resultó ser menor al de las tarjetas especializadas en V.35, por ejemplo la NI PCI-6250[5] de la compañía National Instruments ronda los 1000 USD. Por estos motivos esta opción fue también descartada.

[3] Xilinx - http://www.xilinx.com/
[4] FarSite Comunications - http://www.farsite.com/
[5] National Instruments - http://www.ni.com/

Capitulo 4

ELECCIÓN DE SOFTWARE

4.1 Elección del Sistema Operativo

La elección del Sistema Operativo (SO) en el cual se desarrolló el analizador SnS fue una decisión que derivó de la elección del hardware. Cuando se optó por la tarjeta Sangoma S5141, los fabricantes especificaron que el proyecto sería viable únicamente si se desarrollaba sobre el SO Linux, debido a que las librerías (APIs) que aportaban los fabricantes estaban implementadas en lenguaje de programación C para Linux. El potencial de un Sistema Operativo de código abierto brindó la posibilidad desarrollar el software para el proyecto. Cabe destacar la ventaja del hecho de no haber tenido que pagar licencias de software, algo no menos importante en el desarrollo de un proyecto al plantear los costos y por tanto la viabilidad económica del mismo.

Como ventaja adicional se presentó la oportunidad de manejar un Sistema Operativo actualmente muy utilizado en empresas del medio así como también la oportunidad de incursionar en el mundo de código abierto.

4.2 Elección del Software Analizador

El software Ethereal es un analizador de protocolos probado y disponible en el medio. Es utilizado para efectuar análisis, así como también solucionar distintos tipos problemas que se

presentan en redes de comunicaciones, para desarrollo de software y protocolos y como herramienta didáctica en la educación. Cuenta con todas las características de un analizador de protocolos estándar.

Ethereal está desarrollado bajo licencia de código abierto y se ejecuta sobre la mayoría de los Sistemas Operativos. A continuación se listan algunos de las características más relevantes del software.

- Es mantenido bajo Licencia GNU
- Es capaz de leer datos almacenados en un archivo
- Presenta una interfaz flexible y amigable para el usuario
- Posee diferentes opciones de filtrado
- Soporta formatos de archivo estándar
- Se ejecuta en más de 20 plataformas
- Soporta más de 750 protocolos
- Presenta la capacidad de leer archivos de captura de más de 20 productos
- Posee la gran ventaja de ser un analizador de libre acceso y de constante actualización de su lista de protocolos.

Durante las primeras pruebas de capturas realizadas con el analizador Hewlett Packard propiedad de ANTEL, los datos capturados fueron guardados en formato hexadecimal. En una segunda instancia fueron agregados en forma manual los encabezados del formato PCAP[6], creando un archivo capaz de ser procesado por el software Ethereal. Pudo ser comprobado que si se indicaba el protocolo inicial, en este caso particular CHDLC, el Ethereal se encargaba de los protocolos de capas superiores.

En la página oficial del programa analizador Ethereal se asegura que el software es capaz de analizar diferentes protocolos, entre los cuales se incluyen HDLC, CHDLC, Frame Relay y PPP, los cuales son específicamente los protocolos de interés para el proyecto.

Uno de los puntos claves en la elección del software analizador consistió en confirmar que era posible analizar los datos a partir de un archivo. Por lo tanto, luego de la primer instancia de pruebas realizada en ANTEL mencionada anteriormente, pudo ser confirmado que el software Ethereal satisfacía este requerimiento.

En una segunda etapa, luego de realizadas las primeras capturas con el analizador SnS, surgió la necesidad de desplegar la mayor cantidad posible de información útil para el usuario, por lo tanto se buscó un formato capaz de presentar el sentido de las tramas capturadas. Una vez desplegada la información en pantalla, se tornó necesario que el Ethereal incluyera en la información el sentido de las tramas capturadas, o sea si la trama tenía origen en el DTE o en el DCE.

[6] El formato de archivo PCAP se describe en el ANEXO V.

Se realizaron consultas en la página de Ethereal y fue posible encontrar un formato de archivo que permite desplegar en pantalla la información necesaria para el correcto análisis. Ese formato es el Network Associates Sniffer[7].

4.3 Interfaz de usuario en Windows

El control del analizador SnS se centró en la opción de manejo remoto que presenta el mismo a través del puerto Ethernet, desde una PC portátil, con sistema operativo Windows.

Una vez iniciado el programa, la interfaz gráfica del mismo solicita al usuario que seleccione el protocolo de capa de enlace elegido, seguido de la condición de fin de captura. Si la condición de fin de captura elegida es por tiempo se debe especificar el tiempo que durará la captura. Se selecciona el nombre del archivo con que será guardada la captura y finalmente se debe lanzar la captura.

Una vez cumplida la condición de fin de captura, el usuario debe copiar el archivo con los datos a su computador personal para su posterior análisis con el software Ethereal.

El lenguaje de programación Visual Basic, permitió desarrollar una interfaz gráfica GUI, amigable y de muy sencillo manejo para el usuario, por medio de la cual es posible realizar todas las tareas antes mencionadas. A continuación se puede visualizar la interfaz GUI del analizador SnS.

[7] El formato de archivo Network Associate Sniffer se explica detalladamente en el ANEXO V.

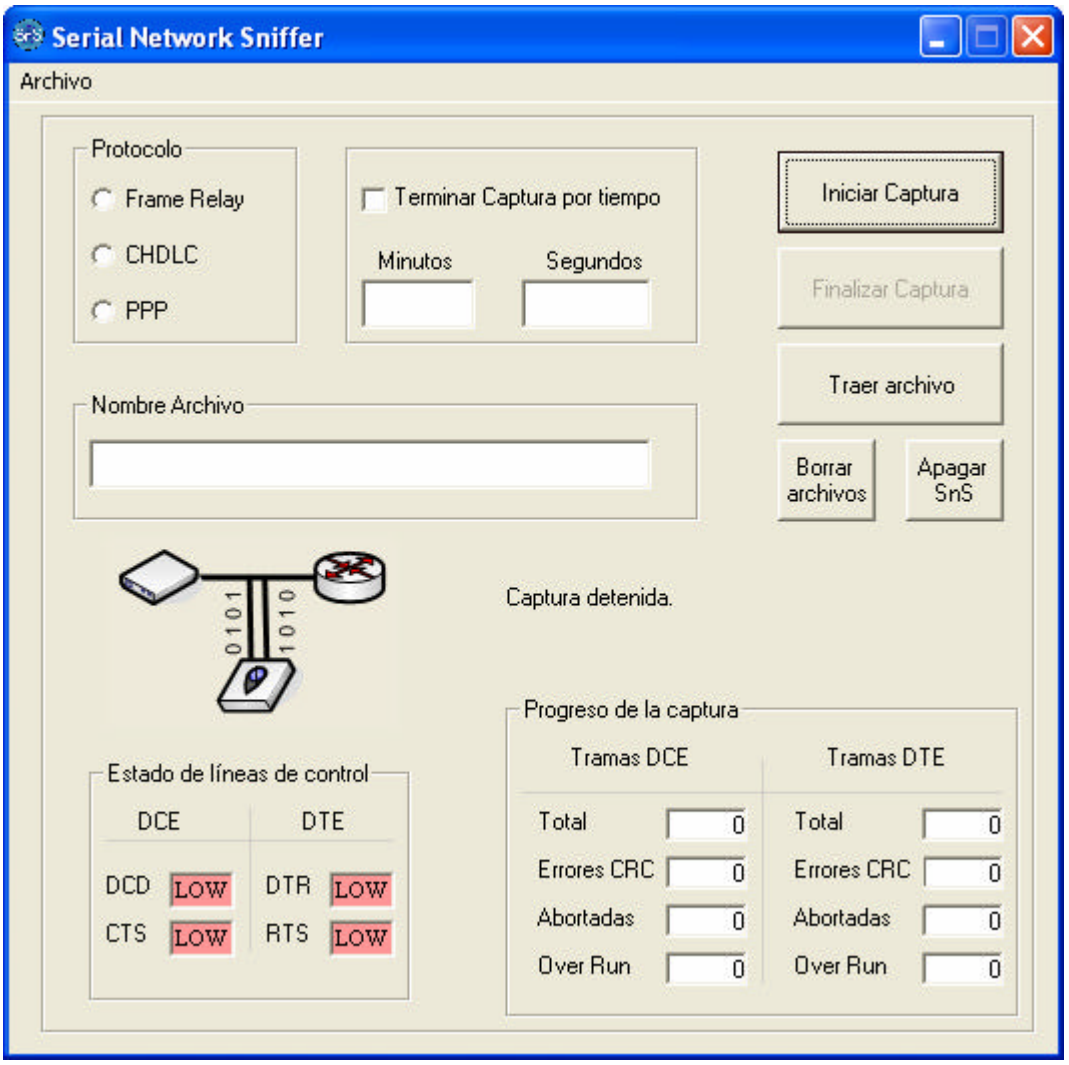


Figura 2: Interfaz GUI del analizador SnS

Esta elección derivó también en la posibilidad de elegir una forma de lanzar comandos remotamente desde Windows a Linux. Esto se logra a través de los programas PUTTY, PLINK y PSCP[8].

4.4 Control de la capacidad del disco

Como se mencionó anteriormente, el analizador SnS almacena los datos capturados en el disco duro. Si una captura iniciada demorara en finalizar el tiempo suficiente como para llenar la

[8] Programas de código abierto - http://www.putty.nl/

capacidad disco, puede resultar en la pérdida de los datos capturados si no es manejada correctamente la señal de disco lleno del sistema operativo.

Para evitar perder la captura, antes de llenar el disco se desarrolló en lenguaje Perl un *script* que da fin a la captura. La elección de desarrollar el *script* en lenguaje Perl fue tomada como una opción didáctica para introducir un lenguaje usado para estos fines de control

Capitulo 5

HARDWARE DEL ANALIZADOR SNS

5.1 Tarjeta adquisidora Sangoma S5141.

5.1.1 Proceso de Adquisición

La adquisición de datos se realiza sobre la base de una tarjeta modelo S5141 del fabricante Sangoma. La mencionada tarjeta realiza la captura en crudo (*raw*) de los datos desde el DTE y DCE e inserta la maraca de tiempo o *time stamp* dentro del ambiente del Kernel de la versión de Linux utilizada. Posteriormente la aplicación de usuario toma los datos con una estructura determinada y los escribe en un archivo.

La tarjeta Sangoma es una interfaz serial que posee dos puertos, los cuales admiten diferentes configuraciones. En el caso del analizador SnS ambos puertos fueron configurados únicamente para recibir datos con el estándar V.35, utilizando el reloj externo proveniente de las señales generadas por el DCE y DTE.

La tarjeta serial S5141 trabaja con el *driver* llamado *wanpipe*[9], sobre el cual se utilizó un *stack* multiprotocolo. Si se seleccionan las APIs multiprotocolo, la tarjeta trabaja con el protocolo *raw* HDLC sobre el cual es posible desarrollar aplicaciones que son capaces de trabajar con distintos protocolos.

La capa *wanpipe* API hace uso de la arquitectura de los *sockets* de Linux, en donde Sangoma desarrolla sus módulos. Estos módulos fueron instalados con el *driver* para manejar los datos

[9] El código fuente se puede encontrar en la página oficial de Sangoma.

capturados y pasarlos a la aplicación de forma segura. A estos *drivers* se le realizaron las modificaciones necesarias que permitieron insertar las marcas de tiempo por software.

Los módulos desarrollados en lenguaje de programación C, a los cuales no se aconseja su cambio, tienen definida una estructura de datos. Cuando la tarjeta solicita una interrupción, devuelve ese formato. La estructura con la cual son pasados los datos a la aplicación de usuario fue modificada con el fin de agregar a la estructura dos variables enteras. Antes de pasar los datos al usuario, el Sistema Operativo setea estas variables, una con segundos y otra con micro segundos, utilizando la estructura *timeval* de la librería *Time* de C. Esta es la forma en que las marcas de tiempo son insertadas por software.

En el hardware de la tarjeta adquisidora Sangoma S5141, dentro de la estructura que devuelve, existe un campo *time stamp* el cual consiste en un contador de 0 a 65535 y cuya precisión es de milisegundos. Este campo no es utilizado en el proyecto del analizador SnS debido a que la precisión de un milisegundo no es suficiente.

Una vez que el Kernel ha manejado los datos, la aplicación de usuario se hace cargo de los mismos lo antes posible para que no exista la posibilidad de que estos sean sobrescritos por datos recientemente capturados.

Los retardos comprometedores se concentran en el cálculo de la *time stamp* insertada por software y cada vez que la aplicación interactúa con el Kernel de Linux para la transmisión de datos.

5.1.2 Descripción Técnica

La tarjeta Sangoma S5141, pose dos puertos seriales independientes, los cuales según el fabricante alcanzan velocidades de hasta 4Mbps el primario y 512kbps el secundario. Es posible también trabajar en modo de solo recepción, siendo este el caso del analizador SnS, en el cual el reloj del puerto secundario trabaja a la misma velocidad que el puerto primario. Sobre este punto los fabricantes aseguran que es posible trabajar a 2.048Mbps en ambos puertos cuando estos son seteados en modo de solo lectura de datos en crudo. La configuración se realizó mediante una aplicación que se corre luego de compilado el *driver*.

Ficha técnica

El fabricante de la tarjeta adjunta las siguientes especificaciones técnicas sobre la misma.

- Puerto primario: puerto serial de 4Mbps el cual soporta los estándares V.35, X.21, RS232.
- Puerto secundario: puerto serial de 512kbps si es configurado para trabajar en forma independiente. Esta velocidad se incrementa hasta 2.048Mbps si trabaja en modo recepción en conjunto con el primario. El puerto secundario soporta los mismos estándares que el puerto primario.
- Consumo: 550 mA a 5V y 60 mA a 12V.

- Compatible con PCI de 32 bits y 64 bits.
- Configuración por software.
- Dimensiones: 144 mm x 99 mm.
- Temperatura de operación: 0 a 45 C.

Figura 3: Tarjeta Sangoma S5141

Módulos y Drivers

El *driver* fue compilado bajo el ambiente Linux[10]. El mismo carga los módulos[11] necesarios para el manejo del hardware. Los datos que son entregados del hardware al Kernel tienen la siguiente estructura.

Error_flag	Time_stamp	Reservado	Datos	CRC

Tabla 1: Estructura datos originales

Error_flag definida como char:

- Bit 0: La trama en la entrada fue desechada.
- Bit 1: La trama en la entrada tiene error de CRC.
- Bit 2: La trama tiene error de overrun.

Time_stamp, 2 bytes:

- Es insertada por el hardware con un contador de 0 a 65535, con precisión de milisegundos.

Reservado: son 13 bytes reservados por el fabricante

[10] Como compilar *drivers* ver ANEXO VIII

[11] En el ANEXO VI se detalla que un módulo en linux

Antes que la aplicación de usuario saque los datos, el módulo agrega dos nuevos campos, el campo de segundos y el de microsegundos, pasando a la aplicación la siguiente estructura:

Error_flag	Time_stamp	sec	usec	Reservado	Data	CRC

Tabla 2: Estructura datos modificada

Las *time stamp* son insertadas utilizando la librería *Time* de C, formato GNU.

La marca de tiempo es insertada en el archivo *sdla_chdlc.c* en la función que maneja la interrupción

```
static void rx_intr (sdla_t* card){
        ……..
        ……..
        do_gettimeofday(&tv);
        api_rx_hdr->sec=tv.tv_sec;
        api_rx_hdr->usec=tv.tv_usec;
        ……..
        ……
}
```

En el archivo *sdla_chdlc.h* se define los campos *sec y usec* para la estructura:

```
typedef struct {
        unsigned char   error_flag      PACKED;
        unsigned short  time_stamp      PACKED;
        unsigned int    sec             PACKED;
        unsigned int    usec            PACKED;
        unsigned char   reserved[5]     PACKED;
} api_rx_hdr_t;
```

El módulo original pasaba al usuario la estructura sin las marcas de tiempo insertadas por software. Para lograr que las mismas fueran insertadas por software se modificó la estructura del modulo. Finalmente se obtuvieron dos estructuras de datos, una con la cual el módulo saca los datos del hardware y otra estructura que es pasada a la aplicación de usuario. De esta forma se logró pasar al usuario la estructura con las marcas de tiempo ya insertadas por software.

En el módulo original, escrito en lenguaje C por los desarrolladores de Sangoma, fue redefinida la estructura de salida de datos hacia el usuario. Antes de cargar los datos a la estructura, se invoca la función *time* para calcular el tiempo y cargar las variables *sec* y *usec*, dejando la estructura con los datos prontos para ser leída por la aplicación de usuario.

No queremos dejar de mencionar la gran cooperación recibida por parte de los desarrolladores de Sangoma cuando fue necesario modificar el módulo para adaptarlo a los requerimientos del proyecto SnS.

La modificación realizada al módulo implicó que las marcas de tiempo sean insertadas al final de la trama capturada, además de agregar un error que depende fuertemente de las tareas que esté realizando el SO.

Una de las críticas a Ethereal consiste en que cuando se realizan capturas utilizando cualquier software *sniffer*, se plantea como defecto que las *time stamp* son insertadas por software, de esta forma no teniendo la misma precisión que al ser insertadas por hardware y por tanto esto puede ocasionar diferencias.

Configuración de la tarjeta Sangoma S5141

La configuración de la tarjeta Sangoma S5141 puede ser realizada mediante una aplicación diseñada por el fabricante para este fin. Otra opción es la configuración por medio de un archivo el cual es creado en forma manual. Se recomienda utilizar la aplicación.

La aplicación GUI permite crear los archivos de configuración, los cuales son creados en el directorio **/etc/wanpipe/** y bajo el nombre de **wanpipe#.conf**

Deberá ser ejecutada la herramienta de configuración **wancfg_legacy**[12], donde una interfaz GUI guiará al usuario para crear el archivo de configuración. Para el caso del SnS se proveerán los archivos de configuración ya definidos.

Dentro del archivo de configuración deben ser seleccionados, entre otros, los siguientes campos:

- Protocolo.
- Hardware, modelo de la tarjeta.
- Puerto primario o secundario.
- Condición de solo recepción.
- Reloj (*Clock*) externo.
- MTU (Maximum Transfer Unit).
- Interfaz de red: Interfaz 1 ó 2.
- Nombre de la Interfaz.
- Modo de operación, lectura en crudo.
- Parámetros del protocolo CHDLC.
- *Script* de inicio y fin.

Existen además otras opciones en la configuración, las cuales se sugiere dejar con la configuración por defecto. Como ejemplo se transcribe el archivo de configuración realizado, en el cual se aclaran todos los parámetros seleccionados.

[12] La herramienta de configuración es para todos los modelos, existe otra wancfg que es para las nuevas tarjetas VoIP.

NOTA: Ver en Anexo III un archivo de configuración del puerto primario con los parámetros seleccionados

5.2 Cable T de interconexión

5.2.1 Conexión T

Como mencionamos anteriormente, el hardware para la adquisición de datos es la tarjeta Sangoma S5141. La tarjeta posee dos puertos seriales donde cada uno de ellos puede trabajar de forma independiente o como en el caso del analizador SnS pueden ser configurados en modo de solo recepción.

Para poder capturar los datos a transmitirse entre el DCE y el DTE a través del estándar V35 se realizó una conexión en T. El *pinout* del estándar V.35 se encuentra bien definido, de ahí fue necesario cablear cada línea a la tarjeta adquisidora de datos.

Como se mencionó en varias oportunidades, para la adquisición de datos son utilizados ambos puertos, en el puerto primario son capturados los datos desde el DTE y desde el puerto secundario son capturados los datos provenientes del DCE. Para que esto sea posible es necesario intercalar el analizador SnS en la línea de transmisión de los equipos DTE y DCE.

Un esquema de la conexión se presenta en la siguiente figura.

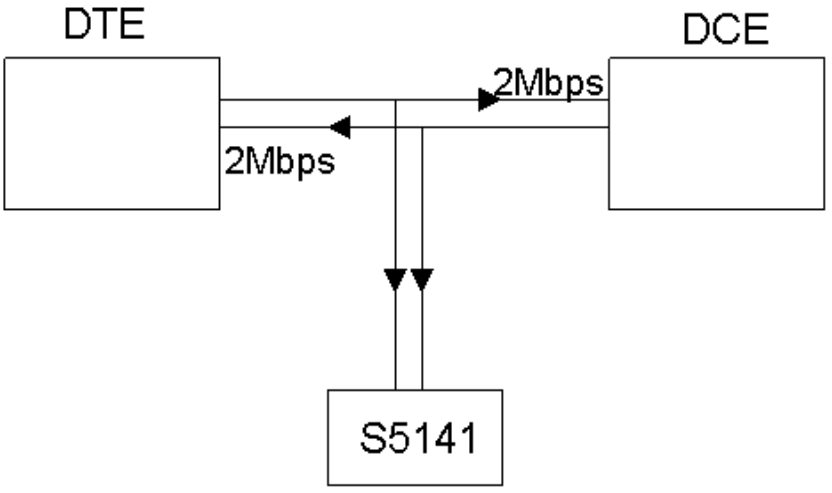


Figura 4 : Esquema de la conexión con el cable T.

La siguiente figura detalla la conexión de los equipos DTE y DCE al analizador SnS. Las señales se diferencian en provenientes desde el DTE, que son conectadas al puerto 1 de la tarjeta Sangoma y las señales que se originan en el DCE, las cuales son conectadas al puerto 2 de la misma.

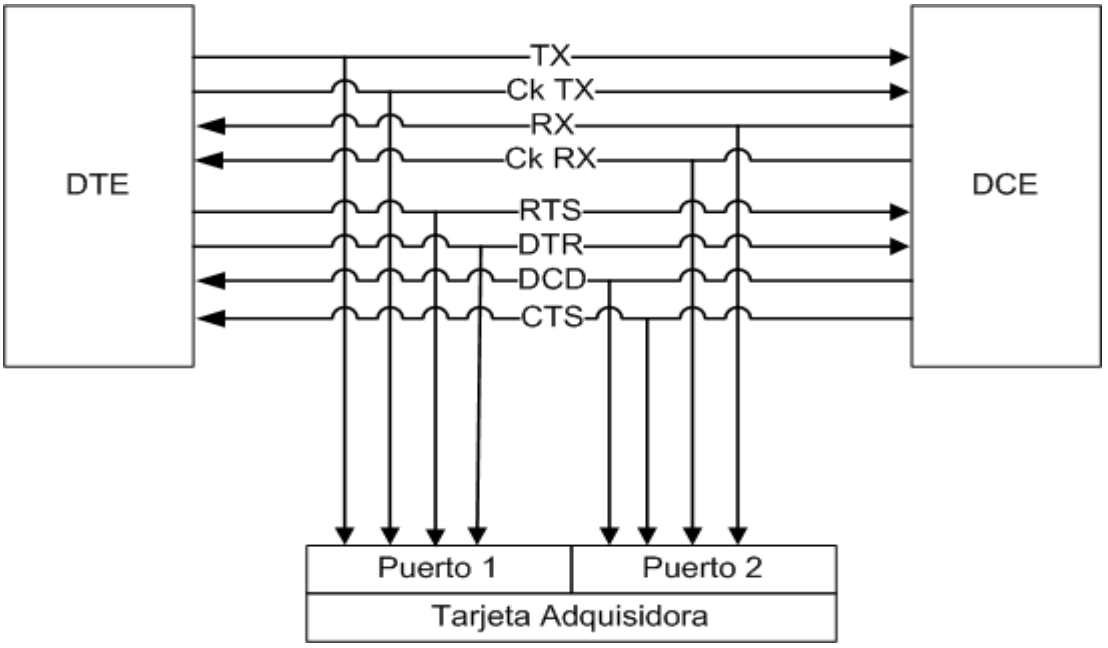


Figura 5: Resumen conexión T

Los *pines* CTS y DCE deben ser cableados en ambos puertos para que se encuentren aptos para la recepción de datos.

El cableado se realiza soldando *pin* a *pin*, dada la estabilidad de los voltajes no es necesario realizar ninguna adaptación electrónica.

El cable tipo T tiene físicamente tres conectores, dos conectores M34, uno hembra y otro macho para interponerse en la línea y un conector DB37 que va a la tarjeta adquisidora.

NOTA: La Norma V.35 es detallada en el ANEXO I.

5.3 Tarjeta madre del analizador

La tarjeta madre VIA EPIA Mini-ITX es una de las *mothers* más pequeñas disponibles en el mercado. Su plataforma x86 optimiza su potencial sin sacrificar la flexibilidad del diseño. El

diseño de la tarjeta se basa en un procesador VIA Eden ESP sin ventilador, como su nombre lo indica (VIA EPIA 5000 Fanless), debido a que sus bajos requerimientos de consumo no lo hacen necesario. La tarjeta trae incorporado un procesador multimedia VIA C3. La placa VIA EPIA Mini-ITX es la plataforma ideal para una ilimitada variedad de proyectos basado en una plataforma PC.

Figura 6: Tarjeta VIA EPIA 5000 Fanless

Midiendo apenas 170mm x 170mm, la tarjeta madre VÍA de Mini-ITX de EPIA es 30% más pequeña que la más pequeña de las plataformas ATX, mientras que mantiene las características del chasis de ATX. También proporciona el ambiente de proceso más pequeño, incluyendo una configuración sin ventilador en el procesador. A su vez es posible trabajar a mayor frecuencia, pero en ese caso utilizando ventilación forzada. Posee gráficos integrados VIA con características de aceleración 2D/3D, acompañada con audio. Incluye una tarjeta de red Ethernet 10/100, salida TV, una ranura adicional del PCI, y un sistema completo de características de I/O las cuales proporcionan opciones amplias de la conectividad y de extensión.

5.3.1 Especificaciones técnicas

Procesador	***-VIA Eden ESP 5000 Processor - 100/133MHz Front Side Bus - Bajo consumo - Sin ventilador***
Chipset	- VIA Apollo PLE133 - VT8601A North Bridge - Características gráficas 4X integradas
Memoria	Dos sockets de 168-pin para memoria DIMM
Gráficos	- AGP4X con acelerador gráfico 2D/3D
Slot de expansión	- 1 x PCI Slot
IDE	ATA 100/66
Fuente de potencia	- ATX Power Supply Compliant
Puertos de E/S	- 3 Audio Jacks - Line out, Line-in, Mic-in - 4 puertos USB - 1 puerto paralelo EPP/ECP - 1 puerto serial - 2 puertos PS/2 para teclado y ratón - 1 puerto RJ45 LAN - 1 slot PCI, soporta dos tarjetas mediante extensión
Factor	- Mini-ITX - 17cm x 17cm - Características chasis ATX

Tabla 3: VIA EPIA, Especificaciones técnicas

5.4 Chasis y fuente

En la elección del *chasis* se tuvo en cuenta la necesidad de que el proyecto fuera reproducible, por lo que se optó por uno disponible en el mercado aunque esto implicase elevar el costo del

equipo SnS. Finalmente se optó por importar un paquete que incluía el *chasis* con fuente y ventilador y la tarjeta madre.

El paquete mencionado, incluyendo el *chasis* y tarjeta madre, fue adquirido en la empresa *CauseOutlet*[13] e importado por una empresa del medio.

La elección consistió en adquirir el *chasis* que mejor se ajustara a los requerimientos luego de seleccionado el hardware adquisidor.

A continuación se listan las características más destacables del *chasis* seleccionado.

- Factor mini-itx para la tarjeta madre.
- Salida para una tarjeta PCI.
- Espacio para un disco duro.
- Fuente acorde al consumo.

Para dimensionar la potencia de la fuente se estimó el consumo total. Los cálculos en cuanto a consumos de potencia se detallan a continuación.

- 1 (una) tarjeta madre EPIA 5000 (Fanless) Mini-ITX: 10,45 Watts[14]
- 128 MB memoria DIM: 7 Watts
- 1 (un) disco duro IDE: 20 Watts
- 1 (una) tarjeta Sangoma S5141: 10 Watts

La potencia total consumida, según los cálculos antes detallados, es de 47,45 Watts.

Las fuentes disponibles para el gabinete elegido eran de 60 Watts, 90 Watts y 120 Watts. Se optó por la de 60 Watts ya que según los cálculos previos, el consumo de potencia no superaría los 60 Watts.

Podríamos haber optado por la fuente de 90 Watts, la cual era la aconsejada si se agregaba además una unidad óptica, CD-ROM o similar, pero debido a que el SnS no cuenta con ninguno de estos módulos, esta elección no tenía sentido.

A continuación se presentan imágenes del *chasis* TRAVLA elegido y la fuente de 60 Watts para el mismo.

[13] Cause Outlet – Shopping on-line www.caseoutlet.com
[14] Calculadora para potencia consumida:
http://resources.mini-box.com/online/powersimulator/powersimulator.html

Figura 7: Vista del chasis

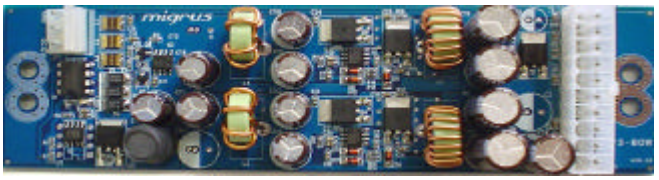


Figura 8: Fuente 60 W

Otra consideración fue la disipación de calor de los componentes. Se pudo tomar en cuenta al elegir el *chasis* que la tarjeta adquisidora no quedara por encima de la tarjeta madre, para evitar que la temperatura del procesador perjudicara el funcionamiento de la tarjeta adquisidora. En el correr del proyecto se pudo verificar que el problema de disipación luego de armado el SnS lo provocó el disco duro, no se pensó que este componente disipara tanto calor.

Una solución planteada, para implementar como mejoras del proyecto en un futuro, consiste en sustituir el disco duro (3.5") por un disco duro de *laptop* (2,5"). Esta opción presenta dos grandes ventajas. Una de ellas consiste en que un disco de *laptop* disipa menos calor, la otra ventaja es que los discos de *laptop* están diseñados para ofrecer mayor durabilidad ante posibles golpes y movimientos continuos, dado que el *laptop* es un equipo portátil. Como también el equipo SnS es portátil creemos que esta sería una buena opción.

Capitulo 6

SOFTWARE DEL ANALIZADOR SNS

6.1 Introducción

El programa SnS realiza la captura de los datos provenientes de las dos interfaces V.35 del analizador. Ambas están conectadas de forma de escuchar los datos que transmiten los equipos a analizar, DCE y DTE. La aplicación toma los datos a través de *sockets* y es responsable de sacarlos del hardware con la velocidad necesaria para que los *sockets* no se llenen y por tanto se descarten paquetes. Con los datos que obtiene de cada *socket* son creados dos archivos. Luego que finaliza la captura de datos se crea un tercer archivo, el cual luego será analizado con el software Ethereal. El mencionado archivo contiene los datos capturados ordenados cronológicamente.

El software se puede dividir en dos capas, capa de aplicación y capa de Kernel del sistema operativo. Esta última es quien trabaja sobre el hardware extrayendo los datos hacia la aplicación e insertando las marcas de tiempo luego de atender la interrupción que el hardware genera al recibir una nueva trama.

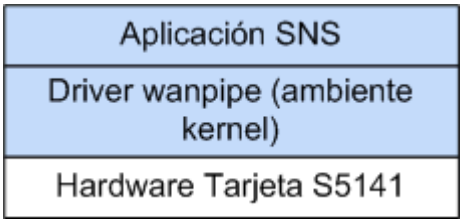


Figura 9: Capas del software

En las siguientes secciones se describe el diseño del software de la aplicación. Primero se realiza una descripción general y luego se profundiza en los procesos de mayor relevancia que esta utiliza.

NOTA: El código de la aplicación SnS se encuentra escrito en lenguaje C. Además se utilizó Visual Basic para la aplicación de Windows y leguaje Perl para realizar los *scripts.*

6.2 Software SnS

La aplicación SnS es invocada al correr el comando *sns* seguido de diferentes parámetros. La aplicación necesita los siguientes datos iniciales:

- Protocolo de capa de enlace:
  - CHDLC
  - Frame Relay
  - PPP
- Condición de fin de captura:
  - Manual
  - Por tiempo
  - Cantidad Paquetes
- Nombre del archivo final.

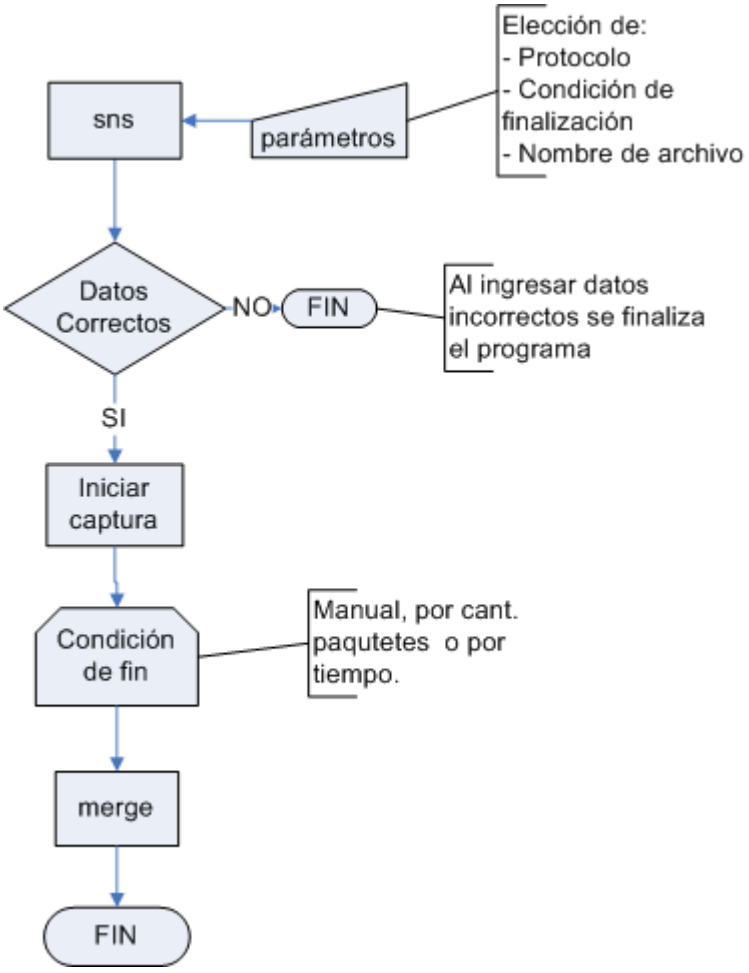


Figura 10: Esquema de la aplicación sns

Existen dos formas de pasar los valores iniciales. Una es de forma interactiva con el usuario desde consola. La otra consiste en pasarlos por parámetros cuando es llamada la aplicación.

Una tercera opción consiste en ejecutar la aplicación en forma remota desde Windows. Esta opción se detalla en el final del capítulo.

Si se corre en consola el comando *sns*, son desplegados en pantalla los parámetros que pueden ser pasados a la aplicación. Si el comando escrito incluye *sns –v* se ingresa al modo interactivo donde son solicitados al usuario los datos iniciales. En el modo interactivo no son tomados en cuenta más parámetros que los solicitados.

```
sns
        opciones:
                -min  <minutos>       minutos a capturar
                -seg  <segundos>      segundos a capturar
                -v                    modo interactivo
                -paq                  cantidad paquetes
                -prot   <1|2>         protocolo de capa enlace
                                              1:CHDLC o PPP
                                              2:Frame Relay

                -archivo  <nombre>    nombre del archivo de captura
```

Figura 11 : Salida en pantalla sin parámetros

Otra opción consiste en ingresar los datos por parámetros al iniciar el programa, salteando así la interacción con el usuario.

Una vez que los datos son confirmados por el usuario se inicia la captura en forma inmediata.

Al cumplirse la condición de fin, se cierran los dos archivos intermedios que contienen los datos capturados del DTE y del DCE respectivamente. Se lanza un nuevo proceso (MERGE) que crea el archivo final siguiendo el formato *Network Associates Sniffer,* el cual puede ser interpretado por el software analizador Ethereal. Este archivo contiene las tramas capturadas en los dos archivos anteriormente mencionados, entrelazadas según sus marcas de tiempo.

6.3 *Software de Captura*

La aplicación se divide en tres ramas principales o procesos independientes utilizando el comando *fork()*. Se tiene entonces, además del proceso principal, dos procesos secundarios que corren ocultos realizando la captura uno por puerto.

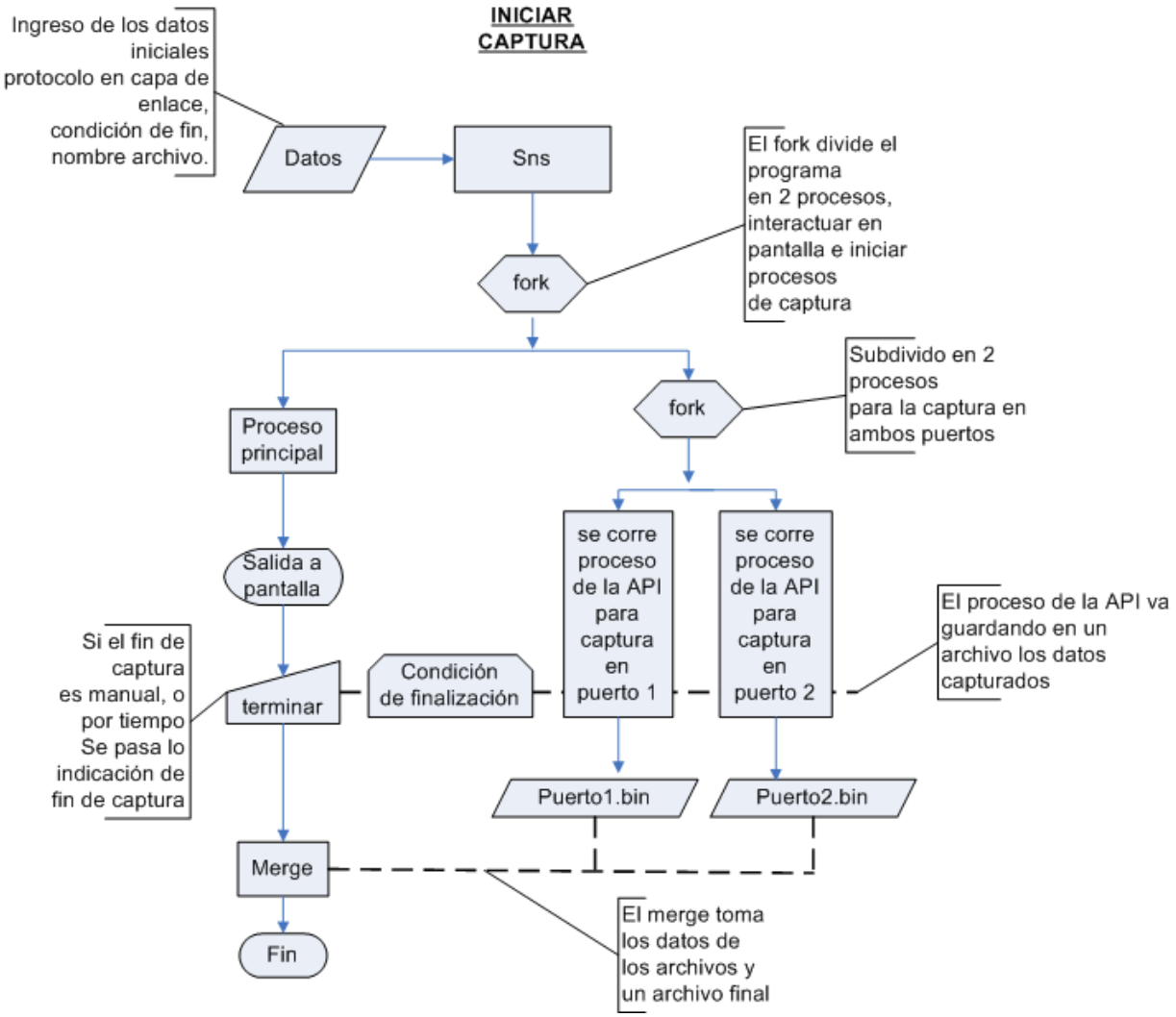


Figura 12 : Iniciar Captura

La rama principal interactúa con el usuario solicitando los datos iniciales, aunque estos pueden ser ingresados por parámetros al ejecutar el programa. Cuando el usuario inicia la captura, el proceso principal crea dos procesos independientes en paralelo, que manejan cada uno un puerto. A medida que los datos del puerto son capturados, son guardados en un archivo con formato *pcap*. Cuando los procesos finalizan, el archivo es cerrado.

En el transcurso de la captura se trabaja con dos formatos de archivo, formato *pcap* y formato *Network Associates Sniffer*. El formato *pcap* es utilizado para guardar los archivos intermedios durante el proceso de captura, debido a la simpleza de los encabezados de los paquetes a escribir por cada trama capturada. El formato *Network Associates Sniffer* es utilizado para el formato del archivo final. Esta forma de trabajo incrementa la performance del código de escritura de los archivos intermedios, dado que el formato *Network Associates Sniffer* tiene una estructura de mayor tiempo de procesamiento y no es necesario recargar el código mientras se capturan las tramas desde la línea.

El proceso principal espera por la condición de fin de captura. Si el usuario decide no esperar el tiempo necesario para que se cumpla la condición de fin establecida inicialmente, se brinda la opción de finalizar la captura en forma manual. En ambos caso, el proceso principal entenderá que se cumplió la condición de fin de captura y terminará los procesos de captura de cada puerto enviando la señal *TERM*. Aguardará un instante para dar tiempo a que se cierren los archivos de captura de cada puerto y comenzará el merge de los datos. El resultado final es un archivo con formato basado en la estructura de archivo de Network Associates Sniffer (DOS-Based). Se seleccionó este formato debido a que permite agregar el campo de sentido de tramas, no siendo esto posible con el formato *pcap*.

Cada proceso de captura, es responsable de obtener los datos desde el hardware a través de *sockets* provistos por las APIs de Sangoma y escribirlos en los archivos *puerto1.bin* y *puerto2.bin* respectivamente. Los archivos son escritos de a tramas, esto asegura que cuando se mata el proceso no quede una trama por la mitad.

La condición de fin de captura ofrece tres opciones. Por tiempo, en donde el usuario del analizador le indica por cuanto tiempo va a capturar paquetes. En forma manual el usuario indica el inicio y el fin. Por cantidad de paquetes, el usuario puede indicar cuantos paquetes desea capturar. Se deja el código abierto para futuras implementaciones de inicio y fin de captura.

La forma en que el programa principal controla las condiciones de fin depende de si el mismo es ejecutado en modo interactivo o no.

6.3.1 Interactivo

Luego de iniciada la captura el programa principal es detenido mediante un *timer* si la condición de fin es por tiempo. Si la condición elegida es por cantidad de paquetes, la captura finalizará automáticamente al capturar la cantidad de paquetes ingresados. Finalmente la opción manual que puede ejecutarse siempre con la combinación de teclas *Control+C*.

Se realiza en este modo, cada 4 segundos, una salida a pantalla con la cantidad de paquetes capturados desde que se inicializó la captura. Si la condición de fin es por tiempo, se despliega también el tiempo restante.

6.3.2 No interactivo

Cuando se ejecuta el programa en modo no interactivo, saltea el código para modo interactivo para pasar a un bucle que se ejecuta una vez por segundo. Dentro del bucle se realizan consultas a un archivo. Si en el archivo se encuentra escrita la letra *"n"* se continúa con la captura. Si por el contrario se lee la letra *"s"*, se da como cumplida la condición de fin y se continúa con el código.

El archivo es siempre iniciado escribiendo la letra n. Si la captura es por tiempo, dentro del bucle que se ejecuta cada 1 segundo, se chequea si se cumplió con el tiempo seteado, si transcurrió un tiempo mayor al solicitado por el usuario, se escribe en el archivo una *"s"* y se finaliza la captura. Si la condición de fin es manual o queremos interrumpir la captura se corre

el comando *stopsns* desde consola. Este comando escribe una *"s"* en el archivo finalizando la captura.

Cumplida la condición, la rama principal identifica los procesos y envía la señal *TERM*, matando los procesos de captura. Cada proceso es capaz de manejar esta señal independientemente, de tal forma que antes de terminar el proceso cierra el archivo que estaba creando en el directorio temporal **/home/sns/analizador/tmp**. El formato de estos archivos es *pcap*, de modo que puedan ser abiertos con Ethereal. El archivo *puerto1*.bin corresponde a los datos provenientes desde el DTE y el archivo *puerto2.bin* corresponde a los datos provenientes desde el DCE.

NOTA: Se determina mediante el pinout de la T que el puerto secundario del hardware adquiera los paquetes provenientes del DCE y puerto primario desde el DTE.

6.3.3 Proceso de Captura

Cuando se ejecuta la captura, la rama principal lanza dos procesos para la captura de datos en cada puerto. Esos procesos permanecen trabajando independientemente de los demás. Hacen uso de librerías API aportadas por los fabricantes de Sangoma y se basa en un programa de ejemplo de transmisión y recepción escrito en código C.

Se optó por realizar procesos para la captura debido a la necesidad de minimizar el tiempo de procesamiento. Se intentó trabajar con hilos POSIX (IEEE) para la captura desde ambos puertos, pero el tiempo de atención entre un hilo y otro originaba pérdidas de paquetes. Esto llevó finalmente a trabajar con procesos. Al trabajar con esta estructura de programación pudimos verificar que ya no se perdían paquetes.

Cada proceso permanece en *loop* esperando a leer una trama. Cuando llega una trama, el programa la lee, guarda los datos en un archivo y vuelve a esperar la siguiente trama.

El proceso **chdlc_api** recibe varios parámetros. Por ejemplo, es obligatorio indicar cuál configuración del *driver* se va a utilizar, siendo opcional indicar la cantidad de paquetes a capturar. Entre otros parámetros se indica el protocolo de capa de enlace, el cual únicamente es utilizado en la función de encabezado de paquete *pcap*, ya que no influye sobre la captura pues la lectura de datos de la tarjeta se realiza en forma cruda.

Este proceso puede correr independientemente ejecutándose desde línea de comando. Una vez ubicados en el directorio donde se encuentra el ejecutable, el procedimiento es el siguiente:

./chdlc_api -i wp1chdlc -c wanpipe1 -r -rxfile puerto1.bin -protocolo 1

A continuación se describen los parámetros involucrados:

-i: indica la configuración de la interfaz.
-c: indica la configuración de la tarjeta, se encuentra dentro del archivo **wanpipe1.conf**
-r: indica modo de solo de lectura.
-rxfile: le indicamos que es el archivo final de la captura.
-**protocolo:** para los encabezados *pcap* que agrega a cada paquete capturado. Le indicamos el protocolo.

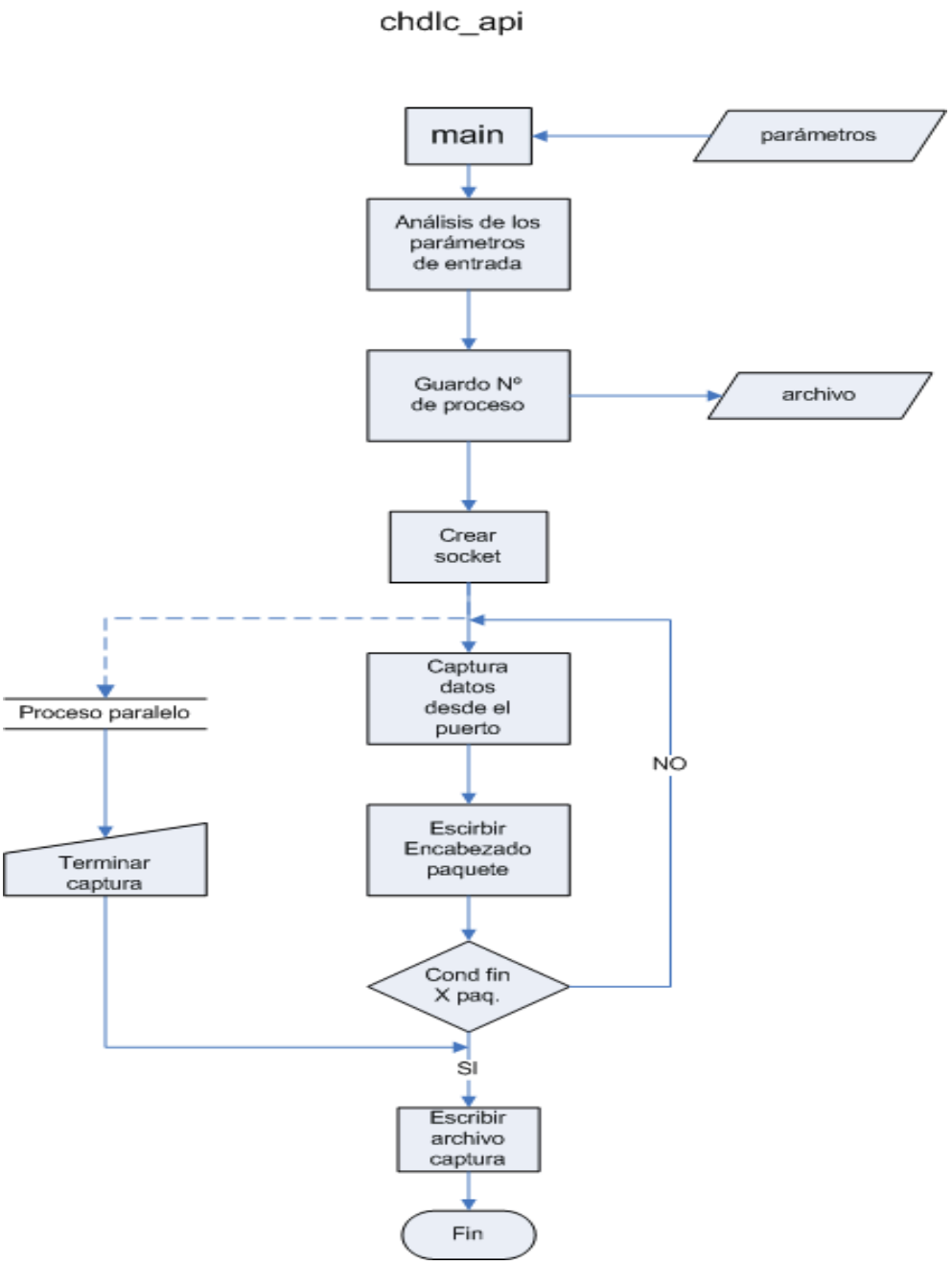


Figura 13 : CHDLC API

Los parámetros son validados, en caso de errores se sale del programa. El paso siguiente es guardar en un archivo el número de proceso. Este número de proceso es utilizado por el proceso de la rama principal para terminar la captura, enviando una señal *TERM*

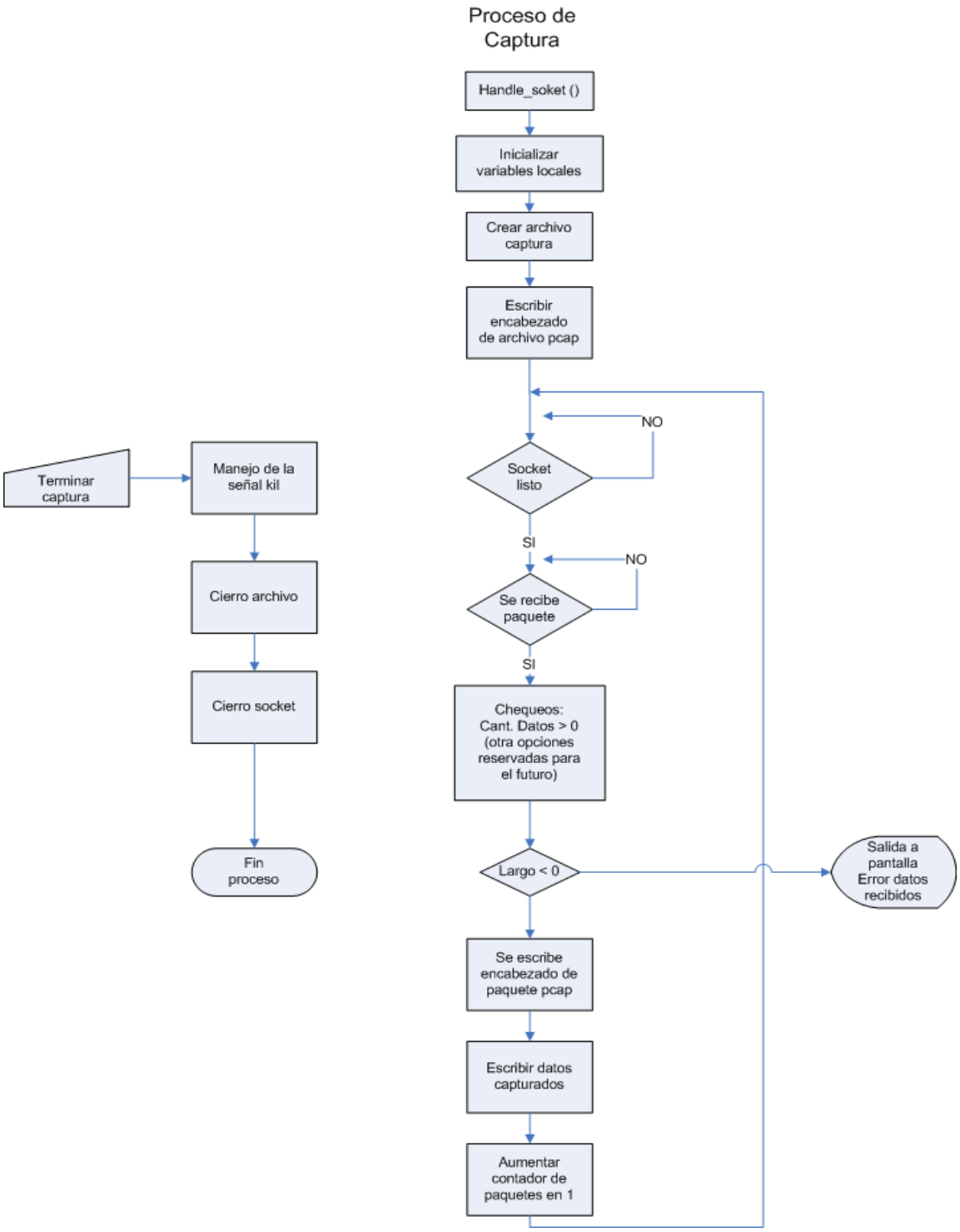


Figura 14 : Proceso de Captura

La librería API crea un *socket* para poder identificar y manejar el puerto. Se crea el archivo con el nombre pasado en el parámetro **rxfile** y posteriormente se escribe el encabezado de archivo *pcap*.

Si la creación del *socket* finaliza con éxito se pasa a un bucle infinito que comienza a leer del puerto. Lee los datos de una trama desde el *socket*, escribe el encabezado de paquete *pcap* y luego la trama capturada. Se incrementa el contador de paquetes capturados en una unidad. Si tenemos la condición de fin por cantidad de paquetes se chequea, si se cumple se termina el bucle, cerramos el archivo y termina el proceso de captura. En el caso de que no exista condición de fin de captura por cantidad de paquetes la aplicación tiene implementado una función para manejar la recepción de la señal *TERM* que es enviada al proceso.

En el proceso de captura, cada tres segundos, se va guardando a un archivo la cantidad de tramas capturadas, errores de CRC, tramas abortadas y *overrun*. El archivo es guardado en el directorio **/home/sns/analizador/tmp** con el nombre ***errores_dce.dat*** para los datos del proceso que captura desde el DCE, y ***errores_dte.dat*** para el DTE.

Para evitar realizar otra consulta al reloj del sistema el período de tres segundos es chequeado con las marcas de tiempo de las tramas. Si la marca de tiempo del paquete es tres segundos mayor a la de la trama anterior se guarda los contadores de cantidad de tramas capturadas, errores de CRC y *overrun* al archivo correspondiente.

Cuando la condición de fin de captura es manual o por tiempo, para terminar la captura la rama principal lee el número de proceso del archivo y mediante el comando ***kill*** **nºproceso** se envía la señal *TERM* para que termine el proceso. Antes de terminar el proceso se deberá cerrar el archivo y el *socket* correspondiente.

6.3.4 Merge

Cada proceso, cuando recibe su señal *TERM*, maneja la mencionada señal y cierra el archivo. En la rama principal invoca a la función **pcaps2fnetas**, la cual toma los dos archivos **puerto1.bin** y **puerto2.bin** y crea el archivo final con el nombre indicado por el usuario, en el directorio **/home/sns/analizador/capturas**.

El archivo final es guardado con formato *Network Associates Sniffer (DOS-Based)*. Este formato permite agregar una bandera en el encabezado para identificar el sentido de las tramas. Cuando el archivo es analizado por el software Ethereal, este se fija en la bandera y según el valor le da el sentido **DTE->DCE o DCE->DTE.**

Para realizar el *merge* se crea un archivo con el nombre especificado por el usuario y luego se escribe el encabezado de archivo.

Para ingresar los datos se van chequeando los archivos de cada puerto, se leen uno a uno los paquetes y se escriben en el archivo final en orden cronológico. Cuando no hay más paquetes que leer se escribe el final de archivo.

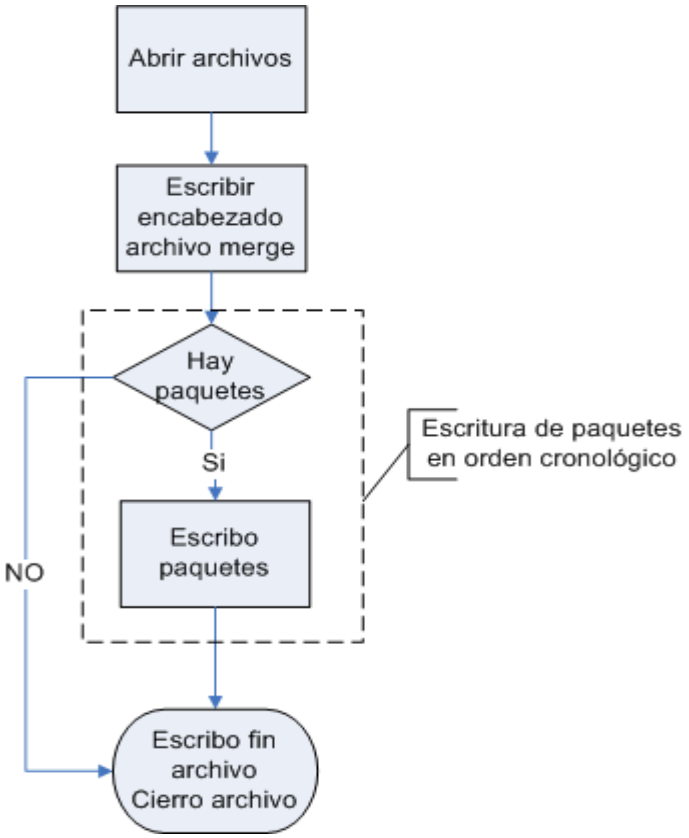


Figura 15 : *merge*

En general los formatos de archivos que entiende el Ethereal, comienzan por un encabezado de archivo, luego un encabezado por paquete y puede terminar con fin de archivo.

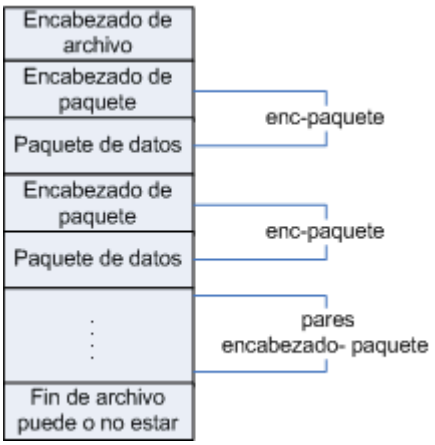


Figura 16 : Encabezado Gral. *pcap*

NOTA: Los formatos de archivo se detallan en el Anexo V.

6.3.5 Descripción del Merge

Esta parte de la aplicación se corre finalizada la captura. Para llevar a cabo esta tarea no se tienen exigencias de tiempo de procesamiento como era el caso de la captura.

El primer paso dentro del *merge* es escribir el encabezado de archivo, para lo cual es necesario leer el primer paquete capturado. De esta forma se determina la fecha de inicio de la captura.

Se escribe el primer paquete y luego se van recorriendo ambos archivos para ir extrayendo el siguiente paquete según tenga menor marca de tiempo y luego escribirlo en el archivo.

La figura siguiente muestra el proceso de lectura de los archivos y como éstos son escritos dentro del archivo final.

El proceso de escribir paquetes en el archivo final, se realiza en varias etapas. Se deberá recordar que los archivos de origen se encuentran con formato pcap. Para poder analizar cual es el siguiente paquete, se debe leer el encabezado de paquete. Dentro de él se encuentran los datos de marca de tiempo y largo de los datos. Con el campo marca de tiempo se determina el siguiente paquete a escribir. Una vez determinado el paquete, se procede a cambiar de formato, se cambia el formato de fecha. Se determina el sentido, si el paquete es del puerto primario se setea con sentido DTE a DCE y si pertenece al puerto secundario se setea con sentido DCE a DTE. Luego se agrega a que tipo de formato de capa de enlace corresponde según los datos

suministrados por el usuario. Si el usuario ingresa mal los datos del protocolo de capa de enlace los encabezados de paquetes no corresponderán con los datos guardados. Luego de guardar los parámetros[15] del encabezado de paquete, se guardan los datos de la captura. A continuación del bucle se vuelve a leer el encabezado del siguiente paquete del mismo archivo, para comparar su marca de tiempo con la de los restantes paquetes.

Una vez guardados todos los paquetes de ambos archivos, **puerto1.bin** y **puerto2.bin**, se guarda el registro de fin de archivo.

[15] Por los campos ingresados ver el Anexo V, encabezados de paquetes.

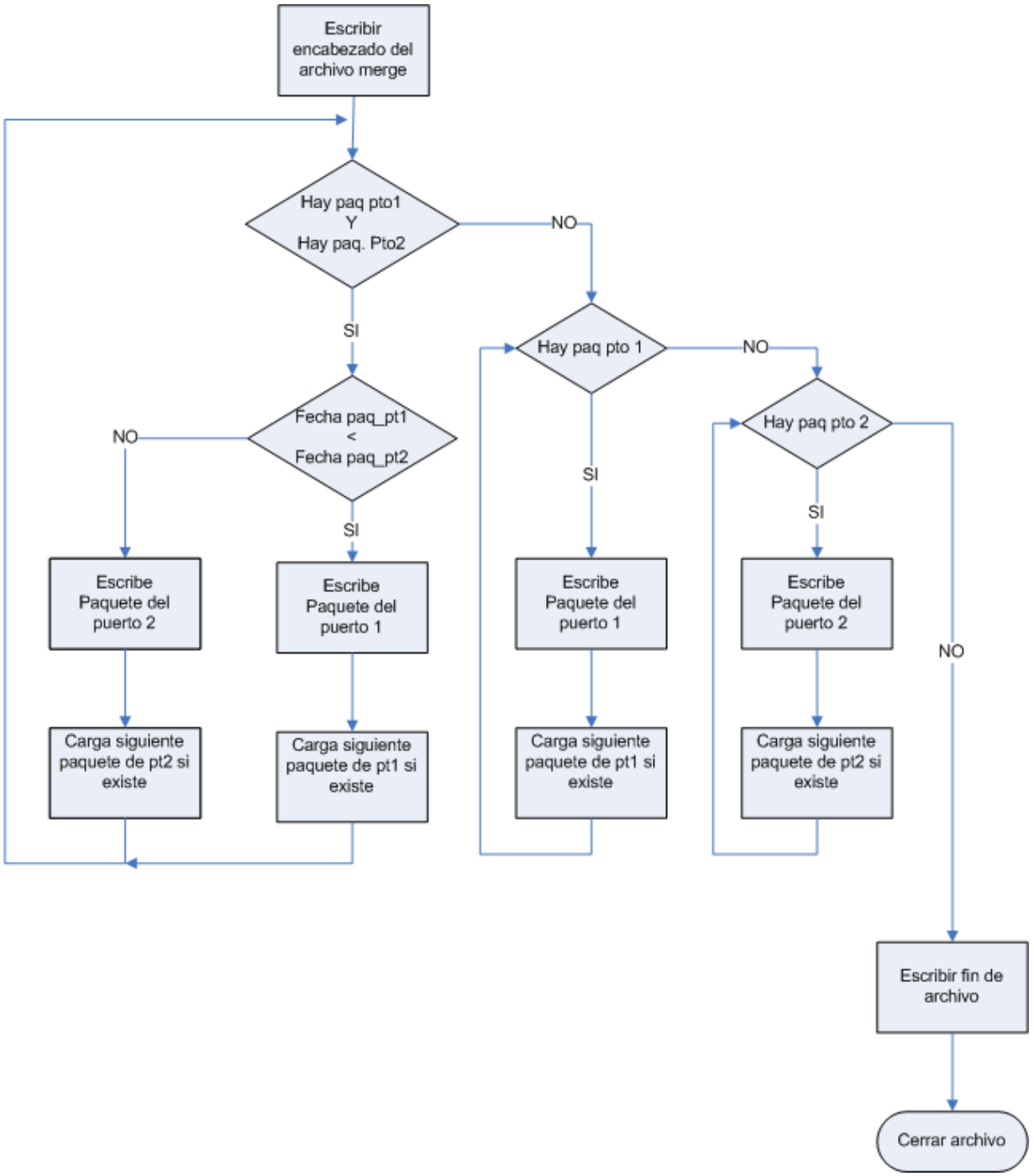


Figura 17 : Detalle del *merge*

6.4 Líneas de Control

Para monitorear las líneas de control se utilizó una librería de la tarjeta Sangoma. Corriendo el comando ***wanpipemon_legacy -i <nombre_interfaz> –c xm*** se despliega en pantalla el estado de las líneas de control. En nuestro caso cambiamos las salidas para que muestre según la interfaz (puerto 1 o puerto 2 de la tarjeta) el estado de las líneas. Para el puerto 1 son las líneas de control DTR y RTS, para el puerto 2 desplegamos DCD y CTS.

También se creó un *script* el cual hace que este comando se ejecute en dos oportunidades, una para cada interfaz. El comando es ***statuslctrl*** y debe ejecutarse con el parámetro ***–v*** para que realice la salida a pantalla. Si se ejecuta sin el parámetro ***–v***, redirecciona la salida al archivo ***status.dat*** en el directorio **/home/sns/analizador/tmp.**

6.5 Interacción con Windows

La ejecución del programa principal SnS por parámetros brinda la posibilidad de que el mismo sea invocado desde el sistema operativo Windows. Esto representa una gran ventaja debido a que Windows continúa siendo el sistema operativo más utilizado. Para pasar los comandos de Windows se utiliza el protocolo *ssh* y los programas PUTTY, PLINK y PSCP.

En la página oficial de PUTTY se encuentran disponibles para bajar en forma libre todos los comandos necesarios.

Para facilitar al usuario la utilización de estos comandos se diseñó una interfaz gráfica en lenguaje de programación Visual Basic. La elección del lenguaje Visual Basic para desarrollar la interfaz se basó en su reconocida calidad de programa amigable para desarrollar aplicaciones graficas para Windows.

La interfaz grafica desarrollada permite al usuario seleccionar el protocolo en la capa de enlace. Si la opción seleccionada de fin de captura es por tiempo, se ingresa el mismo y se selecciona finalmente el nombre del archivo. Estos valores seleccionados en la interfaz gráfica inicializan variables que luego son pasadas como parámetros cuando se ejecuta el comando *sns*.

Los botones de la interfaz ejecutan remotamente los comandos en el analizador SnS. El botón para iniciar captura ejecuta el comando *sns* [parámetro1.....parámetro n]. Estos parámetros, tal

como se explicó anteriormente, son seteados al seleccionar cada una de las opciones en la ventana de la interfaz gráfica.

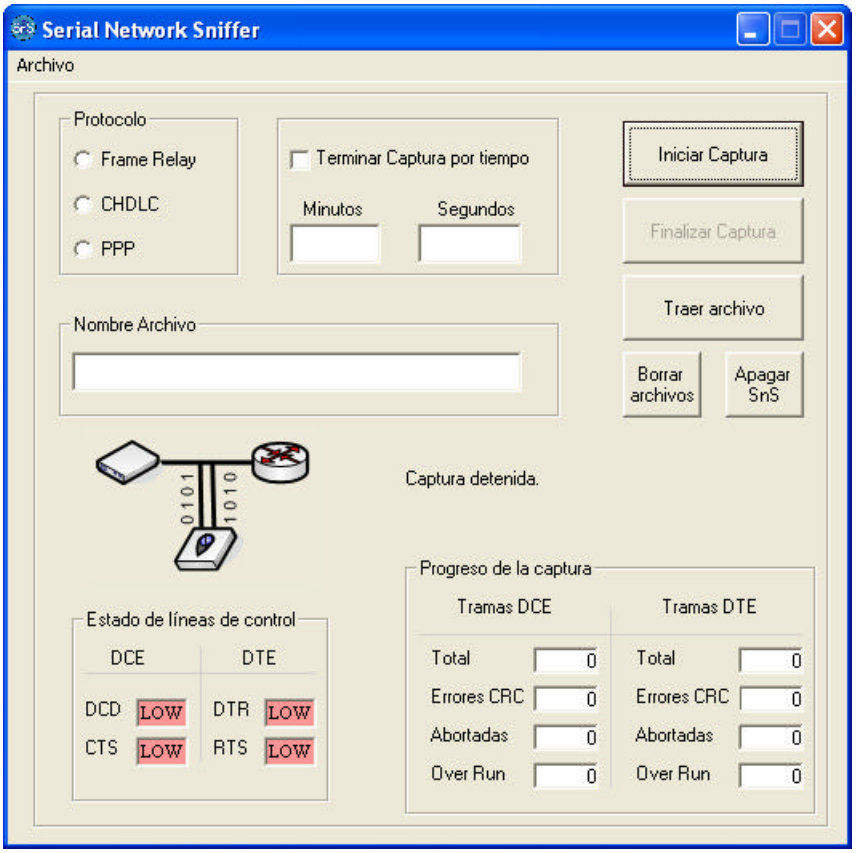


Figura 18: Interfaz gráfica para Windows

La opción de captura por cantidad de paquetes no se encuentra contemplada en esta versión de la interfaz grafica.

Al finalizar la captura el archivo es copiado en la PC del usuario. Una vez concluida la transferencia del archivo al PC se ejecuta el programa analizador Ethereal.

Se brinda la posibilidad al usuario de liberar espacio en disco borrando todos los archivos de captura del directorio **/home/sns/analizador/capturas**, así como también la opción de apagar el analizador SnS desde la interfaz.

En la parte inferior izquierda de la interfaz se puede visualizar el estado de las líneas de control. Esto se logra corriendo remotamente en el analizador, cada 4 segundos, el comando ***statuslctrl***. Este comando realiza una consulta del estado de las líneas de control y pasa a un archivo el valor de las mismas. Éste archivo es copiado desde el analizador e interpretado por la interfaz, desplegando el estado de las líneas de control.

En el proceso de captura cada 3 segundos se va guardando a un archivo la cantidad de tramas capturadas. Este archivo es leído cada 4 segundos por la interfaz gráfica. En el archivo se encuentran los valores de tramas capturadas, errores de CRC, tramas abortadas y *overrun*.

A modo de resumen, la interfaz gráfica desarrollada oculta, de forma amigable para el usuario del analizador SnS, el proceso de correr los comandos remotamente en consola
.

Capitulo 7

ANÁLISIS DE CAPTURAS Y MARCAS DE TIEMPO

7.1 Caracterización de la precisión del analizados SnS.

Como resultado principal comprobamos que el analizador cumple con los requerimientos mínimos de no perder tramas y de tener precisión superior al milisegundo en las marcas de tiempo. A continuación resumimos las características de precisión del SnS y mencionamos los factores que inciden en la misma.

7.1.1 Características y factores que determinan la precisión.

El SnS inserta las marcas de tiempo por software, es decir lo hace el sistema operativo al momento de atender la interrupción solicitada por la tarjeta adquisidora luego de recibir cada trama.[16]

El tiempo que el sistema operativo demora en atender la interrupción se llama latencia. Esta latencia varía en función de las demás tareas que se encuentre realizando el sistema operativo. La latencia variable es la principal limitante en cuanto a la precisión de las marcas de tiempo. Su rango de variación y por lo tanto su efecto de error es mayor si la utilización del canal es alta. A velocidades de 2.048Mbps y con una utilización cercana al 100%, estamos en el caso más exigente para el que está diseñado el analizador y la precisión es de 1ms. A porcentajes

[16] Las marcas de tiempo insertadas por la tarjeta adquisidora tienen menor precisión que las insertadas por el sistema operativo.

menores de utilización del canal o a velocidades más bajas, la precisión mejora, hasta llegar a 0,1ms.

La latencia depende fuertemente de la demanda de otros procesos al sistema operativo. De haber procesos altamente demandantes de recursos[17] la latencia aumenta e incluso puede ocasionar pérdida de tramas. Por este motivo recomendamos no ejecutar tareas externas al analizador durante una captura de datos.

El hecho de que la marca de tiempo es colocada al terminar de recibir la trama debe tomarse en cuenta al realizar capturas en paralelo con otros analizadores que coloquen la marca de tiempo al comienzo de la trama. Para comparar las marcas de tiempo debe descontarse el tiempo de serialización[18] en las marcas del SNS. Este tiempo es igual al largo de la trama en bits, incluyendo los bits adicionales debidos al *bit stuffing*[19], dividido la frecuencia de reloj del canal.

Otro efecto a tener en cuenta en el caso de realizar un análisis con varios analizadores simultáneamente es el de la sincronización de los mismos. Cada analizador tiene un reloj independiente afectando esto a las marcas de tiempo en dos puntos relevantes. El primero es que el origen de tiempos será distinto, ocasionando un defasaje constante entre las marcas de tiempo de uno y otro. Como segundo punto observamos que la frecuencia de reloj diferirá levemente, ocasionando una deriva en las marcas de tiempo (el reloj de un analizador adelantará respecto al reloj del otro) Estos efectos pueden corregirse fuera de línea, como lo hicimos en las pruebas que realizamos para testear la precisión de las marcas de tiempo, pero en los análisis normales esto no es relevante.

Las pruebas que realizamos consistieron en capturar en serie con un analizador HP del Laboratorio de AntelData los datos correspondientes a distintas transferencias de archivos y ejecuciones de *pings* realizados a través de una interfaz V.35. Luego comparamos las marcas de tiempo insertadas por el HP y por el SNS. La descripción en detalle de esas pruebas la hacemos las secciones siguientes. Basándonos en esas pruebas caracterizamos de la siguiente forma la precisión de las marcas de tiempo del SNS:

Con el cuidado de no exigir al analizador con procesos altamente demandantes de recursos, se observan las siguientes características.

- Capacidad de capturar la totalidad de las tramas de la interfaz V.35 hasta una velocidad de 2.048Mbps.
- Precisión de las marcas de tiempo mejor a 1 ms.
- Precisión de las marcas de tiempo de 0,1 ms en las siguientes circunstancias:
  A 2.048Mbps si el canal no está utilizado al máximo.

[17] Un proceso altamente demandante es por ejemplo una transferencia de archivo en modo encriptado a través de la interfaz *fastethernet*

[18] Tiempo de srialización es el tiempo que tarda en colocarse la trama en el canal

[19] *Bit stuffing* es la inserción de bits extra. En HDLC luego de la aparición de cinco bits 1 consecutivos dentro de la parte de carga de la trama, se inserta un bit 0. Así se impide que accidentalmente dentro de la carga de una trama figuren secuencias de bits similares a las delimitadoras de trama las que consisten en un 0, seis 1 y un 0

A 64kbps con máxima utilización del canal.

- Las marcas de tiempo son insertadas al terminar de recibir cada trama.

7.2 Pruebas realizadas y observaciones importantes

La forma más sencilla de probar un equipo de medida, como por ejemplo el analizador SnS, es realizar un conjunto de mediciones en distintas situaciones y comparar los resultados con las mediciones efectuadas por un equipo de medición más preciso.

En nuestro caso tuvimos acceso a equipos analizadores muy precisos, modelos HP Advisor, los cuales son propiedad de ANTEL. Los mismos nos fueron de suma utilidad en las pruebas.

En primera instancia confirmamos la exactitud de los analizadores HP Advisor. Para ello realizamos capturas con dos equipos en serie y examinamos luego la concordancia de las marcas de tiempo insertadas. La exactitud observada fue de 3 microsegundos y una muy leve diferencia de frecuencia de reloj.

Realizamos pruebas a distintas velocidades de transferencia y con distintos tamaños de tramas. Para ello utilizamos 2 *routers* conectados a través de una interfaz v.35 de la cual capturamos los datos. A su vez a cada *router* conectamos por la interfaz Ethernet una PC. El tráfico lo generamos transfiriendo archivos entre las PCS y realizando *pings* entre los *routers*. De los resultados de estas pruebas obtuvimos el error del analizador SnS que detallamos en la sección anterior.

Para comparar las marcas de tiempo de uno y otro tuvimos en cuenta los efectos mencionados en la sección anterior: inserción de la marca de tiempo al comienzo de la trama por el HP y al final por el SNS (corregimos el tiempo de serialización), diferencias en orígenes de tiempo y relojes no sincronizados.

Como prueba complementaria chequeamos la precisión de las marcas de tiempo insertadas directamente por la tarjeta Sangoma S5141 del SnS. Se observó una diferencia de frecuencia importante entre los relojes de la Sangoma S5141 y el reloj del HP. La deriva es de 1segundo en 1hora. Esto en la magnitud pequeña de las marcas de tiempo es importante. Corrigiendo esta deriva, la precisión coincide con lo indicado por el fabricante que es de 1milisegundo.

En las secciones siguientes detallamos cada prueba realizada.

7.3 Exactitud del analizador HP Advisor

El testeo de la precisión de las marcas de tiempo de nuestro analizador SnS como vimos fue realizado haciendo diversas capturas con el HP Advisor en paralelo y luego comparando los resultados. Debido a que las marcas de tiempo que inserta el HP son extremadamente precisas, las diferencias que tuvimos con él significan errores en las marcas de tiempo del SnS, es decir asumimos como verdaderas las marcas insertadas por el HP Advisor.

Para tener total confianza en las marcas de tiempo del HP Advisor verificamos la exactitud del mismo.

Realizamos dos capturas de transferencias de archivos a velocidades de 64kbps y de 2.048Mbps respectivamente. Utilizamos dos equipos HP Advisor en serie. Luego analizamos los archivos de captura chequeando que no hubieran perdido tramas y comparando entre ellos las marcas de tiempo que insertaron.

Después de efectuada la captura de datos, exportamos la información de las marcas de tiempo a un archivo de texto mediante el programa de análisis que es provisto con los equipos. Las comparamos entre sí utilizando una planilla electrónica. Las diferencias de las marcas de tiempo quedaron comprendidas en un entorno de más menos 3 micro segundos. 3 micro segundos es el tiempo que insume transmitir 6 bits a una tasa de 2.048Mbps por la interfaz V.35, por lo que a los efectos de testear el SnS consideramos el resultado del HP Advisor como el valor verdadero de la marca de tiempo.

Como resultados fundamentales encontramos que no pierden tramas y que la exactitud es de 3 microsegundos.

A continuación describimos las pruebas realizadas a los analizadores HP.

7.3.1 Análisis de captura de transferencia de archivo a 64kbps.

Hicimos coincidir el origen de tiempos (la marca de tiempo para la primer trama capturada) y observamos en la segunda marca una diferencia de apenas 1ns, pero ya en la tercera la diferencia alcanzaba los 16us. Las diferencias cada vez se acrecentaban más hasta llegar a la ultima trama de la captura (habían transcurrido 93 segundos aprox.) que tuvo una diferencia de 774us. Las diferencias entre marcas de tiempo en general aumentaban trama a trama, pero en las que disminuía lo hacía por un valor menor a 3us. El aumento gradual en las diferencias suponemos que se debe a una diferencia en la frecuencia del reloj que utilizan como base de tiempos. Luego se aprecia un error muy pequeño del orden de microsegundos que tiene un comportamiento que a los efectos del alcance de nuestro análisis suponemos aleatorio. A

continuación se presenta la grafica de las diferencias entre marcas de tiempo en función de las marcas insertadas por uno de los HP ADVISOR.

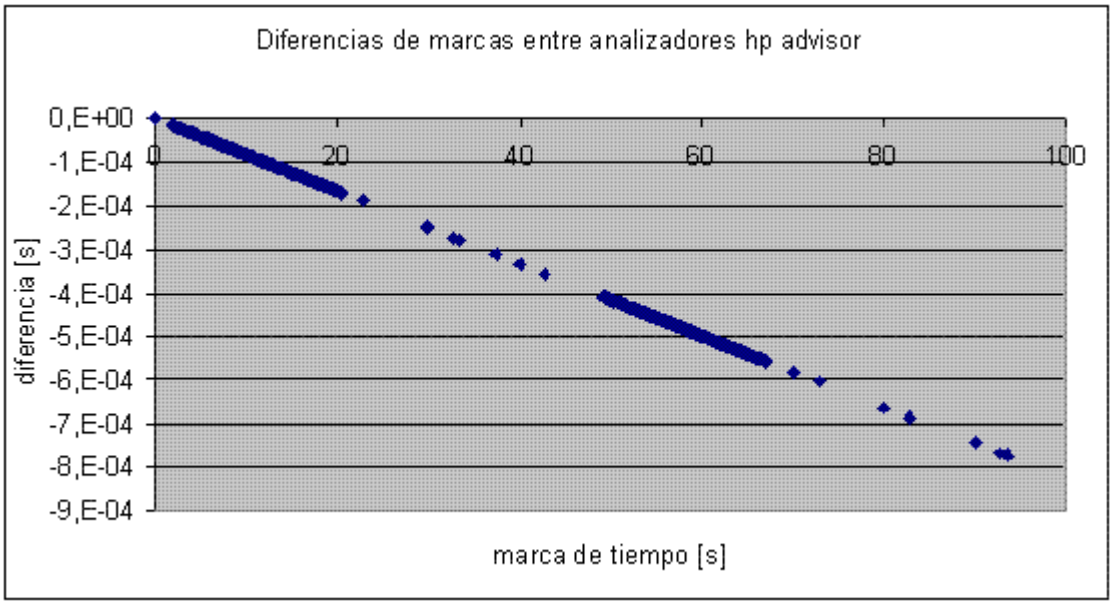


Figura 19: Error HP transferencia archivo a 64kbps

Calculamos el defasaje de relojes aproximando las diferencias entre las marcas de tiempo por una recta, utilizando el método de mínimos cuadrados y haciendo coincidir la recta con la primera marca de tiempo (ya que allí ajustamos el eje de tiempos de modo que el defasaje es cero). Obtuvimos una pendiente para la recta de -8,289838056 us defasaje / segundos de comenzada la captura.

Corregimos el defasaje de relojes y calculamos nuevamente las diferencias entre marcas de tiempo, es decir el apartamiento de las diferencias originales a la recta de aproximación. Interpretamos estas diferencias como el error en las marcas de tiempo. A continuación se presenta la gráfica del error y damos valores descriptivos del mismo como ser error máximo, y desviación estándar.

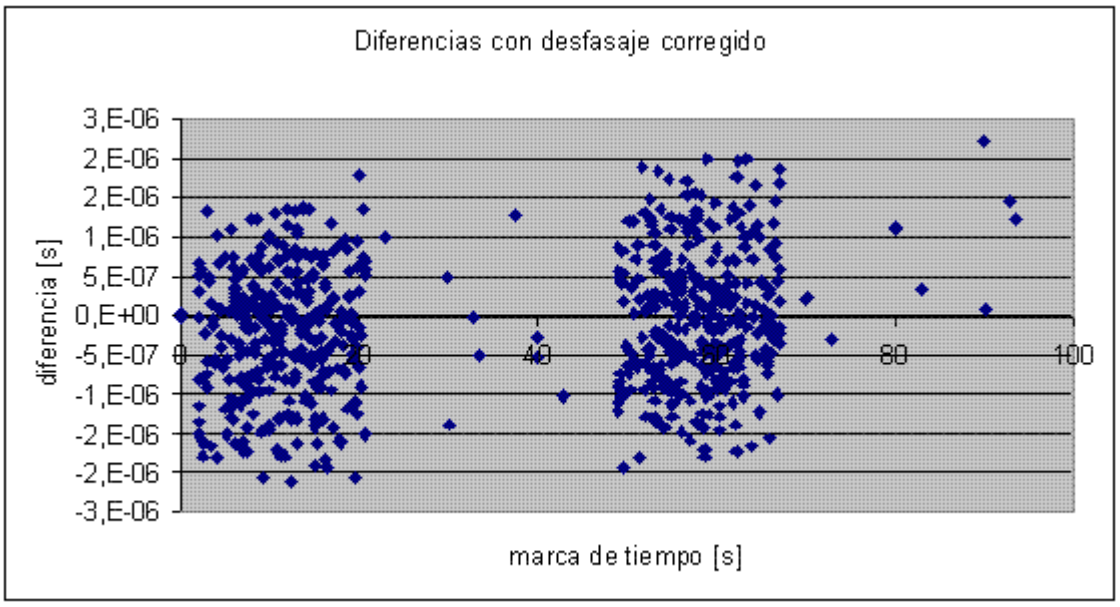


Figura 20: Error HP transferencia archivo a 64kbps (con frecuencia corregida)

La precisión del HP es muy buena, del orden de 3 microsegundos. Vemos que el error quedó comprendido en:

Máximo	3us
Mínimo	-3us

Tabla 4: Diferencias entre HPs

7.3.2 Análisis de captura de transferencia de archivo a 2.048Mbps.

Los datos obtenidos de la captura se trataron en forma similar a los de la captura anterior. A continuación presentamos los resultados.

Las diferencias entre marcas de tiempo presentan igual que en el caso anterior el efecto de corrimiento originado a causa de la diferencia de frecuencia de los relojes de los equipos HP. A continuación presentamos el gráfico de las mismas.

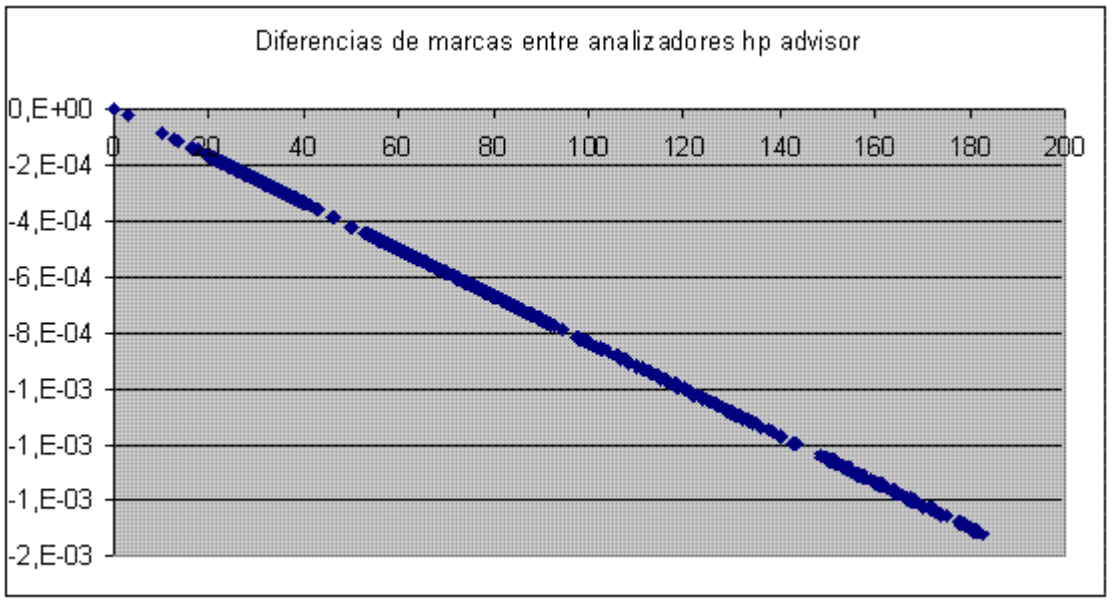


Figura 21: Error HP transferencia archivo a 2.048Mbps

Obtuvimos una pendiente para la recta de aproximación por mínimos cuadrados de -8,334130031E-06 muy similar a la del caso anterior que recordamos fue de -8,289838056 us defasaje / segundos de comenzada la captura.

Corregimos la diferencia en frecuencia de relojes y calculamos nuevamente las diferencias entre marcas de tiempo. Se obtuvieron los siguientes resultados.

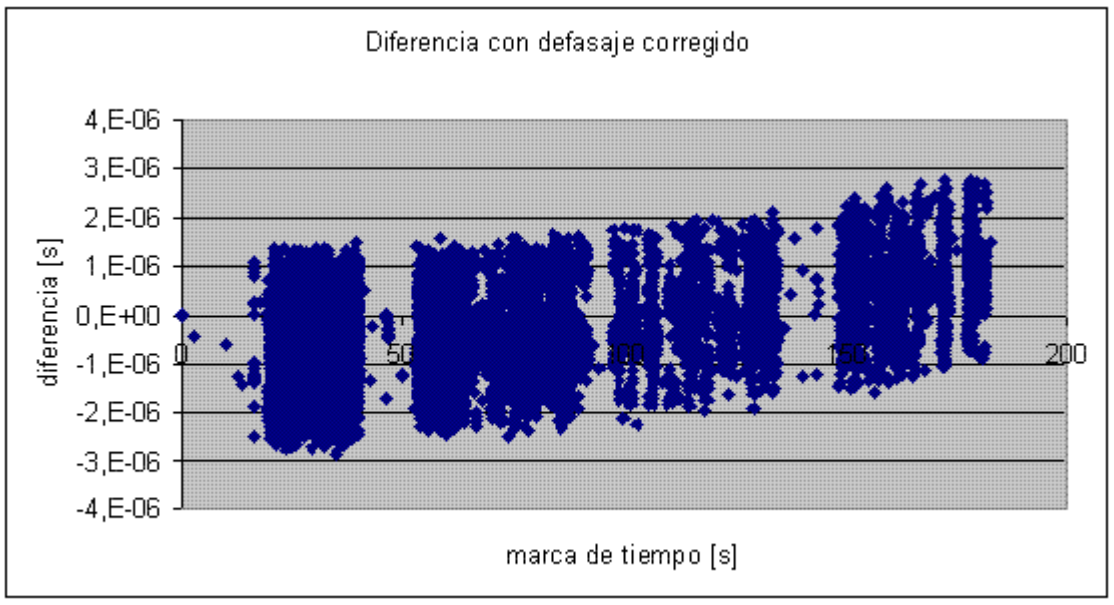


Figura 22: Error HP transferencia archivo a 2.048Mbps (con frecuencia corregida)

En este caso observamos que luego del primer grupo de marcas, se produce un defasaje creciente, atribuible a que en ese momento la frecuencia de alguno de los HP tuvo una leve variación.

Vemos que de todas formas el error quedó comprendido en los mismos valores que en el caso anterior:

Máximo	3us
Mínimo	-3us

Tabla 5: Diferencias entre HPs

7.4 Inserción de marcas de tiempo al comienzo por el HP y al final de la trama por el SnS.

Del análisis de la diferencia entre marcas de tiempo de tramas consecutivas, se verificó que el HP inserta las marcas de tiempo al inicio de cada trama y el analizador SnS lo hace al final. A continuación evidenciamos este hecho.

Si comparamos la diferencia de *time stamps* de tramas consecutivas de distintos tamaños, por ejemplo la número 3 y 4 de la tabla 4, vemos que en el caso del analizador HP este tiempo corresponde a la diferencia de tiempo de comienzos de transmisión (asumimos un tiempo despreciable a estos efectos de inactividad del canal entre tramas) es decir lo que demora en transmitirse la trama 3 y en el caso del SnS es el tiempo de transmisión de la trama 4.

Nº trama	Largo [bytes]	HP time stamp [s]	HP Diferencia [s]	SNS time stamp [s]	SNS Diferencia [s]
1	1506	80,8695418830	80,8695418830	81,0547149180	81,0547149180
2	46	81,0581519600	0,1886100770	81,0603954790	0,0056805610
3	46	81,0642299650	0,0060780050	81,0664305680	0,0060350890
4	1506	81,0703239440	0,0060939790	81,2555477620	0,1891171940
5	1506	81,2589797970	0,1886558530	81,4442124360	0,1886646740

Tabla 6: Ubicación de las marcas de tiempo

Como ejemplo presentamos las tramas número 3 y 4. La diferencia de *time stamps* del HP es de 0, 0060939790s, que es aproximadamente lo que le toma a la trama 3 ser transmitida ((46+1bandera) bytes*8bits/64.000bps=0,005875s), del orden de 0,00609. La diferencia entre *time stamps* del SnS es de 0,1891171940, aproximadamente lo que le toma a la trama 4 ser transmitida ((1506+1bandera) bytes*8bits/64.000bps=0,188375s).

7.5 Efecto en el orden aparente de las tramas debido a la inserción de las marcas de tiempo al comienzo o al final de las tramas.

A continuación se presenta un ejemplo de captura en la cual se muestra la diferencia de insertar la marca de tiempo en distintos momentos, afectando el orden de las tramas guardadas en el archivo.

Analizador HP Advisor				**Analizador SnS**		
Origen	**Bytes**	**Protocolo**		**Origen**	**Bytes**	**Protocolo**
DTE	1506	IP	?	DTE	1506	IP
DTE	326	ICMP	?	DTE	326	ICMP
DCE	1506	IP	?	DCE	1506	IP
DCE	326	ICMP	?	DCE	326	ICMP
DTE	1506	IP	?	DTE	1506	IP
DTE	326	ICMP	?	DTE	326	ICMP
DCE	1506	IP	?	DTE	26	SLARP
DTE	26	SLARP	?	DCE	1506	IP
DCE	326	ICMP	?	DCE	326	ICMP

Tabla 7: Cruzamiento de tramas

Cuando una trama "*corta*" es transmitida en un sentido y en el otro se está transmitiendo una trama *"larga"*, insertar la marca de tiempo al inicio o al final ocasiona un cruzamiento cuando son guardadas en el archivo. En la figura se muestra que el HP Advisor inserta las marcas de tiempo en ambos canales a secuencias de paquetes distintas que el analizador SnS.

En la figura siguiente se muestra la secuencia de la tabla superior gráficamente.

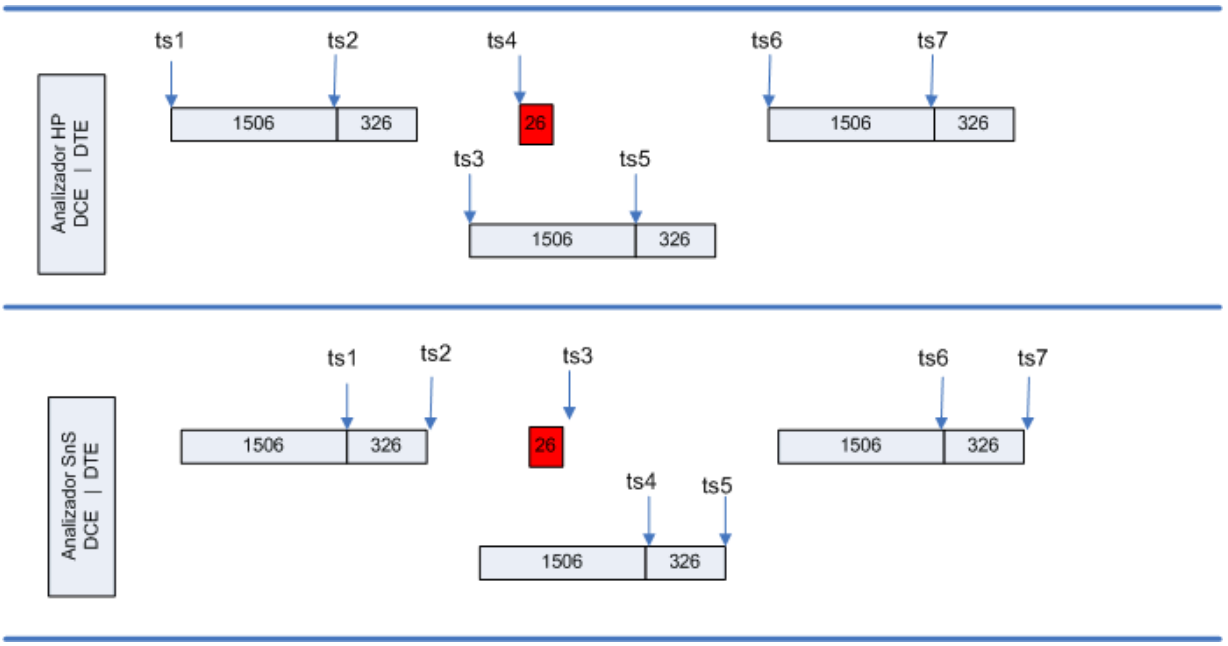


Figura 23: Cruzamiento de tramas

Cuando se analice el archivo de captura, las tramas se verán con distintas marcas de tiempo y algunas tramas también en distinto orden. Los analizadores utilizan distintos criterios para insertar las marcas, el HP el momento en que comienza la transmisión y el SnS el momento en que la trama termina de transmitirse. Si se desea puede cambiarse las marcas de tiempo introducidas por el SnS restando el tiempo de transmisión de las mismas, (largo de trama en bits / frecuencia de reloj de transmisión) y las tramas quedan casi en su totalidad en el mismo orden.

Observemos que no es posible garantizar, por más preciso que sean los analizadotes, que el orden en que se verán las tramas en ellos sea el mismo. Podría suceder que DCE y DTE transmitan tramas con diferencias de tiempo entre si menores a la precisión de los analizadores. En esa situación un analizador puede insertar primero la marca de tiempo a la trama transmitida por el DCE y el otro a la transmitida por el DTE, ocasionando que se vean en distinto orden.

7.6 Error aparente debido al bit stuffing.

Cuando las tramas son transmitidas por el canal, se las delimita con una bandera de un cero, seis unos y un cero: "01111110". Dentro de la carga de las tramas pueden ocurrir esas cadenas de bits, de no hacer *bit stuffing*, estas cadenas se interpretarían erróneamente como delimitadoras de tramas. El *bit stuffing* consiste en agregar un 0 luego de la aparición de 5 unos consecutivos. De esta forma nunca se transmiten como datos estas banderas. El receptor luego examina el contenido de la trama que recibe y siempre que encuentre 5 unos seguidos descartará el siguiente bit.

Vemos que el *bit stuffing* introduce bits extra a la trama, por lo que debe tenerse en cuenta al calcular el tiempo que demora la trama en ponerse en el canal.
Cada ocurrencia de 5 unos implica un bit extra. El tiempo de poner un bit en el canal es el inverso de la velocidad del reloj de la transmisión. El caso más significativo se da a la menor frecuencia, o sea a 64kbps. El tiempo de bit es de 1/64E3 = 0,015625 ms.

Para estudiar la relevancia de este error analizamos una trama correspondiente a una transmisión de un archivo. Se encontraron 300 ocurrencias de 5 unos, por lo tanto esa trama en el canal cuenta con 300bits extras debidos al *bit* stuffing. Entonces el tiempo extra insumido en la transmisión es de 300 * 0,015624 ms = 4,6872 ms. Este es un valor a tener en cuenta dado que es 4,6 veces la precisión de la marca de tiempo por hardware que indica el fabricante.

El error en esta marca es de -3,3ms respecto al promedio del error.

En las tramas de 50 bytes, podemos asumir que la influencia del *bit stuffing* es muy poca (sabemos que es menor a 50 por el tiempo de bit es decir menor a 0,75ms). El error promedio constatado en esas tramas es 2ms respecto al promedio del error.

Si restamos el tiempo extra insumido por el *bit stuffing*, de forma de corregir la marca de tiempo de la trama en análisis, resulta que el error disminuye en 3,3ms, es decir queda 1,4ms apartado del promedio.

Podemos asumir que el defasaje entre los equipos es el promedio del error de las tramas cortas de 50bytes (para las que despreciamos el efecto del *bit stuffing*). Restando ese defasaje de 2ms a la marca corregida por *bit stuffing* de la trama larga que en el ejemplo es de 1,4ms obtenemos un error de 0,6ms.

Observamos que para comparar las marcas de tiempo del SnS que son insertadas al final de la trama con las del HP que lo son al comienzo, es necesario incluir el *bit stuffing*, sobre todo trabajando a baja velocidad (64kbps) donde los tiempos de bit son mayores.

Estadísticamente el efecto del *bit stuffing* esta en función del largo de la trama, para tramas largas se tiene mayor incidencia ya que aumenta la probabilidad de ocurrencias de secuencias de cinco unos. Para calcular el error asumimos que en las tramas cortas de 50 bytes podemos despreciar el efecto del *bit stuffing* y calcularemos el error en esas tramas.

7.7 Pruebas realizadas para la cuantificación de los errores

Se diseñó una batería de pruebas a realizar en el laboratorio de ANTELDATA para comparar la performance del Analizador SnS con el analizador HP Advisor propiedad de ANTEL. La finalidad de esta batería de pruebas es realizar un análisis comparativo de las marcas de tiempo insertadas por el analizador SnS y la precisión con que estas son insertadas.

Las pruebas fueron realizadas conectando el analizador SnS en serie con el analizador HP Advisor y utilizando el protocolo de capa de enlace CHDLC y FRAME RELAY.

La batería de pruebas consistió en realizar capturas de *pings* y de transferencias de archivos.

Los *pings* y las transferencias de archivos se realizaron entre dos *routers* Cisco 1700, también propiedad de ANTEL. Se trabajó a 2.048Mbps y 64kbps para poder testear el equipo con diferentes niveles de exigencia.

Para la transferencia de archivos se utilizaron dos PCs en los extremos de la conexión, los cuales transferían archivos entre sí, encontrándose los *routers* y analizadores entre ambas PCs, tal cual se muestra en la figura.

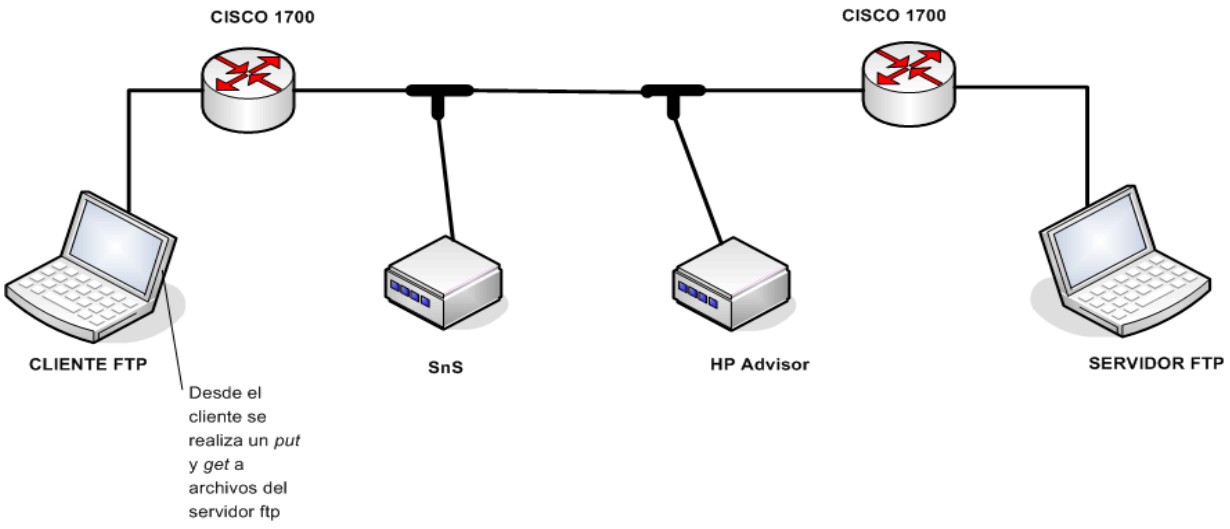


Figura 24: Esquema prueba transferencia de archivos

Los archivos se transfirieron por ambos canales de la interfaz V.35, siendo estos de un tamaño de 40MB aproximadamente, para exigir la conexión. Además se realizaron *pings* en el transcurso de la transferencia para confirmar que el orden de la comunicación se mantenía en los datos capturados.

En el analizador HP se podía visualizar la utilización de ambos canales superando el 95%, y llegando en ocasiones al 100%.

En todas las pruebas realizadas se capturaron la misma cantidad de paquetes de ambos puertos con el equipo SnS y el HP Advisor.

7.8 Análisis de Pings

En los pings se utilizaron siempre los mismos bytes de carga, por lo que el efecto del *bit stuffing* en la variación de largo de la trama puede despreciarse.

7.8.1 Prueba 1: 1000 pings de 36 bytes a 2.048Mbps

Gráfico del error. Se resta a la marca del analizador SnS la marca de tiempo del HP. Para centrar el inicio del error se restan a todas las diferencias un t_0 fijo igual a la primera diferencia de tiempos.

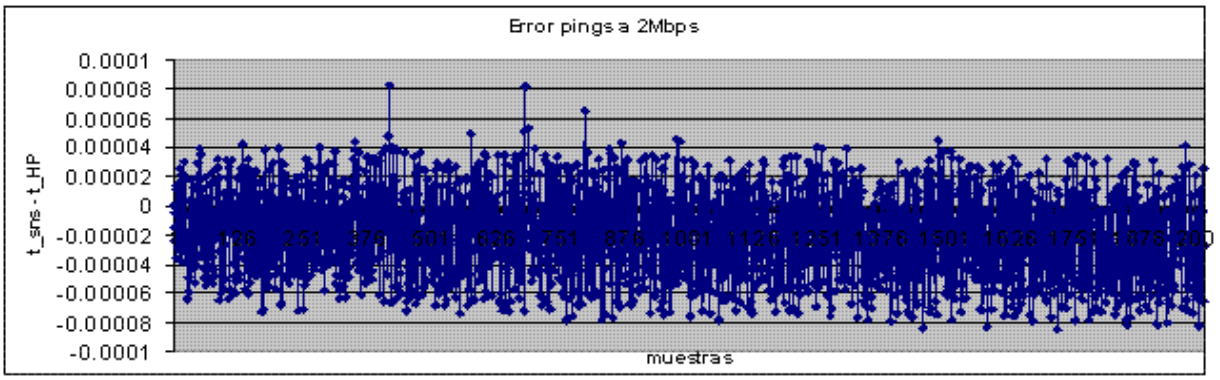


Figura 25: Error de marcas de tiempo de *pings* a 2.048Mbps

Tramas invertidas	0	máximo	8.2255E-05
		mínimo	-8.4638E-05
		promedio	-1.8964E-05

Tabla 8: Datos relevados, prueba 1

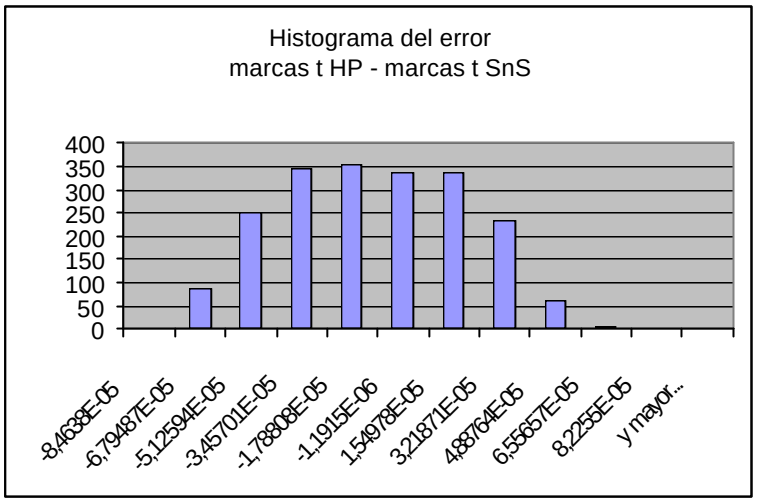


Figura 26: Histograma prueba 1

El error cometido a 2.048Mbps es menor a 0,1 milisegundos. Se repitió la prueba y los valores dieron dentro del mismo orden.

7.8.2 Prueba 2: 1000 pings de 36 bytes a 64kbps

Como en las otras pruebas debe de restarse a las marcas de tiempo del SnS la duración del largo de trama, de modo de hacer que la marca de tiempo corresponda al comienzo de la transmisión de la trama y poder compararla con las marcas insertadas por el HP ADVISOR.

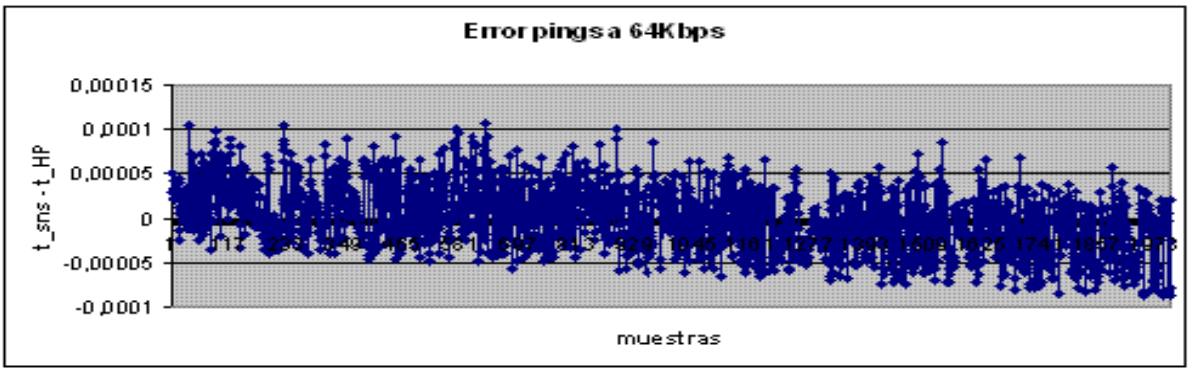


Figura 27: Error de marcas de tiempo de *pings* a 64kbps

A continuación el resumen de los resultados. En la captura invirtió una trama que corresponde a corresponde a un paquete SLARP que tiene 26 bytes mientras que los demás paquetes que

son originados por los pings tienen 42 bytes, pero con la corrección del la duración de la transmisión de la trama, se corrige y se tienen 0 tramas invertidas.

NOTA: Siempre se toma como verdad los datos del analizador HP ADVISOR

Tramas invertidas	0	Máximo	0,000106831
		Mínimo	-0,000087718
		Prom (err. abs.)	0,0000292733

Tabla 9: Datos relevados 1, prueba 2

A la diferencia de marcas de tiempo se le resta un *offset* para centrar el comienzo en el valor 0, el motivo es mejorar la visualización en el gráfico. Este *offset* se atribuye a un comienzo desfasado de las capturas.

Se puede apreciar un defasaje creciente con el tiempo en las marcas de tiempo, el cual puede atribuirse a que la unidad de tiempo de ambos analizadores no es la misma. Esta prueba permitió apreciar el defasaje creciente debido a que los 1000 *pings* realizados demoraron un tiempo considerable (10 min. aprox.).

Recalculamos el error ahora con la diferencia de frecuencia corregida.

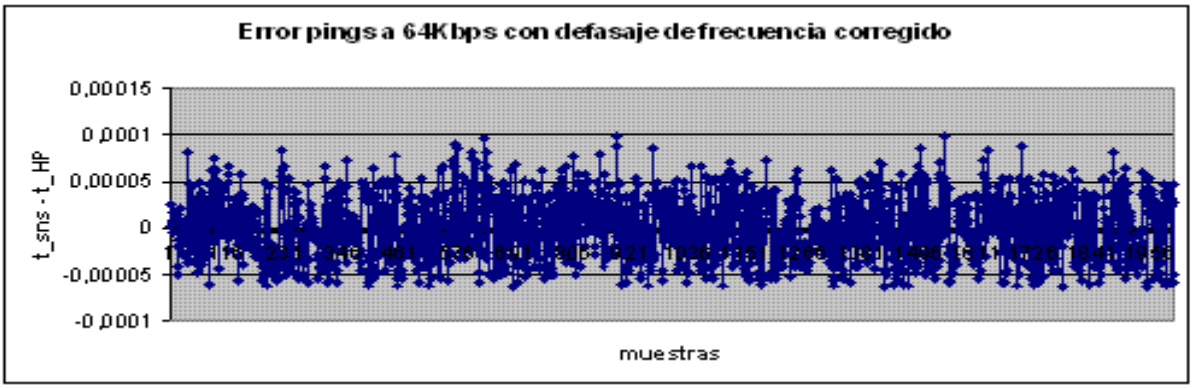


Figura 28: Error de marcas de tiempo de *pings* a 64kbps

Nuevos valores:

err. máx.	9,96677E-05
err. min.	-6,39747E-05
Prom. (err. abs)	2,72006E-05

Tabla 10: Datos relevados 2, prueba 2

Igual que en la prueba anterior la exactitud es superior a 0,1 milisegundo

7.9 Análisis de capturas de transferencias de archivos

Al comparar los archivos de captura notamos que las tramas se presentaban en distinto orden. Esto es debido fundamentalmente a que el HP inserta las marcas de tiempo al comienzo de cada trama y el SnS lo hace al final, como se mencionó anteriormente.

Se procede a estudiar por separando las tramas por origen de DCE y DTE. Detallamos a continuación los resultados de las distintas pruebas.

7.9.1 Procedimiento para el cálculo del error.

Para ajustar las marcas de tiempo del SnS al criterio utilizado por el HP se corrigieron restando el tiempo de transmisión de la trama, de forma de simular que el analizador SnS insertara las marcas de tiempo al comienzo de cada trama.

Debido a que el tiempo de inicio de captura es distinto en el HP que en el SnS, todas las marcas de tiempo quedan desfasadas generando un error aparente. Para ajustar el tiempo de inicio de cada captura se resta a cada error la media.

NOTA: Los datos de la batería de pruebas se adjuntan en el CD.

7.9.2 Análisis de datos desde el DCE

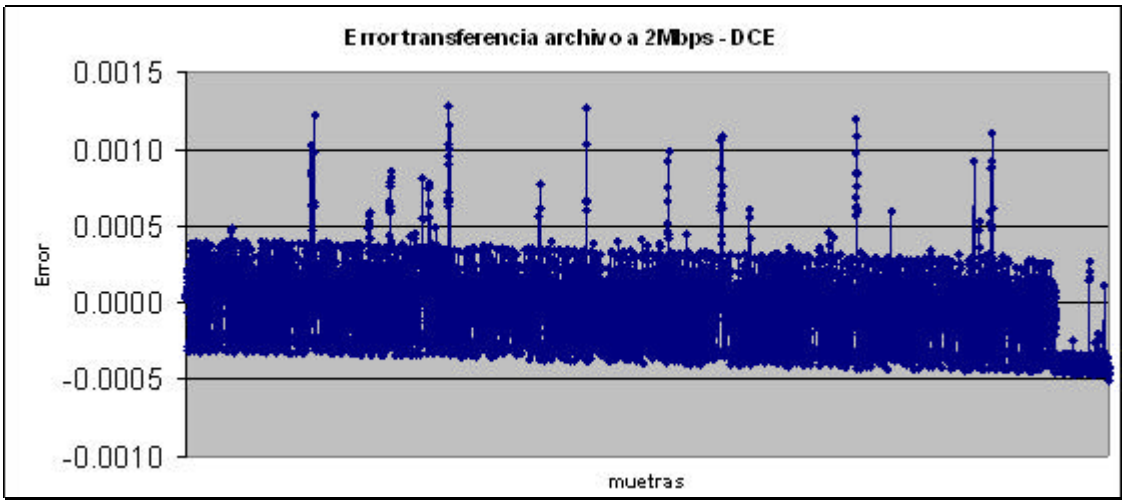


Figura 29: Error transferencia archivo a 2.048Mbps DCE

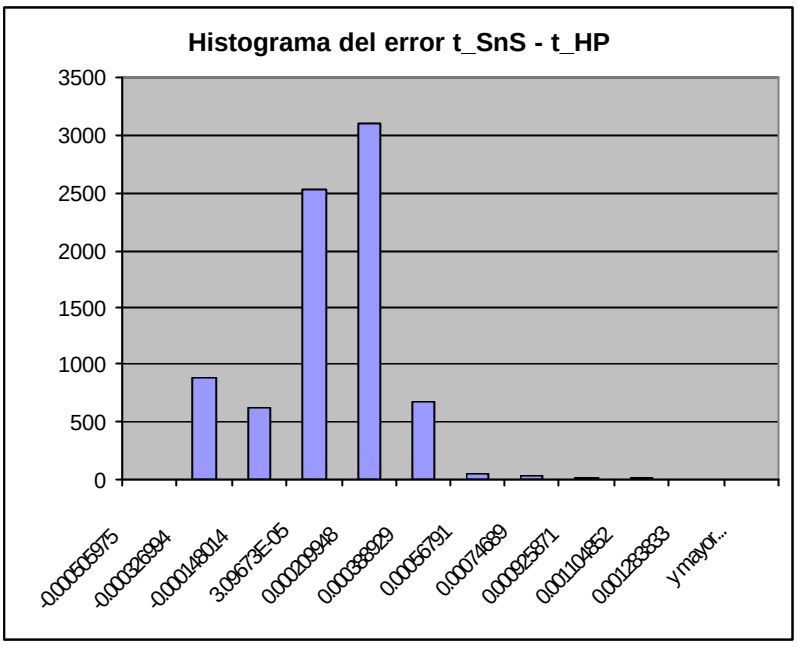


Figura 30: Histograma transferencia archivo DCE

0.0012838329	máximo
-0.0005059751	mínimo

Tabla 11: Datos relevados transferencia archivo DCE

7.9.3 Análisis de datos desde el DTE

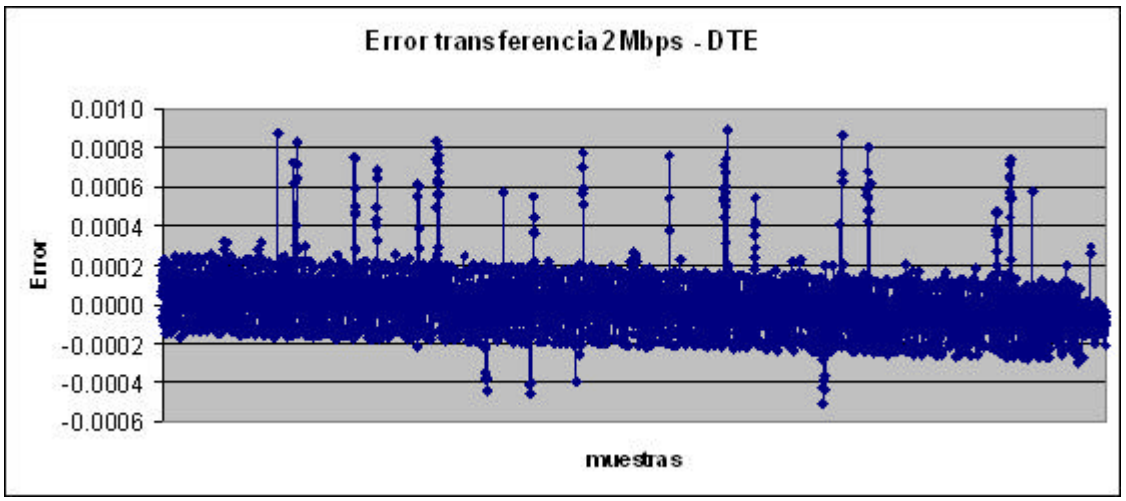


Figura 31: Error transferencia archivo a 2.048Mbps DTE

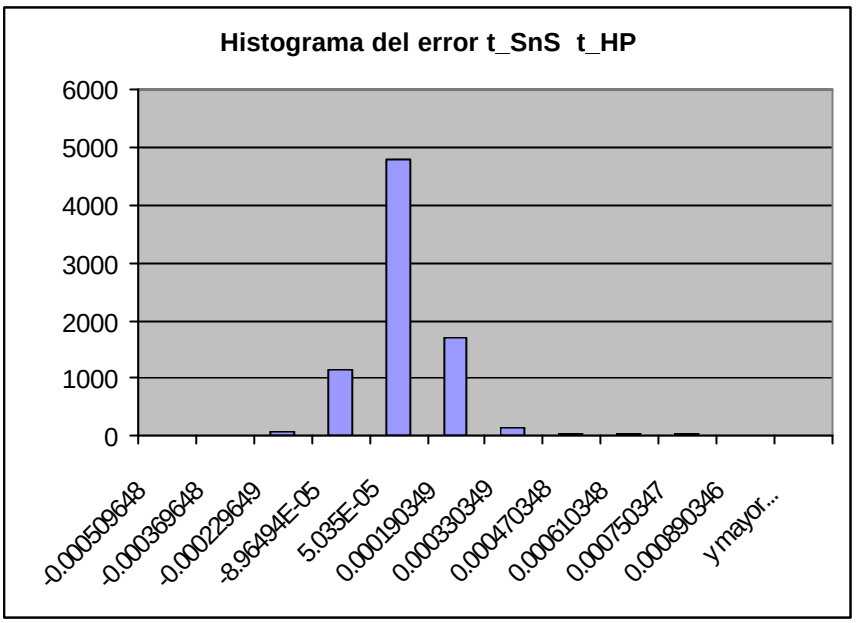


Figura 32: Histograma transferencia archivo DTE

0.0008903464	máximo
-0.0005096476	mínimo

Tabla 12: Datos relevados transferencia archivo DTE

Se observa que disminuye la precisión respecto al caso de la transferencia a 64kbps, aumenta levemente el error promedio y en mayor medida el error máximo. Posiblemente se deba que a 2.048Mbps el analizador debe realizar mayores accesos a disco y eso influya aumentando la variación de la latencia del sistema operativo al momento de atender las interrupciones y setear las marcas de tiempo.

7.9.4 Consideración del bit stuffing.

Teniendo en cuenta el error en la comparación con el HP debido a la falta de estimación del *bit stuffing*, analizamos los errores en las tramas cortas de 44 bytes en las que asumimos un efecto despreciable del mismo. Vemos que el error mejora pero el efecto del *bit stuffing* no es relevante a esta velocidad, si lo es a velocidades menores.

-0,00046696	Mínimo
0,00132285	Máximo
0,0002265	Promedio del error absoluto

7.9.5 Transferencia de archivos a 64kbps

A esta velocidad en la que el tiempo de bit es de 1/64 ms (0,015ms) es necesario tomar en cuenta la corrección de *bit stuffing*.

Aquí resumiremos los errores analizando únicamente tramas cortas y asumiendo que el efecto en las marcas de tiempo tramas cortas del *bit stuffing* es despreciable.

Analizamos tramas de 68bytes.
El error es menor que en el caso de 2.048Mbps:

err. min.	-0,00010481
err. máx.	5,3262E-05
Prom. (abs. err.)	4,653E-05

7.10 Error en marcas de tiempo por hardware

Como se comentó en la introducción, el SnS tiene la facultad de setear las marcas de tiempo por hardware. En esta sección mostraremos que el error introducido es mayor al de las marcas por software, lo que nos llevó a utilizar las marcas por software para el diseño final del SnS.

Para averiguar el error de las marcas por hardware se realizó una transferencia de archivo a 64kbps y se la capturo con el SnS y el HP simultáneamente. Luego comparamos las marcas de tiempo.

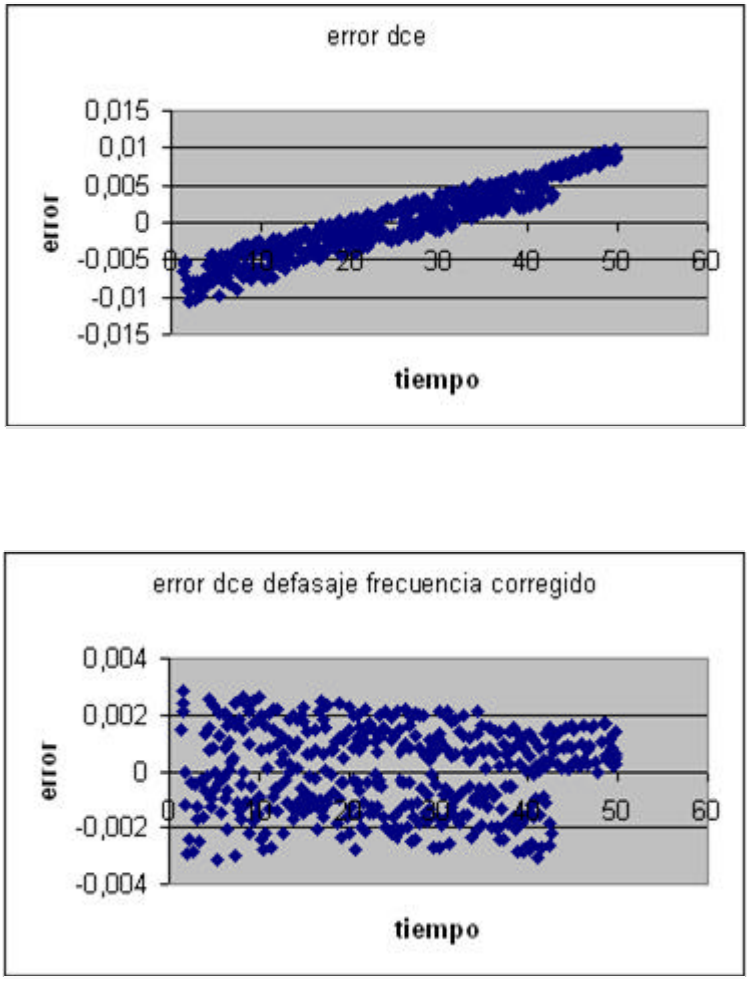


Figura 33: Error con marcas de tiempo por hardware

El error medido coincide con la especificación del fabricante de que las marcas por hardware tienen precisión de un milisegundo. La tarjeta indica la marca de tiempo por hardware con un contador cíclico de 2 bytes que expresan los milisegundos transcurridos.

7.10.1 Resultados.

Observamos que las marcas de tiempo se colocan al final de la trama igual que cuando son por software. Por lo que para compararlas con las del HP ADVISOR hubo que restarles el tiempo que demora en ponerse en el canal cada trama (Como vimos antes es función de la frecuencia de reloj y del largo de trama). En las gráficas no se corrigió el efecto del *bit stuffing* que a 64kbps es muy considerable, el efecto del mismo se visualiza por la distribución no uniforme del error, algunos comprendidos en una franja por encima del promedio y otros en una por debajo.

En la primera gráfica, que es la diferencia habiendo únicamente corregido el tiempo de transmisión de cada trama, también observamos la diferencia de relojes utilizados por el SnS y el HP. Esta diferencia es mayor que en los casos anteriores porque ahora las marcas del SnS son colocadas por la tarjeta adquisidora, con su propio hardware y no por el PC. Evidentemente el PC tiene mejor reloj que la tarjeta adquisidora.

La diferencia de relojes es de 0,00030076 s/s
Para fijar ideas, el tiempo para desfasarse un segundo es de aproximadamente 3.300 segundos, es decir casi 1 hora.

Luego de corregir el defasaje en frecuencia recalculamos el error. Vemos que el error aparente oscila entre 4 y -4 ms. Este es debido al efecto del *bit stuffing* que altera el largo de la trama en el canal.

Se observa un error hacia un lado si las tramas son pequeñas y hacia el otro en el caso de tramas largas. Para evitar el efecto del *bit stuffing* en el largo de las tramas, analizamos el error en las tramas pequeñas de 50bytes. Así el error se reduce a 1ms, es decir se encuentra acotado a un intervalo aproximadamente de (-1ms, 1ms). Esta es la exactitud que asegura el fabricante. A continuación presentamos los resultados exactos obtenidos, que muestran que la exactitud es un poco menor a 1ms.

El error estimado entonces para las marcas de tiempo por hardware es el siguiente:

Promedio del abs.(err.)	0,00051032	
Error min.		-0,00117166
Error máx.		0,00118011

Capitulo 8

ANÁLISIS DE COSTOS

8.1 Análisis de Costos del Analizador SnS

En el presente capítulo se realiza un análisis de costos del proyecto SnS. Cabe mencionar que se discriminan los costos de los componentes de hardware adquiridos en el mercado local con los adquiridos en el exterior, debido a que a estos últimos debe sumarse el costo de importación. No obstante se presentan tablas comparativas con los costos de los materiales involucrados, referidos a un mismo origen, para realizar una comparación objetiva del costo final del analizador SnS con respecto a algunos de los analizadores disponibles en el mercado.

Con respecto a los componentes utilizados en el equipo SnS que no fueron adquiridos durante el transcurso del proyecto, también se proveerá el valor actual de los mismos para poder realizar una estimación real del costo final del analizador.

NOTA: Los costos de los analizadores y materiales detallados en este capítulo fueron obtenidos a partir de consultas vía mail realizadas a fabricantes y representantes de las distintas marcas.

8.1.1 Materiales

A continuación se detallan los costos en origen de los componentes de hardware del analizador SnS.

COMPONENTES DEL ANALIZADOR SnS			
DESCRIPCIÓN	**MARCA**	**MODELO**	**COSTO ORIGEN**
Mother con procesador de bajo consumo.	VIA EPIA	5000 Fanless	USD 100,00
Gabinete con fuente.	Travla C150		USD 116,00
Memoria DIM 128MB.	Kingstone		USD 31,00
Disco Duro 10GB.	Western Digital		USD 40,00
Tarjeta WAN, PCI.	Sangoma	S5141	USD 570,00
		TOTAL:	**USD 857,00**

Tabla 13: Componentes del SnS

Los componentes de hardware que finalmente fue necesario importar se limitaron a la tarjeta Sangoma S5141, la *mother* para la mini PC descripta en el cuadro anterior y el gabinete de mini PC con fuente incluida. La tarjeta Sangoma fue importada por medio de un proveedor de ANTEL. Los demás componentes fueron adquiridos por medio de la empresa Case Outlet, en un paquete que consistía en el gabinete más la *mother*, lo cual implicó un costo final menor que si los mismos hubieran sido adquiridos individualmente.

COSTO FINAL EN ORIGEN: GABINETE + MOTHER	
DESCRIPCIÓN	**COSTO ORIGEN**
GABINETE + MOTHER	USD 205,00
SHIPPING COST (A Miami, EE.UU.)	USD 12,03
TOTAL :	**USD 217,03**

Tabla 14 : Costo en origen de gabinete y *mother*.

En el caso del analizador SnS no fue necesario adquirir los cables V.35 para transmisión de datos, dado que fueron provistos en calidad de préstamo por ANTEL. El precio de este tipo de cables en el mercado, al la fecha del inicio del proyecto, oscila entre USD 90,00 y USD 105,00.

8.1.2 Costos de Importación

El costo final de importación del paquete mencionado anteriormente, efectuada con la empresa Miami Box, implicó un costo final al proyecto de USD 448.04. El detalle se puede apreciar en la Tabla 13.

DETALLE DE IMPORTACIÓN DE GABINETE + MOTHER (MIAMI BOX)	
DESCRIPCIÓN	**COSTO**
COSTO DEL PAQUETE EN ORIGEN	USD 217,03
FLETE	USD 48,01
GASTOS IMPORTACIÓN	USD 157,74
VIMALCOR	USD 10,50
HONORARIOS DESPACHO	USD 14,76
COSTO TOTAL EN DESTINO :	**USD 448,04**

Tabla 15 : Importación de gabinete y mother

El costo final de importación de la tarjeta Sangoma S5141 con el proveedor de ANTEL representó un costo para el proyecto de USD 1150.00.

8.1.3 Costo del Analizador SnS

En este punto tratamos de dar un costo real del analizador SnS. Es claro que en el momento de reproducir el analizador deberá tenerse en cuenta las horas que consume a un técnico capacitado reproducir el equipo, estimamos unas 16hs para el armado del hardware, instalación de sistema operativo, *drivers* y la configuración de los mismos. También deben sumarse materiales menores para su armado como ser conectores, cables, etc.

En la Tabla 14 se detalla el costo de reproducir el equipo SnS en forma unitaria, si solamente se tienen en cuenta los costos de los materiales necesarios.

DETALLE DE COSTOS DE LOS COMPONENTES DEL EQUIPO SnS	
DESCRIPCIÓN	**COSTO**
TARJETA SANGOMA S5141	USD 1150,00
GABINETE + MOTHER	USD 450,00
DISCO DURO (10GB)	USD 40,00
MEMORIA DIM 128K	USD 31,00
CABLES V.35 (2 unidades)	USD 200,00
COSTO FINAL DEL EQUIPO SnS:	**USD 1871,00**

Tabla 16 : Costos de componentes del SnS

8.1.4 Comparación con Equipos Similares Existentes

En el cuadro siguiente se listan analizadores disponibles en el mercado. Fueron seleccionados aquellos que presentaban características similares al analizador SnS. Cabe destacar que tanto el analizador Fluke como el HP presentan mayores prestaciones que el equipo SnS.

El cálculo del costo de los analizadores en destino se realizó tomando en cuenta que el costo de importación de equipos electrónicos de este tipo, habiendo consultado a varias empresas de nuestro medio, se aproxima al valor en origen multiplicado por un factor de aproximadamente 2.5.

Finalmente se calculó el valor aproximado en destino, multiplicando el valor en origen por un factor de 2.1, siendo este factor el cobrado por la empresa con la que fueron importados los materiales del analizador SnS

OPCIONES DE ANALIZADORES EN EL MERCADO				
DESCRIPCIÓN	**MARCA**	**MODELO**	**COSTO ORIGEN**	**COSTO DESTINO APROX.**
Captura y transmite datos a 2,048Mbps. Portatil, Multiinterfaces incluida E1.	FE Test	ParaScope 2000	USD 8.995,00	USD 18.889,50
Conexión via PCMCIA	LOGIX	Mocha WAN Analizer	USD 3.000,00	USD 6.300,00
Conexión vía Ethernet	FLUKE	OptiView T1 E1 WAN Analizer	USD 5.995,00	USD 12.589,50
Agilent Network Analizer, multiinterface, interfaz de usuario incluida	Agilent (HP)	J6800A	USD 22.000,00	USD 46.200,00

Tabla 17 : Costos de analizadores del mercado

A continuación se muestra un gráfico comparativo de precios de analizadores (costo en origen) con características similares al equipo SnS.

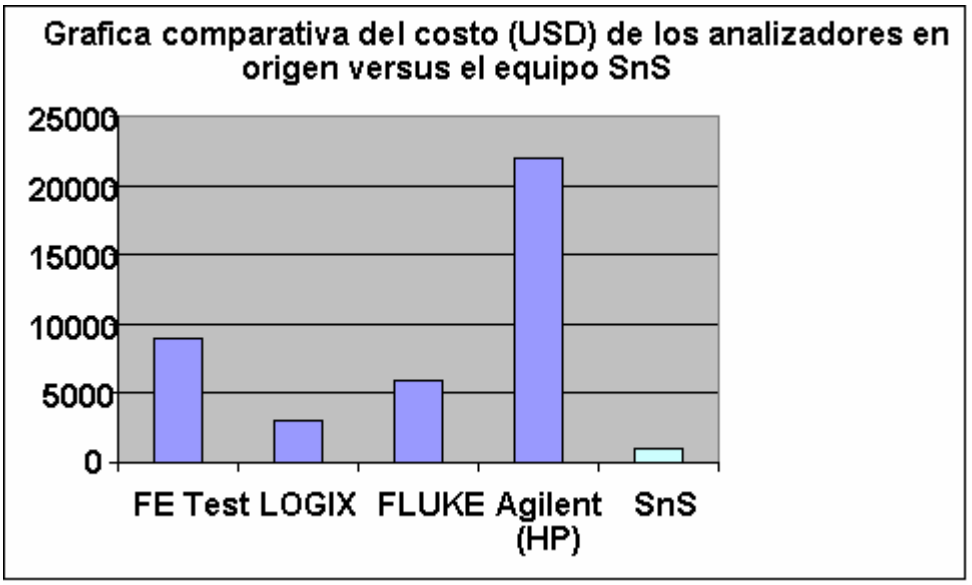


Figura 34: Grafico comparativo de costos

CONCLUSIONES

En el presente capítulo se resumen algunas de las consideraciones sobre el desarrollo del proyecto, el producto alcanzado y las posibles extensiones del mismo que avizoramos.

1.1 El desarrollo del proyecto.

El proyecto del analizador SnS surgió de modo de satisfacer el interés de Álvaro García, ingeniero por la contraparte de ANTEL, en poseer un analizador de bajo costo, portátil y fácilmente reproducible que sustituyera los analizadores de alto costo HP que son utilizados hoy en día en ANTEL.

Asumimos la responsabilidad de lograr un producto final y no solo un prototipo. Obtuvimos la confianza de la gerencia de ANTEL con una presentación formal del analizador que proyectamos.

En las primeras conversaciones con la contraparte de ANTEL se manejó un monto máximo de 1.000 USD. Finalmente el monto estipulado en un principio debió elevarse casi al doble al tener en cuenta los costos de importación del hardware necesario para el desarrollo del proyecto, debido a que en plaza no se cuenta con el equipamiento necesario. Esto derivó en volver a defender la propuesta ante la contraparte de ANTEL y la conveniencia del proyecto económicamente replanteado.

Para la realización del proyecto fue necesario efectuar dos importaciones. La importación de la mini PC y la importación de la tarjeta adquisidora. Aunque los costos de importación de la tarjeta corrieron por parte de ANTEL, debimos realizar la búsqueda de posibles empresas importadoras. La importación de la mini PC corrió enteramente por nuestra parte. Luego de analizar diferentes opciones de importación, se escogió una empresa del medio que se encargara de la compra, traslado y de los trámites correspondientes a la importación. Estas tareas finalmente insumieron más tiempo del que habíamos asignado en un principio en el plan de trabajo.

Debido a las necesidades de desarrollo durante el transcurso del proyecto debimos pasar de simples usuarios Linux a conocer las opciones que brinda un sistema operativo de código

abierto, como ser cargar módulos personalizados al Kernel de Linux y el manejo de *sockets.* Para la personalización del Kernel se contó con el apoyo en línea de quienes desarrollan el software de la tarjeta adquisidora Sangoma.

Inicialmente las marcas de tiempo fueron insertadas por hardware con precisión de un mili segundo, lo cual no era suficiente al trabajar con frecuencias de 2MHz. Para aumentar la precisión y gracias a trabajar sobre un sistema operativo de código abierto, se modificó el driver de la tarjeta de modo que sea el sistema operativo quien coloque la marca de tiempo, en el momento que atiende la interrupción que se genera cuando la tarjeta recibe una trama.

Durante su desarrollo, el proyecto SnS recorrió diferentes etapas, las cuales clasificamos en niveles de la siguiente forma.

Nivel físico: Se realizó el estudio y elección del hardware así como también el cable de conexión T.

Nivel Sistema Operativo: Estudio y desarrollo de módulos para el proyecto.

Nivel Aplicación: Desarrollo de una aplicación que manejara la los datos obtenidos.

Nivel Interfaz-Usuario: Enfoque y desarrollo de una interfaz amigable para el usuario.

Nivel Desarrollador-Cliente: Se interactuó con el cliente final, se defendió el proyecto y su presupuesto.

El hecho de haber recorrido los niveles antes mencionados deja en claro lo difícil que resulta abarcar todas las áreas en forma vertical.

Con la satisfacción de haber cumplido con las especificaciones técnicas acordadas y dejando la puerta abierta a futuras incorporaciones en cuanto a facilidades damos por concluido el proyecto SnS, analizador de protocolos para la interfaz V.35.

1.2 El producto final

El analizador SnS es un equipo portátil, capaz de adquirir datos de dos interfaces V.35 a una velocidad máxima de 2.048Mbps. Es un producto versátil, de costo sensiblemente menor al de los analizadores, con prestaciones similares, existentes en el mercado. La plataforma se basa en una mini PC con sistema operativo Linux con ambiente gráfico, de forma que permite visualizar el análisis de los datos en él.

A diferencia de los analizadores que dispone ANTEL, los cuales incorporan más de un tipo de interfaz, el SnS cuenta solamente con la interfaz V.35, logrando así reducir costos y dimensiones.

Dentro de los requerimientos realizados por el Ing. Álvaro García por la contraparte de ANTEL, se nos solicitó que fuera desplegada la frecuencia del reloj a la cual era realizada la captura. Se investigó en manuales y no se encontró la información buscada, lo que llevó a consultar a los fabricantes, quienes atendieron la consulta con mucho interés pero desconocían la solución al problema y se encuentran estudiando el caso.

El hecho de trabajar sobre una plataforma PC estándar x86 brindó múltiples facilidades a la hora de desarrollar aplicaciones.

1.3 Desarrollos a futuro.

1.3.1 Disminución del tamaño del archivo de captura

Si se pretende analizar una comunicación por varias horas y guardar todos los datos que se transfieran, se obtiene un archivo de captura de tamaño considerable.

Calculemos el tamaño del archivo vs. el tiempo de captura en horas. Una hora tiene 3.600 segundos. Por cada segundo, si el enlace a analizar es a 2.048Mbps y se utiliza a pleno, se transmiten aproximadamente 250Kbytes por segundo por canal. Por lo tanto entre ambos canales tenemos 0,5 MBytes por segundo. El archivo de captura aumenta entonces en 3.600 * 0,5 MBytes o sea a una tasa de 1.8 Gbyte por hora.

En muchos casos podría obtenerse un archivo mucho más pequeño que contenga todas las tramas que interesa analizar. Esto podría implementarse mediante filtros de captura, similar a los que utiliza *libpcap*. Cada trama capturada sería filtrada inspeccionando su protocolo y luego solo en caso de que corresponda sería guardada a disco.

1.3.2 Iniciar captura por eventos

Sería interesante también poder iniciar la captura luego de determinado evento, por ejemplo un error de CRC. Podría implementarse con un buffer circular en memoria RAM, en el que constantemente se guardarían las tramas capturadas. Al detectar una trama con error en CRC se guardaría a disco el contenido del buffer, por ejemplo de 100 tramas. Luego se continuaría adquiriendo datos hasta una condición de fin la cual podría ser un *timer*.

1.3.3 Sugerencias

Otro posible desarrollo sería lograr integrar la tarjeta Sangoma S5141 al Ethereal, de modo de poder iniciar y detener la captura en forma manual a través de la mencionada interfaz. Para ello se debe modificar la librería *libpcap*.

Álvaro García tiene interés en incorporar al analizador SnS la posibilidad de generar tráfico en forma personalizada, por ejemplo en cuanto al largo de las tramas

La funcion *do_gettimeofday()* es invocanda desde adentro de una interrupción, verificar si es una función reentrante para poder invocarla desde una interrupción en forma segura.

Para detener las capturas se escribe en un archivo una bandera “s”, la cual es leída, se pude mejorar para terminar utilizando el comando kill y manejar la señal *TERM* enviada.

.

ANEXO I

NORMAS REFERENTES A LA INTERFAZ V.35

1.1 Interfaz V.35

1.1.1 Introducción

La interfaz V.35 (ITU) es un modo estandarizado de conectar dos equipos, DCE y DTE, en forma sincrónica, digital, serial y modo *full duplex*.

Las velocidades de línea habituales son N x 64kbps, con N desde 1 (64kbps) a 32 (2048kbps) y debe ser utilizado un cable de interconexión corto.

En este anexo intentamos dejar en claro de que hablamos cuando se nombra la interfaz V.35.

1.1.2 Descripción

Este tipo de interfaz se emplea comúnmente para interconectar equipos a través de cables cortos (de longitud menor a los 10 m). Se trata de una interfaz digital binaria, esto implica que los símbolos transmitidos son dos, estado activo o inactivo. Es *full duplex*, en general simétrico, de modo que el *throwput* en un sentido es el mismo que en el sentido opuesto. Sincrónica, implica que continuamente se transmiten símbolos, a una cadencia que es conocida como la de reloj; en esta interfaz la información de elemento de tiempo de reloj se transmite en

general por el equipo DCE al DTE. Se transmiten dos relojes, uno para facilitar la recepción de datos por parte del DTE y otro (en general a la misma velocidad y sin defasaje) para que este también lo utilice para la transmisión. Habitualmente la velocidad de línea es de Nx64kbps, con N = 1 (64kbps) y N = 32 (2048kbps).

Esta interfaz se encuentra comúnmente en los servicios de acceso a Internet y en los circuitos digitales punto a punto como la interfaz entre la red de transporte y la red del usuario.

Ejemplificamos el empleo de este tipo de interfaz en el siguiente esquema típico de conexión a Internet de una Red de Usuario.

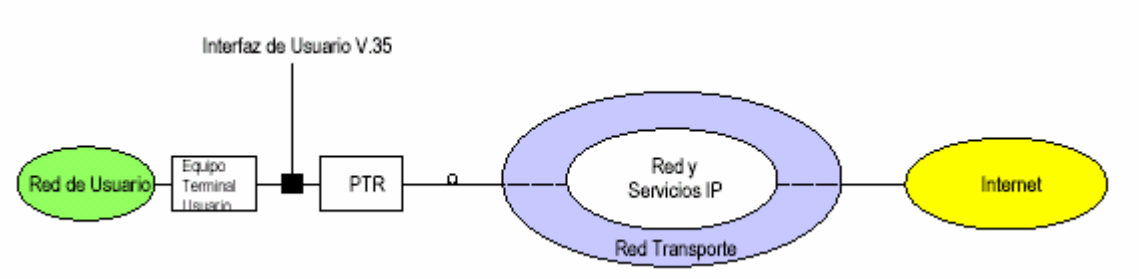


Figura 35: Esquema de Conexión V.35

El equipo encargado de la terminación de la red de transporte de la Administración (PTR en el diagrama) es un dispositivo ubicado en dependencias del cliente. Este equipo ofrece una interfaz V.35 hacia el cliente, visible en general como un conector M34 hembra. El dispositivo PTR es llamado DCE (Data Circuit Equipment). El equipo Terminal de Usuario, es llamado DTE (Data Terminal Equipment) y habitualmente es suplido junto con un cable que ofrece un conector M34 macho para realizar la conexión al DCE. De aquí en más seguiremos la nomenclatura del ITU-T y nos referiremos a los equipos que participan en la conexión como DCE y DTE

1.2 Recomendaciones y Normas que especifican la Interfaz

Es una interfaz totalmente estandarizada. A nivel eléctrico, mecánico y de funcionamiento se rige por recomendaciones de ITU-T (Sector de Normalización de las Telecomunicaciones de la Unión Internacional de las Telecomunicaciones), anteriormente llamada CCITT, e ISO. En cuanto a los protocolos de capa de enlace empleados hay libertad, pero se observa en general que están basados en el protocolo HDLC.

Las recomendaciones ITU involucradas son las V.35, V.28, V10, V11 que definen la parte eléctrica, V24 el funcionamiento y el estándar ISO 2593l la definición mecánica del conector.

La recomendación que da nombre a la interfaz es la Recomendación V.35 de ITU-T (10/84). La misma versa sobre la *"Transmisión de datos a 48 kbit/s por medio de circuitos en grupo primario de 60 a 108 kHz"*. Claramente esta recomendación no aplica completamente a lo que hoy día conocemos como interfaz V.35, dado que hace referencia a la transmisión de datos por circuitos de tipo telefónicos con una señal piloto de 104,08 KHz. De todas formas no restringe a la utilización de otro tipo de módems y se puede tomar en cuenta por ejemplo la especificación de los circuitos de enlace.

Número	Función
102	Tierra de señalización o retorno común
103 ∅	Transmisión de datos
104 ∅	Recepción de datos
105	Petición de transmitir
106	Preparado para transmitir
107	Aparato de datos preparado
109	Detector de señales de línea recibidas por el canal de datos
114 ∅	Temporización para los elementos de señal en la transmisión
115 ∅	Temporización para los elementos de señal en la recepción

Tabla 18: Especificación de Circuitos de enlace

Actualmente para la definición eléctrica de estos circuitos se remite a las normas V.10 y V.11.

1.3 Recomendaciones V.10 y V.11

La recomendación V.10 versa sobre los circuitos no balanceados, los cuales son empleados en la interfaz V.35 para las líneas de control.

La recomendación V.11 lo hace sobre los circuitos balanceados, que son empleados para la transmisión de datos y de elementos de tiempo (reloj).

La compatibilidad entre señales de tipo V.10 y de tipo V.11 se afirma en el anexo A de la V.11 en la sección Compatibilidad de circuitos de enlace conformes con las Recomendaciones V.10 y V.11 en una misma interfaz, "...Las características eléctricas de esta Recomendación han sido concebidas para permitir el uso de circuitos asimétricos (véase la Recomendación V.10) y simétricos en una misma interfaz. Por ejemplo, los circuitos simétricos pueden utilizarse para temporización y datos, mientras que los circuitos asimétricos pueden utilizarse para funciones de control asociadas de circuitos..."

La interfaz V.35 combina señales balanceadas y no balanceadas. Las líneas de control, por ejemplo DTR, DSR, DCD, RTS y CTS son señales no balanceadas, retornando por un único hilo de tierra común de señal, (Signal Ground). Estas señales prácticamente no varían, tienen poca frecuencia, dado que mientras los equipos se están transmitiendo datos estas señales de control permanecen constantes.

Las señales de datos y reloj son balanceadas, disponiéndose dos hilos (un par) para cada señal, estas señales tienen la frecuencia de la velocidad de la interfaz.

La recomendación UIT-T V.11 (10/96) define las características eléctricas de los circuitos de enlace simétricos de doble corriente que funcionan con velocidades binarias de hasta 10Mbit/s. Aquí define las señales utilizadas, que son de voltajes diferenciales. El generador y las cargas son de tal forma de que la interferencia con circuitos de enlace adyacentes sea mínima. Este tipo de circuitos de enlace permite alcanzar velocidades mayores que con un enlace no balanceado.

Los estados lógicos se definen por la diferencia de tensión entre los hilos del circuito de enlace. El hilo A es positivo al B, corresponde a el elemento binario 0 (trabajo) o el estado CERRADO para circuitos de control y de temporización. A la inversa para el elemento binario 1 (reposo) o el estado ABIERTO para circuitos de control y de temporización, el punto de salida A es negativo con respecto al punto B.

Especifica condiciones sobre el cable, el generador, las cargas. Define pruebas, etc.

Nos limitaremos unicamente a dar voltajes y corrientes de referencia. Por mayor información remitirse a las normas referidas.

1.3.1 Voltajes y Corrientes de Referencia

	Parámetro	Especificación
Requerimientos comunes	Maximun differential input voltage	12 V
	Maximun common mode input voltage	10 V
	Line Rate, internal clock	56Kbps a 2.048Kbps configurable
Señales balanceadas	Input sensitivity	< 200 mV
	Source impedance	< 100 ohms
	Output voltage Vo	1.10 Vpp +- 20%
	DC line offset	0.4 V max
	Output short circuit current	< 150 mA
	Output leakage current Iol	< 100 uA
Señales no balanceadas	Input sensitivity	< 200 mV
	Output drive	> 90 % de Vo dentro de 450 ohms

Tabla 19: Voltajes y Corrientes de Referencia V.35

1.4 Interfaz Física

l conector del DCE está estandarizado por la norma ISO 2593, también conocido como M34 Winchester. Al DCE (en general el punto de terminación de la red) le corresponde el conector hembra. El conector no tiene porqué estar en el mismo bloque físico que el DCE, puede estar distante, unido a través de un cable. Habitualmente el DTE (por ejemplo el *router* ubicado en la dependencia del cliente) viene provisto de un cable con un conector M34 macho para conectarse al DCE.

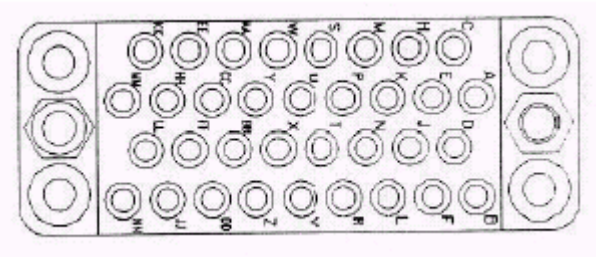


Figura 36: Conector M34

Esta claramente establecida la correspondencia entre los pines del conector y los circuitos de enlace. Se toma como referencia el conector hembra ubicado del lado del DCE. Se puede observar la tabla con tal referencia. Los números de circuitos son según la norma V.24.

1.4.1 Funcionamiento y descripción de los circuitos básicos

El significado de las señales en los circuitos de enlace esta especificado en la recomendación V.24. A continuación damos la condición básica para la transferencia de datos por el circuito de datos.

Para que el DTE transfiera datos por el circuito 103 (circuito balanceado de transmisión de datos), debe estar presente un estado CERRADO en los cuatro circuitos siguientes, si existen: circuito 105 (RTS petición para transmitir del DTE), circuito 106 (CTS Preparado para transmitir, del DCE), circuito 107(DSR Data Set Ready, Aparato de datos preparado, desde el DCE) y circuito 108/1 ó 108/2 (con el significado Conecte el aparato de datos a la línea en el caso 108/1 y en el caso Circuito 108/2 DTR, Data Terminal Ready, – Terminal de datos preparado)

Señal	PIN	Nombre	Número del circuito de enlace	Denominación del circuito de enlace	Tierra	Datos		Control		Temporización	
						Del DCE	Hacia el DCE	Del DCE	Hacia el DCE	Del DCE	Hacia el DCE
			1	2	3	4	5	6	7	8	9
Frame Ground	A	FG	101								
Signal Ground	B	SG	102	Tierra de señalización o retorno común	X						
			102a	Retorno común del DTE	X						
			102b	Retorno común del DCE	X						
			102c	Retorno común	X						
Send Data	P/S	SDA / SDB	103	Transmisión de datos			X				
Receive Data	R/T	RDA / RDB	104	Recepción de datos		X					
Request To Send	C	RTS	105	Petición de transmitir					X		
Clear To Send	D	CTS	106	Preparado para transmitir				X			
Data Set Ready	E	DSR	107	Aparato de datos preparado				X			
			108/1	Conecte el aparato de datos a la línea					X		
Data Terminal Rady	H	DTR	108/2	Terminal de datos preparado					X		
Received Line Signal Detected	F	RLSD	109	Detector de señales de línea recibidas por el canal de datos				X			
			110	Detector de la calidad de las señales de datos				X			
			111	Selector de velocidad binaria (DTE)					X		
			112	Selector de velocidad binaria (DCE)				X			
			113	Temporización para los elementos de señal en la transmisión (DTE)							X
Transmit Clock A / B	Y / A	TCA / TCB	114	Temporización para los elementos de señal en la transmisión (DCE)						X	
Receive Clock A / B	V / X	RCA / RCB	115	Temporización para los elementos de señal en la recepción (DCE)						X	
Remote Loopback	N		140	Conexión en bucle/Prueba de mantenimiento					X		
Local Loopback	L		141	Conexión en bucle local					X		
Test Indicators	NN		142	Indicador de prueba				X			

			191	Respuesta vocal transmitida					X			
			192	Respuesta vocal recibida				X				

Tabla 20: Funcionamiento y descripción de los circuitos básicos

1.4.2 Sobre los Cables

Las recomendaciones no especifican el cable a emplear, sino que da una orientación a las limitaciones operacionales del cable en función de sus parámetros, longitud, simetría y resistencia de terminación.

En la recomendación V.11 ejemplifican con un cable compuesto por cables telefónicos trenzados. De colocarse una terminación al circuito de enlace, este incrementa el largo máximo a una frecuencia alta dada. Por ejemplo a una frecuencia de 1Mbps puede llegar a 100 metros y sin terminación a 10 metros.

1.4.3 Sincronización y Reloj

En función de la necesidad la interfaz V.35 entregará o no al usuario un reloj de referencia. Si es necesaria la sincronización con la red de la Administración, el equipo de usuario se comportara como DTE y tomará el reloj del DCE.

1.4.4 Codificación

La interfaz V.35 soporta habitualmente el método de codificación NRZ y NRZI.

ANEXO II

PROTOCOLOS DE REFERENCIA

1.5 Protocolos de Capa de Enlace

En cuanto a los protocolos de capa de enlace, típicamente se emplea HDLC (High Level Data Link Control, protocolo estandarizado por ISO) u otro protocolo basado en él, como ser PPP (Point to Point Protocol, protocolo IETF) el cual es bastante generalizado.

1.5.1 Protocolo HDLC

HDLC es un protocolo de propósito general, provee una comunicación entre transmisor y receptor que puede ser de mejor esfuerzo o confiable, dependiendo del modo de HDLC que sea usado. El intercambio de información de datos y de control se realiza a través de tramas. Cada pieza de datos proporcionada por un usuario de HDLC, es encapsulada un en una trama HDLC, mediante el agregado de un encabezado y una cola. El encabezado consta de campos de dirección y de control. La cola contiene un CRC que detecta errores durante la transmisión. Las tramas están separadas por campos de delimitación.

Las secuencias delimitadoras constan de 8 bits, de la forma 01111110. Son transmitidos entre cada trama y también siempre que no haya datos para transmitir. En algunas implementaciones cuando no hay nada para transmitir, se deje la línea en reposo transmitiendo solamente el símbolo binario 1 (en caso de que este se corresponda con 0V).

Se previenen de la interpretación de datos a transmitir como secuencias delimitadoras con una técnica conocida como *0-bit insertion*, que es nada más que al momento de codificar los datos, insertar un 0 luego de una aparición de 5 elementos unos seguidos. Cuando se decodifica el contenido de la trama recibida, simplemente se descarta el siguiente *bit* a la aparición de 5 unos seguidos.

1.5.2 Protocolo PPP

PPP es un protocolo propuesto por el IETF y derivado de HDLC. Originalmente fue concebido para el encapsulamiento y transporte de tráfico IP sobre conexiones serie de naturaleza asíncrona y síncrona. PPP añade capacidades para la asignación y gestión de direcciones IP, multiplexación de protocolos de red, configuración del enlace, *tests* de calidad de línea, detección de errores, negociación y compresión.

1.5.3 Protocolo Frame Relay

Frame Relay consiste en un protocolo WAN que opera en Capa de Enlace de Datos del modelo de referencia OSI. Originalmente fue diseñado para utilizar a través de la interfaz ISDN (Integrated Services Digital Network) aunque hoy en día es utilizado en muchas otras interfaces de red.

Sus especificaciones fueron definidas por ANSI, fundamentalmente como medida para superar la lentitud de X.25, eliminando la función de los conmutadores, en cada "salto" de la red (control de errores y de flujo).

Frame Relay es un ejemplo de tecnología de conmutación de paquetes. En las redes que utilizan esta tecnología, las estaciones terminales comparten el medio de transmisión de la red de manera dinámica, así como el ancho de banda disponible. Los paquetes de longitud variable se utilizan en transferencias más eficientes y flexibles. Posteriormente, estos paquetes se conmutan entre los diferentes segmentos de la red hasta que llegan a su destino. Las técnicas de multiplexado estadístico controlan el acceso a la red en una red de conmutación de paquetes. La ventaja de esta técnica es que permite un uso más flexible y eficiente de ancho de banda. La mayoría de las LAN más aceptadas en la actualidad, como Ethernet y Token Ring, son redes de conmutación de paquetes.

Frame Relay normalmente opera a través de instalaciones WAN que ofrecen servicios de conexión. Como se dijo anteriormente, Frame Relay es estrictamente una arquitectura de Capa 2, en tanto que X.25 también proporciona servicios de Capa 3 (la capa de red). Frame Relay supera en desempeño y eficiencia la transmisión a X.25, y la mencionada tecnología resulta apropiada para las aplicaciones WAN actuales, como la interconexión LAN.

La gran ventaja de este protocolo radica en su sencillez y se puede emplear a velocidades de hasta 34Mbit/s. Otra ventaja no menos importante, es la capacidad de compartir el ancho de banda de forma dinámica, para la consolidación del tráfico, lo que lo hace económicamente muy atractivo frente al empleo de líneas arrendadas.

En el ámbito internacional, la tecnología Frame Relay fue estandarizada por la ITU-T (Unión Internacional de Telecomunicaciones, Sector Telecomunicaciones). En Estados Unidos, Frame Relay es un estándar de ANSI (Instituto Nacional Americano de Estándares).

1.5.4 Formato de la Trama Frame Relay

El formato de trama Frame Relay se muestra en la Figura 33.

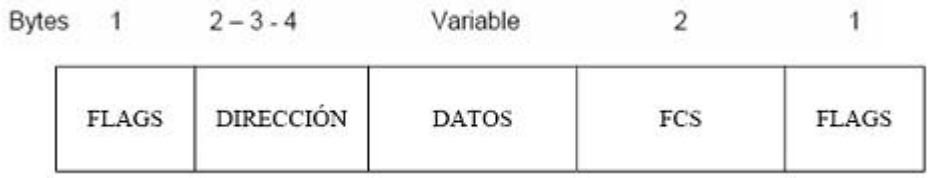


Figura 37: Trama Frame Relay

El campo **FLAGS** (1 byte) delimita el comienzo y el final de la trama. Su valor, 01111110, es el mismo que en las tramas LAP-B / HDLC.

A continuación del campo *flags*, están los 2, 3 ó 4 bytes del campo de **DIRECCIÓN.** Por defecto se usan 2 bytes, de los cuales los primeros 10 bits de estos dos bytes representan el ID del actual circuito, denominado el DLCI (Data Link Connection Identifier). Su significado es equivalente al campo número de circuito virtual en X.25 (LAP-B). El valor de 10 bits de DLCI (para el caso de dirección de 2 bytes) es el corazón de la cabecera Frame Relay. Identifica la conexión lógica que esta multiplexada en el canal físico. En el modo de direccionamiento básico (que es, no extendido por el LMI), los DLCI tienen significado local; o sea, los dispositivos extremos a dos diferentes extremos de una conexión pueden usar un DLCI para referirse a esa misma conexión. Como se comentó anteriormente, los DLCI permiten la multiplexación de varias conexiones lógicas de retransmisión de tramas a través de un único canal. Como en X.25, el DLCI tiene significado local: cada extremo de la conexión lógica asigna su propio DLCI de acuerdo con los números libres, debiendo realizar la red la conversión correspondiente entre ellos. Alternativamente, el uso del mismo DLCI por parte de ambos extremos requeriría algún tipo de gestión global de los valores de DLCI.

El campo **DATOS,** de longitud variable, consiste de un número entero de bytes. La red puede soportar un máximo tamaño de campo de información de 8.189 bytes, aunque generalmente se utilizan 1600 bytes. El gran tamaño de trama (comparado con los 128 de X.25) es necesario para prevenir el uso de segmentación y reensamblado en el equipo del usuario.

Finalmente se incorpora el código de control de errores de la trama (**FCS**: Frame Control Sequence) de 16 bits y el *flag* (01111110) de fin de trama.

1.5.5 Aplicaciones

Los dispositivos conectados a una WAN Frame Relay caen dentro de una de dos categorías generales:

1.5.6 DTE (Equipo Terminal de Datos)

Los DTE, en general, se consideran equipos de terminales de una red específica y, por lo general, se localizan en las instalaciones de un cliente. De hecho, pueden ser propiedad del cliente. Algunos ejemplos de los dispositivos DTE son computadoras personales, *routers* y puentes.

1.5.7 DCE (Equipo de Comunicación de Datos)

Los DCE son dispositivos para ínter conectividad de redes propiedad de la compañía de larga distancia. El propósito del equipo DCE es proporcionar los servicios de temporización y conmutación en una red, que son en realidad los dispositivos que transmiten datos a través de la WAN. En la mayoría de los casos, éstos son *switches* de paquetes.

La conexión entre un dispositivo DTE y un DCE consta de un componente de la capa física y otro de la capa de enlace de datos. El componente físico define las especificaciones mecánicas, eléctricas y de procedimiento para la conexión entre dispositivos. El componente de la capa de enlace de datos define el protocolo que establece la conexión entre el dispositivo DTE, que puede ser un *router* y el dispositivo DCE, que puede ser un *switch*

.

ANEXO III

EJEMPLO DEL ARCHIVO DE CONFIGURACIÓN DEL DRIVER

En el presente Anexo III se muestra un ejemplo de archivo de configuración del driver.

```
#================================================
#       WANPIPE1 Configuration File
#       Sangoma Technologies Inc.
#================================================

[devices]
wanpipe1 = WAN_CHDLC, Comment

[interfaces]
wp1chdlc = wanpipe1, , API, Comment

[wanpipe1]
CARD_TYPE     = S51X
S514CPU       = A
AUTO_PCISLOT          = YES
PCISLOT       = 0
PCIBUS        = 0
Firmware      = /etc/wanpipe/firmware/cdual514.sfm
CommPort      = PRI
Receive_Only  = YES
Connection    = Permanent
LineCoding    = NRZ
LineIdle = Flag
Interface     = V35
Clocking      = External
BaudRate      = 1540000
MTU           = 4098
UDPPORT       = 9000
TTL           = 255
IGNORE_FRONT_END  = NO

[wp1chdlc]
MULTICAST             = NO
```

```
IGNORE_DCD              = YES
IGNORE_CTS              = YES
IGNORE_KEEPALIVE    = YES
HDLC_STREAMING                  = YES
KEEPALIVE_TX_TIMER          = 10000
KEEPALIVE_RX_TIMER          = 11000
KEEPALIVE_ERR_MARGIN    = 5
SLARP_TIMER             = 0
TRUE_ENCODING_TYPE          = NO
```

NOTA: Para editar este archivo existe una herramienta GUI, que se corre desde consola, facilitando enormemente la elección de todos los parámetros, es muy aconsejable utilizarla.

ANEXO IV

LIBRERÍAS DE SANGOMA

1.1 API – Multiprotocolo

La capa wanpipe API, está diseñada para tomar los beneficios de la arquitectura de sockets de Linux. La Interfaz de programación de aplicaciones, API, representa un método para lograr abstracción en la programación, con las capas inferiores.

La configuración del driver en el modo API, es usado para recibir datos en forma cruda (*raw*), en donde la comunicación no tiene porque tener formato IP u otro protocolo de capa superior, por lo que la interfaz es configurada sin una dirección de red. Los paquetes son capturados, se agregan los campos de *time stamp* y son pasados al software de la aplicación.

Cuando el desarrollo es basado en las APIs, se tiene un *stack* multiprotocolo, desarrollado sobre el driver wanpipe HDLC. Los protocolos en los cuales se puede trabajar sobre el *driver* incluyen Frame Relay, X25/LAPB, entre otros. La arquitectura fue pensada para ir agregando protocolos uno encima del otro o trabajar en forma independiente.

Si se quisiera trabajar sobre X25, se puede implementar el *stack* de protocolos de diferentes maneras.

API --> X25 --> LAPB --> Frame Relay --> HDLC (adaptador)
ó
API --> X25 --> LAPB --> HDLC (adaptador)

Cuando se utiliza la arquitectura API en la tarjeta Sangoma, ésta deberá ser configurada en su driver en modo *raw* y especificar que va a trabajar con las API.

NOTA: Recordamos que para configurar el driver existe la herramienta wancfg_legacy, que se corre desde consola luego de instalado.

Las interfaces que utilizamos para tal propósito son las siguientes.

En el desarrollo del software de aplicación se toma como base el código ejemplo **chdlc_api.c** aportados por el fabricante.

El código se puede dividir en dos etapas bien diferenciadas, la primera consiste en crear el *socket* y la segunda en manejarlo, en esta última se extraen los datos y se procesan. Además es aportada una carpeta con librerías, dentro de estas se encuentran funciones de validación de parámetros, como lo son el nombre de interfaz, nombre dado a la tarjeta, entre otras.

1.1.1 Creación del Socket.

En esta etapa, se crea un *socket*, se reserva memoria y se asocia la memoria al *socket* mediante las funciones siguientes.

socket();
Crea un socket y devuelve un entero para identificarlo.

bind();
Asocia el socket con la dirección de memoria de la interfaz.

También se realiza una asociación de la configuración del driver con el *socket.*

1.1.2 Manejo del Socket

Básicamente en esta etapa se deja el *socket* a la espera para recibir un dato y cuando esto sucede, se realizan las tareas necesarias.

select (sock +1 ,&ready, &write, &excepcion,&time)

Esta función permite manejar el socket indicado de forma sencilla. A los parámetros de la función se le llaman descriptores, estos son:

1) el **soket+1** con el que se quiere trabajar.
2) recibir una señal si hay datos para leer
3) un descriptor si podemos pasar a escribir al socket
4) este informa si ha ocurrido una excepción
5) el último es utilizado para indicar cuanto tiempo se espera hasta que suceda algo.

Es necesario en nuestro caso estar listo para un suceso de leer dato desde el puerto, manejar la excepción nombrándola *oob* (Out of Band), y esperar indefinidamente hasta que suceda algo pasando el parámetro NULL.

select (sock + 1,&ready, &write, &oob, NULL)

Cuando **select()** sale, se deberá interrogar a todos los descriptores uno por uno con el macro **FD_ISSET**

FD_ISSET(sock,&oob)

Se verifica que no tenemos ninguna excepción

FD_ISSET(sock,&write)

No esta implementado en nuestro código pero se deja abierto para en un futuro poder escribir al *socket*.

FD_ISSET(sock,&ready)

Esta es la que nos interesa, una vez aquí debemos manejar los datos

recv(sock, Rx_data, MAX_RX_DATA, 0)

Una vez que la tarjeta pide una interrupción, la función *recv* carga los datos del *socket* en **Rx_data**, para luego trabajar con ellos.

1.1.3 Paquete de datos recibido

El paquete de datos recibido tiene la siguiente estructura:

Encabezado	***Datos***
16 bytes	X datos

Dentro del encabezado tenemos un campo **error_flag**, esta bandera puede tener los valores:

error_flag:

bit 0: la trama fue abortada
bit 1: la trama tiene error de CRC
bit 2: la trama tiene error de *overrun*

Para manejar estas banderas se implementa el siguiente código:

switch (.error_flag)

```
case 0:
        /* El paquete recibido es bueno*/
       break;
case RX_FRM_ABORT:
       /* Trama abortada*/
       break;
case RX_FRM_CRC_ERROR:
       /* Trama con error de CRC */
       break;
case RX_FRM_OVERRUN_ERROR:
       /* Trama con error de overrun */
       break;
 default:
        /* Queda para libre implementación*/
          break;
```

Dado esto, se puede ver que queda el código abierto para la futura implementación del manejo de los errores de CRC. Álvaro García, por parte de ANTEL, propuso como implementación a futuro, que dado un error de CRC el SnS comenzara la captura.

Todo el código de manejo del *socket* se repite en un bucle infinito hasta que se cumple alguna condición de fin.

Antes de terminar la aplicación se debe cerrar el *socket*. La siguiente instrucción lo realiza.

close (sock)

ANEXO V

SOFTWARE ETHEREAL

En el presente anexo se describe el software analizador Ethereal y los formatos de archivos de capturas utilizados.

1.1 Ethereal

Ethereal es un software analizador de protocolos de red y tiene licencia de código abierto de acuerdo con la licencia GNU (General Public License). Es ampliamente utilizado, tanto en el ambiente profesional como en el educativo para solucionar problemas, analizar y desarrollar protocolos. Tiene una interfaz gráfica amigable y la capacidad de analizar una enorme variedad de protocolos (más de 750). Incluso la variedad de los tipos de capa de enlace es muy variada, siendo el uso más común el de Ethernet, pero también es capaz de analizar como en nuestro caso Frame Relay, PPP o CHDLC.

Ethereal permite capturar los datos directamente de la red o leer a partir de un archivo con los datos capturados.

1.1.1 Captura directa de la red.

Las interfaces de red de las cuales puede capturar datos están limitadas entre otras cosas por el sistema operativo de la PC en la cual corre, el hardware, los *drivers* que estén utilizando y la versión de *libpcap/WinPcap* que tenga instalada.

El analizador SnS utiliza la tarjeta S5141 de Sangoma para adquirir los datos de las interfaces de tipo V.35. Las versiones disponibles de *libpcap* reconocen la interfaz como interna al sistema operativo y le presenta los datos al Ethereal con un encabezado del sistema operativo. Además de ese inconveniente, únicamente permite capturar datos provenientes de un puerto a la vez siendo que en nuestro caso debemos inspeccionar ambos a la vez, debimos crear una aplicación que capture los datos y cree un archivo para que luego Ethereal lo analice.

1.1.2 Los formatos de archivos de captura

Definen la estructura de los archivos que contienen los datos capturados de las interfaces de forma de brindar información extra. Es de interés conocer la interfaz de la qué fueron tomados los datos, el tipo de capa de enlace, y en que momento se transmitieron, esto último es conocido como *time stamps*.

Los datos, se pueden almacenar físicamente en distinta forma según la arquitectura del sistema. En función del orden en que se escriben las palabras, por ejemplo si el byte menos significativo de un número de 32 bits se encuentra al principio o al final, se denomina *little endian* o *big endian*.

"Little Endian" significa que el byte de menor peso se almacena en la dirección más baja de memoria y el byte de mayor peso en la más alta. El *"Big Endian"* lo hace a la inversa.

NOTA: El protocolo TCP usa el formato *"Big Endian"*, por lo que los sistemas que usan *"Little Endian"* deben convertir los datos al crear los paquetes TCP/IP.

Para identificar si el archivo que analizamos está en *Little Endian* o *Big Endian* y poder interpretar de forma correcta los datos guardados, surge el número mágico. Además el número mágico es utilizado por Ethereal para reconocer el tipo de archivo.

1.1.3 Análisis de un archivo de captura.

Ethereal es capaz de abrir archivos de capturas de un gran número de programas analizadores, o sea un gran número de formatos de archivos.

Para nuestro propósito era necesario guardar en un mismo archivo de captura, las tramas correspondientes originadas por el DTE y el DCE, y agregar cierta información que permita luego identificarla el sentido, ya que a diferencia de otro tipo de capa de enlace, en HDLC en su modalidad de punto a punto, no es necesario que las tramas intercambiadas porten información de dirección que identifique al equipo que las origina, sin esta información en la trama se tiene que utilizar alguna bandera dentro de la información del archivo para que el analizador reconozca el sentido.

1.2 Selección del formato de archivo de captura

Como mencionamos anteriormente, Ethereal es capaz de trabajar con archivos de capturas de diferentes formatos y por supuesto de diferentes protocolos de capa de enlace, de red y demás capas superiores.

El formato de archivo "nativo" es el *libpcap*. Este es el formato de archivo de captura más difundido. Además de ser utilizado por el programa Ethereal, es utilizado por el programa *Tcpdump,* comúnmente encontrado en las distribuciones de los sistemas operativos Linux.

El formato *libpcap,* soportado por las versiones al momento de Ethereal, no permite la distinción en un mismo archivo de paquetes provenientes de distintas interfaces, por lo que no es posible distinguir entre una trama originada por el DCE y una originada por el DTE. Al momento, el grupo de desarrollo de Ethereal trabaja en una extensión del formato *libpcap,* lo que permitirá entra otras cosas discriminar dentro de un mismo archivo a que interfaz corresponde cada trama.

Así es que se trabajó con otros formatos de archivos que permitan tal discriminación y a su vez mantengan al menos una precisión en las marcas de tiempo de microsegundos.

El único formato capaz de ello al momento es el Network Associates Sniffer (DOS-based). Puede indicar los tipos de capa de enlace, Frame Relay y CHDLC (luego Ethereal distingue si se trata de CHDLC o PPP) y tiene precisión en las *time stamps* de un microsegundo.

1.3 Formatos PCAP y Network Associates Sniffer

A continuación describimos ambos formatos, el PCAP debido a su amplio uso y a que se guardan los archivos intermedios, de las capturas de ambos puertos y el Network Associates Sniffer, que es el formato en que guardamos la captura de datos en el archivo final.

1.3.1 Formato PCAP

Este archivo consta de un encabezado general (encabezado de archivo). Luego cada trama o paquete capturado tiene un encabezado propio (encabezado de paquete) seguido por los datos del paquete correspondiente.

1.3.2 Encabezado de Archivo

Se compone de siete campos, ocupando un total de 24 bytes

- Nº mágico: 4 bytes.
- Versión mayor: 2 bytes (puede estar invertida)
- Versión menor: 2 bytes (puede estar invertida)
- Zona horaria: 4 bytes
- Cifras significativas: 4 bytes
- Largo máximo del paquete: 4 bytes
- Tipo de capa de enlace: 4 bytes

1.3.3 Encabezado de Paquete

Se compone de cuatro campos, ocupando un total de 16 bytes.

Marca de tiempo
- Segundos: 4 bytes
- Fracción de segundos: 4 bytes
- Tamaño de paquete capturado: 4 bytes
- Tamaño de paquete guardado: 4 bytes

1.3.4 Campos de Encabezado de Archivos

Número Mágico

Consta de cuatro bytes con el siguiente valor, en hexadecimal: A1 B2 C3 D4

HEXA	BINARIO
A1 B2 C3 D4	10100001101100101100001111010100 Litlle Hendían
D4 C3 B2 A1	11010100110000111011001010100001 Big Endian

Varía según la arquitectura del sistema pudiendo encontrarse en orden inverso

Versión Mayor

Indica la última versión conocida de *libpcap* para la cual es compatible el formato. Consta de dos bytes. 0x0004 es el valor correspondiente a capturas realizadas con la versión actual de libpcap

Versión Menor

Indica la primera versión de *libpcap* compatible el formato de archivo. Consta de dos bytes. 0x0002 es el valor correspondiente a capturas realizadas con la versión actual de *libpcap*

Zona Horaria

Puede ser de interés analizar en el mismo momento una comunicación en distintas partes del mundo, muy probablemente con distintas zonas horarias. Para comparar tiempos, debemos referir en las time stamps a un instante dado, absoluto. El tiempo absoluto entonces lo expresamos con el tiempo local, en la marca de tiempo y un *offset* según la región donde se haya realizado la captura, que es la zona horaria. El valor de zona horaria son las horas de diferencia entre la hora local y la hora en la línea del meridiano de Greenwitch.

Consta de cuatro bytes. En nuestro ejemplo, no se utiliza. Tenemos entonces, 0x 00000000 (4 bytes)

Cifras Significativas

Expresa la precisión de las marcas de tiempo. Las marcas de tiempo se guardan como dos campos. El primero: los segundos después del 1/1/1970. El segundo campo: es la cantidad de unidades de tiempo según el valor de las cifras significativas que debe sumarse a los segundos. Es decir, si las cifras significativas indican precisión de micro segundos, entonces el segundo campo indica la cantidad de microsegundos que deben adicionarse a los segundos para obtener el tiempo en que se capturó la trama. Si este campo se omite, se interpreta con precisión de milisegundos. Consta de cuatro bytes.

Largo Máximo de Paquete

Los datos correspondientes a paquetes guardados en el archivo de captura nunca pueden ser de más bytes que el valor de largo máximo de paquete. Si el paquete transmitido por la red, supera el largo máximo, este no es guardado totalmente, si no que se eliminan los últimos bytes de modo de que la cantidad de bytes guardada no supere el largo máximo de paquete. Consta de cuatro bytes.

Capa de Enlace

Indica el tipo de capa de enlace de la cual se capturaron los datos, Ethernet, PPP en HDLC, etc. Esta información, además de su valor en si misma, permite al analizador de protocolos comenzar a "direccionar" la trama capturada adecuadamente. Es decir, conociendo el tipo de capa de enlace, puede comenzar a interpretar los bits capturados, analizando los bits involucrados en el protocolo de enlace. El protocolo de enlace, en general indica que protocolo encapsula, permitiendo entonces continuar "ascendiendo" en los niveles de protocolos.

1.3.5 Campos del Encabezado de Paquete

Cada paquete capturado tiene su propio encabezado, con información concerniente a cada paquete como ser marca de tiempo, cifras significativas, tamaño de paquete real y tamaño de paquete guardado

Marca de Tiempo

Indica el tiempo en que el paquete fue tomado por la interfaz. Este tiempo lo proporciona el hardware adquisidor o el sistema operativo cuando el procesador atiende interrupción generada por ese puerto de entrada al tener un nuevo paquete. La primera forma es exacta y la precisión de la segunda esta determinada por la potencia del procesador, las instrucciones que necesite para obtener el tiempo y la carga de otros procesos que estén corriendo.

En el caso general del ethereal, que es capturar paquetes a través de una interfaz común de red, la marca de tiempo se pone por software.

La marca de tiempo se divide en dos campos. El primero corresponde a los segundos, siguiendo el formato de tiempo GNU Linux, se toman los segundos posteriores al primero de enero de 1970. El segundo campo es la cantidad de unidades de tiempo del valor de las cifras significativas que se hayan especificado en el encabezado de archivo que deben adicionarse al primer campo de segundos.

La marcha de tiempo se interpreta como la suma en tiempo de ambos campos.

(1) Segundos
Segundos absolutos (tiene compensación zona horaria) posteriores al 1º de enero de 1970 (año de referencia utilizado por GNU) respecto a Grenwitch.

Tipo de campo: unsigned long de 4 bytes. En el ejemplo: 0x 41 B3 C1 6A corresponde a 5/12/2004 6:18 p.m. Interpretado como unsigned long es la cantidad de segundos 1102299498. La fecha se obtiene de la siguiente manera: dividimos este número entre la cantidad de segundos que tiene un año. Sumamos a 1970 y obtenemos el año 2004, en el cual se realizo la captura. Los segundos restantes nos terminan de indicar la fecha que es 05/12/2004 6:18 p.m.

(2) Fracción de Segundo

Tipo de campo: unsigned long de 4 bytes. Es la cantidad de unidades de tiempo según lo expresado en las cifras significativas que debe sumarse a los segundos.

Tamaño de Paquete Capturado

Tipo de campo: unsigned long de 4 bytes. Es el tamaño del paquete en la red.

Tamaño paquete guardado.

Tipo de campo: unsigned long de 4 bytes. Es la cantidad de bytes del paquete que han sido guardados en el archivo. En ocasiones, para reducir el tamaño del archivo de captura, se guarda únicamente el comienzo del paquete, que es donde en general están los encabezados de los distintas PDU, que encapsuladas una en otra, conforman el paquete capturado.

1.4 Ejemplo de archivo de captura

Ejemplificamos el formato *pcap* con un archivo real. Se trata de una captura a través de una interfaz Ethernet de una PC conectada a Internet a través de un módem Adsl. El archivo es de formato *libpcap*, más precisamente Ethereal lo identifica como de formato RedHat Linux 6.1 *libpcap* (*Tcpdump*).

1.4.1 Contenido del archivo (expresado en hexadecimal)

```
00000000  D4C3 B2A1 0200 0400 0000 0000 0000 0000 FFFF 0000
00000014  0100 0000 5D90 3B43 4AF7 0B00 3C00 0000 3C00 0000
00000028  00D0 090C 4818 0007 84ED D438 8864 1100 5375 000E
0000003C  C021 0920 000C D7E8 FF9B 46C7 7A84 0000 0000 0000
00000050  0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
00000064  5D90 3B43 2D45 0C00 2200 0000 2200 0000 0007 84ED
00000078  D438 00D0 090C 4818 8864 1100 5375 000E C021 0A20
0000008C  000C 46C7 7A84 46C7 7A84 6890 3B43 707C 0000 3C00
000000A0  0000 3C00 0000 00D0 090C 4818 0007 84ED D438 8864
000000B4  1100 5375 000E C021 0921 000C D7E8 FF9B 46C7 7A84
000000C8  0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
000000DC  0000 0000 0000 6890 3B43 23DB 0000 4D00 0000 4D00
000000F0  0000 0007 84ED D438 00D0 090C 4818 8864 1100 5375
00000104  0039 0021 4500 0037 0236 0000 8011 4349 C9D9 863F
00000118  C828 DCF5 042E 0035 0023 071C B2CF 0100 0001 0000
0000012C  0000 0000 0579 6168 6F6F 0363 6F6D 0000 0100 0168
```

1.4.2 Significado de los datos

D4C3	B2A1	0200	0400	0000	0000	0000	0000	HF	0000
número mágico		versión menor	versión mayor	zona horaria		cifras significativas		largo máximo del paquete	
0100	0000	5D90	3B43	4AF7	0B00	3C00	0000	3C00	0000
tipo de enlace		segundos		fracción de segundos		tamaño paquete capturado		tamaño paquete guardado	
Continúa el archivo con los datos de un paquete, un encabezado de paquete, datos de paquete, encabezado de paquete, datos de paquete, etc.									

Tabla 21: Encabezado libpcap

Interpretando en Little Endian (Intel)
Como signed o unsigned de 4 bits
Segundos: 5D 90 3B 43 = 1127977053
Fracción de segundos (en microsegundos): 784202

El tercer encabezado de paquete, muestra una marca de tiempo de
Segundos: 68903B43 =1127977064
Fracción: 707C0000 = 31856

La diferencia con el primero es: 1127977064,031856 – 1127977053,784202 = 10,247654

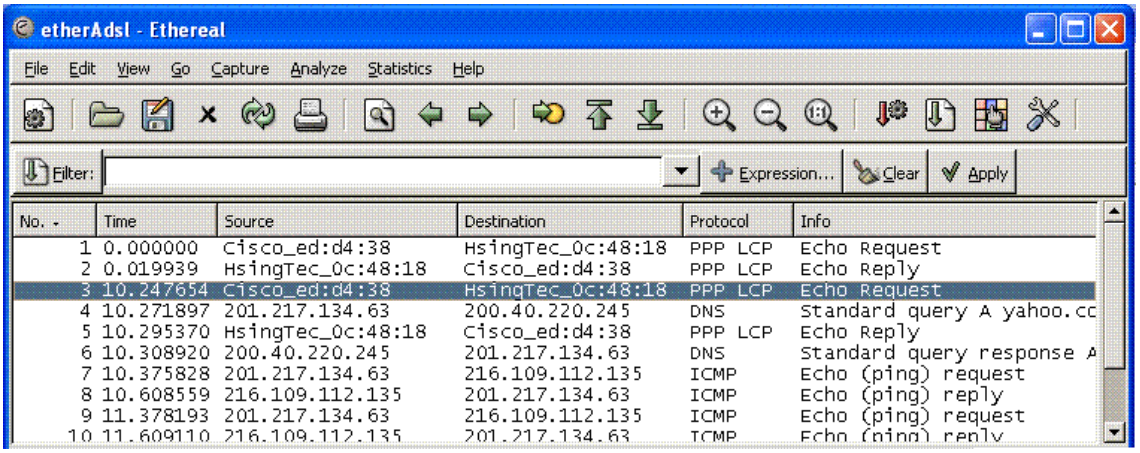


Figura 38: Imagen de análisis Ethereal

Imagen de análisis con Ethereal, en el que se muestra en el campo Time la diferencia de tiempo con el primer paquete capturado. Confírmese la forma de calcular el tiempo con lo mostrado por Ethereal.

1.5 Network Associates Sniffer (DOS-based)

Este formato de archivo permite incorporar a las tramas capturadas la información del sentido de las mismas, es decir, si son originadas por el DTE o por el DCE. Esto se logra mediante el campo (1 byte) frame error status que es uno de los campos del encabezado de trama FRAME2_REC. El valor hexadecimal "80" corresponde a una trama originada por el DTE y el valor 0x00 a una originadaza por el DCE.

Para que Ethereal decodifique correctamente los datos, es necesario indicarle el tipo de capa de enlace. (Aunque para algunos protocolos no es necesario ya que utiliza un método heurístico, si lo es para el caso de Frame Relay en el que se basa en la información de los encabezados extras a la captura). Más precisamente se indica la capa de enlace mediante el campo "major version" del encabezado de archivo **VERSION_REC** de la siguiente manera: El valor 0x03 corresponde a Frame Relay y el valor 0x04 corresponde a PPP y CHDLC.

Para obtener el formato adecuado de archivo Network Associates Sniffer nos basamos en el código fuente de Ethereal y en el manual del analizador Network Associates Sniffer (DOS-based).

El código correspondiente a la decodificación por parte de Ethereal de este tipo de archivos de captura se encuentra en el archivo *ngsniffer.c* en la carpeta *wiretap* de las fuentes del Ethereal.

NOTA: Para que el Ethereal muestre en pantalla el sentido de las tramas, es necesario agregar dos columnas a la interfaz de *Hardware src addr* y *Hardware dest addr (esto es realizado a nivel de usuario de ethereal y se explica como hacerlo en el manual de usuario)*

1.5.1 Formato

De la misma forma que el archivo *pcap*, este formato de archivos coloca cierta información general al comienzo del archivo, luego información particular a una trama capturada y los datos capturados de esa trama. Luego continúa con la información de otra trama y los datos capturados de esa trama y así sucesivamente. Este formato incluye una sección de fin de archivo.

1.5.2 Componentes del archivo.

1. Número mágico – Igual significado que en *pcap*.
2. Record header – Informa que tipo de encabezado se encuentra a continuación. En este caso avisa que sigue un "Version_rec"
3. Version_rec – Ethereal obtiene de aquí el tipo de capa de enlace y la fecha en que comenzó la captura.
4. Record header – Informa que continua un encabezado de tipo "Rec_header"

5. Rec_header – Encabezado con datos propios de la trama. En nuestro caso es del tipo Frame2_rec.
6. Datos capturados (un paquete) – Los datos capturados (sin las banderas delimitadoras de tramas y con el "bit-stuffing" (bits de relleno)
7. Para un siguiente paquete retornamos a 5, y así sucesivamente hasta el último paquete.
8. Cierre de archivo

1.5.3 Descripción de cada componente del archivo.

a) Número Mágico

Consiste en el siguiente valor fijo.
'T', 'R', 'S', 'N', 'I', 'F', 'F', ' ', 'd', 'a', 't', 'a', ' ', ' ', ' ', ' ', 0x1a}
En hexadecimal corresponde: 54 52 53 4E 49 46 46 20 64 61 74 61 20 20 20 20 1A

b) Record_header

Tipo de registro 2 bytes

Ejemplo: 0100 indica que es un registro tipo vers_rec

Largo de registro 4 bytes

c) Version_rec

```
struct vers_rec {
        gint16  maj_vers;       /* version mayor */                     0400
        gint16  min_vers;       /* version menor */                     0000
        gint16  time;           /* tiempo en formato DOS*/              0000
        gint16  date;           /* tiempo en formato DOS */ 9FEB
        gint8   type;           /* tipo del siguiente registro */       04
        guint8  network;        /* tipo de red*/                        07
        gint8   format;/* version del formato*/                 01
        guint8  timeunit;       /* unidad de marca de tiempo*/          01
        gint8   cmprs_vers;     /* version de la compresión */          00
        gint8   cmprs_level;    /* nivel de compresión */               00
        gint16  rsvd[2];        /* reservado */                 0000
};
```

d) Record_header

(Este corresponde al paquete capturado, previo a los datos)
Tipo de registro (2 bytes)
0400 indica que es un registro tipo frame2_rec

Largo de registro (4 bytes)

Formas de ingresarlo:

```
buf[0] = REC_FRAME2;
buf[1] = 0x00;
buf[2] = (char)((largoPaqueteCapturado + largoEstructura Frame2_rec) %256);
buf[3] = (char)(( largoPaqueteCapturado + largoEstructura Frame2_rec) /256);
buf[4] = 0x00;
buf[5] = 0x00;
```

e) Frame2_rec

time_low (2 bytes)
time_med (2 bytes)
time_hight (2 bytes)
size (2 bytes)
fs (1 byte) frame error status, pero lo utilizamos para el sentido de tramas. Los valores hexa "80" o "00" setean el sentido.
Flags (1 byte)
True- size (2 bytes)
Reserved (2 bytes)

Para pasar las marcas de tiempo de formato *pcap* a formato Network Associates Sniffer se utiliza el siguiente algoritmo.

```
t = (double) timestamp_Seg + timestamp_uSeg /1.0e9
t = (t – «0_HorasDelDiaComienzoCapt»)*1.0e6 / 0.838096
t_low = (unsigned short)(t-(double)((unsigned long32)(t/65536.0))*65536.0)
t_med = (unsigned short)((unsigned long)(t/65536.0) % 65536)
t_high = (unsigned short)(t/4294967296.0)
```

f) Datos

Acá se agregan los datos de un solo paquete.
Para volver a poner otro paquete se agregan los encabezados 4 y 5

g) Cierre de archivo

Son 6 bytes: "REC_EOF", 00, 00, 00, 00, 00.

El valor REC_EOF se define en 03

ANEXO VI

KERNEL Y MÓDULOS DE LINUX

En un sistema Linux la interacción final con dispositivos la realizan los controladores o el Kernel. Un dispositivo sólo podrá ser utilizado si el mismo es soportado por el Kernel o si existe un controlador para el dispositivo, además debe ser configurado apropiadamente. Las fuentes en C de cada versión del Kernel cuentan con controladores para diversos tipos de dispositivos. Cuando se compila una versión, algunos de esos controladores pueden unirse con el Kernel mismo (estáticamente), algunos pueden dejarse como módulos para cargarse o descargarse cuando la parte estática del Kernel este operando, otros pueden ser excluidos del proceso de compilación y por lo tanto no podrán ser usados ni aún cuando el Kernel esté operando.

1.1 Módulos

En Linux, un módulo es un conjunto de rutinas que realizan funciones a nivel de sistema y que pueden cargarse y descargarse dinámicamente desde el núcleo cuando sea requerido. Los módulos con frecuencia contienen controladores de dispositivos y están fuertemente ligados a la versión del núcleo. La mayoría de los módulos construidos con una versión específica del núcleo no podrán ser cargados de forma apropiada en un sistema que corra un núcleo cuya versión sea distinta.

ANEXO VII

CONTENIDO DEL CD

Aplicación SnS

- Código Fuente interfaz para Windows

Código Fuente aplicación ***sns***

Datos capturados de prueba de laboratorio

Documentación

Herramientas:

- Windows
  - putty.exe - Cliente ssh. Permite abrir una consola en un PC con servidor *ssh*
  - pscp.exe - Permite copiar desde un archivo, mediante *ssh*
  - plink.exe – Ejecuta comandos en la PC con servidor *ssh*
  - SSHSecureShellClient-3.2.9.exe – Paquete completo, para abrir consola en PC con servidor *ssh* y además transferir archivos. Su ambiente gráfico es amigable.
  - Tightvnc – cliente *vnc*
  - Aplicación SnS
  - Ethereal

Linux

- Driver Sangoma
  Librerías necesarias – para Mandrake 10.1 por ejemplo *flex* para compilar el *driver*.

Manuales:

- Manual de Usuario

ANEXO VIII

INSTALACIÓN DEL SISTEMA OPERATIVO

1.1 Instalación Mandrake 10.1

1.1.1 Pasos a seguir para la correcta instalación

1. Se deberá *bootear* desde el CD 1 de instalación de la versión *10.1* de *Mandrake Linux*.
2. Se sugiere elegir la opción *español* en idioma.
3. Aceptar los términos de *Mandrake 10.1*.
4. *Clickear* la opción *Instalar*.
5. Se deberá seleccionar el nivel de seguridad estándar.
6. En la opción de *Particionamiento* seleccionar personalizado.
7. A continuación se deberá seleccionar el punto de *Montaje* **/home**.
   Ejemplo: seleccionar 27% / y 71% **home**.
   Sugerencia: asignar el doble de *swap* de RAM
   Clickear Hecho
8. Se deberá formatear las particiones.
9. Instalar a continuación los CD1, CD 2 y CD3.
10. En *Opciones* se deberá proceder a la *Selección de grupos de paquetes:*
    Estación de Trabajo: (paquetes a seleccionar)
    - ✓ Estación de trabajo Oficina (Opcional)
    - ✓ Estación Internet (Opcional)
    - ✓ Computadora de red (Cliente)
    - ✓ Configuración
    - ✓ Herramientas para consola
    - ✓ Desarrollo

    Servidor:
    - ✓ Servidor, contrafuegos, router
    - ✓ Computadora/Servidor de red

    Entorno Gráfico:
    - ✓ Estación de trabajo *GNOME*
11. Se deberá instalar los paquetes a medida que se vayan solicitando.
12. Elegir la contraseña para *root*.

13. Seleccionar *usuario* y *contraseña.*
14. Continuar la instalación con las opciones que vienen seleccionadas por defecto.
15. En la opción de configuración de la red LAN, el usuario deberá seleccionar:
    - ✓ Conexión a la red local
    - ✓ Configurar IP de modo manual:
        - o Elección de la dirección IP (192.168.0.11)
        - o Máscara de red (255.255.255.0)
    - ✓ Definir el nombre de la máquina (*fcr sniffer)*
16. No aceptar la opción *descargar actualizaciones.*
17. Finalmente para concluir la instalación se deberá reiniciar la PC.

1.1.2 Instalación de la fuente del Kernel

El usuario deberá ir a *Inicio? Sistema? Configuración? Empaquetado? Instalar Software*
Ventana: Instalación de paquetes de software
Seleccionar: **Kernel –source-2.6-2.6.8.1-12 mdk**
Luego deberá aceptar la opción *Instalar.*

1.1.3 Instalación Flex

Podrá accederse en la carpeta *Drivers*.

Rpm-i flex .x.x.x.rpm

ANEXO IX

INSTALACIÓN DE LOS DRIVERS

Ir a la carpeta *Utilitarios? Drivers* y copiar el directorio al **home**.

Se deberán ejecutar los comandos:

cp wanpipe x.x.x.x.tgz /home/frcsniffer/
cd/home/frcsniffer
tar xvfz wanpipe. x.x.x.x.tgz (comando para descomprimir)
cd wanpipe
./setup install

Sugerencia: correr el comando como *root*

Continuar con las opciones de instalación.

ANEXO X

PINOUT CABLE T

Los dos puertos de la tarjeta están dispuestos sobre un conector DB37. Descripción del *pinout* de la tarjeta Sangoma S5141 y las señales que maneja son resumidas en la siguiente tabla.

PIN	PUERTO 1	PUERTO 2	PIN V35
1	RTS		
2	CTS		C
3	GND	GND	B
4	DCD		H
10		RTS	
11		CTS	D
12		DCD	F
18	TXB		
19	TXA		
20	RXB		S
21	RXA		P
22	TX Ck A		
23	TX Ck B		
24	RX Ck A		Y
25	RX Ck A		AA
28		TXB	
29		TXA	
30		RXB	T
31		RXA	R
32		TX Ck A	
33		TX Ck B	
34		RX Ck A	V
35		RX Ck B	X

Para nuestro diseño utilizamos los pines de recepción de cada puerto.

Para las señales de datos y reloj utilizamos los pines RX y RX Ck en ambos puertos. La configuración del *driver* permite ignorar las señales de control para la recepción de datos.

Los pines de control son utilizados para verificar el estado de las señales CTS, RTS, DTR Y DCD.

GLOSARIO

ANTEL - Administración Nacional de Telecomunicaciones. Es el nombre que recibe la compañía estatal uruguaya de telecomunicaciones.

API - (Application Programming Interface) Interfaz de Programación de Aplicaciones, interfaz de programación de la aplicación) es un conjunto de especificaciones de comunicación entre componentes software.

DTE – (Data Terminal Equipment). Equipo terminal de datos, nombre que suele recibir en una comunicación el ordenador que recibe o envía los datos.

DCE - (Data Communication Equipment). Equipo de comunicación de datos, nombre que suele recibir en una comunicación el módem utilizado por un ordenador para conectarse con otro equipo.

Driver - Programa diseñado para hacer que un dispositivo concreto de hardware interactúe con un sistema operativo o software.

Ethereal -Ethereal es un analizador de protocolos, utilizados para solucionar problemas de red, análisis, desarrollo de software y protocolos, y como una herramienta didáctica para educación. Cuenta con todas las características estándar de un analizador de protocolos.

Ethernet - Norma o estándar (IEEE 802.3) que determina la forma en que los puestos de la red envían y reciben datos sobre un medio físico compartido que se comporta como un bus lógico, independientemente de su configuración física. Originalmente fue diseñada para enviar datos a 10Mbps, aunque posteriormente ha sido perfeccionado para trabajar a 100Mbps, 1Gbps o 10Gbps y se habla de versiones futuras de 40Gbps y 100Gbps

Frame Relay - Estándar industrial, protocolo de capa de enlace del ISO/OSI que maneja múltiples circuitos virtuales utilizando encapsulamiento HDLC entre dispositivos interconectados.

Fork() - "fork" es un software derivativo a partir del código fuente de otro

HDLC – (High Level Link Control). Protocolo de comunicaciones de datos punto a punto entre dos elementos.

ITU - International Telecommunication Union (en español Unión Internacional de Telecomunicaciones, UIT) es el organismo especializado de las Naciones Unidas encargado de regular las telecomunicaciones, a nivel internacional, entre las distintas Administraciones y Empresas Operadoras.

Kernel -el Kernel (también conocido como núcleo) es la parte fundamental de un sistema operativo. Es el software responsable de facilitar a los distintos programas acceso seguro al hardware de la computadora.

Linux - Versión de libre distribución (gratis) del sistema operativo *Unix*, con contribuciones de programadores de todo el mundo.

Mandrake - Mandrake Linux es una distribución Linux enfocada a principiantes o usuarios medios, propiedad de Mandrakesoft.

PCI - PCI son las siglas de varios términos: Peripheral Component Interconnect, un tipo de bus.

PCMCIA - Una tarjeta PCMCIA es un dispositivo normalmente utilizado en computadoras portátiles para expandir sus capacidades.

Perl - Perl (Practical Extraction and Report Language) es un lenguaje de programación inspirado en herramientas de UNIX.

PPP - Son las siglas para Protocolo Punto a Punto definido en el RFC 1661.

Sangoma – Fabricante de hardware, de la tarjeta S5141.

SnS – Analizador de protocolos Serial Network Sniffer.

ssh - Es el nombre de un protocolo y del programa que lo implementa. Este protocolo sirve para acceder a máquinas a través de una red, de forma similar a como se hace con *telnet*.

vnc - Son las siglas de Virtual Network Computing.

V.35 - Es un estándar de la International Telecommunication Union (ITU) para intercambios de datos sincrónicos de alta velocidad.

BIBLIOGRAFÍA

"El lenguaje de programación C" Segunda edición
Brian Wl Kernighan, Dennis M. Ritchie, editorial Prentice Hall

"Como programar en C/C++" Segunda edición
H.M. Deitell, P.J. Deitell, editorial Prentice Hall

"Redes de Computadoras" Tercera Edición
Andrew S. Tanenbaum, editorial Prentice Hall

"Configuraciones y Normas"
www.cisco.com

"Soporte y documentación de APIs"
www.sangoma.com
ftp://ftp.sangoma.com

"Analizador y documentación de protocolos"
www.ethereal.com

"Definiciones del Glosario"
http://es.wikipedia.org/wiki/Portada

"Recomendaciones de interfaz V.35"
http://www.itu.int/home/index-es.html

"Protocolos"
http://www.ieee.org/portal/site

"Tenaria, Interfaz V.35 Nx64 circuito digital punto a punto"
http://www2.retena.es/textos/pdf_tenaria/tenar_16.pdf

"Adaptadores de PCI a PCMCIA"
http://www.mobl.com/expansion/

"Módulos Stand Alone"
http://www.rabbitsemiconductor.com/

"Kits FPGA Xilinx"
http://www.xilinx.com/

"Tarjetas PCMCIA con interfaz V.35"
http://www.farsite.com/

"National Instruments"
http://www.ni.com/

"Cliente ssh Putty "
http://www.putty.nl/

"Cause Outlet – Shopping on-line"
http://www.caseoutlet.com

"Calculadora para potencia consumida"
http://resources.mini-box.com/online/powersimulator/powersimulator.html

Serial Network Sniffer

Manual de Usuario

Contenido

1 Presentación del Analizador SnS

El analizador de protocolos SnS es un equipo analizador que posee una gran versatilidad. Al tener las interfaces de una PC estándar y Sistema Operativo, puede ser utilizado conectando un monitor, un teclado y ratón (opcional). También brinda la opción de ser operado desde un cliente *ssh* o utilizando la interfaz gráfica SnS para Windows a través de su puerto Ethernet.

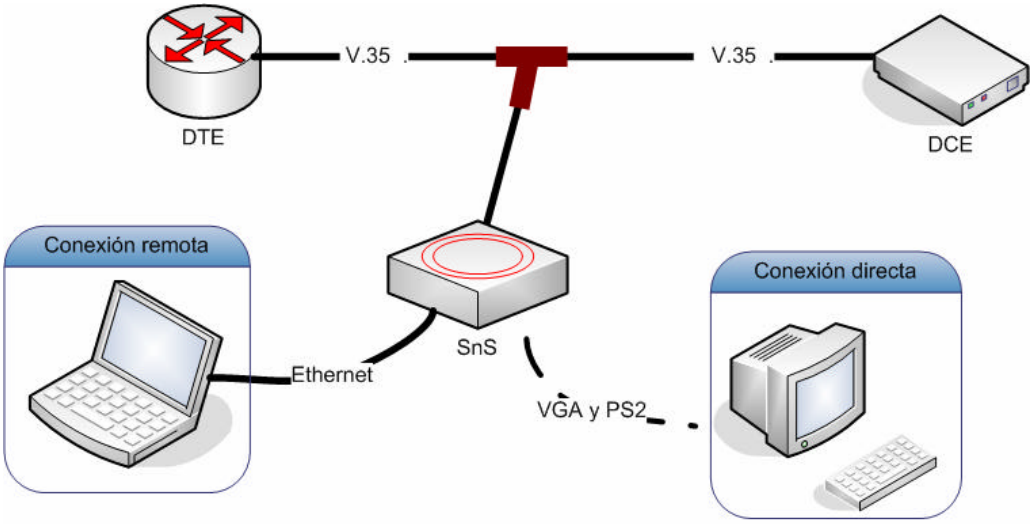


Figura 1: Presentación analizador SnS

2 Analizador SnS

A continuación se presentan los elementos que componen el analizador SnS y se identifican sus componentes.

Figura 2: Analizador SnS

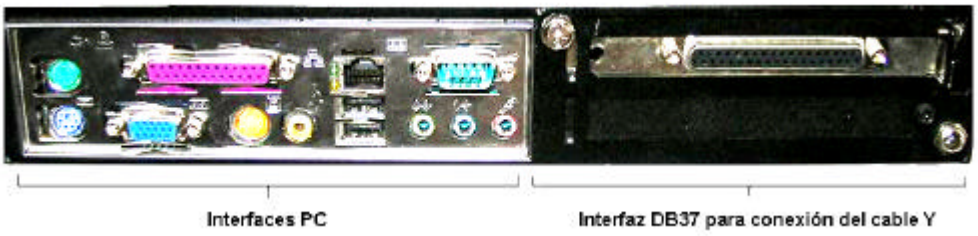


Figura 3: Vista posterior 1

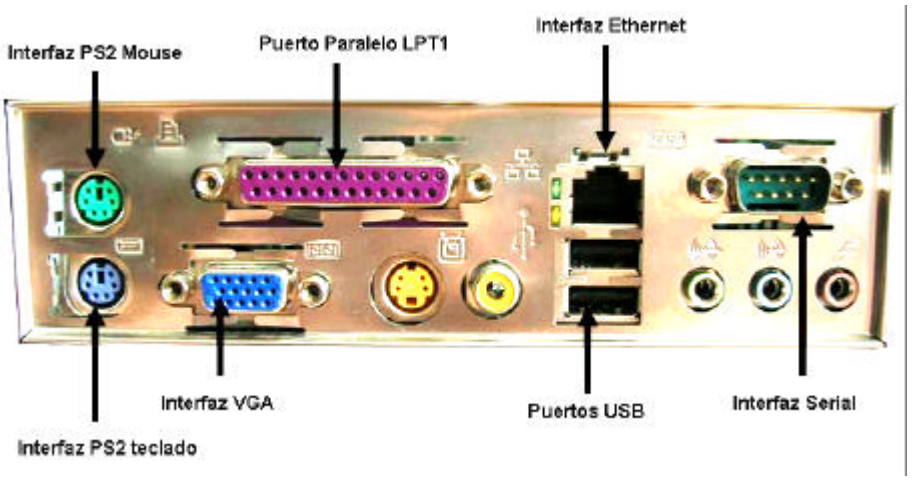


Figura 4: Vista posterior 2

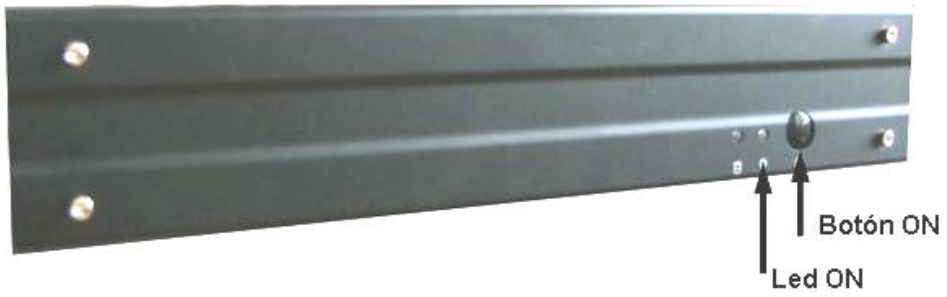


Figura 5: Vista anterior

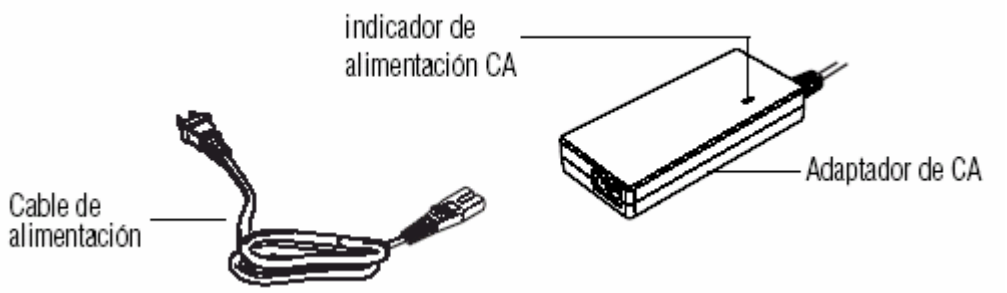


Figura 6: Cable de alimentación y adaptador de CA

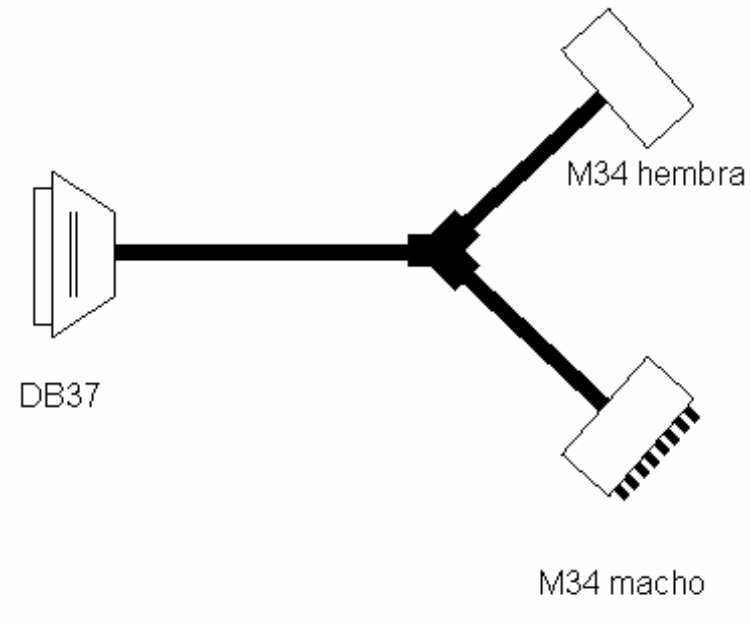


Figura 7: Cable T

3 Cómo Iniciar el Analizador SnS

Como primer paso se debe conectar la interfaz V35 de los equipos que desea analizar (DTE y DCE) utilizando el cable Y suministrado como se muestra en la Figura 8.

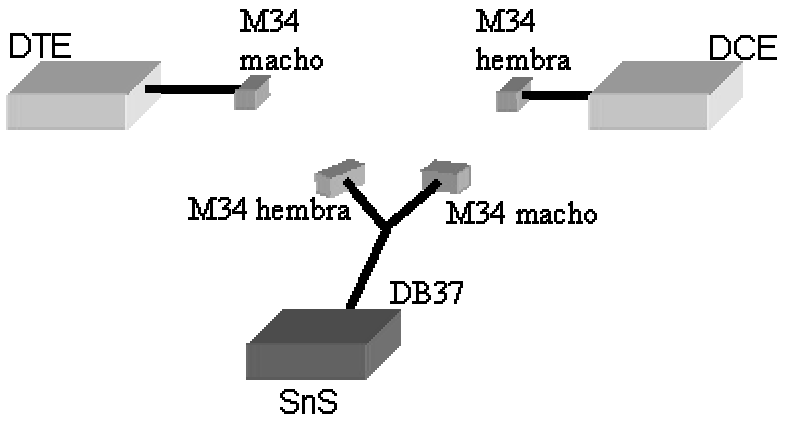


Figura 8: Conexión del analizador

A continuación, conectar la ficha DB37 al analizador SnS y ajustarla para evitar posibles movimientos que puedan dañar el equipo.

El paso siguiente es conectar la fuente del analizador SnS y encenderlo a través del botón ON. Un *led* azul indicará si el equipo se ha encendido correctamente

4 Opciones de manejo del software analizador

El analizador es una PC con sistema operativo LINUX, por lo que puede ser utilizado como PC si se le conecta un monitor y teclado También puede ser manejado remotamente accediendo a él por medio de la aplicación gráfica para Windows o utilizando un cliente *ssh* para Windows. Se supone que el control se realiza desde una PC con sistema operativo Windows. En el caso que se realice desde un sistema operativo Linux puede utilizar el cliente *ssh* que trae instalado.

4.1 *Trabajando directamente sobre el Analizador SnS*

Como se mencionó anteriormente, para trabajar directamente se puede conectar un monitor, teclado y ratón (opcional). Al prender el analizador se accederá directamente al escritorio del usuario *sns*. Para acceder a los programas de captura y análisis se encuentran dos *links* de acceso directo en dicho escritorio.

4.2 *Conexión remota*

Para la conexión remota es necesario que el equipo desde el cual se accederá al analizador tenga conectividad TCP/IP al mismo a través de la interfaz Ethernet.
La IP seteada en el analizador es la 192.168.0.11 con máscara 255.255.255.0, por lo que la IP de la PC de la cual se accederá debe pertenecer a la misma red, por ejemplo puede ser utilizada la IP 192.168.0.12 máscara 255.255.255.0.

NOTA: Ver anexo II.

5 Interfaz Gráfica desde Windows

La interfaz gráfica es una herramienta que pretende simplificar el manejo remoto del analizador, presentando al usuario un entorno amigable.

NOTA: La interfaz grafica SnS debe ser instalada como se explica en el Anexo I.

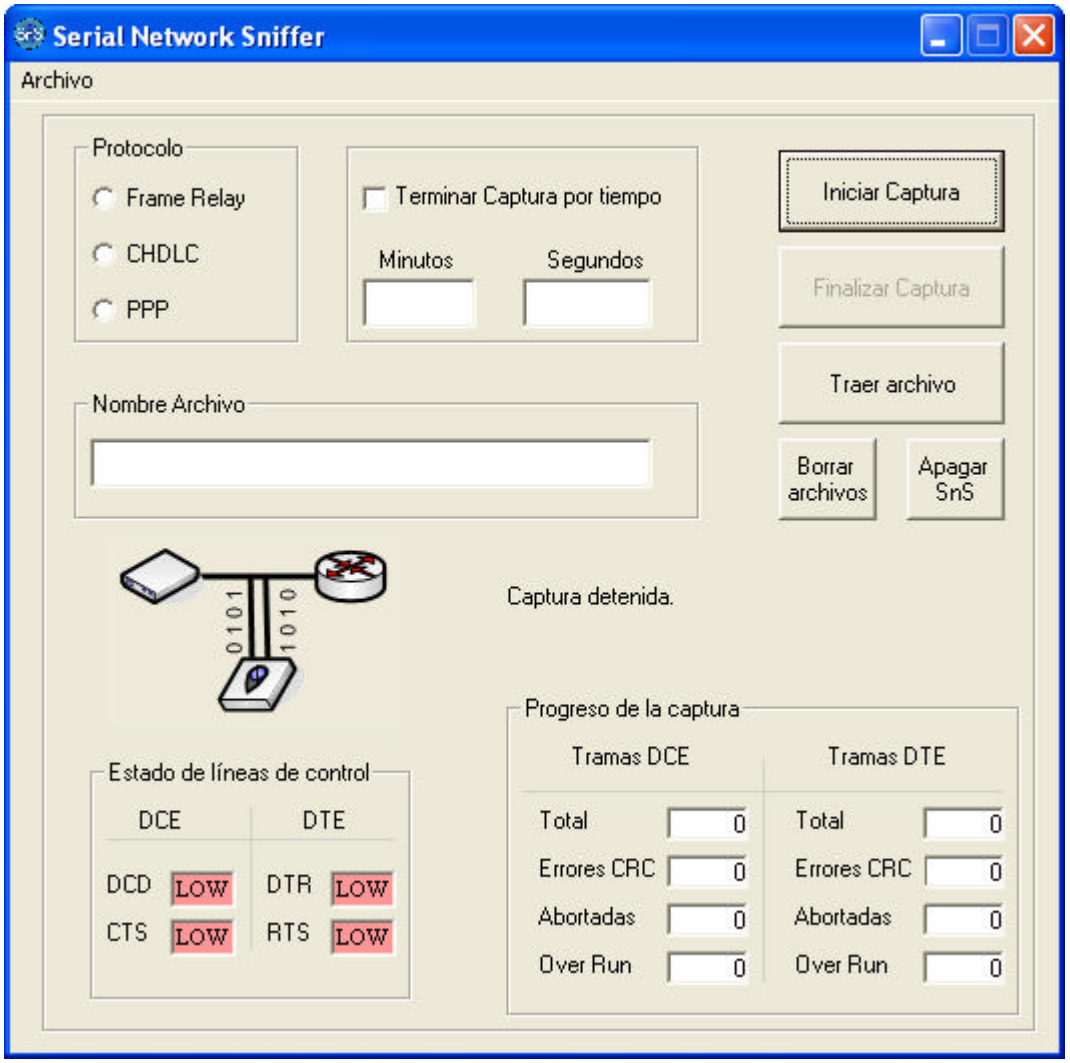


Figura 9: Interfaz gráfica para Windows

La interfaz gráfica se encuentra dividida en los sectores:

- Elección de protocolo de Capa de enlace.
- Si la captura es por tiempo, elección del tiempo de captura.
- Nombre de archivo para la captura.
- Control del analizador y captura.
- Estado de Líneas de Control
- Estadísticas de la captura

5.1 *Elección del protocolo*

El protocolo en la capa de enlace debe ser indicada en el programa antes de iniciar la captura. Es necesario setear el protocolo para que pueda ser analizado correctamente los datos de la captura.

Se pueden elegir entre los protocolos Frame Relay, Cisco HDLC y PPP. Seleccionar este parámetro es obligatorio. En caso de no ser elegido, el programa le pedirá al usuario que elija el protocolo.

5.2 *Captura por tiempo*

Este parámetro es opcional, si la captura a realizar es por un tiempo determinado. Antes de ser ingresado dicho tiempo de captura es necesario habilitar la opción haciendo un clic en la casilla *"Terminar captura por tiempo",* el paso siguiente es ingresar los minutos y segundos que durará la captura.

La opción de iniciar captura por tiempo no es obligatorio para iniciar una captura, solo establece la condición de terminar una captura luego de cumplido el tiempo seteado.

5.3 *Nombre de archivo*

El nombre del archivo es un campo obligatorio. En el caso de no ser ingresado, el programa seteará por defecto un nombre de archivo. El nombre por defecto comenzará con *Captura* seguido por la fecha y hora de la captura.

Si la captura fue efectuada el día 5 del mes 3 del año 2006, a la hora 01:10:47, el nombre del archivo será el siguiente: Captura20060305011047

El nombre de archivo elegido será el nombre con el cual será guardado en el analizador y en la PC bajo el directorio: **C.\SnS\capturas**

Si el usuario así lo desea, pude elegir el nombre del archivo con el cual desea que el mismo sea guardado.

5.4 *Control de captura*

Dentro del cuadro *control de captura* podemos tanto iniciar la captura como detenerla. Luego de ser indicado el protocolo de capa de enlace, el tiempo de captura, si corresponde y el nombre de archivo, se le debe indicar al analizador que comience la captura presionando el botón *Iniciar Captura.*

Al presionar el botón de *Iniciar Captura* el botón de *Traer archivo* se deshabilita, hasta que la captura haya finalizado. Esto se debe ya sea porque se cumplió el tiempo de captura o se detuvo manualmente utilizando el botón de *Finalizar Captura.*

Si se inicializa la captura sin indicar el tiempo de captura, el usuario deberá finalizar voluntariamente la captura. Si se ingresó un tiempo, el usuario también podrá detener la captura antes de transcurrido el tiempo seleccionado.

Una vez finalizada la captura, debido a que se presiono el botón *Finalizar Captura* o se agotó el tiempo seleccionado, se habilitará el botón de *Traer archivo*.

El botón de *Traer archivo* copiará los datos de la captura desde el analizador a la carpeta **C:\SnS\capturas**.

Una vez que se haya copiado el archivo se abrirá automáticamente el programa Ethereal con el archivo copiado.

El botón *Borrar Archivos* borra todos los archivos de capturas “del analizador”, los archivos que se copiaron a la carpeta **C:\SnS\capturas** no son modificados. Por este motivo se recomienda liberar espacio del analizador presionar este botón cada vez que se esté seguro que no se necesitarán más los datos que se encuentran en el analizador SnS.

El botón *Apagar SnS*, envía el comando *poweroff* al analizador. Luego de unos segundos se apagará el analizador SnS.

5.5 *Estado de Líneas de Control*

Cuando es lanzada la captura el analizador chequea cada 4 segundos el estado de las líneas de control. Cuando las líneas de control están en estado activo se visualiza en la interfaz en color verde y con la palabra HIGH. En el caso que alguna de las líneas de control baje, se visualizaran con color rojo y la palabra LOW.

5.6 *Estadísticas de la captura*

De la misma forma que el analizador chequea cada cuatro segundos el estado de las líneas de control, también realiza una consulta de tramas capturadas, tramas perdidas, tramas con error de CRC y tramas con error de Over Run.

El chequeo se realiza también cada 4 segundos. Estos valores son contadores, *no identifican la trama*, para identificar la trama con errores hay que abrir el archivo de captura con el analizador Ethereal.

6 Captura por línea de comando

El software de captura se inicializa corriendo el programa *sns*, el cual se encuentra en el directorio **/home/sns/analizador**.

De ahora en adelante nos centraremos en explicar la captura inicializada vía *ssh*.

6.1 *Iniciar la captura por ssh*

El trabajo remoto utilizando una conexión *ssh* brinda la ventaja de operar el analizador desde cualquier sitio, con pocos requerimientos de ancho de banda entre usuario y analizador. Como contrapartida presenta la desventaja de tener que bajar el archivo desde el analizador a la PC utilizando el programa *Secure File Transfer Client* (Se puede utilizar otro programa que el usuario prefiera). Luego se deberá correr el *Ethereal* para analizar el archivo.

A continuación se muestra un ejemplo de como loguearse vía *ssh* al analizador SnS utilizando el programa *Secure Shell Client*.

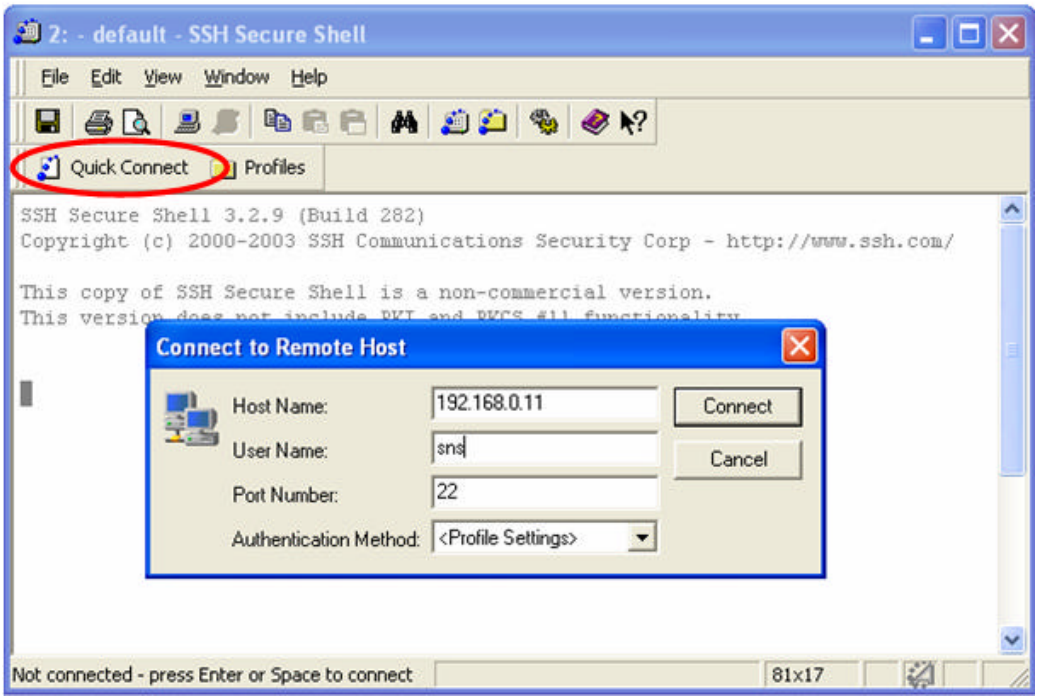


Figura 10: Secure Shell Client

Se debe abrir desde el menú de Inicio el cliente *ssh* Secure Shell, con lo cual se abrirá una ventana igual a la de la Figura 10. Para elegir el servidor donde se quiere conectar el usuario tiene que hacer un *clic* en *Quick Connect*. Esta opción es indicada en la imagen con un ovalo en rojo. Se abrirá una ventana en donde se debe indicar la IP del servidor y el usuario. En este ejemplo se seteó la IP 192.168.0.11 y usuario *sns* (Valores por defecto del analizador).

Al presionar la botón *Connect* se deberá ingresar la *password* que por simplicidad es igual al nombre de usuario *sns.*

Al loguearse, el usuario se encontrará en el directorio **/home/sns/analizador**. El comando que inicializa el analizador es el comando *sns*.

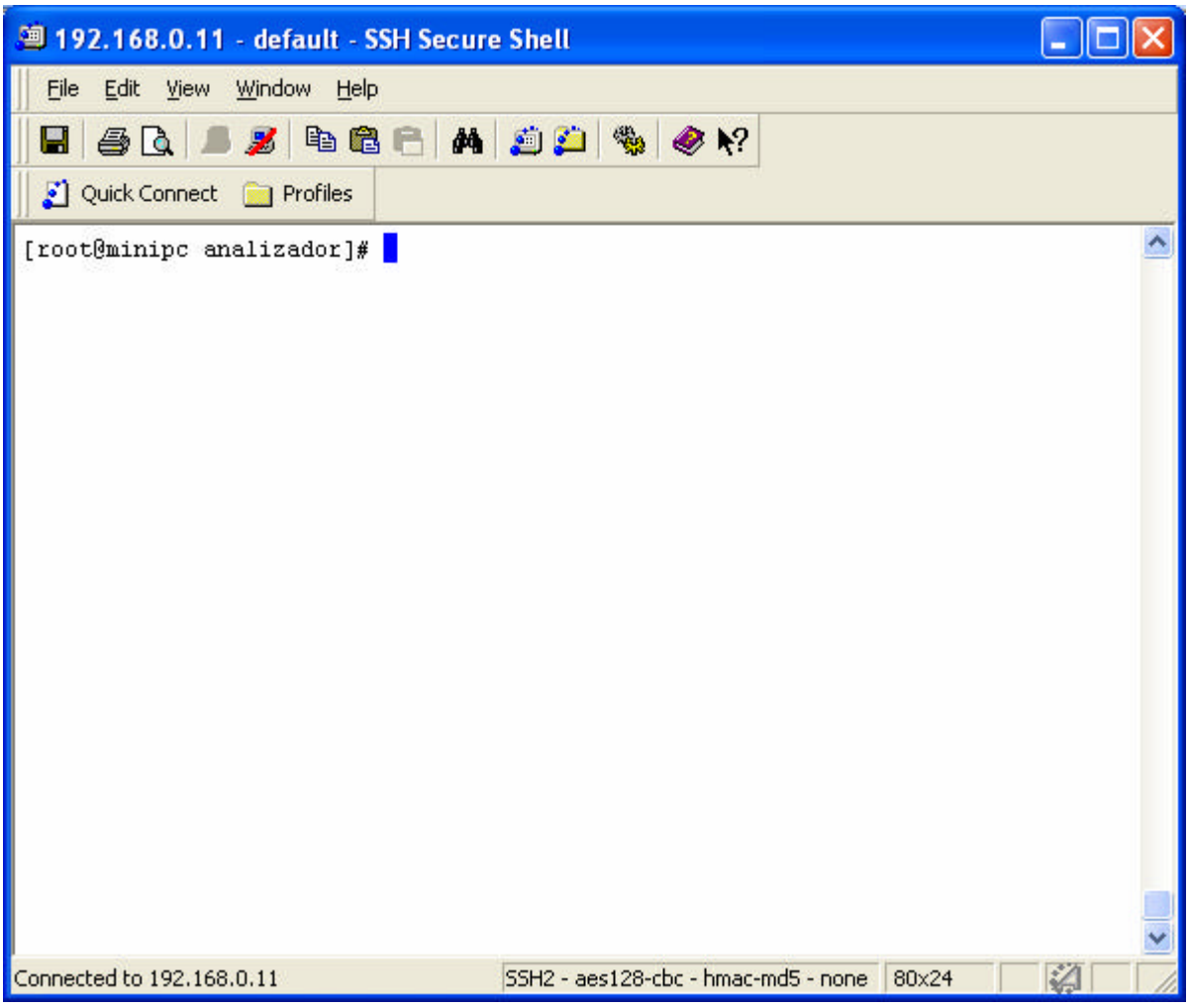


Figura 11: Ejemplo de vista de consola.

Si se ejecuta el comando *sns* se desplegará en pantalla los parámetros que se le deben pasar para iniciar la captura.

```
[sns@minipc analizador]$ sns

<options>:
        -min  <minutos>          # minutos a capturar
        -seg  <segundos>         # segundos a capturar
        -v                       # modo interactivo
        -paq                     # cantidad paquetes
        -prot    <1|2>           # protocolo de capa enlace
                                          1:CHDLC o PPP
                                          2:Frame Relay
        -archivo  <nombre>       # nombre del archivo de captura

PROGRAMA CANCELADO.
```

7 Iniciar una captura

La captura pude ser iniciada de modo interactivo, donde el analizador solicita los parámetros necesarios para la captura o los mismos pueden ser pasados al programa en la misma línea en la cual se ejecuta el comando *sns*.

Ejemplo:

- **sns -min 3 -seg 30 -archivo captura1.bin -prot 1**
- **sns -paq 10 -archivo captura1.bin -prot 1**
- **sns -archivo captura1.bin -prot 1**

En el primer ejemplo se pide capturar por tres minutos y medio. El archivo será guardado en **/home/sns/analizador/capturas** con el nombre **captura1.bin**. El nombre es totalmente arbitrario. Se aconseja trabajar con nombres sin espacios.

El parámetro **–prot** indica el protocolo de capa de enlace, siendo 1 para CHDLC y PPP y el valor 2 para Frame Relay.

Es obligatorio que los parámetros **–archivo** y **–prot** sean pasados.

En el segundo ejemplo se pide al analizador que capture los primeros diez paquetes. Cuando se cumpla la condición de 10 paquetes en alguno de los dos puertos la captura finalizará. Puede ocurrir que, por ejemplo, sean capturados 5 paquetes desde el DTE y 10 del DCE, cumpliéndose la condición de fin de captura.

En el tercer ejemplo se inicializa la captura sin indicar la condición de fin. Para finalizar la captura se debe presionar **Ctrl+C**.

En todas las situaciones se puede forzar el fin de la captura presionando **Ctrl+C**.

Una forma distinta de iniciar la captura es utilizando el modo interactivo, de la forma:

sns -v

Al comienzo, en pantalla se puede visualizar "Bienvenido a SnS Network Serial Sniffer", como se muestra a continuación. Luego se despliegan las opciones que se encuentran disponibles para seleccionar en la configuración del analizador.

```
================================
||          Bienvenido a SnS          ||
||        Serial Network Sniffer      ||
================================
```

Opciones:

1) Protocolo de la interfaz 1: CHDLC o PPP, 2: FRAME RELAY:
2) Condición de finalización 1: Cant. Paq, 2: Manual, 3: Por tiempo:
3) Nombre final del archivo de captura (Ejemplo: captura.bin):

7.1 *Descripción de las opciones*

7.1.1 Protocolos

Los protocolos disponibles en esta versión son: CHDLC (Cisco HDLC), PPP y Frame Relay.

Se deberá ingresar la opción 1 para CHDLC o PPP y 2 para el protocolo Frame Relay, por lo que es necesario tener la información del protocolo en capa de enlace antes de iniciar la captura.

7.1.2 Condición para finalizar captura

Se puede seleccionar la condición de fin de captura por cantidad de paquetes capturados, de forma manual o por tiempo transcurrido.

7.1.2.1 Por cantidad de paquetes capturados

Si se selecciona la condición de fin de captura por cantidad de paquetes, se desplegará un submenú en el cual debe ingresarse la cantidad de paquetes que desea capturar desde la línea (X paquetes desde el DCE) y desde el terminal (Z paquetes desde el DTE).
Una vez que el analizador capturó el número de paquetes ingresados para línea y terminal, se finalizará la captura.

7.1.2.2 Terminación manual

Seleccionada esta opción, una vez iniciada la captura, el usuario podrá finalizar la misma en el momento que así lo desee presionando la combinación de teclas **Ctrl+C.**

7.1.2.3 Por tiempo

En esta opción se desplegaran dos submenús para ingresar los minutos y los segundos que durará la captura.

En cada una de las opciones es posible terminar la captura presionado **Ctrl+C**.

Una vez finalizada la elección de las opciones se desplegarán en pantalla los datos ingresados y se solicitará confirmación: **[y/n]**

Si se ingresa el valor "n", el programa finalizará. En el caso que deba ingresar datos diferentes, debe presionar "n" y luego correr nuevamente el programa.

Si se confirma los datos (opción "y") se dará comienzo a la captura.

En todos los casos en el modo interativo se muestra a pantalla cada 4 segundos un estado de la captura. Se muestra la cantidad de paquetes capturados al momento, cantidad de errores de CRC, con Over Run, y tramas abortadas.

Una vez finalizada la captura, se habrá creado un archivo con los datos en el directorio **/home/sns/analizador/capturas**, con el nombre proporcionado por el usuario al ejecutar el software de captura.

Para el análisis de los datos capturados con el Ethereal se deberá bajar el archivo ubicado en **/home/sns/analizador/capturas/nombre_archivo** a la PC utilizando la aplicación *Secure File Transfer Client*.

NOTA: Como transferir archivos con el programa *Secure File Transfer Client* puede ver el anexo III.

8 Transferencia del archivo del analizador a la PC portátil

Antes de comenzar el análisis de datos con el Ethereal deben ser agregadas dos columnas, en las cuales se mostrará el sentido de las tramas. Ver anexo II.

Finalizada la captura, los datos son guardados en **/home/sns/analizador/capturas** con el nombre de archivo ingresado por el usuario.

Una vez que se tenga el archivo se debe ejecutar el programa Ethereal y luego abrir el archivo con los datos.

8.1 *Conexión ssh*

Se deberá transferir el archivo utilizando el programa *Secure File Transfer Client* desde el analizador a la PC de trabajo. Ver anexo III.

8.2 *Conexión ftp*

El analizador tiene instalado un servidor *ftp.* Para ingresar se debe hacer *ftp* 192.168.0.11 (IP del analizador) y entrar con el usuario *anonymous* sin *password*.

Los permisos son de solo lectura. Para copiar el archivo desde el analizador a la PC se debe ejecutar el comando *get* "*nom_archivo"*, en este caso *get* es el comando para bajar el archivo a la PC y *nom_archivo* es el nombre del archivo que se quiere bajar.

En la Figura 12 se muestra un ejemplo de ejecución del comando *ftp* desde DOS:

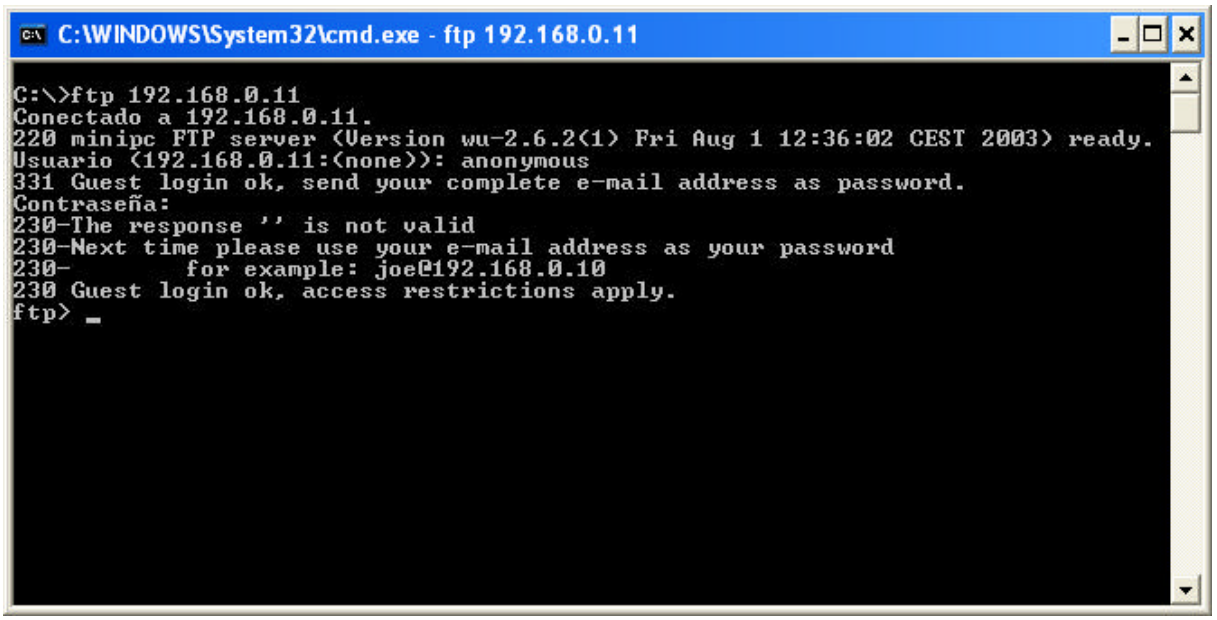


Figura 12: Ejemplo de ejecución del comando *ftp*

Cuando el programa solicita el nombre de usuario se debe escribir *anonymous* y dejar la contraseña en blanco para luego presionar *enter*.

Al ingresar se estará directamente sobre el directorio en donde son guardadas las capturas. Si se ejecuta el comando *dir* se listan los archivos. Si se desea bajar el archivo se deberá ejecutar el comando *get*.

9 Apagar y desconectar el Analizador

La forma correcta de apagar el analizador se realiza de la misma forma en que se apagan los sistemas operativos conocidos.

Si se desea hacerlo desde consola se deberá ejecutar el comando **poweroff.**

Si elige hacerlo a través de la interfaz gráfica, de forma no remota, se deberá ir a *Inicio→Salir*.

El estado del *led* azul indicará cuando se haya apagado el analizador SnS.

Una vez que el analizador fue apagado, se puede proceder a desconectar los cables.

10 Si surgen problemas

Si enciende el analizador siguiendo las instrucciones y no enciende el *led* azul, asegúrese que lo conectó correctamente a la fuente de energía.

Se esta intentando acceder por *ssh* y no responde ejecute un *ping* desde la PC de la cual se desea acceder al analizador:

- Si el *ping* responde, verifique que escribió correctamente el usuario y la IP.
- Si el *ping* no responde, espere uno minutos y vuelva a ejecutar el *ping.* Puede ser que el analizador no haya sido apagado correctamente la vez anterior y se encuentre realizando un chequeo de disco. Si continúa sin responder conecte un monitor al analizador para ver los errores.

Si corre el programa y no captura datos, asegúrese que conectó el cable Y al analizador correctamente o el de los routers al cable Y.

ANEXO I

1 Instalaciones

1.1 *Instalación de la interfaz gráfica SnS*

La instalación se encuentra dentro del CD en **Herramientas\windows**. El mismo es un archivo comprimido **SnS.exe** auto extraíble. El archivo instalará el programa en **C:\SnS**. Además serán creadas dos carpetas; *capturas* y *herramientas*.

Dentro de la carpeta *capturas* se guardarán los datos luego de traer el archivo con la aplicación. En la carpeta *herramientas* se guardan varios programas. Es importante no borrar ninguno de estos archivos, debido a que los mismos son indispensables para la aplicación.

1.2 *Instalación del cliente ssh SSHSecureShellClient*

La instalación se encuentra en el CD en **Herramientas\windows**. Se deberá realizar doble *clic* e instalar. La última versión puede ser descargada gratuitamente desde la página oficial **http://ftp.ssh.com/pub/ssh/**

Luego de la instalación deberá tener instalado dos programas*: "Secure File Transfer Client"* y el "*Secure Shell Client*".

Secure File Transfer Client: Programa sencillo para la transferencia de archivos basados en el protocolo *ssh*.

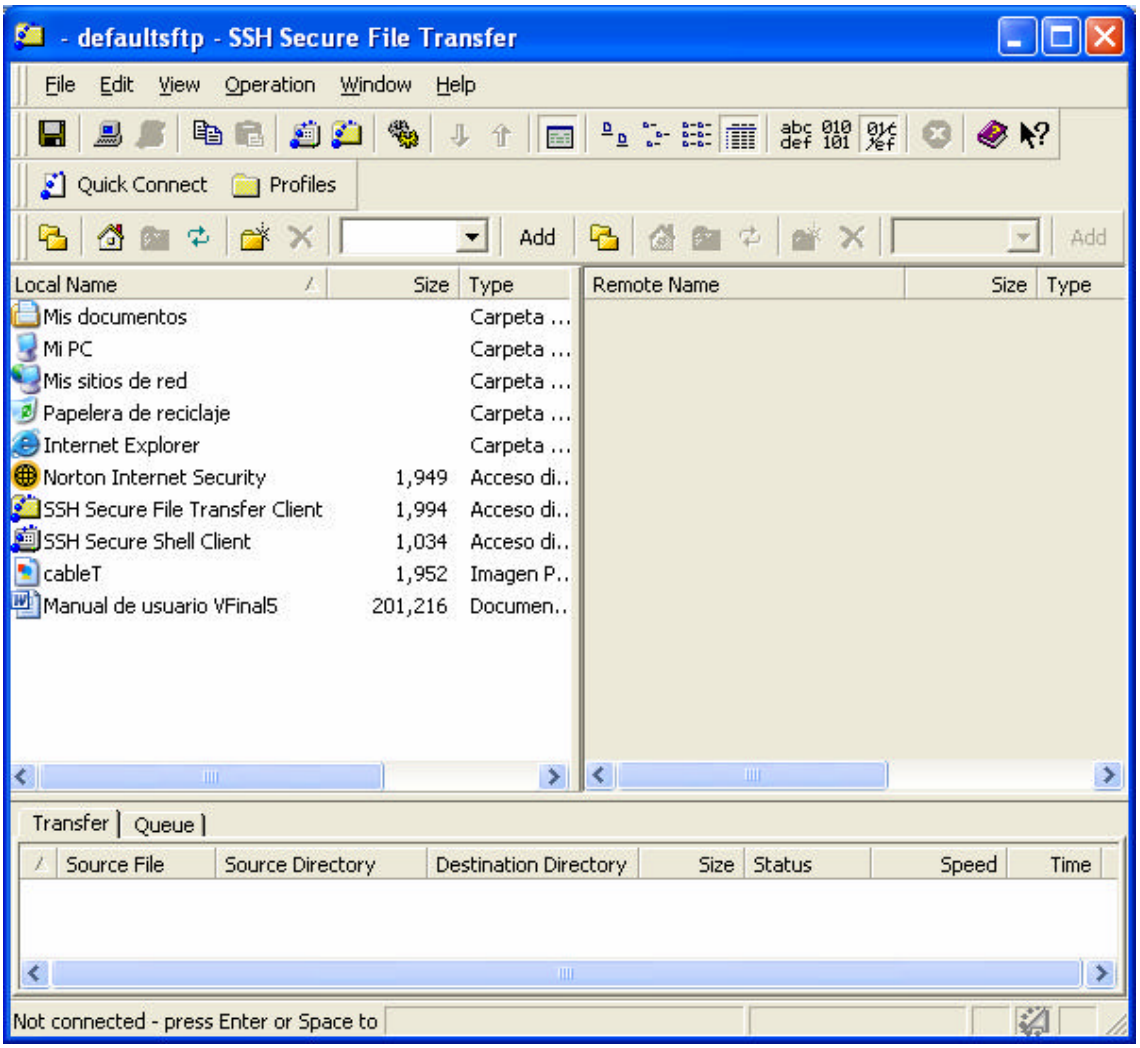


Figura 13: Interfaz gráfica del Secure File Transfer Client

Secure Shell Client: Cliente *ssh*, el cual permite abrir una consola desde la PC que contenga el servidor.

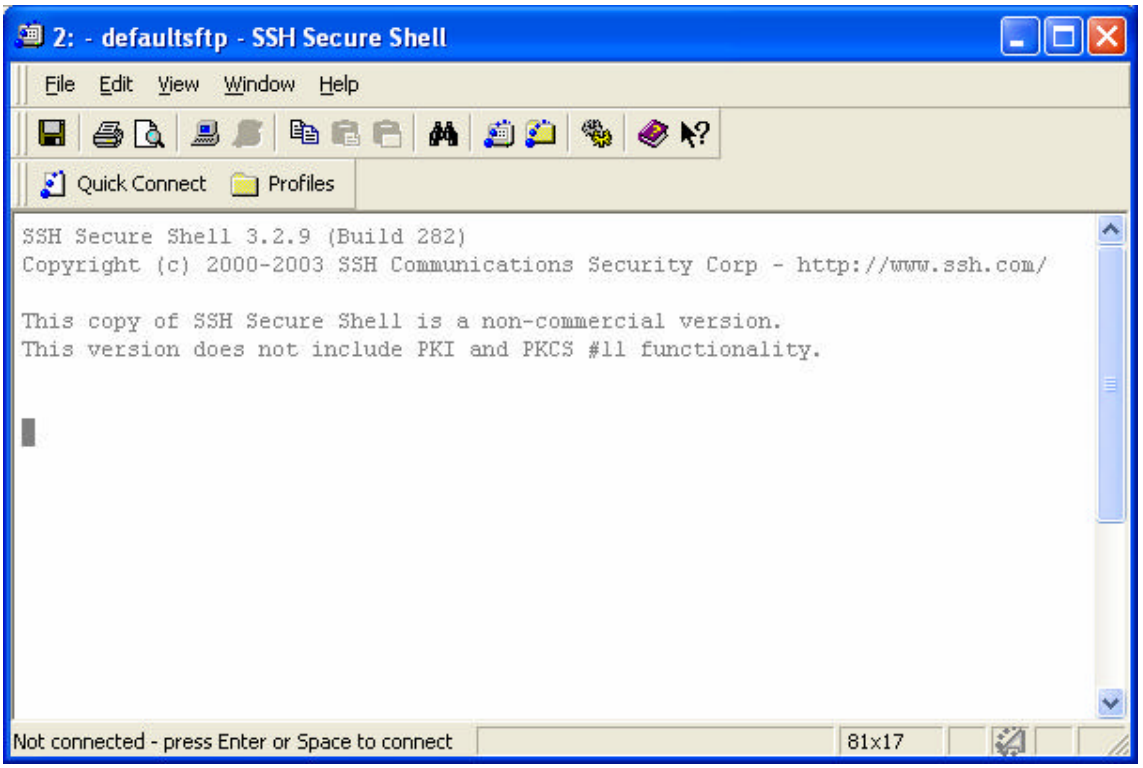


Figura 14: Secure Shell Client

ANEXO II

1 Configuración de la dirección IP en Windows

Se debe Ir a: Inicio → *Panel de control* → *Conexiones de red*

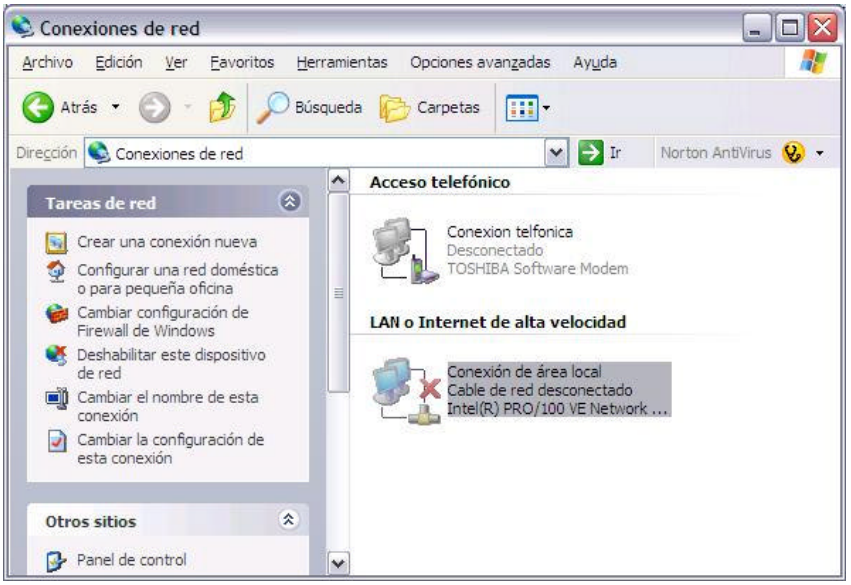


Figura 15: Configuración de la IP (1).

Botón derecho sobre *Conexión de área local* → *Propiedades*

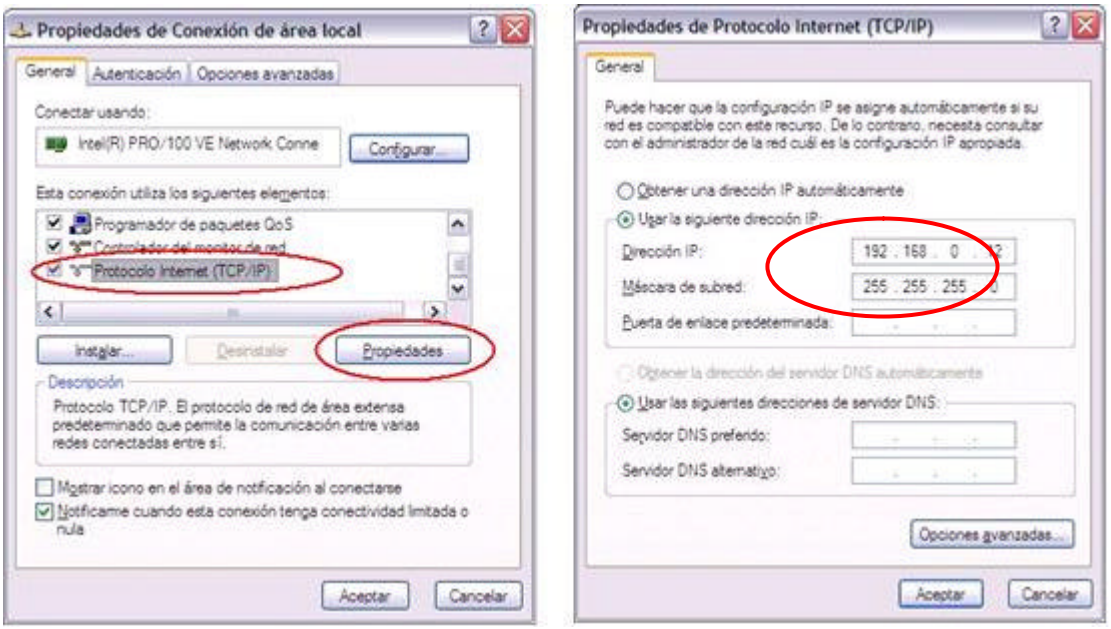


Figura 16: Configuración de la IP (2).

Dentro de *Propiedades de Conexión de área local*, se deberá cambiar las propiedades de Protocolo de Internet (TCP/IP), y setear la IP con la cual se trabajará, recordando que debe pertenecer a la red 192.168.0.0. En este ejemplo se seteó la IP 192.168.0.12, con mascara 255.255.255.0

ANEXO III

1 Configuración del software Ethereal

La instalación del Ethereal para Windows es proporcionada también en el CD. Para poder visualizar el sentido de las tramas se deberá agregar dos columnas.

NOTA: El analizador ya trae configurado el Ethereal de modo que despliegue las mencionadas columnas.

Luego de instalar el Ethereal, ejecutarlo y agregar las columnas de la siguiente Forma: ir a *Edit→Preferences*

Se abrirá la ventana mostrada en la Figura 17. En la columna de la derecha seleccionar: *Columns → New*

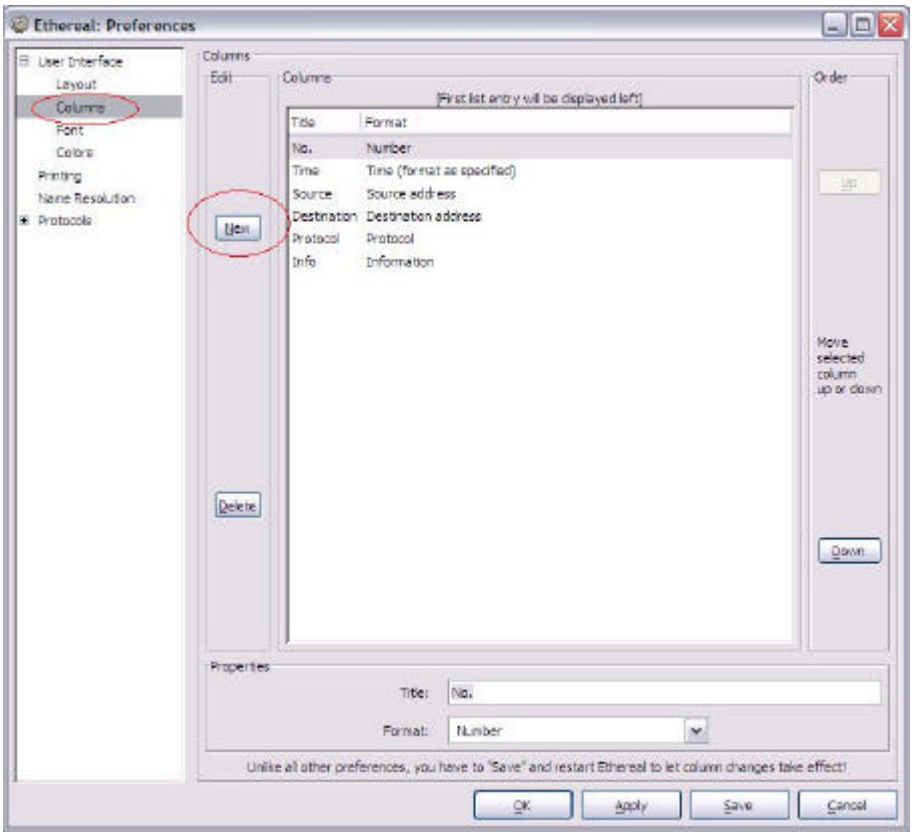


Figura 17: Configuración del Ethereal (1).

Agregar el *Title: "Origen"* y dentro de *Format* seleccionar *"Hardware src addr"* Agregar la segunda columna, volviendo a presionar el botón *New* y escribir el *Title: "Destino"* y dentro de *Format* seleccionar *"Hardware dest addr".*

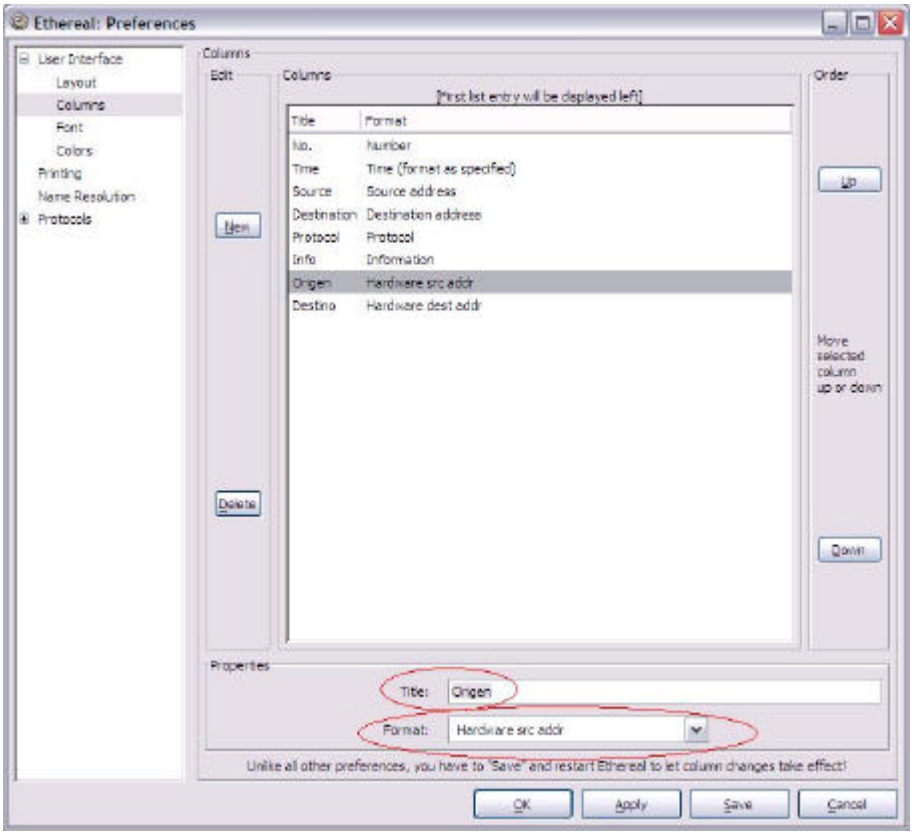


Figura 18: Configuración del Ethereal (2).

Para mejorar la visualización, es posible llevar la columna *Origen* para la tercera posición, luego de *Time*, utilizando el botón *UP*. Hacer lo mismo con la columna destino y llevarlo a la cuarta posición.

Es importante guardar la configuración presionando el botón *Save* y *Ok*. Para poder visualizar los cambios se debe reiniciar el Ethereal, cerrándolo y volviéndolo a abrir.

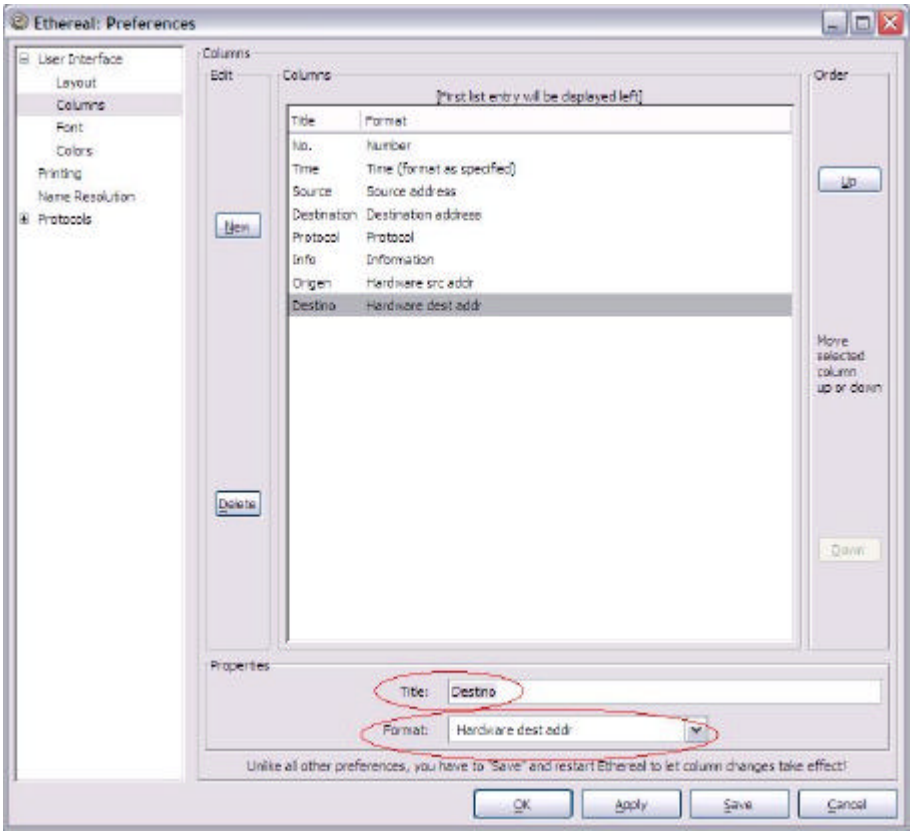


Figura 19: Configuración del Ethereal (3).

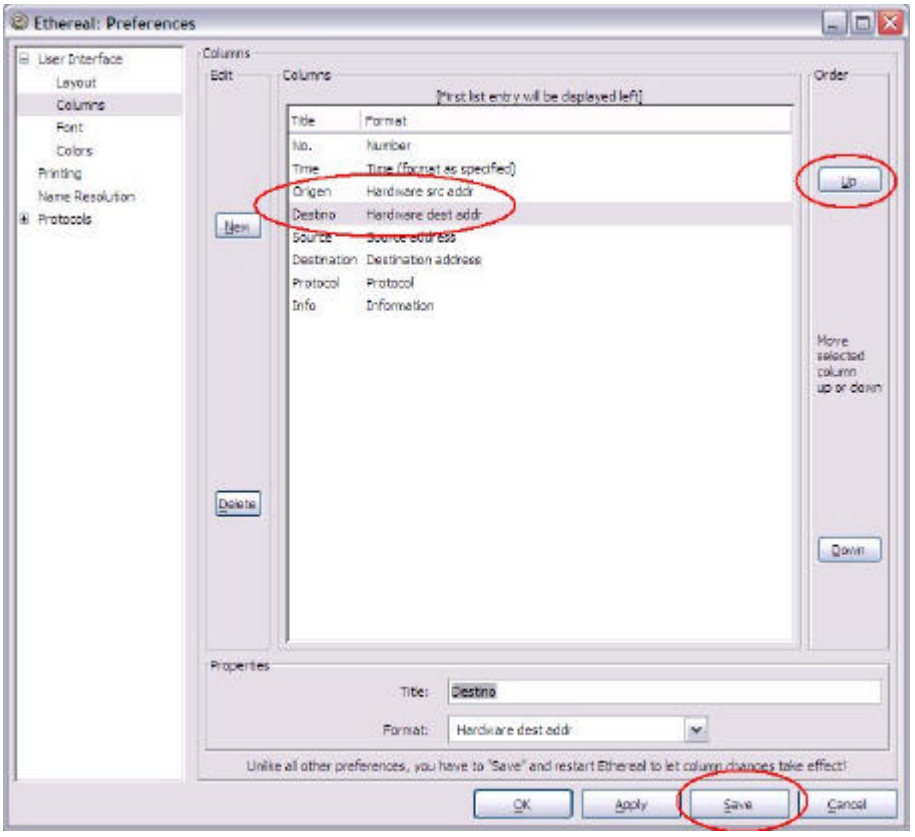


Figura 20: Configuración Ethereal (4).

ANEXO IV

1 Transferencia de archivos

NOTA: Este programa es aplicable para PCs con sistema operativo Windows.

Para transferir los archivos desde el directorio **/home/sns/analizador/capturas/**, desde el analizador hacia la PC con SO Windows, se recomienda utilizar el programa *Secure File Transfer Client*. Su instalación se encuentra en el CD que se adjunta. Su utilización es sencilla, a continuación se presenta como utilizarlo.

Al ejecutar el programa *SSH Secure File Transfer* se abrirá la ventana

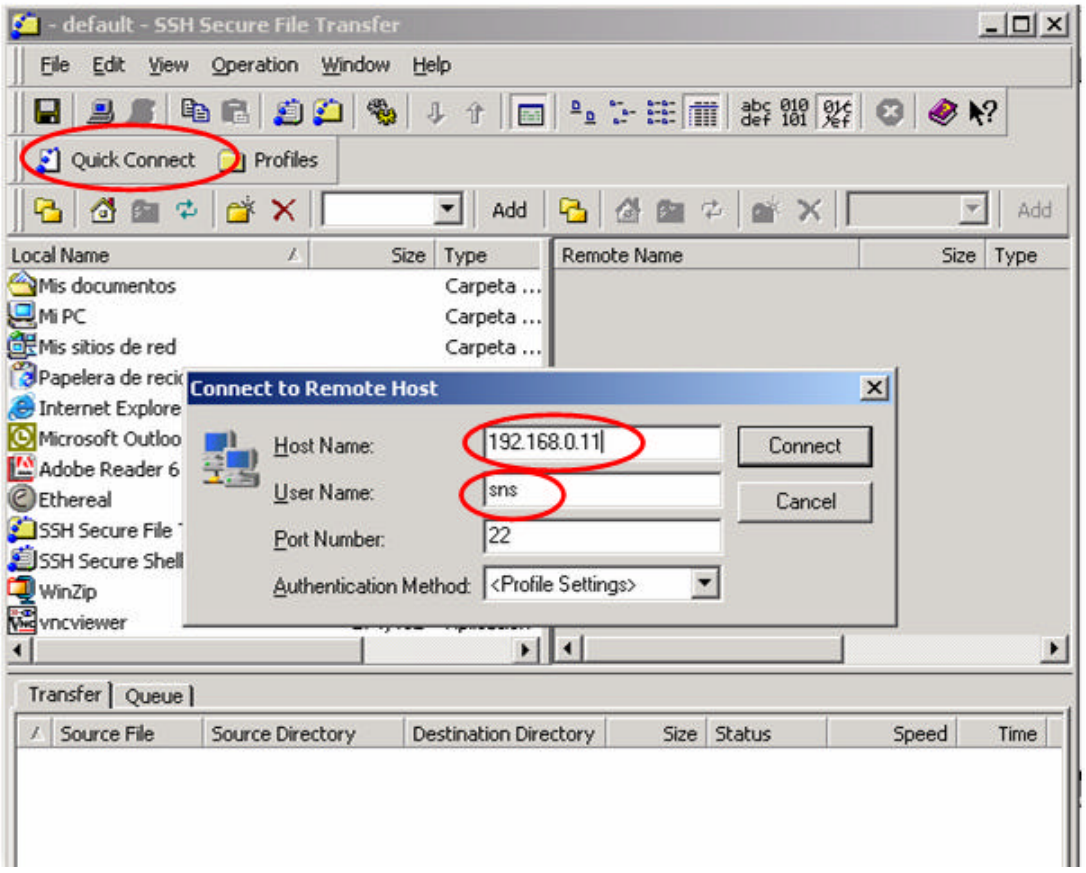


Figura 21: Ejecución de *ssh*

Se debe realizar *clic* en *Quick Connect* e indicar que nos conectaremos con el Analizador SnS. Para esto debemos ingresar la IP 192.168.0.11, con usuario *sns*, a continuación hacer *clic* en *Connect*. Puede aparecer en pantalla un cartel que pregunta si se quiere agregar el *"host"* dentro de los *host* conocidos, a lo cual se deberá aceptar. A continuación aparecerá una ventana pidiendo la clave de usuario, en donde ingresamos la clave *sns.*

Figura 22: Ingreso de clave *ssh*.

Luego de conectados, se podrá ver a la izquierda las carpetas de la PC a la cual estamos conectados y a la derecha encontramos las carpetas correspondientes al analizador.

Se podrá visualizar la carpeta Analizador. Haciendo doble *clic* se accederá a ella y se podrán ver las carpetas *tmp* y *capturas.* Dentro de esta última se encontrará el archivo con los datos de la captura.

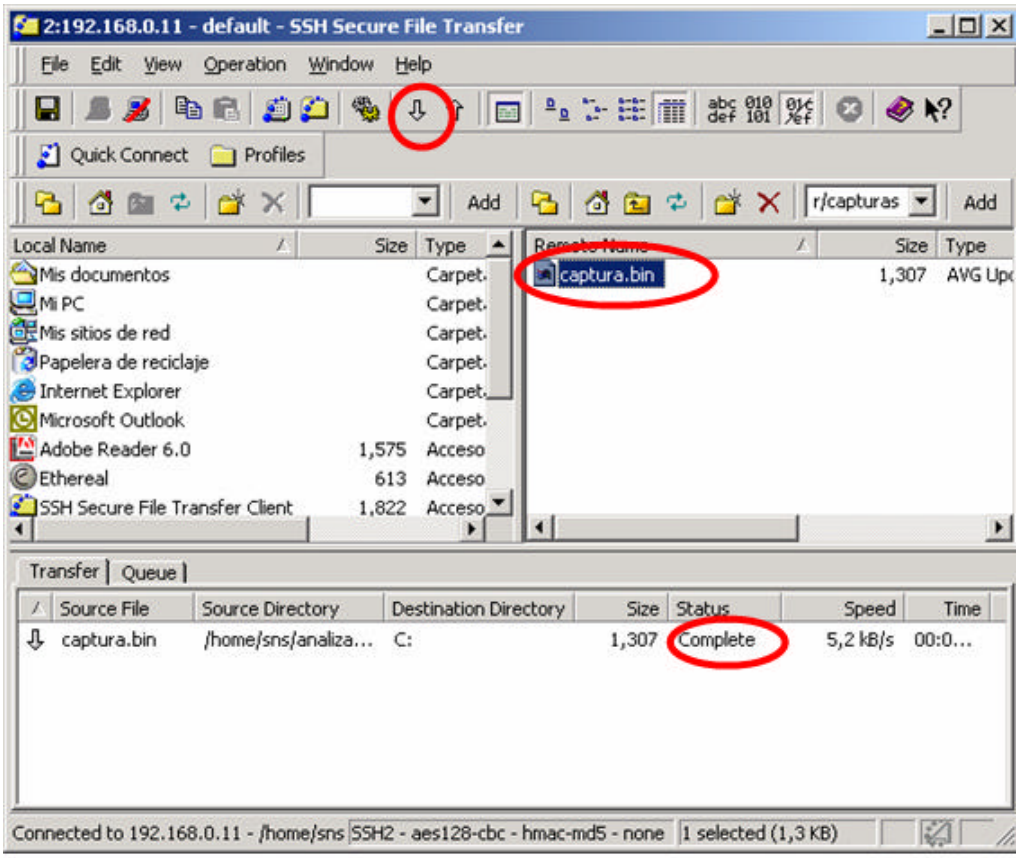


Figura 23: *ssh*

Seleccionado el archivo de captura, se deberá bajar a la PC, haciendo *clic* en la opción *"Download"*. Se pedirá la ruta donde el usuario quiera que sea guardado, lo cual queda a total elección del usuario.

NOTA: Antes de ser analizado el archivo bajado a la PC, se deberá verificar que haya sido transferido en su totalidad.

ANEXO V

1 Comandos de Linux que pueden ser de utilidad

La siguiente lista es simplemente una reseña de los comandos que pueden ser de utilidad. No intenta en absoluto abarcar Linux.

ls - Lista los archivos y directorios donde se ejecuta el comando.

cp – Copia de archivos
sintaxis: ***cp origen/nombrearchivo destino/nombrearchivo***

cd. – Cambia de directorio en que se está trabajando.
sintaxis: ***cd directorio***
cambia al subdirectorio
sintaxis: ***cd..***
cambia al directorio superior.

ps aux - Muestra los procesos que se encuentran corriendo.

kill - Mata un proceso
sintaxis***: kill nºproceso***
El numero de proceso se obtiene ejecutando el comando ***ps aux***

pwd – Muestra la dirección absoluta del directorio en donde se está trabajando.

Proyecto de Grado, Facultad de Ingeniería, Universidad de la República, 22 de marzo de 2006

SnS
Serial Network Sniffer

Gabriel Botti, Francisco Pérez, Paola Sciarra

I. INTRODUCCIÓN

El proyecto de grado SnS (Serial Network Sniffer) consistió en el diseño e implementación de un analizador de protocolos de redes de datos, cuya interfaz es la V.35.

El objetivo del proyecto fue lograr una herramienta para un primer diagnóstico del problema, que sustituyera los analizadores de alto costo para el caso de análisis únicamente de interfaces v.35. Se presenta así el analizador SnS, específico para interfaz V.35, a un costo menor a los 2.000 USD.

Al presentarse un problema de comunicación entre dos equipos con interfaz V.35, se puede acceder al lugar con un equipo portátil y económico.

SnS es un equipo, que dadas sus dimensiones y peso lo hacen sencillo de transportar. Analiza los datos que están comunicando los equipos, identificando la causa de los problemas de red a partir de la capa de enlace.

II. ORGANIZACIÓN DEL DOCUMENTO.

En el primer y segundo ítem se explican las características generales del proyecto y se introduce la estructura básica del documento.

En el ítem 3 se estudian los fundamentos, tanto teóricos como prácticos, en los que se sustentó o se basó la elección del hardware y software utilizado en el equipo SnS.

En los ítem 4 y 5 se detalla el diseño del analizador desarrollado. Se describe el diseño del hardware del equipo y su principal componente, la tarjeta adquisidora de datos Sangoma S5141. Se detalla el cable de interconexión T utilizado y el software de captura desarrollado.

En el ítem 6 se presenta un análisis comparativo de las capturas entre el analizador SnS y el Advisor de HP utilizado actualmente por el cliente. Se explican las pruebas realizadas y se analiza la precisión con que el SnS inserta las marcas de tiempo.

El análisis de los costos relacionados con el proyecto se desarrolla en el ítem 7.

Se dedica al final del documento un ítem correspondiente a las conclusiones del proyecto.

III. DESCRIPCIÓN DEL ANALIZADOR SNS

El proyecto de grado SnS surgió con el interés de obtener un analizador de protocolos para la interfaz ITU V.35 que cumpliera con ciertos requisitos básicos como el bajo costo y la portabilidad del mismo por parte de los técnicos que lo utilicen.

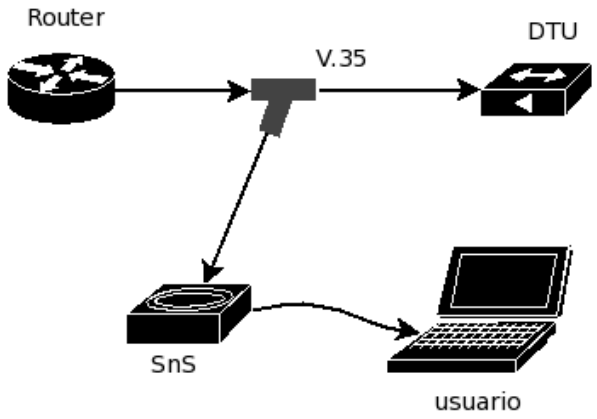
Por ser un proyecto de grado, la duración total del mismo insumió los tres semestres destinados a ello. En cuanto a los requerimientos económicos, en una primera instancia se planteó un monto máximo de 1000 USD para la compra de los componentes del analizador, debiendo luego ampliar dicho monto debido a los costos que insumió la importación del hardware

A estos requisitos se sumaron los requisitos técnicos inherentes al analizador mismo. A continuación se listan los requerimientos técnicos solicitados por el cliente.

1. Lectura de dos canales o puertos.
2. Velocidad máxima de lectura de al menos 2.048Mbps por canal sincrónico.
3. Interfaz ITU V.35
4. Cada trama deberá tener una marca de tiempo, insertada con una precisión mayor a 1ms.
5. Formato de datos HDLC.

6. Tamaño y peso que permitan la portabilidad del equipo analizador.
7. Conexión vía Ethernet a una PC portátil.
8. Almacenamiento de datos adquiridos en un archivo.
9. Análisis de datos con un programa que permita ser accedido por una interfaz gráfica.

El equipo SnS es capaz de adquirir datos en forma continua, leyendo de los dos canales simultáneamente a una velocidad máxima de 2Mbps por canal sincrónico. Se insertan marcas de tiempo a cada trama. La conexión a una PC portátil se realiza mediante la interfaz Ethernet. La interconexión entre el analizador y los equipos DTE y DCE es a través de un cable tipo T, el cuál no afecta la transmisión existente.



El formato de datos capturados corresponde a tramas HDLC (High-Level Data Link Control), incluyendo entonces a CHDLC (Cisco HDLC), Frame Relay y PPP (Point to Point Protocol). Los datos adquiridos son almacenados en un archivo para luego ser analizados con el software Ethereal. El formato del mencionado archivo es tal que el software Ethereal lo reconoce y por tanto es capaz de analizar. Por las características de los datos en cuanto a: precisión de marcas de tiempo mayor a un milisegundo, indicar el sentido de las tramas y las capas de enlace antes mencionadas, los datos son guardados con el formato Network Associates Sniffer.

En su primera versión final, el equipo SnS se basa en una plataforma de PC comercial con una placa compacta y moderna, siendo sus dimensiones 300 mm x 280 mm x 56mm y un peso aproximado de 2,5Kg, incluyendo el cable T de interconexión.

El componente principal de hardware del analizador SnS es la tarjeta adquisidora de datos Sangoma S5141, capaz de adquirir datos de los puertos primario y secundario simultáneamente a una velocidad máxima de 2 Mbps. La tarjeta fue configurada con el formato de trama HDLC, a partir del cual se realizaron las capturas de formatos de tramas CHDLC, PPP y Frame Relay.

El sistema operativo (SO) elegido para desarrollar el software del analizador es LINUX en la distribución Mandrake 10.1. Sobre el mencionado SO se desarrolló el software de captura, utilizando para esto el lenguaje de programación C y las librerías (APIs) del fabricante Sangoma.

El software desarrollado realiza la captura de datos de ambos puertos en forma simultánea e independiente, a una velocidad máxima de 2Mbps. Una vez iniciada la captura, el programa principal es capaz de capturar los datos de ambos puertos y según la condición seleccionada de fin de captura atenderá la interrupción. Los datos son guardados en archivos independientes, con los cuales se realiza el merge de los datos según las marcas de tiempo insertadas. Una vez efectuado y guardado el archivo final con los datos, el usuario podrá analizar la captura con el software analizador Ethereal.

Como se mencionó anteriormente, para el análisis de los datos capturados se utiliza el programa Ethereal, el cual es un analizador reconocido tanto en el ámbito académico como profesional por su gran versatilidad y por su calidad de software libre.

En el analizador SnS se encuentran instalados un servidor vnc y un servidor ssh. Estos servidores brindan al usuario la posibilidad de controlar el analizador remotamente como también realizar transferencias de archivos a una PC portátil mediante una conexión de red, lo cual implica la ventaja de trabajar en forma remota si el usuario así lo prefiere. Una opción puede ser la de conexión remota vía Internet, siendo ésta opción sumamente ventajosa hoy en día por ser las conexiones a Internet de tan fácil acceso.

El analizador, al estar basado en una plataforma PC, brinda además la posibilidad de conexión de monitor y teclado y por lo tanto de ser controlado directamente.

Para inicializar el programa SnS, el cual da inicio a la captura, se debe invocar al mismo pasándole diferentes parámetros. Dentro de los parámetros que deberá seleccionar el usuario, se encuentra el formato de trama, la condición de fin para la captura y el nombre del archivo en el cual será guardada la captura para su posterior análisis.

El analizador SnS puede ser controlado remotamente desde una PC portátil con sistema operativo Windows conectado al analizador vía Ethernet. Para esto se brinda una interfaz grafica instalada en la PC con sistema operativo Windows. Por medio de ésta interfaz se lanzan los comandos al analizador utilizando el protocolo ssh. Los comandos ejecutados remotamente dan inicio a la captura, setean la condición de fin de captura y copian el archivo final desde el analizador SnS hacia la PC portátil, para su posterior análisis con el Ethereal.

Si el usuario lo desea, podrá abrir una consola del analizador desde el PC portátil utilizando el protocolo ssh. Desde esta consola el usuario puede iniciar la captura, pasando los parámetros necesarios al programa SnS. Uno de los parámetros consiste en la opción de trabajar en modo interactivo, en el cual se le pregunta al usuario el formato de trama, condición de fin y nombre del archivo.

Entre las opciones que presenta esta versión del analizador, con respecto al formato de trama a seleccionar, se encuentran disponibles los protocolos CHDLC, PPP y Frame Relay. En cuanto a las opciones de finalización de captura es posible elegir la finalización manual, o sea en el instante que el usuario así lo decida, la finalización por tiempo así como también se presenta la opción de seleccionar la cantidad de paquetes a capturar. El usuario debe elegir el nombre del archivo en el cual será guardada la captura para su posterior análisis, sin que sea necesario agregar una extensión particular de archivo.

IV. HARDWARE

El hardware del analizador SnS es una PC portátil (mini computadora) con una tarjeta PCI que proporciona una interfaz V.35. Como se mencionó anteriormente la tarjeta utilizada captura los datos de ambos puertos a 2.048Mbps.

El chasis final del analizador SnS, incluye una tarjeta madre de bajo consumo con un slot PCI, una tarjeta adquisidora de datos con interfaz V.35, un disco duro para instalar el sistema operativo y el software de captura.

La tarjeta madre trae incorporada una interfaz Ethernet, por medio de la cual es posible el manejo remoto del analizador desde otra computadora, mediante un servicio de tipo vnc y/o ssh.

Al ser una PC con características estándar se tienen dos alternativas con respecto a la ubicación del software analizador. Una opción consiste en instalar el software analizador en la mini PC, como es el caso del SnS, lo cual tiene la ventaja de que los datos capturados son analizados directamente, sin tener que transferirlos por la red. La segunda alternativa es instalar el software analizador en una PC portátil, con lo cual los datos capturados son transferidos a la PC del usuario. La ventaja de esta opción reside en que los requerimientos de procesador y disco de la mini PC son menores.

A. Tarjeta adquisidora Sangoma S5141.

1) Proceso de Adquisición

La adquisición de datos se realiza sobre la base de una tarjeta modelo S5141 del fabricante Sangoma. La mencionada tarjeta realiza la captura en crudo (raw) de los datos desde el DTE y DCE e inserta las marcas de tiempo o *time stamp* dentro del ambiente del Kernel de la versión de Linux utilizada. Posteriormente la aplicación de usuario toma los datos con una estructura determinada y los escribe en un archivo.

La tarjeta Sangoma es una interfaz serial que posee dos puertos, los cuales admiten diferentes configuraciones. En el caso del analizador SnS ambos puertos fueron configurados únicamente para recibir datos con el estándar V.35, utilizando el reloj externo proveniente de las señales generadas por el DCE y DTE.

La tarjeta serial S5141 trabaja con el driver llamado wanpipe , sobre el cual se utilizó un stack multiprotocolo. Si se seleccionan las APIs multiprotocolo, la tarjeta trabaja con el protocolo raw HDLC sobre el cual es posible desarrollar aplicaciones que son capaces de trabajar con distintos protocolos.

La capa wanpipe API hace uso de la arquitectura de los sockets de Linux, en donde Sangoma desarrolla sus módulos. Estos módulos fueron instalados con el driver para manejar los datos capturados y pasarlos a la aplicación de forma segura. A estos drivers se le realizaron las modificaciones necesarias que permitieron insertar las marcas de tiempo por software.

Los módulos desarrollados en lenguaje de programación C, tienen definida una estructura de datos. Cuando la tarjeta solicita una interrupción, devuelve ese formato. La estructura con la cual son pasados los datos a la aplicación de usuario fue modificada con el fin de agregar a la estructura dos variables enteras. Antes de pasar los datos al

usuario, el Sistema Operativo setea estas variables, una con segundos y otra con micro segundos, utilizando la estructura timeval de la librería Time de C. Esta es la forma en que las marcas de tiempo son insertadas por software.

En el hardware de la tarjeta adquisidora Sangoma S5141, dentro de la estructura que devuelve, existe un campo time stamp el cual consiste en un contador de 0 a 65535 y cuya precisión es de milisegundos. Este campo no es utilizado en el proyecto del analizador SnS debido a que la precisión de un milisegundo no es suficiente.

Una vez que el Kernel ha manejado los datos, la aplicación de usuario se hace cargo de los mismos lo antes posible para que no exista la posibilidad de que estos sean sobrescritos por datos recientemente capturados.

Los retardos comprometedores se concentran en el cálculo de la time stamp insertada por software y cada vez que la aplicación interactúa con el Kernel de Linux para la transmisión de datos.

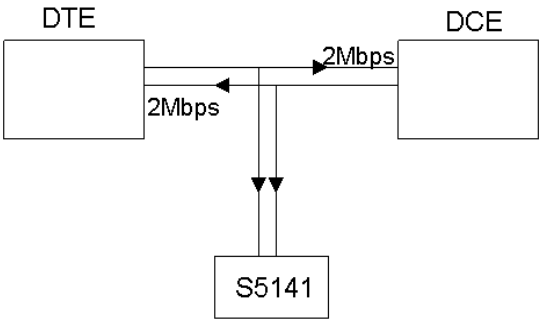
B. Conexión T

Como mencionamos anteriormente, el hardware para la adquisición de datos es la tarjeta Sangoma S5141. La tarjeta posee dos puertos seriales donde cada uno de ellos puede trabajar de forma independiente o como en el caso del analizador SnS pueden ser configurados en modo de solo recepción.

Para poder capturar los datos a transmitirse entre el DCE y el DTE a través del estándar V35 se realizó una conexión en T. El pinout del estándar V.35 se encuentra bien definido, de ahí fue necesario cablear cada línea a la tarjeta adquisidora de datos.

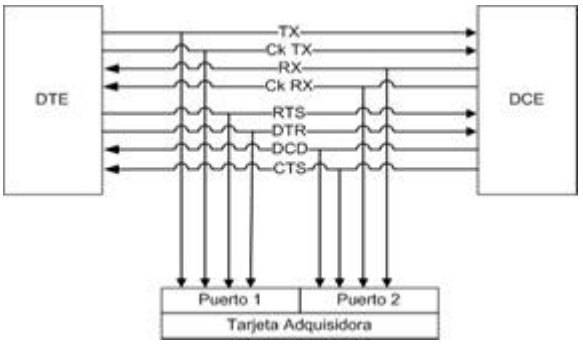
Como se mencionó en varias oportunidades, para la adquisición de datos son utilizados ambos puertos, en el puerto primario son capturados los datos desde el DTE y desde el puerto secundario son capturados los datos provenientes del DCE. Para que esto sea posible es necesario intercalar el analizador SnS en la línea de transmisión de los equipos DTE y DCE.

Un esquema de la conexión se presenta en la siguiente figura.



Esquema de la conexión con el cable T.

La siguiente figura detalla la conexión de los equipos DTE y DCE al analizador SnS. Las señales se diferencian en provenientes desde el DTE, que son conectadas al puerto 1 de la tarjeta Sangoma y las señales que se originan en el DCE, las cuales son conectadas al puerto 2 de la misma.



Las líneas de control son cableadas con el fin de consultar su estado y mostrarlas al usuario..

El cableado se realiza soldando pin a pin, dada la estabilidad de los voltajes no es necesario realizar ninguna adaptación electrónica.

El cable tipo T tiene físicamente tres conectores, dos conectores M34, uno hembra y otro macho para interponerse en la línea y un conector DB37 que va a la tarjeta adquisidora.

C. Tarjeta madre del analizador

La tarjeta madre VIA EPIA Mini-ITX es una de las mothers más pequeñas disponibles en el mercado. Su plataforma x86 optimiza su potencial sin sacrificar la flexibilidad del diseño. El diseño de la tarjeta se basa en un procesador VIA Eden ESP sin ventilador, como su nombre lo indica (VIA EPIA 5000 Fanless), debido a que sus bajos requerimientos de consumo no lo hacen necesario. La

tarjeta trae incorporado un procesador multimedia VIA C3. La placa VIA EPIA Mini-ITX es la plataforma ideal para una ilimitada variedad de proyectos basado en una plataforma PC.

D. Chasis y fuente

En la elección del chasis se tuvo en cuenta la necesidad de que el proyecto fuera reproducible, por lo que se optó por uno disponible en el mercado aunque esto implicase elevar el costo del equipo SnS. Finalmente se optó por importar un paquete que incluía el chasis con fuente y ventilador y la tarjeta madre.

V. SOFTWARE

A. El Sistema Operativo

La elección del Sistema Operativo (SO) en el cual se desarrolló el analizador SnS fue una decisión que derivó de la elección del hardware. Cuando se optó por la tarjeta Sangoma S5141, los fabricantes especificaron que el proyecto sería viable únicamente si se desarrollaba sobre el SO Linux, debido a que las librerías (APIs) que aportaban los fabricantes estaban implementadas en lenguaje de programación C y para SO Linux. El potencial de un Sistema Operativo de código abierto brindó la posibilidad desarrollar el software para el proyecto. Cabe destacar la ventaja del hecho de no haber tenido que pagar licencias de software, algo no menos importante en el desarrollo de un proyecto al plantear los costos y por tanto la viabilidad económica del mismo.

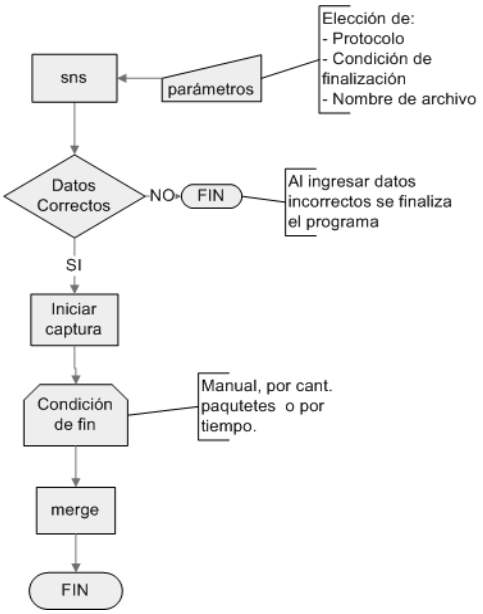
B. Aplicación SnS

El programa SnS realiza la captura de los datos provenientes de las dos interfaces V.35 del analizador. Ambas están conectadas de forma de escuchar los datos que transmiten los equipos a analizar, DCE y DTE. La aplicación toma los datos a través de sockets y es responsable de sacarlos del hardware con la velocidad necesaria para que los sockets no se llenen y por tanto se descarten paquetes. Con los datos que obtiene de cada socket son creados dos archivos. Luego que finaliza la captura de datos se crea un tercer archivo, el cual luego será analizado con el software Ethereal. El mencionado archivo contiene los datos capturados ordenados cronológicamente.

El software se puede dividir en dos capas, capa de aplicación y capa de Kernel del sistema operativo. Esta última es quien trabaja sobre el hardware extrayendo los datos hacia la aplicación e insertando las marcas de tiempo luego de atender la interrupción que el hardware genera al recibir una nueva trama.

La aplicación SnS es invocada al correr el comando sns seguido de diferentes parámetros. La aplicación necesita los siguientes datos iniciales:

1. Protocolo de capa de enlace:
   - CHDLC
   - Frame Relay
   - PPP
2. Condición de fin de captura:
   - Manual
   - Por tiempo
   - Cantidad Paquetes
3. Nombre del archivo final.



Esquema de la aplicación sns

Los datos iniciales pueden pasarse como parámetros al ejecutar la aplicación o en caso de escoger el modo interactivo (sns –v) son solicitados uno a uno por la aplicación. En caso de no indicar ningún (sns) o utilizarlos inconsistentemente se despliega la ayuda.

Si se corre en consola el comando sns, son desplegados en pantalla los parámetros que pueden ser pasados a la aplicación. Si el comando escrito incluye sns –v se ingresa al modo interactivo donde son solicitados al usuario los

datos iniciales. En el modo interactivo no son tomados en cuenta más parámetros que los solicitados.

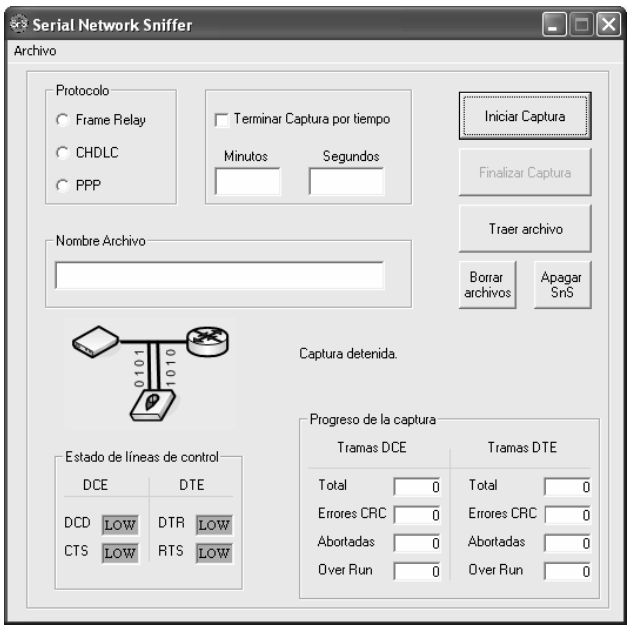
La aplicación Windows utiliza el modo por parámetros, pero esto es oculto al usuario. Se detalla más adelante

El control del analizador SnS se centró en la opción de manejo remoto que presenta el mismo a través del puerto Ethernet, desde una PC portátil, con sistema operativo Windows ya que este será el caso más utilizado por el cliente.

Una vez iniciado el programa, la interfaz gráfica del mismo solicita al usuario que seleccione el protocolo de capa de enlace elegido, seguido de la condición de fin de captura. Si la condición de fin de captura elegida es por tiempo se debe especificar el tiempo que durará la captura. Se selecciona el nombre del archivo con que será guardada la captura y finalmente se debe lanzar la captura.

Una vez finalizada la captura, el usuario debe copiar el archivo con los datos a su computador personal para su posterior análisis con el software Ethereal.

El lenguaje de programación Visual Basic, permitió desarrollar una interfaz gráfica GUI, amigable y de muy sencillo manejo para el usuario, por medio de la cual es posible realizar todas las tareas antes mencionadas. A continuación se puede visualizar la interfaz GUI del analizador SnS.



La aplicación de captura en el analizador se controla mediante la ejecución de comandos remotos desde Windows a Linux. Esto se logra a través de los programas PUTTY, PLINK y PSCP.

La aplicación principal luego de recibir los datos en forma correcta da comienzo a la captura creando dos procesos secundarios mediante el comando fork(). Estos procesos corren en background y cada uno realiza la captura de uno de los puertos, utilizando la API provista por el fabricante de la tarjeta Sangoma S5141, y guardando los datos capturados en un archivo con formato pcap.

Se optó por realizar procesos para la captura debido a la necesidad de minimizar el tiempo de procesamiento. Se intentó trabajar con hilos POSIX (IEEE) para la captura desde ambos puertos, pero el tiempo de atención entre un hilo y otro originaba pérdidas de paquetes. Esto llevó finalmente a trabajar con procesos. Al trabajar con esta estructura de programación pudimos verificar que ya no se perdían paquetes.

Cada proceso permanece en loop esperando a leer una trama. Cuando llega una trama, el programa la lee, guarda los datos en un archivo y vuelve a esperar la siguiente trama.

En el transcurso de la captura se trabaja con dos formatos de archivo, formato pcap y formato Network Associates Sniffer. El formato pcap es utilizado para guardar los archivos intermedios durante el proceso de captura, debido a la simpleza de los encabezados de los paquetes a escribir por cada trama capturada. El formato Network Associates Sniffer es utilizado para el formato del archivo final. Esta forma de trabajo incrementa la performance del código de escritura de los archivos intermedios, dado que el formato Network Associates Sniffer tiene una estructura de mayor tiempo de procesamiento y no es necesario recargar el código mientras se capturan las tramas desde la línea.

El proceso principal espera por la condición de fin de captura. Si el usuario decide no esperar el tiempo necesario para que se cumpla la condición de fin establecida inicialmente, se brinda la opción de finalizar la captura en forma manual. En ambos caso, el proceso principal entenderá que se cumplió la condición de fin de captura y terminará los procesos de captura de cada puerto enviando la señal TERM. Aguardará un instante para dar tiempo a que se cierren los archivos de captura de cada puerto y comenzará el merge de los datos. El resultado final es un archivo con formato basado en la estructura de archivo de Network Associates Sniffer (DOS-Based). Se seleccionó este formato debido a que permite agregar el campo de sentido de tramas, no siendo esto posible con el formato pcap.

Cada proceso de captura, es responsable de obtener los datos desde el hardware a través de sockets provistos por las APIs de Sangoma y escribirlos en los archivos puerto1.bin y puerto2.bin respectivamente. Los archivos son escritos de a tramas, esto asegura que cuando se mata el proceso no quede una trama por la mitad.

La condición de fin de captura ofrece tres opciones. Por tiempo, en donde el usuario del analizador le indica por cuanto tiempo va a capturar paquetes. En forma manual el usuario indica el inicio y el fin. Por cantidad de paquetes, el usuario puede indicar cuantos paquetes desea capturar. Se deja el código abierto para futuras implementaciones de inicio y fin de captura.

1) Realización del Merge

Al finalizar la captura el programa principal envía señales TERM a los procesos. Estos la manejan cerrando el archivo y finalizando.

A partir de los dos archivos puerto1.bin y puerto2.bin el archivo final es guardado con formato Network Associates Sniffer (DOS-Based). Este formato permite agregar una bandera en el encabezado para identificar el sentido de las tramas. Cuando el archivo es analizado por el software Ethereal, este se fija en la bandera y según el valor le da el sentido DTE->DCE o DCE->DTE.

Para ingresar los datos se van chequeando los archivos de cada puerto, se leen uno a uno los paquetes y se escriben en el archivo final en orden cronológico. Cuando no hay más paquetes que leer se escribe el final de archivo.

En general los formatos de archivos que entiende el Ethereal, comienzan por un encabezado de archivo, luego un encabezado por paquete y puede terminar con fin de archivo.

C. El Software Analizador de Protocolos

El software Ethereal es un analizador de protocolos probado y disponible en el medio. Es utilizado para efectuar análisis, así como también solucionar distintos tipos problemas que se presentan en redes de comunicaciones, para desarrollo de software y protocolos y como herramienta didáctica en la educación. Cuenta con todas las características de un analizador de protocolos estándar.

Ethereal está desarrollado bajo licencia de código abierto y se ejecuta sobre la mayoría de los Sistemas Operativos. A continuación se listan algunos de las características más relevantes del software.

1. Es mantenido bajo Licencia GNU
2. Es capaz de leer datos almacenados en un archivo
3. Presenta una interfaz flexible y amigable para el usuario
4. Posee diferentes opciones de filtrado
5. Soporta formatos de archivo estándar
6. Se ejecuta en más de 20 plataformas
7. Soporta más de 750 protocolos
8. Presenta la capacidad de leer archivos de captura de más de 20 productos
9. Posee la gran ventaja de ser un analizador de libre acceso y de constante actualización de su lista de protocolos.

Pudo ser comprobado que si se indica el protocolo inicial, en este caso particular CHDLC, Ethereal se encarga de los protocolos de capas superiores.

Ethereal es capaz de analizar diferentes protocolos, entre los cuales se incluyen HDLC, CHDLC, Frame Relay y PPP, los cuales son específicamente los protocolos de interés para el proyecto.

Se buscó un formato capaz de presentar el sentido de las tramas capturadas o sea si la trama tenía origen en el DTE o en el DCE.

Se realizaron consultas en la página de Ethereal y el formato de archivo que permite desplegar en pantalla la información de sentido, que tenga precision en marcas de tiempo de microsegundos o superior y que acepte capas de enlace Frame Relay, PPP o CHDLC es el Network Associates Sniffer.

D. Control de la capacidad del disco

Como se mencionó anteriormente, el analizador SnS almacena los datos capturados en el disco duro. Si una captura iniciada demorara en finalizar el tiempo suficiente como para llenar la capacidad disco, puede resultar en la pérdida de los datos capturados si no es manejada correctamente la señal de disco lleno del sistema operativo.

Para evitar perder la captura, antes de llenar el disco se desarrolló en lenguaje Perl un script que da fin a la captura.

VI. PRECISIÓN DEL ANALIZADOS SNS.

El analizador cumple con los requerimientos mínimos de no perder tramas y de tener precisión superior al milisegundo en las marcas de tiempo. A continuación resumimos las características de precisión del SnS y mencionamos los factores que inciden en la misma.

A. Características y factores que determinan la precisión.

El SnS inserta las marcas de tiempo por software, es decir lo hace el sistema operativo al momento de atender la interrupción solicitada por la tarjeta adquisidora luego de recibir cada trama.

El tiempo que el sistema operativo demora en atender la interrupción se llama latencia. Esta varía en función de las demás tareas que se encuentre realizando el sistema operativo. La latencia variable es la principal limitante en cuanto a la precisión de las marcas de tiempo. Su rango de variación y por lo tanto su efecto de error es mayor si la utilización del canal es alta. A velocidades de 2Mbps y con una utilización cercana al 100%, estamos en el caso más exigente para el que está diseñado el analizador y la precisión es de 1ms. A porcentajes menores de utilización del canal o a velocidades más bajas, la precisión mejora, hasta llegar a 0,1ms.

La latencia depende fuertemente de la demanda de otros procesos al sistema operativo. De haber procesos altamente demandantes de recursos la latencia aumenta e incluso puede ocasionar pérdida de tramas. Por este motivo recomendamos no ejecutar tareas externas al analizador durante una captura de datos.

El hecho de que la marca de tiempo es colocada al terminar de recibir la trama debe tomarse en cuenta al realizar capturas en paralelo con otros analizadores que coloquen la marca de tiempo al comienzo de la trama. Para comparar las marcas de tiempo debe descontarse el tiempo de serialización en las marcas del SNS. Este tiempo es igual al largo de la trama en bits, incluyendo los bits adicionales debidos al bit stuffing , dividido la frecuencia de reloj del canal.

Otro efecto a tener en cuenta en el caso de realizar un análisis con varios analizadores simultáneamente es el de la sincronización de los mismos. Cada analizador tiene un reloj independiente afectando esto a las marcas de tiempo en dos puntos relevantes. El primero es que el origen de tiempos será distinto, ocasionando un defasaje constante entre las marcas de tiempo de uno y otro. Como segundo punto observamos que la frecuencia de reloj diferirá levemente, ocasionando una deriva en las marcas de tiempo (el reloj de un analizador adelantará respecto al reloj del otro) Estos efectos pueden corregirse fuera de línea, como lo hicimos en las pruebas que realizamos para testear la precisión de las marcas de tiempo, pero en los análisis normales esto no es relevante.

Las pruebas que realizamos consistieron en capturar en serie con un analizador HP Advisor los datos correspondientes a distintas transferencias de archivos y ejecuciones de pings. Luego comparamos las marcas de tiempo insertadas por el HP y por el SNS.

Con el cuidado de no exigir al analizador con procesos altamente demandantes de recursos, se observan las siguientes características.

1. Capacidad de capturar la totalidad de las tramas de la interfaz V.35 hasta una velocidad de 2.048Mbps.
2. Precisión de las marcas de tiempo mejor a 1 ms.
3. Precisión de las marcas de tiempo de 0,1 ms en las siguientes circunstancias:
   A 2Mbps si el canal no está utilizado al máximo.
   A 64Kbps con máxima utilización del canal.
4. Las marcas de tiempo son insertadas al terminar de recibir cada trama.

B. Pruebas realizadas y observaciones importantes

La forma más sencilla de probar un equipo de medida, como por ejemplo el analizador SnS, es realizar un conjunto de mediciones en distintas situaciones y comparar los resultados con las mediciones efectuadas por un equipo de medición más preciso.

En nuestro caso tuvimos acceso a equipos analizadores muy precisos, modelos HP Advisor.

En primera instancia confirmamos la exactitud de los analizadores HP Advisor. Para ello realizamos capturas con dos equipos en serie y examinamos luego la concordancia de las marcas de tiempo insertadas. La exactitud observada fue de 3 microsegundos y una muy leve diferencia de frecuencia de reloj.

Realizamos pruebas a distintas velocidades de transferencia y con distintos tamaños de tramas. Para ello utilizamos 2 routers conectados a través de una interfaz v.35 de la cual capturamos los datos. A su vez a cada router conectamos por la interfaz Ethernet una PC. El tráfico lo generamos transfiriendo archivos entre las PCS y realizando pings entre los routers. De los resultados de estas pruebas obtuvimos el error del analizador SnS que detallamos en la sección anterior.

Para comparar las marcas de tiempo de uno y otro tuvimos en cuenta los efectos mencionados en la sección anterior: inserción de la marca de tiempo al comienzo de la trama por el HP y al final por el SNS (corregimos el tiempo de serialización), diferencias en orígenes de tiempo y relojes no sincronizados.

VII. COSTO DEL ANALIZADOR SNS

En este punto damos el costo del analizador SnS. Es claro que en el momento de reproducir el analizador deberá tenerse en cuenta las horas que consume a un técnico capacitado reproducir el equipo, estimamos unas 16hs para el armado del hardware, instalación de sistema operativo, drivers y la configuración de los mismos. También deben sumarse materiales menores para su armado como ser conectores, cables, etc. Los costos presentados en la siguiente tabla son en origen a la fecha 22 de marzo de 2006 .

COMPONENTES DEL ANALIZADOR SnS			
DESCRIPCIÓN	**MARCA**	**MODELO**	**COSTO**
Mother con procesador de bajo consumo.	VIA EPIA	5000 Fanless	USD 100,00
Gabinete con fuente.	Travla C150		USD 116,00
Memoria DIM 128MB.	Kingstone		USD 31,00
Disco Duro 10GB.	Western Digital		USD 40,00
Tarjeta WAN, PCI.	Sangoma	S5141	USD 570,00
Cable T			USD 150
		TOTAL	**USD 857,00**

VIII. CONCLUSIONES

En el presente capítulo se resumen algunas de las consideraciones sobre el desarrollo del proyecto, el producto alcanzado y las posibles extensiones del mismo que avizoramos.

A. El desarrollo del proyecto.

El proyecto del analizador SnS surgió de modo de satisfacer el interés del cliente en poseer un analizador de bajo costo, portátil y fácilmente reproducible que sustituyera los analizadores de alto costo HP que son utilizados hoy en día por el cliente.

Debido a las necesidades de desarrollo durante el transcurso del proyecto debimos pasar de simples usuarios Linux a conocer las opciones que brinda un sistema operativo de código abierto, como ser cargar módulos personalizados al Kernel de Linux y el manejo de sockets. Para la personalización del Kernel se contó con el apoyo en línea de quienes desarrollan el software de la tarjeta adquisidora Sangoma.

Inicialmente las marcas de tiempo fueron insertadas por hardware con precisión de un mili segundo, lo cual no era suficiente al trabajar con frecuencias de 2MHz. Para aumentar la precisión y gracias a trabajar sobre un sistema operativo de código abierto, se modificó el driver de la tarjeta de modo que sea el sistema operativo quien coloque la marca de tiempo, en el momento que atiende la interrupción que se genera cuando la tarjeta recibe una trama.

Durante su desarrollo, el proyecto SnS recorrió diferentes etapas, las cuales clasificamos en niveles de la siguiente forma.

Nivel físico: Se realizó el estudio y elección del hardware así como también el cable de conexión T.

Nivel Sistema Operativo: Estudio y desarrollo de módulos para el proyecto.

Nivel Aplicación: Desarrollo de una aplicación que manejara los datos obtenidos.

Nivel Interfaz-Usuario: Enfoque y desarrollo de una interfaz amigable para el usuario.

Nivel Desarrollador-Cliente: Se interactuó con el cliente final, se defendió el proyecto y su presupuesto.

El hecho de haber recorrido los niveles antes mencionados deja en claro lo difícil que resulta abarcar todas las áreas en forma vertical.

B. El producto final

El analizador SnS es un equipo portátil, capaz de adquirir datos de dos interfaces V.35 a una velocidad máxima de 2Mbps. Es un producto versátil, de costo sensiblemente menor al de los analizadores, con prestaciones similares, existentes en el mercado. La plataforma se basa en una mini PC con sistema operativo Linux con ambiente gráfico, de forma que permite visualizar el análisis de los datos en él.

El SnS cuenta solamente con la interfaz V.35, logrando así reducir costos y dimensiones.

Dentro de los requerimientos se nos solicitó que fuera desplegada la frecuencia del reloj a la cual era realizada la captura. Se investigó en manuales y no se encontró la información buscada, lo que llevó a consultar a los fabricantes, quienes atendieron la consulta con mucho interés pero desconocían la solución al problema y se encuentran estudiando el caso.

El hecho de trabajar sobre una plataforma PC estándar x86 brindó múltiples facilidades a la hora de desarrollar aplicaciones.

C. Desarrollos a futuro.

1) Disminución del tamaño del archivo de captura

Si se pretende analizar una comunicación por varias horas y guardar todos los datos que se transfieran, se obtiene un archivo de captura de tamaño considerable.

Calculemos el tamaño del archivo vs. el tiempo de captura en horas. Una hora tiene 3.600 segundos. Por cada segundo, si el enlace a analizar es a 2Mbps y se utiliza a pleno, se transmiten aproximadamente 250Kbytes por segundo por canal. Por lo tanto entre ambos canales tenemos 0,5 MBytes por segundo. El archivo de captura aumenta entonces en 3.600 * 0,5 MBytes o sea a una tasa de 1.8 Gbyte por hora.

En muchos casos podría obtenerse un archivo mucho más pequeño que contenga todas las tramas que interesa analizar. Esto podría implementarse mediante filtros de captura, similar a los que utiliza libpcap. Cada trama capturada sería filtrada inspeccionando su protocolo y luego solo en caso de que corresponda sería guardada a disco.

2) Iniciar captura por eventos

Sería interesante también poder iniciar la captura luego de determinado evento, por ejemplo un error de CRC. Podría implementarse con un buffer circular en memoria RAM, en el que constantemente se guardarían las tramas capturadas. Al detectar una trama con error en CRC se guardaría a disco el contenido del buffer, por ejemplo de 100 tramas. Luego se continuaría adquiriendo datos hasta una condición de fin la cual podría ser un timer.

3) Sugerencias

Otro posible desarrollo sería lograr integrar la tarjeta Sangoma S5141 al Ethereal, de modo de poder iniciar y detener la captura en forma manual a través de la mencionada interfaz. Para ello se debe modificar la librería libpcap.

Álvaro García tiene interés en incorporar al analizador SnS la posibilidad de generar tráfico en forma personalizada, por ejemplo en cuanto al largo de las tramas