UNIVERSIDAD DEL AZUAY

FACULTAD DE ADMINISTRACION DE EMPRESAS

ESCUELA DE INGENIERIA DE SISTEMAS

ANÁLISIS DE PROTOCOLOS BASICOS REDES IP

MONOGRAFIA PREVIA A LA OBTENCION DEL TITULO DE INGENIERO DE SISTEMAS

ALUMNO:

JUAN E. CAMPOVERDE M.

Cuenca, a 14 de Febrero del 2003

Los conceptos y criterios vertidos en el presente documento son de estricta responsabilidad del autor.

Juan E. Campoverde M.

AGRADECIMIENTO

A todo el personal docente de la Universidad del Azuay y de manera muy especial para los señores Ingeniero Fernando Balarezo é Ingeniero Francisco Vázquez, personas que con su gran capacidad hicieron posible llegar a feliz termino este proyecto de tesis.

También agradezco a todo el grupo maravilloso de personas de la Universidad Politécnica de Madrid, por su tiempo y comprensión para con todos los ecuatorianos que tuvimos privilegio de contar como vuestros alumnos.

Y un agradecimiento muy especial a mis Señores Padres, por su gran entrega y apoyo durante todo momento de mi vida.

Juan Eugenio Campoverde Mora

DEDICATORIA

Doy gracias a DIOS, por haberme permitido el terminar otra etapa mas de mi vida y, dedico este trabajo monográfico a toda mi familia, y de manera muy especial a mis queridos Padres José y Mariana, por que con su consejo y apoyo; hicieron posible el culminar con éxito esta carrera universitaria.

También dedico este trabajo a todas aquellas personas que supieron brindarme su ayuda durante algún momento de mi vida.

Gracias.

Juan E. Campoverde M.

INDICE

1. PROTOCOLOS

1.1 INTRODUCCION

Los tres últimos siglos han estado dominados, cada uno de ellos, por una tecnología. El siglo XVIII fue la época de los grandes sistemas mecánicos que acompañaron a la Revolución Industrial. El siglo XIX fue la era de las máquinas de vapor. En el siglo XX, la tecnología clave ha sido la obtención, procesamiento y distribución de la información. Entre otros avances, hemos visto la instalación de la redes telefónicas mundiales, la invención del radio y la televisión, el nacimiento y crecimiento sin precedentes de la industria de las computadoras y el lanzamiento de satélites de comunicación.

Debido al rápido progreso de la tecnología, estas áreas están convergiendo rápidamente, y las diferencias entre juntar, transportar, almacenar y procesar información desaparecen con rapidez. Las organizaciones con cientos de oficinas que se extienden sobre una amplia área geográfica esperan

ser capaces de examinar la situación, aun de sus más remotos puestos de avanzada, oprimiendo un botón. Al crecer nuestra habilidad para obtener, procesar y distribuir información, también crece la demanda de técnica de procesamiento de información más avanzadas.

Para la comunicación de la información una herramienta software tecnológica son los protocolos.

Cuando tenemos dispositivos de hardware, separados geográficamente, existirán procedimientos para control de cada dispositivo implementados por procesos de software. Como los procesos ejecutan en hardware separado, deben intercambiar mensajes para coordinar la acción y obtener SINCRONIZACIÓN.

Para realizar el intercambio de mensajes debemos diseñar (cuidadosamente) los procedimientos o protocolos.

La principal característica, es la habilidad para

trabajar en un ambiente donde los periodos y secuencia de eventos es desconocida y se esperan errores en la transmisión de datos.

El termino protocolo lo usamos para describir el intercambio de información entre procesos.

1.2 PROTOCOLO ARP (Protocolo de resolución de direcciones)

Aunque cada máquina de Internet tiene una (o más) direcciones de IP, éstas no pueden usarse para enviar paquetes porque el *hardware* de la capa de enlace de datos no entiende las direcciones de Internet. Hoy día, la mayoría de los *hosts* están conectados a alguna LAN mediante una tarjeta de interfaz que sólo entiende direcciones de LAN. Por ejemplo, todas las tarjetas Ethernet que se han fabricado vienen equipadas con una dirección de Ethernet de 48 bits. Los fabricantes de tarjetas Ethernet solicitan un bloque de direcciones de una autoridad central para asegurarse de que dos tarjetas no puedan tener la misma dirección (para evitar conflictos en caso de que las dos tarjetas aparezcan en la misma LAN). Las tarjetas envían y reciben marcos con base en direcciones de Ethernet de 48 bits. No saben nada sobre las direcciones de IP de 32 bits.

Ahora surge la pregunta: ¿cómo se proyectan las direcciones de IP en las direcciones de capa de enlace de datos, como las de Ethernet? Para explicar el funcionamiento de esto, usemos el ejemplo de la **figura 1**, en la que se ilustra una universidad pequeña con varias redes clase C. Aquí tenemos dos Ethernet, uno en el departamento de informática, con una dirección IP de 192.31.65.0 y otra en el de ingeniería electrónica, con dirección de IP de 192.31.63.0. Éstas se conectan mediante un anillo FDDI del campus con una dirección IP de 192.31.60.0. Cada máquina de un Ethernet tiene una dirección de Ethernet única, etiquetada *E1* a *E6*, y cada máquina del anillo FDDI tiene una dirección FDDI, etiqueta *F1* a *F3*.

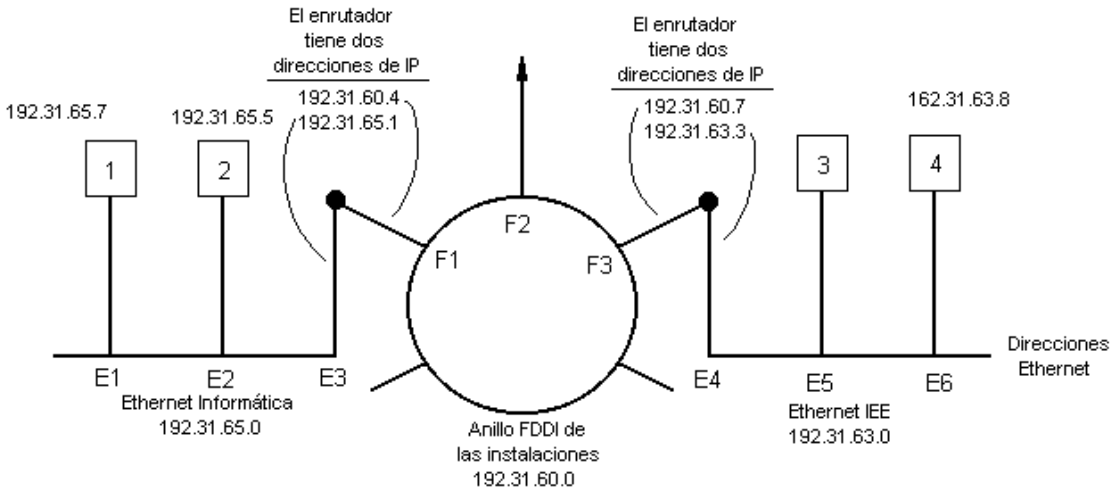


FIGURA 1. Tres redes clase C interconectadas; dos Ethernets y un anillo FDDI.

Comencemos por ver la manera en que un usuario del *host* 1 envía un paquete a un usuario del *host* 2. Supongamos que el transmisor sabe el nombre del receptor pretendido, posiblemente algo como *maria@eagle.cs.uni.edu*. El primer paso es encontrar la dirección IP del *host* 2, conocidos como *eagle.cs.uni.edu.* Esta búsqueda la lleva a cabo el sistema de nombres de dominio (DNS, *Domain Name System*). Por lo tanto, simplemente supondremos que el DNS devuelve la dirección IP del *host* 2 (192.31.65.5).

El *software* de capa superior del *host* 1 construye ahora un paquete con 192.31.65.5 en el campo de *dirección de destino* y lo entrega al *software* de IP para su transmisión. El *software* de IP puede ver la dirección y saber si el destino está en su propia red, pero necesita una manera de encontrar la dirección Ethernet del destino. Una solución es tener un archivo de configuración en algún lugar del sistema que proyecte las direcciones IP en las direcciones Ethernet. Esta solución ciertamente es posible pero, en organizaciones con miles de máquinas, el mantenimiento de estos archivos es una tarea susceptible a errores y con alto consumo de tiempo.

Una mejor solución es el *host* 1 envíe un paquete de difusión por el Ethernet preguntando: "¿quién es el dueño de la dirección IP 192.31.65.5?"La difusión llegaría a cada máquina del Ethernet 192.31.65.0, y cada una revisará su propia dirección IP. Sólo el *host* 2 responderá con su dirección Ethernet (*E2*). De esta manera, el *host* 1 aprende que la dirección de IP 192.31.65.5 está en el *host* con la dirección Ethernet *E2*. El protocolo para preguntar esto y obtener la respuesta se llama **ARP** *(Address Resolution Protocol,* **protocolo de resolución de direcciones**). Casi todas las máquinas de Internet lo ejecutan. Este protocolo se define en el RFC 826.
La ventaja del ARP sobre los archivos de configuración es la sencillez. El administrador del sistema no necesita hacer mucho, excepto asignar a cada máquina una dirección IP y decidir sobre las máscaras de subred. El ARP hace lo demás.

En este punto, el *software* de IP del *host* 1 construye un marco Ethernet dirigido a *E2*, pone el paquete IP (dirigido a 192.31.65.5) en el campo de carga útil, y lo arroja al Ethernet. La tarjeta de Ethernet del *host* 2 detecta este marco, lo reconoce como un marco para él mismo, lo recoge y causa una interrupción. El operador de Ethernet extrae el paquete IP de la carga útil y lo pasa al *software* de IP, que nota que está correctamente dirigido, y lo procesa.

Son posibles varias optimizaciones para hacer más eficiente el ARP. Por principio de cuentas, una vez que una máquina ha ejecutado el ARP, pone en caché el resultado por si necesita establecer contacto con la misma máquina pronto. La siguiente vez encontrará la proyección en

su propio caché, eliminando por tanto la necesidad de una segunda difusión. En muchos casos, el *host* 2 necesitará devolver una respuesta, lo que lo obliga también a ejecutar el ARP para determinar la dirección Ethernet del transmisor. Esta transmisión ARP puede evitarse habiendo que el *host* 1 incluya su proyección de IP a Ethernet en el paquete ARP. Al llegar la difusión del ARP al *host* 2, se ingresa el par (192.31.65.7, E1) en el caché de ARP del *host* 2 para su uso futuro. De hecho, todas las máquinas que operan con Ethernet pueden ingresar esta proyección en sus cachés de ARP.

Otra optimización es hacer que cada máquina difunda su proyección al arranque. Esta difusión generalmente se hace en la forma de un ARP que busca su propia dirección de IP. No debe haber una respuesta, pero un efecto secundario de a difusión es crear una entrada en el caché de ARP de todos. Si llega una respuesta, es que dos máquinas tienen asignada la misma dirección de IP. La nueva debe informar de ello al administrador del sistema y no debe arrancar.

Para permitir el cambio de las proyecciones, por ejemplo, cuando se descompone una tarjeta Ethernet y se reemplaza por una nueva (y por tanto con una dirección Ethernet nueva), las entradas en el caché de ARP deben terminar su temporización la transcurrir algunos minutos.

Ahora veamos la **figura 1** nuevamente, sólo que esta vez el *host* 1 quiere enviar un paquete al *host* 6 (192.31.63.8). El ARP fallará, porque el *host* 4 no verá la difusión (los enrutadores no reenvían las difusiones de nivel Ethernet). Hay dos soluciones. Primero, el enrutador CS podría configurarse para responder a las solicitudes de ARP para la red 192.31.63.0 (y posiblemente para otras redes locales). En este caso, el *host* 1 creará en el caché de ARP la entrada (192.31.63.8, E3) y felizmente enviará todo el tráfico para el *host* 1 creará en el caché de ARP la entrada (192.31.63.8, E3) y felizmente enviará todo el tráfico para el *host* 4 al enrutador local. Esta solución se llama **ARP apoderado** (*proxy* ARP). La segunda solución es hacer que el *host* 1 vea de inmediato que el dstino está en una red remota y simplemente envíe todo ese tráfico a la dirección Ethernet predeterminada que maneja todo el tráfico remoto, en este caso *E3*. Esta solución no requiere hacer que el enrutador CS sepa a cuáles redes remotas sirve.

De cualquier manera, lo que ocurre es que el *host* 1 empaca el paquete IP en el campo de carga útil de un marco Ethernet dirigido a *E3*. Cuando el enrutador CS recibe el marco Ethernet, retira el paquete IP del campo de carga y busca la dirección IP en sus tablas de enrutamiento, descubriendo que los paquetes para la red 192.31.63.0 deben ir al enrutador 192.31.60.7. Si el enrutador no sabe ya la dirección FDDI de 192.31.60.7, difunde un paquete ARP por el anillo y se entera de que la dirección de su anillo es *F3*. El enrutador inserta entonces el paquete en el campo de carga útil de un marco FDDI dirigido a *F3* y lo pone en el anillo.

En el enrutador EE, el operador de FDDI retira el paquete del campo de carga útil y lo entrega al *software* de IP, el cual descubre que necesita enviar el paquete al 192.31.63.8. Si esta dirección no está en su caché de ARP, difunde una solicitud ARP por el Ethernet EE y se entera de que la dirección de destino es *E6*, por lo que construye un marco Ethernet dirigido a *E6*, pone el paquete en el campo de carga útil, y lo envía a través del Ethernet. Al llegar el marco Ethernet al *host* 4, el paquete se extrae del marco y se entrega al *software* de IP para su proceso.

Una transmisión del *host* 1 a una red distante a través de una WAN funciona en esencia de la misma manera, excepto que esta vez las tablas del enrutador CS le indican que use el enrutador WAN cuya dirección FDDI es *F2*.

Protocolo de resolución de direcciones en reversa

El ARP resuelve el problema de encontrar la dirección de Ethernet correspondiente a una dirección IP dada. A veces tiene que resolverse el problema inverso: dada una dirección Ethernet, ¿cuál es la dirección IP correspondiente? En particular, este problema ocurre al iniciarse una estación de trabajo sin disco. Tal máquina normalmente recibirá la imagen binaria de su sistema operativo de un servidor de archivo remoto, pero ¿cómo conoce su dirección de IP?

La solución es usar el RARP (*Reverse Address Resolution Protocol,* **protocolo de resolución de direcciones en reversa**, definido en el RFC 903). Este protocolo permite que una estación de trabajo recién iniciada difunda su dirección Ethernet y diga, “mi dirección Ethernet de 48 bits es 14.04.05.18.01.25. ¿Sabe alguien por ahí mi dirección de IP?” El servidor RARP ve esta solicitud, busca la dirección de Ethernet en sus archivos de configuración y envía de regreso la dirección de IP correspondiente.

El uso del RARP es preferible a integrar una dirección de IP en la imagen de memoria, pues permite usar la misma imagen para todas las máquinas. Si la dirección de IP estuviera incluido en la imagen, cada estación de trabajo necesitaría su propia imagen.

Una desventaja del RARP es que usa una dirección de destino que contiene únicamente unos (difusión limitada) para llegar al servidor RARP. Sin embargo, tales difusiones no son reenviadas por los enrutadores, por lo que se requiere un servidor RARP en cada red. Para superar este problema, se ha inventado un protocolo alterno de arranque llamado **BOOTP**.

1.3 PROTOCOLO IP (Protocolo de interred)

Un lugar adecuado para comenzar nuestro estudio de la capa de red de Internet es el formato de los datagramas del IP mismos. Un datagrama IP consiste en una parte de cabecera y una parte de texto. La cabecera tiene una parte fija de 20 bytes y una parte opcional de longitud variable. El formato de la cabecera se muestra en la **figura 2.** Se transmite en orden *big endian*: de izquierda a derecha, comenzando por el bit de orden mayor del campo de *versión*. (SPARC es *big endian;* Pentium es *little endian*.) En las máquinas *little endian*, se requiere conversión por *software* tanto para la transmisión como para la recepción.

El campo de *versión* lleva el registro de la versión del protocolo al que pertenece el datagrama. Al incluir la versión en cada datagrama es posible hacer que la transición entre versiones se lleve meses, o inclusive años, ejecutando algunas máquinas la versión vieja y otras la versión nueva.

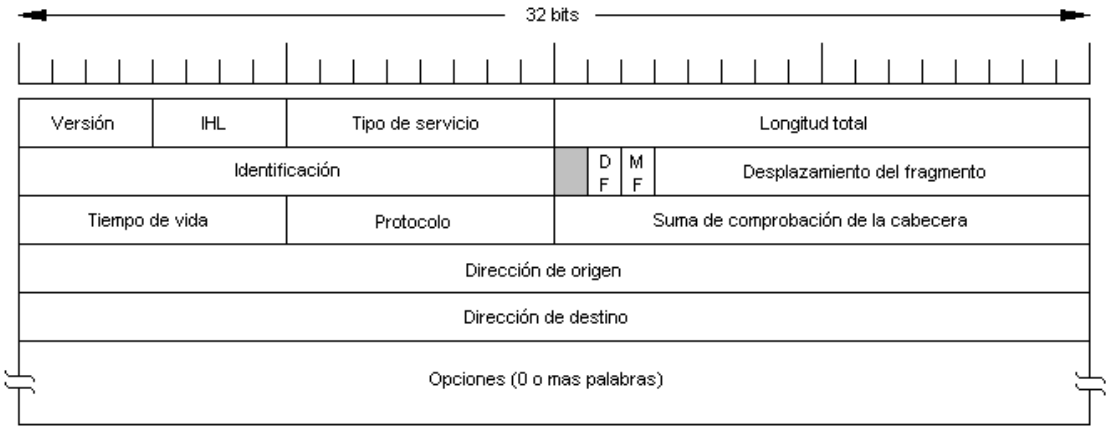


FIGURA 2. La cabecera IP (protocolo de Internet)

Dado que la longitud de la cabecera no es constante, se incluye un campo en la cabecera, *IHL*, para indicar la longitud en palabras de 32 bits. El valor mínimo es de 5, cifra que aplica cuando no hay opciones. El valor máximo de este campo de 4 bits es de 15, lo que limita la cabecera a 60 bytes y, por tanto, el campo de opciones a 40 bytes. Para algunas opciones, por ejemplo para una que registre la ruta que ha seguido un paquete, 40 bytes es muy poco, lo que hace inútil esta opción.

El campo de tipo de *servicio* permite al *host* indicar a la subred el tipo de servidio que quiere. Son posibles varias combinaciones de confiabilidad y velocidad. Para voz digitalizada, la entrega rápida le gana a la entrega precisa. Para la transferencia de archivos, es más importante la transmisión libre de errores que la transmisión rápida.

El campo mismo contiene (de izquierda a derecha) un campo de *precedencia;* tres indicadores, D, T y R; y dos bits no usados. El campo de *precedencia* es una prioridad, de 0 (normal) a 7 (paquete de control de red). Los tres bits indicadores permiten al *host* especificar lo que le interesa más del grupo {retardo (*delay*), rendimiento (*throughput*), confiabilidad (*reliability*)}. En teoría, estos campos permiten a los enrutadores tomar decisiones entre, por ejemplo, un enlace satelital de alto rendimiento y alto retardo o una línea arrendada con bajo rendimiento y poco retardo. En la práctica, los enrutadores actuales ignoran por completo el campo de *tipo de servicio*.

La *longitud total* incluye todo el datagrama: tanto la cabecera como los datos. La longitud máxima es de 65,535 bytes. Actualmente este límites es tolerable, pero con las redes futuras de gigabits se requerirán datagramas más grandes.

El campo de *identificación* es necesario para que el *host* de destino determine a qué datagrama pertenece un fragmento recién llegado. Todos los fragmentos de un datagrama contienen el mismo valor de *identificación*.

A continuación viene un bit sin uso y luego dos campos de 1 bit. *DF* significa no fragmentar (*Don´t Fragment*); es una orden para los enrutadores de que no fragmenten el datagrama, porque el destino es incapaz de juntar las piezas de nuevo. Por ejemplo, al arrancar una computadora, su ROM podría pedir el envío de una imagen de memoria a ella como un solo datagrama. Al marcar el datagrama con el bit *DF*, el transmisor sabe que llegará en una pieza, aún si significa que el datagrama debe evitar una red de paquete pequeño en la mejor trayectoria y tomar una ruta subóptima. Se requiere que todas las máquinas acepten fragmentos de 576 bytes o menos.

MF significa más fragmentos. Todos los fragmentos excepto el último tienen establecido este bit, que es necesario para saber cuándo han llegado todos los fragmentos de un datagrama.

El *desplazamiento del fragmento* indica en qué parte del datagrama actual va este fragmento. Todos los fragmentos excepto el último del datagrama deben tener un múltiplo de 8 bytes, que es la unidad de fragmento elemental. Dado que se proporcionan 13 bits, puede haber un máximo de 8192 fragmentos por datagrama, dando una longitud máxima de datagrama de 65,536 bytes, uno más que el campo de *longitud total*.

El campo de *tiempo de vida* es un contador que sirve para limitar la vida de un paquete. Se supone que este contador cuenta el tiempo en segundos, permitiendo una vida máxima de 255 seg; debe disminuirse en cada salto y se supone que disminuye muchas veces al encolarse durante un tiempo grande en un enrutador. En la práctica, simplemente cuenta los saltos. Cuando el contador llega a cero, el paquete se descarta y se envía de regreso un paquete de

aviso al *host* de origen. Esta característica evita que los datagramas vaguen eternamente, algo que de otra manera podría ocurrir si se llegan a corromper las tablas de enrutamiento.

Una vez que la capa de red ha ensamblado un datagrama completo, necesita saber qué hacer con él. El campo de *protocolo* indica la capa de transporte a la que debe entregarse TCP es una posibilidad, pero también está UDP y algunos más. La numeración de los protocolos es global en toda la Internet, y se define en el RFC 1700.

La *suma de comprobación de la cabecera* verifica solamente la cabecera. Tal suma de comprobación es útil para la detección de errores generados por palabras de memoria errónea en un enrutador. El algoritmo es sumar todas las medias palabras de 16 bits a medida que llegan, usando aritmética de complemento a uno, y luego obtener el complemento a uno del resultado. Para los fines de este algoritmo, se supone que la *suma de comprobación de la cabecera* es cero cuando llega. Este algoritmo es más robusto que una suma normal. Nótese que la *suma de comprobación de la cabecera* debe recalcularse en cada salto, pues cuando menos uno de los campos siempre cambia (el campo de *tiempo de vida*), pero pueden usarse truco para acelerar el cálculo.

La *dirección de origen* y la *dirección de destino* indican el número de red y el número de *host*. El campo de *opciones* se diseñó para proporcionar un recurso que permitiera que las versiones subsiguientes del protocolo incluyeran información no presente en el diseño original, para permitir a los experimentadores probar ideas nuevas y para evitar la asignación de bits de cabecera a información pocas veces necesaria. Las opciones son de longitud variable. Cada una empieza con un código de 1 byte que identifica la opción. Algunas opciones vienen seguidas de un campo de longitud de la opción de 1 byte, y luego de uno o más bytes de datos. El campo de *opciones* se rellena para completar múltiplos de cuatro bytes. Actualmente hay cinco opciones definidas, las que se listan en la **figura 3**, pero no todos los enrutadores reconocen a todas.

La opción de *seguridad* indica qué tan secreta es la información. En teoría, un enrutador militar puede usar este campo para especificar que no se enrute a través de ciertos países que los militares consideren “malos”. En la práctica, todos los enrutadores lo ignoran, por lo que su única función real es la de ayudar a los espías a encontrar la información importante con mayor facilidad.

La opción de *enrutamiento estricto desde el origen* da la trayectoria completa desde el origen hasta el destino como secuencia de direcciones IP. Se requiere que el datagrama siga esa ruta exacta. Esta opción se usa sobre todo cuando los administradores de sistemas envíen

paquetes de emergencia porque las tablas de enrutamiento se han corrompido, o para hacer mediciones de tiempo.

Opción	Descripción
Seguridad	Especifica qué tan secreto es el datagrama
Enrutamiento estricto desde el origen	Indica la trayectoria completa a seguir
Enrutamiento libre desde el origen	Da una lista de los enrutadores que no deben evitarse
Registrar ruta	Hace que cada enrutador agregue su dirección de IP
Marca de tiempo	Hace que cada enrutador agregue su dirección y su marca de tiempo

FIGURA 3. Opciones del IP

La opción de *enrutamiento libre desde el origen* requiere que el paquete pase por los enrutadores indicados en la lista, y en el orden especificado, pero se le permite pasar a través de otros enrutadores indicados en la lista, y en el orden especificado, pero se le permite pasar a través de otros enrutadores en el camino. Normalmente, esta opción sólo indicará algunos enrutadores, para obligar a una trayectoria en particular. Por ejemplo, si se desea obligar a un paquete de Londres a Sydney a ir hacia el oeste en lugar de hacia el este, esta opción podría especificar enrutadores en Nueva York, Los Ángeles y Honolulú. Esta opción es de mucha utilidad cuando las consideraciones políticas o económicas dictan pasar a través de, o evitar, ciertos países.

La opción de *registrar ruta* indica a los enrutadores a lo largo de la trayectoria que agreguen su dirección de IP al campo de opción. Esto permite a los administradores del sistema buscar fallas en los algoritmos de enrutamiento (“¿por qué todos los paquetes de Houston a Dallas pasan por Tokio primero?”). Al establecer inicialmente ARPANET, ningún paquete pasaba nunca por más de nueve enrutadores, por lo que 40 bytes de opciones eran más que suficientes. Como se mencionó antes, ahora esto es demasiado poco.

Por último, la opción de *marca de tiempo* es como la opción de *registrar ruta*, excepto que además de registrar su dirección IP de 32 bits, cada enrutador también registra una marca de tiempo de 32 bits. Esta opción también es principalmente para búsqueda de fallas en los algoritmos de enrutamiento.

Direcciones IP

Cada *host* y enrutador de Internet tiene una dirección de IP, que codifica su número de red y su número de *host*. La combinación es única: no hay dos máquinas que tengan la misma dirección de IP. Todas las direcciones de IP son de 32 bits de longitud y se usan en los campos de *dirección de origen y dirección de destino* de los paquetes IP. Los formatos usados para las direcciones IP se muestran en la **figura 4**. Aquellas máquinas conectadas a varias redes tienen direcciones de IP diferentes en cada red.

Los formatos de clase A, B, C y D permiten hasta 126 redes con 16 millones de *hosts* cada una, 16,382 redes con hasta 64K *hosts*, 2 millones de redes (por ejemplo, LAN) de hasta 254 *hosts* cada una, y multitransmisión, en la cual se dirige un datagrama a múltiples *hosts*. Las direcciones que comienzan con 11110 se reservan para uso futuro. Hay decenas de miles de redes conectadas ahora a Internet, y la cifra se duplica cada año. Los números de red los asigna el **NIC** (*Network Information Center*, **centro de información de redes**) para evitar conflictos.

Las direcciones de red, que son números de 32 bits, generalmente se escriben en **notación decimal con puntos**. En este formato, cada uno de los 4 bytes se escribe en decimal, de 0 a 255. Por ejemplo, la dirección hexadecimal C0290614 se escribe como 192.41.6.20. La dirección de IP menor es 0.0.0.0 y la mayor 255.255.255.255.

Los valores 0 y –1 tienen significado especial, como se muestra en la **figura 5.** El valor 0 significa esta red o este *host*. El valor –1 se usa como dirección de difusión para indicar todos los *hosts* de la red indicada.

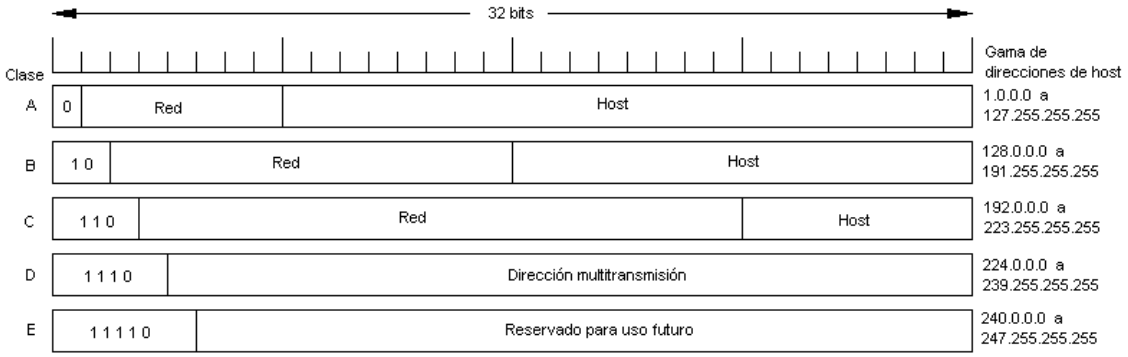


FIGURA 4. Formatos de dirección IP

La dirección de IP 0.0.0.0 es usada por los *hosts* cuando están siendo arrancados, pero no se usa después. Las direcciones de IP con 0 como número de red se refieren a la red actual. Estas direcciones permiten que las máquinas se refieran a su propia red sin saber su número (pero tienen que saber su clase para saber cuántos 0 hay que incluir). La dirección que consiste solamente en unos permite la difusión en la red local, por lo común una LAN. Las direcciones con un número de red propio y solamente unos en el campo de *host* permite que las máquinas envíen paquetes de difusión a LAN distantes desde cualquier parte de Internet. Por último, todas las direcciones de la forma 127.xx.yy.zz se reservan para pruebas de realimentación. Los paquetes enviados a esa dirección no se colocan en el alambre; se procesan localmente y se tratan como paquetes de entrada. Esto permite que los paquetes se

envíen a la red local sin que el transmisor conozca su número. Esta característica también se usa para la detección de fallas en el software de la red.

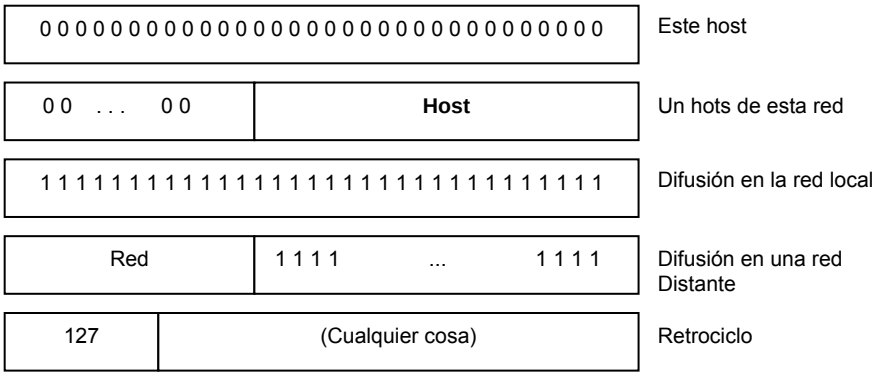


FIGURA 5. Direcciones especiales de IP

Subredes

Como hemos visto, todos los *host* de una red deben tener el mismo número de red. Esta propiedad del direccionamiento IP puede causar problemas a medida que crecen las redes. Por ejemplo, considere una compañía que comienza con una LAN clase C en Internet. A medida que pasa el tiempo, podría adquirir de 254 máquinas, y por tanto necesitar una segunda dirección clase C. De la misma manera, podría adquirir una segunda LAN de un tipo diferente y querer una dirección de IP diferente para ella (las LAN podrían estar unidas mediante un puente para formar una sola red IP, pero los puentes tienen sus propios problemas). Finalmente, podría terminar con muchas LAN, cada una con sus propio enrutador y su propio número de red clase C.

Al crecer la cantidad de redes locales distintas, su administración puede volverse un dolor de cabeza grave. Cada vez que se instala una red nueva, el administrador del sistema tiene que comunicarse con el NIC para obtener un número de red nuevo. Después, este número debe anunciarse mundialmente. Además, mover una máquina de una LAN a otra requiere el cambio de su dirección de IP, lo que a su vez puede significar la modificación de sus archivos de configuración y también el anuncio de la nueva dirección de IP a todo el mundo. Si a otra máquina se le da la dirección de IP recién liberada, esa máquina recibirá el correo electrónico y otros datos dirigidos a la máquina original hasta que la dirección se haya propagado por todo el mundo.

La solución a este problema es permitir la división de una red en varias partes para uso interno, pero aún actuar como una sola red ante el mundo exterior. En la literatura sobre Internet, a estas partes se les llama **subredes**. Como mencionamos en el **capítulo 1**, este uso entra en

conflicto con “subred” cuyo significado es el grupo de todos los enrutadores y líneas de comunicación de una red. Esperamos que el contexto deje en claro el significado de que se trata. Si nuestra compañía en crecimiento tenía inicialmente una dirección clase B en lugar de clase C, puede comenzar simplemente por numerar los *hosts* de 1 a 254. Al llegar la segunda LAN podría decidir, por ejemplo, dividir el número de *host* de 16 bits en un número de subred de 6 bits y un número de *host* de 10 bits, como se muestra en la **figura 6**. Esta división permite 62 LAN (se reservan el 0 y el –1) cada una con hasta 1022 *hosts*.

Fuera de la red, la subred no es visible, por lo que la asignación de una subred nueva no requiere comunicación con el NIC ni la modificación de bases de datos externas. En este ejemplo, la primera subred podría usar direcciones de IP a partir de 130.50.4.1, la segunda subred podría empezar en 130.50.8.1, etcétera.

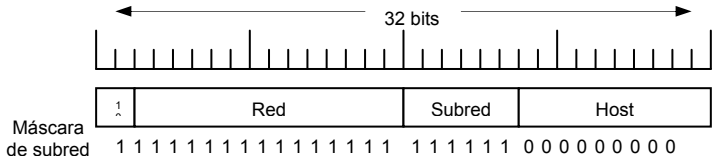


FIGURA 6. Una de las formas de generar una subred clase B

Para ver el funcionamiento de las subredes, es necesario explicar la manera en que se procesan los paquetes IP en un enrutador. Cada enrutador tiene una tabla en la que se lista cierto número de direcciones IP (red, 0) y cierto número de direcciones IP (esta red, *host*). El primer tipo indica cómo llegar a redes distantes. El segundo tipo indica cómo llegar a redes locales. Asociada a cada tabla está la interfaz de red a usar para llegar al destino y cierta información adicional.

Al llegar un paquete de IP, se busca su dirección de destino en la tabla de enrutamiento. Si el paquete es para una red distnate, se reenvía al siguiente enrutador de la interfaz dada en la tabla; si es para un *host* local (por ejemplo, en la LAN del enrutador), se envía directamente al destino. Si la red no está en la tabla, el paquete se reenvía a un enrutador predeterminado con tablas más extensas. Este algoritmo significa que cada enrutador sólo tiene que llevar el registro de otras redes y *hosts* locales, no de pares red-*host*, reduciendo en gran medida el tamaño de la tabla de enrutamiento.

Al introducirse subredes, se cambian las tablas de enrutamiento, agregando entradas con forma de (Esta red, subred, 0) y (esta red, esta subred, *host*). Por tanto, un enrutador de la subred *k* sabe cómo llegar a todas las demás subredes y también cómo llegar a todos los *hosts* de la subred *k*; no tiene que saber los detalles sobre los *hosts* de otras subredes. De hecho, todo lo que se necesita es hacer que cada enrutador haga un ADN booleano con la **máscara**

de subred (**véase la figura 6**) para deshacerse el número de *hosts* y buscar la dirección resultante en sus tablas (tras determinar la clase de red de la que se trata). Por ejemplo, a un paquete dirigido a 130.50.15.6 que llega a un enrutador de la subred 5 se le hace un AND con la máscara de subred de la **figura 6** para dar la dirección 130.50.12.0. Esta dirección se busca en las tablas de enrutamiento para averiguar la manera de llegar a los *hosts* de la subred 3. Así, el enrutador de la subred 5 se ahorra el trabajo de mantener un registro de las direcciones de enlace de datos de *hosts* que no pertenecen a la subred 5. El proceso de subred reduce así el espacio de tabla de enrutamiento creando una jerarquía de tres niveles.

1.4 PROTOCOLO ICMP (Protocolo de control de mensajes de Internet)

La operación de Internet es supervisada cuidadosamente por los enrutadores. Al ocurrir algo inesperado, el **ICMP** (*Internet Control Message Protocol*, **protocolo de control de mensajes de Internet**), que también se usa para probar Internet, informa del suceso. Se ha definido una docena de tipos de mensajes de ICMP; los más importantes se listan en la **figura 7**. Cada tipo de mensajes de ICMP se encapsula en un paquete IP.

El mensaje DESTINO INALCANZABLE (*DESTINATION UNREACHABLE*) se usa cuando la subred o un enrutador no pueden ubicar el destino, o un paquete con el bit *DF* no puede entregarse porque está en el camino una red de “paquete pequeño”.

El mensaje de *tiempo excedido* (*TIME EXCEEDED*) se envía cuando un paquete se descarta debido a que su contador llega a cero. Este suceso es un síntoma de que los paquetes están en ciclo, de que hay un congestionamiento enorme, o de que los valores de temporización son demasiado bajos.

Tipo de mensaje	Descripción
Destino inalcanzable	No pudo entregarse el paquete
Tiempo excedido	Campo de tiempo de vida llegó a cero
Problema de parámetro	Campo de cabecera no válido
Supresión de origen	Paquete de estrangulamiento
Reenvío	Enseña geografía a un enrutador
Solicitud de eco	Pregunta a una máquina si está viva
Respuesta de eco	Sí, estoy viva
Solicitud de marca de tiempo	Igual que la solicitud de eco, pero con marca de tiempo
Respuesta de marca de tiempo	Igual que la respuesta de eco, pero con marca de tiempo

FIGURA 7. Los principales tipos de mensajes ICMP.

El mensaje de PROBLEMA DE PARÁMETRO (*PARAMETER PROBLEM*) indica que se ha detectado un valor ilegal en un campo de cabecera. Este problema indica una falla en el *software* de IP del *host*, o posiblemente en el *software* de un enrutador transitado.

El mensaje SUPRESION DE ORIGEN (*SOURDE QUENCH*) se usaba antes para controlar a los *hosts* que enviaban demasiados paquetes. Al recibir un *host* este mensaje, se esperaba que se refrenara. Este mensaje se usa poco en la actualidad porque, al ocurrir congestionamientos, estos paquetes tienden a echarle más leña al fuego. El control de congestionamiento de Internet se hace ahora en gran medida en la capa de transporte.

El mensaje de REDIRECCIONAMIENTO (*REDIRECT*) se usa cuando un enrutador se da cuenta de que un paquete parece estar mal enrutado el enrutador lo usa para indicar al *host* transmisor el posible error.

Los mensajes de SOLICITUD DE ECO (*ECHO REQUEST*) y RESPUESTA DE ECHO (*ECHO REPLAY*) sirven para ver si un destino dado es alcanzable y está vivo. Al recibir el mensaje de ECO, se espera que el destino devuelva un mensaje de RESPUESTA DE ECO. Los mensajes SOLICITUD DE MARCA DE TIEMPO (*TIMESTAMP REQUEST*) Y RESPUESTA DE MARCA DE TIEMPO (*TIMESTAMP REPLY*) son parecidos, excepto que el tiempo de llegada del mensaje y el tiempo de partida de la respuesta se registran en la respuesta. Este recurso se emplea para medir el desempeño de la red.

Además de estos mensajes, hay cuatro más que tienen que ver con el direccionamiento de Internet, para permitir que los *hosts* descubran sus números de red y para manejar el caso de varias LAN que comparten una sola dirección IP. El ICMP se define en el RFC 792.

1.5 PROTOCOLO TCP (Protocolo de control de la transmisión)

La capa que está sobre la capa de interredes en el modelo TCP/IP se llama usualmente ahora **capa de transporte**. Esta capa se diseñó para permitir que las entidades pares en los nodos de origen y destino lleven a cabo una conversación, lo mismo que en la capa de transporte OSI. Aquí se definieron dos protocolos de extremo a extremo. El primero, **TCP** *(transmission control protocol*, **protocolo de control de la transmisión**) es un protocolo confiable orientado a la conexión que permite que una corriente de bytes originada en una máquina se entregue sin errores en cualquier otra máquina de la interred. Este protocolo fragmenta la corriente entrante de bytes en mensajes discretos y pasa cada uno a la capa de interred. En el destino, el proceso TCP receptor reensambla los mensajes recibidos para formar la corriente de salida. El TCP también se encarga del control de flujo para asegurar que un emisor rápido no pueda abrumar a un receptor lento con más mensajes de los que pueda manejar.

El segundo protocolo de esta capa, el **UDP** (*user datagram protocol,* **protocolo de datagrama de usuario**), es un protocolo sin conexión, no confiable, para aplicaciones que no necesitan la

asignación de secuencia ni el control de flujo del TCP y que desean utilizar los suyos propios. Este protocolo también se usa ampliamente para consultas de petición y respuesta de una sola ocasión, del tipo cliente-servidor, y en aplicaciones en las que la entrega pronta es más importante que la entrega precisa, como las transmisiones de voz o vídeo. La relación entre IP, TCP y UDP se muestra en la **figura 8**. Desde que se desarrolló el modelo, el IP se ha implementado en muchas otras redes.

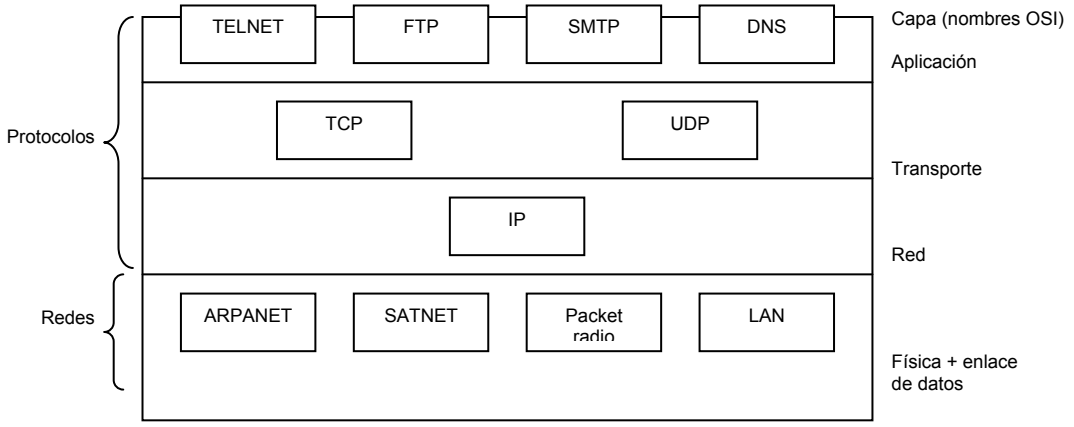


FIGURA 8. Protocolos y redes en el modelo TCP/IP inicial.

Modelo de servicio TCP

El servicio TCP se obtiene haciendo que tanto el transmisor como el receptor creen puntos terminales, llamados sockets. Cada socket tiene un número (dirección) de socket que consite en la dirección IP del *host* y en un número de 16 bits local a ese *host*, llamado **puerto**. Puerto es el nombre en TCP de un TSAP. Para obtener el servicio TCP, debe establecerse explícitamente una conexión entre un socket de la máquina transmisora y un socket de la máquina receptora. Las llamadas de socket se listan en la **figura 9**.

PRIMITIVAS	SIGNIFICADO
SOCKET (ENCHUFAR)	Crear un nuevo punto terminal de comunicación
BIND (LIGAR)	Conecta una dirección local a un socket
LISTEN (ESCUCHAR)	Anuncia la disposición a aceptar conexiones; indica tamaño de cola
ACCEPT (ACEPTAR)	Bloquea al invocador hasta la llegada de un intento de conexión.
CONNECT (CONECTAR)	Intenta activamente establecer una conexión
SEND (ENVIAR)	Envía datos a través de una conexión
RECEIVE (RECIBIR)	Recibe datos de una conexión
CLOSE (CERRAR)	Libera conexión.

FIGURA 9. Primitivas de Socket para TCP

Puede usarse un socket para varias conexiones al mismo tiempo. En otras palabras, dos o más conexiones pueden terminar en el mismo socket. Las conexiones se identifican mediante los identificadores de ambas terminales, es decir (*socket1, socket 2*). No se usan números de circuito virtual ni ningún otro identificador.

Los números de puerto por debajo del 256 se llaman **puertos bien conocidos** (*well-knownports*) y se reservan para servicios estándar. Por ejemplo, cualquier proceso que desee establecer una conexión a un *host* para transferir un archivo usando FTP puede conectarse al puerto 21 del *host* de destino para comunicarse con su *daemon* (proceso reentrante) de FTP, De la misma manera, para establecer una sesión interactiva remota usando TELNET, se usa el puerto 23. La lista de puertos bien conocidos se da en el RFC 1700.

Todas las conexiones TCP son dúplex integral y punto a punto. Dúplex integral significa que el tráfico puede ir en ambos sentidos al mismo tiempo. Punto a punto significa que cada conexión tiene exactamente dos puntos terminales. El TCP no reconoce la multitransmisión ni la difusión.

Una conexión TCP es una corriente de bytes, no una corriente de mensajes. Los límites de mensaje no se conservan de extremo a extremo. Por ejemplo, si el proceso transmisor hace cuatro escrituras de 512 bytes en una corriente TCP, estos datos pueden entregarse al proceso receptor como cuatro bloques de 512 bytes, dos bloques de 1024 bytes, un bloque de 2048 bytes (**véase la figura 10**), o de algún otro modo. No hay manera de que el receptor detecte las unidades en las que se escribieron los datos.

Los archivos de UNIX tienen también esta propiedad. El lector de un archivo no puede saber si el archivo se escribió bloque por bloque, byte por byte, o todo de golpe. Como sucede con los archivos UNIX, el *software* del TCP no tiene idea de lo que significan los bytes y no tiene interés en averiguarlo. Un byte simplemente es un byte.

Cuando una aplicación pasa datos al TCP, el TCP puede enviarlos de inmediato o guardarlos en un *buffer* (a fin de reunir una cantidad mayor de información para enviarla junta), a discreción qué nos estamos refiriendo. Por ejemplo, en el “el usuario proporciona al TCP los datos”, claramente nos estamos refiriendo a la entidad de transporte TCP.

Cuando una aplicación pasa datos al TCP, el TCP puede enviarlos de inmediato o guardarlos en un *buffer* (a fin de reunir una cantidad mayor de información para enviarla junta), a discreción propia. Sin embargo, a veces la aplicación simplemente quiere que los datos se envíen de inmediato. Por ejemplo, supóngase que un usuario está en sesión con una máquina remota. Tras la terminación de una línea de comandos y el retorno de carro, es esencial que la línea se envíe a la máquina remota de inmediato y no se guarde en *buffer* hasta la entrada de

la siguiente línea. Para obligar la salida de datos, las aplicaciones pueden usar la bandera (*flag*) PUSH que ordena al TCP no retrasar la transmisión.

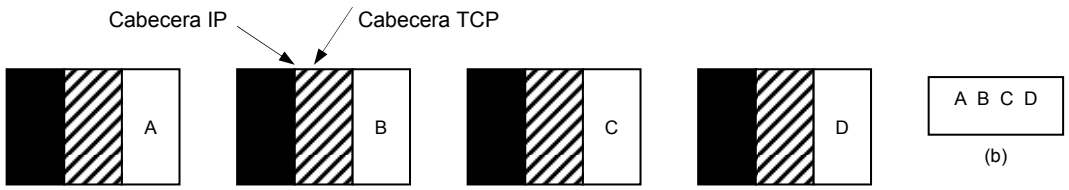


FIGURA 10. (a) Cuatro segmentos de 512 bytes enviados como datagramas IP independientes. (b) Los 2048 bytes de datos entregados a la aplicación con una sola READ.

Algunas de las primeras aplicaciones usaron la bandera PUSH como una especie de marcador para delinear los límites de los mensajes. Aunque este truco a veces funciona, en ocasiones falla porque no todas las implementarios de TCP pasan la bandera PUSH a la aplicación del lado receptor. Es más, si llegan otros PUSH adicionales antes de la transmisión del primero (por ejemplo, por esta ocupada la línea de salida), el TCP está en libertad de juntar todos los datos que entran por PUSH en un solo datagrama IP, sin separación entre las diferentes partes.

Una última característica del servicio TCP que vale la pena mencionar son los **datos urgentes**. Cuando un usuario interactivo pulsa la tecla DEL o CTRL-C para interrumpir un cómputo remoto ya iniciado, la aplicación transmisora pone cierta información de control en la corriente de datos y se la da al TCP junto con la bandera URGENT. Este evento hace que el TCP deje de acumular datos y transmita de inmediato todo lo que tiene para esa conexión.

Al recibirse los datos urgentes en el destino, se interrumpe la aplicación receptora (por ejemplo, recibe una señal en términos de UNIX), para que la aplicación pueda detener lo que estaba haciendo y leer la corriente de datos hasta encontrar los datos urgentes. Se marca el fin de los datos urgentes, por lo que la aplicación sabe cuándo han terminado. El comienzo de los datos urgentes no se marca; es responsabilidad de la aplicación averiguarlo. Este esquema básicamente proporciona un mecanismo burdo de señalamiento, y deja todo lo demás en manos de la aplicación.

El protocolo TCP

En esta sección daremos un repaso general del protocolo TCP; en la siguiente veremos la cabecera del protocolo, campo por campo. Cada byte de una conexión TCP tiene su propio número de secuencia de 32 bits. En un *host* que opera a toda velocidad en una LAN de 10 Mbps, en teoría los números de secuencia podrían volver a comenzar en una hora, pero en la práctica tarda mucho más tiempo. Se usan los números de secuencia tanto para los acuses de

recibo como para el mecanismo de ventana, que utilizan campos de cabecera de 32 bits distintos.

La entidad TCP transmisora y la receptora intercambian datos en forma de segmentos. Un **segmento** consiste en una cabecera TCP fija de 20 bytes (más una parte opcional) seguida de cero o más bytes de datos. El *software* de TCP decide el tamaño de los segmentos; puede acumular datos de varias escrituras para formar un segmento, o dividir los datos de una escritura en varios segmentos. Hay dos límites que restringen el tamaño de segmento. Primero, cada segmento, incluida la cabecera TCP, debe caber en la carga útil de 65,535 bytes del IP. Segundo, cada red tiene una **unidad máxima de transferencia**, o **MTU** (*maximum transfer unit*), y cada segmento debe caber en la MTU. En la práctica, las MTU generalmente son de unos cuantos miles de bytes y por tanto definen el límite superior del tamaño de segmento. Si un segmento pasa a través de una serie de redes sin fragmentarse y luego se topa con una cuya MTU es menor que el segmento, el enrutador de la frontera fragmenta el segmento en dos o más segmentos más pequeños.

Un segmento demasiado grande para transitar por una red puede dividirse en varios segmentos mediante un enrutador. Cada segmento nuevo recibe sus propias cabeceras TCP e IP, por lo que la fragmentación en los enrutadores aumenta la carga extra total (puesto que cada segmento adicional agrega 40 bytes de información de cabecera).

El protocolo básico usado por las entidades TCP es el protocolo de ventana corrediza. Cuando un transmisor envía un segmento, también inicia un temporizador. Cuando llega el segmento al destino, la entidad TCP receptora devuelve un segmento (con datos, si existen, de otro modo sin ellos) que contienen un número de acuse de recibo igual al siguiente número de secuencia que espera recibir. Si el temporizador del transmisor expira antes de la recepción del acuse de recibo, el transmisor envía de nuevo el segmento.

Aunque este protocolo suena sencillo, tiene muchos vericuetos que cubriremos a continuación. Por ejemplo, dado que los segmentos pueden fragmentarse, es posible que llegue una parte del segmento transmitido y que la entidad TCP receptora envíe un acuse de recibo, pero la otra parte se pierda. También pueden llegar segmentos fuera de orden, por lo que los bytes 3072-4095 podrían llegar pero no enviarse acuse de recibo porque los bytes 2048-3071 no han aparecido aún. También pueden retardarse segmentos en tránsito durante tanto tiempo que el transmisor termina de temporizar y retransmite nuevamente. Si un segmento retransmitido toma una ruta distinta a la del original y se fragmenta de manera diferente, pueden llegar esporádicamente partes tanto del original como del duplicado, requiriéndose una administración cuidadosa para lograr una corriente de bits confiable. Por último, siendo tantas

las redes que constituyen la Internet, es posible que un segmento pueda toparse ocasionalmente con una red congestionada (o rota) en alguna parte de su trayectoria.

El TCP debe estar preparado para manejar y resolver estos problemas de una manera eficiente. Se ha invertido una cantidad considerable de esfuerzo en la optimización del desempeño de las corrientes TCP, incluso ante problemas de red. A continuación se estudiarán varios de los algoritmos usados por muchas implementaciones de TCP.

La cabecera de segmento TCP

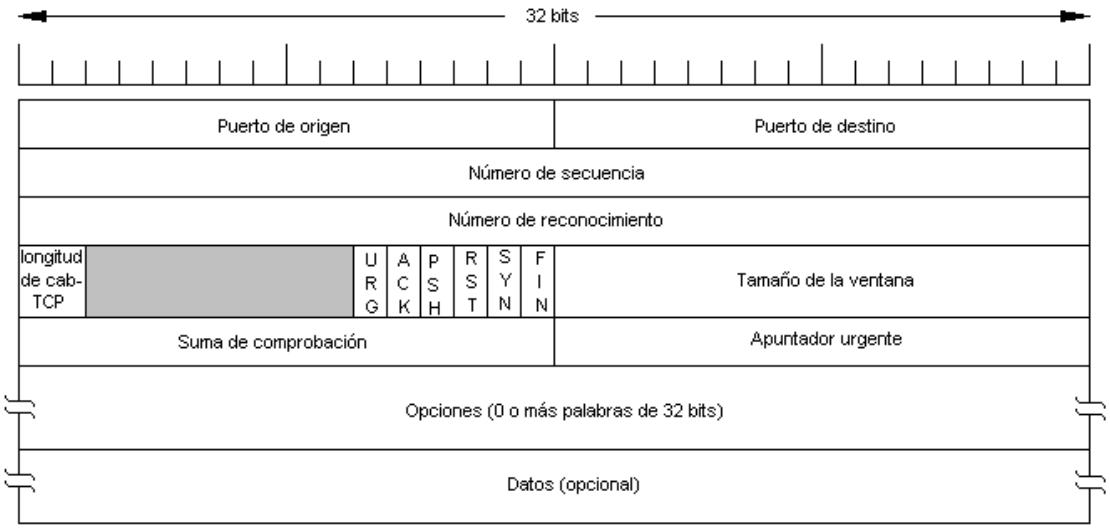


FIGURA 11. Cabecera TCP.

En la **figura 11.** se muestra la distribución de un segmento TCP. Cada segmento comienza con una cabecera de formato fijo de 20 bytes. La cabecera fija puede ir en seguida de opciones de cabecera. Tras la opciones, si las hay, pueden continuar hasta 65.535 – 20 - 20 = 65.515 bytes de datos, donde los primeros 20 se refieren a la cabecera IP y los segundos a la cabecera TCP. Los segmentos sin datos son legales y se usan por lo común para acuses de recibo y mensajes de control.

Realicemos la disección de la cabecera TCP campo por campo. Los campos de *puerto de origen y puerto de destino* identifican los puntos terminales locales de la conexión. Cada *host* puede decidir por sí mismo la manera de asignar sus propios puertos comenzando por el 256. La dirección IP de su *host* forman un TSAP único de 48 bits. Los números de socket de origen y d destino en conjunto identifican la conexión.

Los campos de *número de secuencia y número de acuse de recibo* desempeñan sus funciones normales. Nótese que el segundo especifica el siguiente byte esperado, no el último byte

correctamente recibido. Ambos tienen 32 bits de longitud porque cada byte de datos está numerado en una corriente TCP.

La *longitud de cabecera TCP* indica la cantidad de palabras de 32 bits contenidas en la cabecera TCP. Esta información es necesaria porque el campo de *opciones* es de longitud variable, por lo que la cabecera también. Técnicamente, este campo en realidad indica el comienzo de los datos en el segmento, medido en palabras de 32 bits, pero ese número es simplemente la longitud de la cabecera en palabras, por lo que el efecto es el mismo.

A continuación viene un campo de 6 bits que no se usan. El que este campo haya sobrevivido intacto durante más de una década es testimonio de lo bien pensando que está el TCP. Protocolos menos bien hechos lo habrían necesitado para corregir errores del diseño original.

Ahora vienen seis banderas de 1 bits. *URG* se establece en 1 si está en uso el *apuntador urgente*. El *apuntador urgente* sirve para indicar un desplazamiento en bytes a partir del número actual de secuencia en el que se encuentran datos urgentes. Este recurso sustituye los mensajes de interrupción. Como se mencionó antes, este recurso es un mecanismo rudimentario para permitir que el transmisor envíe una señal al receptor sin implicar al TCP en la razón de la interrupción.

El bit *ACK* se establece en 1 para indicar que el *número de acuse de recibo* es válido. Si el *ACK* es 0, el segmento no contiene un acuse de recibo, por lo que se ignora el campo de *número de acuse de recibo*.

El bit *PSH* indica datos empujados (con PUSH). Por este medio se solicita atentamente al receptor entregar los datos a la aplicación a su llegada y no ponerlos en *buffer* hasta la recepción de un *buffer* completo (lo que podría hacer en otras circunstancias por razones de eficiencia).

Se usa el bit *RST* para restablecer una conexión que se ha confundido debido a una caída de *host* u otra razón; también sirve para rechazar un segmento no válido o un intento de abrir una conexión. Por lo general, si usted recibe un segmento con el bit *RST* encendido, tiene un problema entre manos.

El bit *SYN* se usa para establecer conexiones. La solicitud de conexión tiene *SYN = 1* y *ACK = 0* para indicar que el campo de acuse de recibo incorporado no está en uso. La respuesta de conexión sí lleva un reconocimiento, por lo que tiene *SYN = 1* y *ACK = 1*. En esencia, el bit SYN se usa para denotar CONNECTION REQUEST Y CONNECTION ACCEPTED, usándose el bit ACK para distinguir entre ambas posibilidades.

El bit FIN se usa para liberar una conexión: específica que el transmisor no tiene más datos que transmitir. Sin embargo, tras cerrar una conexión, un proceso puede continuar recibiendo datos indefinidamente. Ambos segmentos, *SYN* y *FIN*, tienen números de secuencia y por tanto tienen la garantía de procesarse en el orden correcto.

El control de flujo en el TCP se maneja usando una ventana corrediza de tamaño variable. El campo de *ventana* indica la cantidad de bytes que pueden enviarse comenzando por el byte que ya sea ha enviado acuse de recibo. Es válido un campo de *ventana* de 0, e indica que se han recibido los bytes hasta *número de acuse de recibo* – 1, inclusive pero que el receptor actualmente necesita un descanso y quisiera no recibir más datos por el momento, gracias. El permiso para enviar puede otorgarse después enviando un segmento con el mismo *número de acuse de recibo* y un campo de *ventana* distinto de cero.

También se proporciona una *suma de comprobación* para confiabilidad extrema. Es una suma de comprobación de la cabecera, los datos y la pseudocabecera conceptual mostrada en la **figura 12**. Al realizar este cálculo, se establece el campo de *suma de comprobación* del TCP en cero, y se rellena el campo de datos con un byte cero adicional si la longitud es un número non. El algoritmo de suma de comprobación simplemente suma todas las palabras de 16 bits en complemento a 1 y luego obtiene el complemento a 1 de la suma. Como consecuencia, al realizar el cálculo el receptor con el segmento completo, incluido el campo de *suma de comprobación*, el resultado debe ser 0.

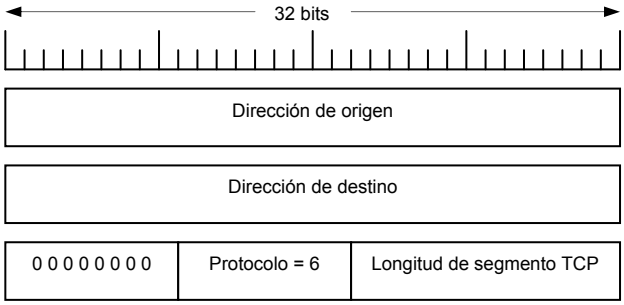


FIGURA 12. Pseudocabecera incluida en la suma de comprobación del TCP

La pseudocabecera contiene las direcciones IP de 32 bits de las máquinas de origen y de destino, el número de protocolo de TCP (6), y la cuenta de bytes del segmento TCP (incluida la cabecera). La inclusión de la pseudocabecera en el cálculo de la suma de comprobación TCP ayuda a detectar paquetes mal entregados, pero hacerlo viola la jerarquía de protocolos puesto que las direcciones de IP que contiene pertenecen a la capa IP, no a la capa TCP.

El campo de *opciones* se diseñó para contar con una manera de agregar características extra no cubiertas por la cabecera normal. La opción más importante es la que permite que cada *host* especifique la carga útil TCP máxima que está dispuesto a aceptar. El uso de segmentos grandes es más eficiente que el de segmentos pequeños, puesto que la cabecera de 20 bytes puede entonces amortizarse entre más datos, pero los *hosts* pequeños tal vez no puedan manejar segmentos muy grandes. Durante el establecimiento de la conexión, cada lado puede anunciar su máximo y ver el de su compañero. El más pequeño de los dos números es el ganador. Si un *host* no usa esta opción, predetermina una carga útil de 536 bytes. Se requiere que todos los *hosts* Internet acepten segmentos TCP de 536 + 20 = 556 bytes.

En las líneas con alto ancho de banda, alto retardo o ambas cosas, la ventana de 64 KB con frecuencia es un problema. En una línea T3 (44.736 Mbps) se requieren sólo 12 mseg para enviar una ventana completa de 64 KB. Si el retardo de propagación de ida y vuelta es de 50 mseg (típico de una fibra transcontinental), el transmisor estará inactivo ¾ del tiempo en espera de acuses de recibo. En una conexión satelital la situación es peor aún. Un tamaño de ventana más grande permitirá al transmisor continuar enviando datos, pero como el campo de *tamaño de ventana* es de 16 bits, es imposible expresar tal tamaño. En el RFC 1323 se propuso una opción de *escala de ventana* que permite al transmisor y al receptor negociar un factor de escala de ventana. Este número permite que ambos lados desplacen el campo de tamaño de ventana hasta 16 bits a la izquierda, permitiendo pro tanto un máximo de 2^{32} bytes. La mayoría de las implementaciones actuales de TCP manejan esta opción.

Otra opción propuesta en el RFC 1106 y ahora de uso difundido es el empleo de la repetición selectiva en lugar del protocolo de regresar *n (go back n)*. Si el receptor recibe un segmento malo y luego una gran cantidad de segmentos buenos, el protocolo TCP normal en algún momento terminará de temporizar y retransmitirá todos los segmentos sin acuse de recibo, incluidos los que se recibieron correctamente. El RFC 1106 introdujo los NAK, para permitir que el receptor solicite un segmento (o segmentos) específico. Tras recibirlo, puede enviar un acuse de recibo de todos los datos que tiene en *buffer*, reduciendo de esta manera la cantidad de datos retransmitidos.

Gestión de una conexión TCP

En el TCP se establecen las conexiones usando el protocolo de acuerdo de tres vías *(three-way handshake)*. Para establecer una conexión, un lado, digamos el servidor, espera pasivamente una conexión entrante ejecutando las primitivas LISTEN Y ACCEPT y especificando cierto origen o bien nadie en particular.

El otro lado, digamos el cliente, ejecuta una primitiva CONNECT especificando la dirección y el puerto IP con el que se desea conectar, el tamaño máximo de segmento TCP que está dispuesto a aceptar y opcionalmente algunos datos de usuario (por ejemplo, una contraseña). La primitiva CONNECT envía un segmento TCP con el bit *SYN* encendido y el bit *ACK* apagado, y espera una respuesta.

Al llegar el segmento al destino, la entidad TCP ahí revisa si hay un proceso que haya ejecutado un LISTEN en el puerto indicado en el campo de *puerto de destino*. Si no lo hay, envía una contestación con el bit *RST* encendido para rechazar la conexión.

Si algún proceso está escuchando en el puerto, ese proceso recibe el segmento TCP entrante y puede entonces aceptar o rechazar la conexión; si la acepta, se devuelve un segmento de acuse de recibo. La secuencia de segmentos TCP enviados en el caso normal se muestra en la **figura 13(a)**.

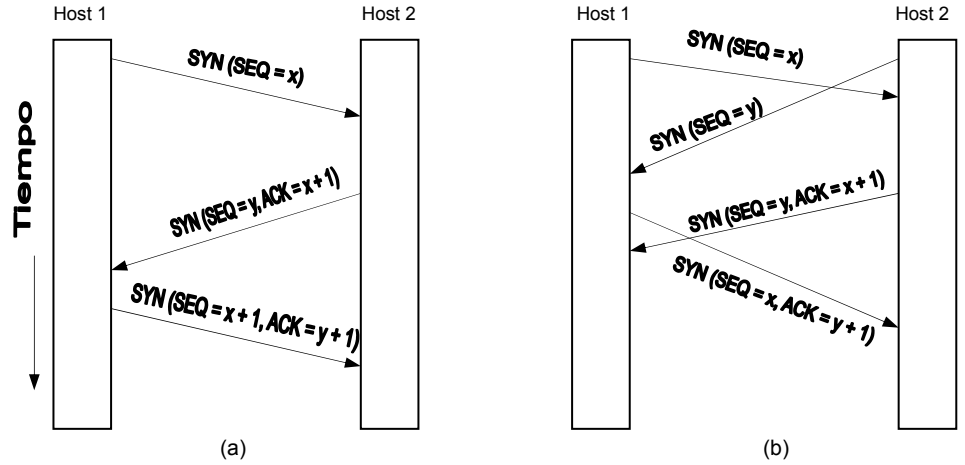


FIGURA 13. (a) Establecimiento de una conexión TCP en el caso normal. (b) Colisión de llamadas.

Nótese que un segmento *SYN* consume 1 byte de espacio de secuencia, por lo que puede reconocerse sin ambigüedades.

En el caso en que dos *hosts* intentan simultáneamente establecer una conexión entre los mismos dos sockets, la secuencia de eventos es la que se ilustra en la **figura 13(b)**. El resultado de estos eventos es que sólo se establece una conexión, no dos, pues las conexiones se identifican por sus puntos terminales. Si el primer establecimiento resulta en una conexión identificada por (*x, y*), al igual que el segundo, sólo se hace una entrada de tabla, es decir, de (*x, y*).

El número de secuencia inicial de una conexión no es 0 por las razones que señalamos antes. Se usa un esquema basado en reloj, con un pulso de reloj cada 4μseg. Por seguridad adicional, al caerse un *host*, no podrá reiniciarse durante el tiempo máximo de paquete (120 seg) para asegurar que no haya paquetes de conexiones previas vagando por Internet.

Aunque las conexiones TCP son dúplex integral, para entender la manera en que se liberan las conexiones es mejor visualizarlas como un par de conexiones símplex. Cada conexión símplex se libera independientemente de su igual. Para liberar una conexión, cualquiera de las partes puede enviar un segmento TCP con el bit *FIN* establecido, lo que significa que no tiene más datos por transmitir. Al reconocerse el *FIN*, ese sentido se apaga. Sin embargo, puede continuar un flujo de datos indefinido en el otro sentido. Cuando ambos sentidos se han apagado, se libera la conexión. Normalmente se requieren cuatro segmentos TCP para liberar una conexión, un *FIN* y un *ACK* para cada sentido. Sin embargo, es posible que el primer *ACK* y el segundo *FIN* estén contenidos en el mismo segmento, reduciendo la cuenta total a tres.

Al igual que con las llamadas telefónicas en las que ambas partes dicen adiós y cuelgan el teléfono simultáneamente, ambos extremos de una conexión *TCP* pueden enviar segmentos *FIN* al mismo tiempo. Ambos se reconocen de la manera normal, y se apaga la conexión. De hecho, en esencia no hay diferencia entre la liberación secuencial o simultánea por parte de los *hosts*.

Para evitar el problema de los dos ejércitos, se usan temporizadores. Si no llega una respuesta a un *FIN* en un máximo de dos tiempos de vida de paquete, el transmisor del *FIN* libera la conexión. Tarde o temprano el otro lado, notará que, al parecer, ya nadie lo está escuchando, y también terminará su temporización. Aunque esta solución no es perfecta, dado el hecho de que teóricamente es imposible una solución perfecta tendremos que conformarnos con ella. En la práctica, pocas veces ocurren problemas.

Los pasos requeridos para establecer y liberar conexiones pueden representarse en una máquina de estados finitos con los 11 estados listados en la **figura 14**. En cada estado son legales ciertos eventos. Al ocurrir un evento legal, debe emprenderse alguna acción. Si ocurren otros eventos, se informa un error.

Estado	Descripción
CLOSED	No hay conexión activa ni pendiente
LISTEN	el servidor espera una llamada
SYN RCVD	Llegó solicitud de conexión; espera ACK
SYN SENT	La aplicación comenzó a abrir una conexión
ESTABLISHED	Estado normal de transferencia de datos
FIN WAIT 1	La aplicación dijo que ya terminó
FIN WAIT 2	El otro lado acordó liberar
TIMED WAIT	Espera que todos los paquetes mueran
CLOSING	Ambos lados intentaron cerrar simultáneamente
CLOSE WAIT	El otro lado inició una liberación
LAST ACK	Espera que todos los paquetes mueran

FIGURA 14. Estados usados en la máquina de estados finitos de administración de conexiones TCP.

Cada conexión comienza en el estado *CLOSED* (cerrado) y deja ese estado cuando hace una apertura pasiva (LISTEN), o una apertura activa (CONNECT). Si el otro lado realiza la acción opuesta, se establece una conexión y el estado se vuelve ESTABLISHED (establecido). La liberación de la conexión puede iniciarse desde cualquiera de los dos lados. Al completarse, el estado regresa a CLOSED.

1. 6 PROTOCOLO HTTP (Protocolo de transferencia de hipertexto)

El protocolo estándar de transferencia de la *Web* es el **HTTP** (*HiperText Transfer Protocol,* **protocolo de transferencia de hipertexto**). Cada interacción consiste en una solicitud ASCII seguida de una respuesta tipo MIME RFC 822. Aunque es muy común el uso del TCP para la conexión de transporte, no es requerido formalmente por el estándar. Si las redes ATM se vuelven lo bastante confiables, las solicitudes y respuestas HTTP podrían transportarse igualmente en mensajes AAL 5.

El HTTP está evolucionando constantemente. Se usan varias versiones y se están desarrollando otras. El material presentado a continuación es relativamente básico y es poco probable que cambie su concepto, pero algunos detalles podrían ser un poco diferentes en las versiones futuras.

El protocolo HTTP consiste en dos elementos bastante diferentes: el grupo de solicitudes de los visualizadores a los servidores y el grupo de respuestas en el otro sentido. Ahora los veremos por partes.

Todas las versiones más recientes de HTTP reconocen dos tipos de solicitud: solicitudes sencillas y solicitudes completas. Una solicitud sencilla es sólo una línea *GET* que nombra la página deseada, sin la versión del protocolo. La respuesta es la página en bruto, sin cabeceras,

sin MIME y sin codificación. Para ver su funcionamiento, intente establecer una conexión Telnet con el puerto 80 de www.w3.org (como se muestra en la primera línea de la **figura 15**) y luego teclee GET/hypertext/WWW/TheProject.html pero esta vez sin HTTP/1.0. Se devolverá la página sin indicación de su tipo de contenido. Este mecanismo es necesario para la compatibilidad hacia atrás; su uso se reducirá a medida que los visualizadores y los servidores basados en solicitudes completas se vuelvan la regla.

Las solicitudes completas se indican por la presencia de la versión del protocolo en la línea de solicitud *GET*, como en la **figura 15**. Las solicitudes pueden consistir en múltiples líneas, seguidas de una línea en blanco para indicar el final de la solicitud, razón por la que se requirió la línea en blanco en la **figura 15**. La primera línea de una solicitud completa contiene el comando (*GET* es sólo una posibilidad), la página deseada y el protocolo/versión. Las líneas subsiguientes contienen cabeceras RFC 822.

C: telnet www.w3.org 80
T: Trying 18.23.0.23
T: Connected to www.w3.org.
T: Escape character is '^]'
C: GET /hypertext/WWW/TheProyect.html HTTP /1.0
C:
S: HTTP /1.0 200Document follows
S: MIME – Version: 1.0
S: server: CERN /3.0
S: Content – Type: text/html
S: Content – Length. 8247
S:
S: <HEAD> <TITLE> The world web consortium (W3C)</TITLE></HEAD>
S: <BODY>
S: <H1>
S: ….
S: ….
S: ….
S: ….
S: …
S: </BODY>

FIGURA 15. Situación ejemplo para obtener una página de la WEB.

Aunque el HTTP se diseñó para usarse en la *Web*, ha sido intencionalmente más general de lo necesario con miras a aplicaciones futuras orientadas a objetos. Por esta razón, la primera palabra de la línea de solicitud completa es sencillamente el nombre del **método** (comando) a ejecutar con la página de la *Web* (u objeto general). Los métodos interconstruidos se listan en la **figura 16**. Después de acceder a objetos generales, también pueden estar disponibles

métodos adicionales específicos para ese objeto. Los nombres son sensibles a mayúsculas y minúsculas, por lo que *GET* es un método legal, pero *get* no.

El método *GET* solicita al servidor que envíe la página (con lo que queremos decir objeto, en el caso más general), codificada adecuadamente en MIME. Sin embargo, si a la solicitud *GET* le sigue una cabecera *If-Modified-Since*, el servidor sólo envía los datos si fueron modificados después de la fecha proporcionada. Usando este mecanismo, un visualizador al que se solicitó una página que está en caché puede solicitarla condicionalmente al servidor, dando la hora de modificación asociada a la página. Si la página en caché aún es válida, el servidor simplemente devuelve una línea de estado anunciado el hecho, eliminando por tanto la carga extra de transferir de nuevo la página.

Método	Descripción
GET	Solicita leer una página de Web
HEAD	Solicita leer la cabecera de una página de Web
PUT	Solicitar almacenar una página de Web
POST	Adiciona a un recurso nombrado (p.ej., página de Web)
DELETE	Elimina la página de Web
LINK	Conecta dos recursos existentes
UNLIK	Rompe una conexión existente entre dos recursos

FIGURA 16. Métodos de solicitud HTTP interconstruidos.

El método *HEAD* simplemente pide la cabecera del mensaje, sin la página. Este método puede servir para obtener la hora de la última modificación, para recolectar información con fines de indicación, o simplemente para probar a validez de un URL. No existen las solicitudes *HEAD* condicionales.

El método *PUT* es el inverso de *GET*: en lugar de leer una página, la escribe. Este método hace posible construir un conjunto de páginas de la *Web* en un servidor remoto. El cuerpo de la solicitud contiene la página y puede codificarse usando MIME, en cuyo caso las líneas que siguen a PUT podrían incluir cabeceras *Content-Type* y de validación de identificación, para demostrar que el solicitante efectivamente tiene permiso de ejecutar la operación solicitada.

Algo parecido a *PUT* es el método *POST*; también lleva un URL pero, en lugar de reemplazar los datos existentes, se “anexa” a ellos en algún sentido generalizado. La publicación de un mensaje en un grupo de noticias y la adición de un archivo a un sistema de boletines electrónicos son ejemplos de anexión en este contexto. Claramente, la intención aquí es hacer que la *Web* asuma la funcionalidad del sistema de noticias USENET.
DELETE hace lo que podría esperarse: elimina la página. Como con *PUT*, la validación de identificación y los permisos desempeñan un papel principal. No hay garantía de que *DELETE* tendrá éxito, puesto que, incluso si el servidor HTTP remoto está dispuesto a borrar la página,

el archivo subyacente puede tener un modo que prohiba al servidor HTTP su modificación o eliminación.

Los métodos *LINK* y *UNLINK* permiten establecer conexiones entre páginas existentes u otros recursos.

Cada solicitud recibe una respuesta que consiste en la línea de estado y, posiblemente, información adicional (por ejemplo, toda o parte de una página de la *Web*). La línea de estado puede tener el código 200 (OK) o cualquiera de varios códigos de error, por ejemplo 304 (no modificado), 400 (lista errónea) o 403 (prohibido).

Los estándares HTTP describen las cabeceras y los cuerpos de mensaje con considerable detalle. Basta decir que son muy parecidos a los mensajes MIME RFC 822, por lo que no los veremos aquí.

Escritura de una página de *Web* en HTML

Las páginas de *Web* se escriben en un lenguaje llamado HTML (*HyperTExt Markup Language,* **lenguaje de marcación de hipertexto**). El HTML permite a los usuarios producir páginas de *Web* que incluyen texto, gráficos y apuntadores a otras páginas de *Web*. Comenzaremos nuestro estudio del HTML con estos apuntadores, puesto que son el pegamento que mantiene unida a la *Web*.

2. ANALIZADOR DE PROTOCOLOS

2.1 INTRODUCCIÓN.

JUICIO DE EXPERTOS

Por: De David Kane

Los succionadores del alcance medio entregan bastantes características para la mayoría de los administradores de la red. Pero una solución high-end, tal como AGILENT ADVISOR SW EDITION puede ser la mejor opción para organizaciones más grandes, porque apoya más protocolos y estructuras de la red.

Los succionadores high-end ofrecen el análisis experto, que en el funcionamiento largo puede reducirlo una compañía los costes de la ayuda. El análisis experto toma la supervisión de un paso más lejos ayudando a la resolución problemas. Después de que se descubra un problema, el software da los detalles específicos sobre él y a veces incluso recomienda los pasos que se tomarán manualmente. Porque proporciona un análisis, un alto nivel de la maestría no es siempre necesario fijar ediciones de la red. Esencialmente, con una solución high-end, joven personal puede desempeñar un papel más importante, problemas de la fijación independientemente por después de las instrucciones proporcionadas por el software.

El AGILENT ADVISOR SW EDITION es una versión software basada en un succionador hardware portable que funcione Windows 98. Al principio, el producto es impresionante. En start-up, ventana principal del ADVISOR da una característica separada de la ayuda, del experto Quickstart, abierto. El experto es parte del sistema de ayuda en línea y ofrece una lista de las preguntas llano-Inglesas tales como "quién está utilizando la mayoría de la anchura de banda de la red?" y "qué protocolos son las estaciones más activas usando?" Al lado de cada uno de estas preguntas están dos acoplamientos: "demuéstreme," cuál describe cómo utilizar la herramienta específica, y "hágala," cuál la lanza realmente. Este sistema de ayuda integrado es probablemente la fuerza más grande del ADVISOR.

Leyendo a través de un almacenador intermediario de la captura del paquete que registre acontecimientos del protocolo, AGILENT hace cada acontecimiento enlazable al sistema de ayuda.

El ADVISOR tiene varias otras características valiosas, tales como filtros bastante personalizados y la capacidad de supervisar sobre dos NICs simultáneamente. El nivel del análisis es también absolutamente impresionante. Desafortunadamente, aunque el análisis

experto del ADVISOR recoge la información detallada, que encontramos muy difícil de tener acceso a estos datos debido al interfaz confuso del programa.

El fallar principal del producto, sin embargo, es su interfaz, que está frustrando particularmente. Las ventanas herramienta-especificas que se abren dentro del programa detraen de su utilidad. El funcionamiento de la herramienta superior de los transmisores, por ejemplo, crea una lista de las varias estaciones usando la mayoría de la anchura de banda de la red. La lista sí mismo podía tener valor sino que por el contrario es una mezcla confusa de las direcciones del IP y de las direcciones del hardware. Las direcciones del hardware son intrínsecamente difíciles de leer (su formato 12-caracteres mira algo como 00:C0:4F:B7:EE:D0). El consejero confunde la materia más lejos agregando la red del ocho-dígito antes de que cada dirección del hardware, yéndose fuera de cualquier delimitador entre los pares de caracteres en la dirección real del hardware y repitiendo la dirección otra vez parenthetically. Una dirección parecería esto en consejero: 00000001-00c04fb7eed0 (00C04FB7EED0). Una lista para una red pequeña de 100 computadoras puede ser insoportable mirar.

Muchas de las exhibiciones gráficas del consejero representan la red como círculo grande, con cada nodo dado un punto de la referencia a lo largo del círculo. Las líneas que conectan cada punto ilustran conexiones entre los nodos; conexiones más grandes una red, y más hechas, más ilegible el gráfico llega a ser.

El interfaz del ADVISOR es idéntico a el que esta usado por sus contrapartes del hardware. La edición del software es también compatible con otros productos de prueba de AGILENT como el AGILNET ADVISOR LAN. Para los administradores ya familiares con la versión del hardware, o mirar para ampliar las capacidades de otros productos de AGILENT actualmente en uso, la edición del software del consejero se parece una lógica, y capaz, opción.

2.2 ANALIZADOR DE PROTOCOLOS ADVISOR SW EDITION

2.2.1 INTRODUCCIÓN

El AGILENT ADVISOR SW EDITION es que un software analizador poderoso diseñado para ayudarle a arreglar y analizar sus redes Ethernet y Fast Ethernet.

Con de ADVISOR SW EDITION, usted puede usar a una computadora personal (PC) provisto con una tarjeta de interfase de red (NIC) o una tarjeta de PCMCIA en lugar de adquirir el hardware ADVISOR LAN.

La industria estándar esta utilizando las Especificaciones los Manejadores de Interfases Red (Network Driver Interface Specification NDIS 5.0) de soporte sacado del estante del NIC y PCMCIA adaptadores de red.

Usted puede usar el ADVISOR SW para:

- Previene los problemas de la red antes de que ellos afecten a los usuarios.
- Resolver rápidamente y eficazmente los problemas de red.
- Optimizar el rendimiento de la red.

Las páginas siguientes proporcionan una apreciación global más detallada de los rasgos del ADVISOR SW EDITION.

- Iniciación con el Analizador Especialista (Expert Analyzer) para ver una apreciación global rápida de la salud, utilización, y la actividad protocolar en su red.
- Examinar la capa física para ver si los nodos en su red se pueden conectar y pueden comunicar.
- Vea quién está enviando el la mayoría el tráfico. Vea qué protocolos ellos están usando. Vea qué estaciones están estableciendo las conexiones.
- Encontrar desde fuera qué red los errores protocolares están ocurriendo en su red.
- Descubre qué nodos están en sus redes.

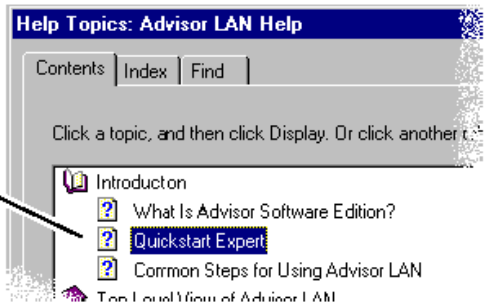
2.2.2 INSTALACION

EMPEZANDO CON UN INICIO RAPIDO DE LA GUIA ESPECIALISTA

(Vea la utilización de salud, y un vistazo a la actividad.)

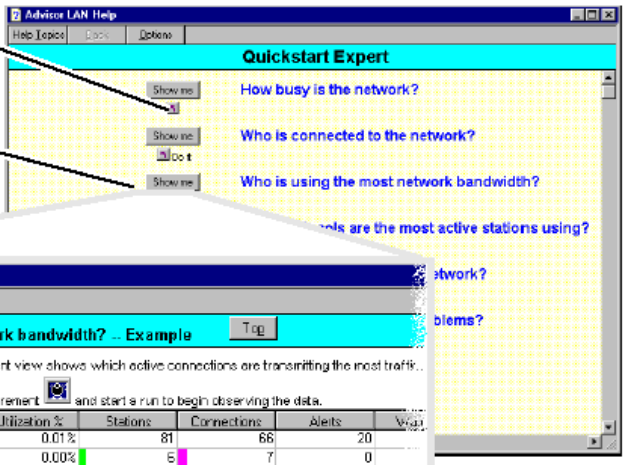
Cuando usted empieza la aplicación de ADVISOR SW EDITION, una ventana de la guía solucionador despliega una lista de problemas comunes de la red que usted puede encontrar con el ADVISOR SW EDITION.

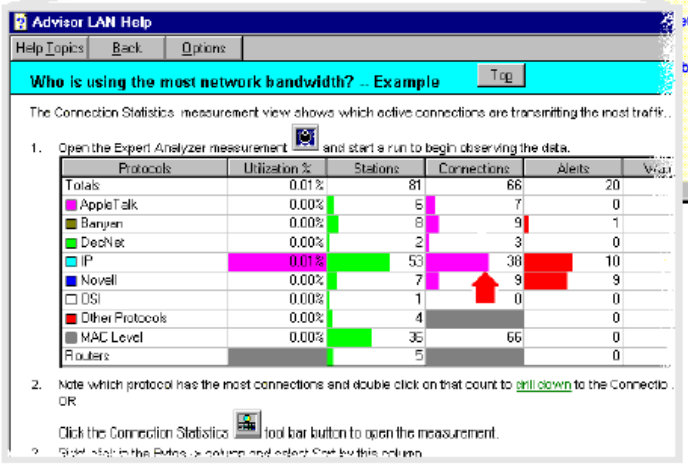
Para desplegar la ventaja para localizar averias
- Arrancar la aplicación
- En el menu HELP
- En linea con la Ventana HELP, use la Viñeta CONTENTS, abrir INTRODUCTION, pulse sobre QUICKSTART EXPERT



Utilice el boton HACER para abrir la ventana de medida apropiada para el solucionador de problemas.

Use el botón SHOW ME para ver un ejemplo de como utilizar la medida para el solucionador de problemas



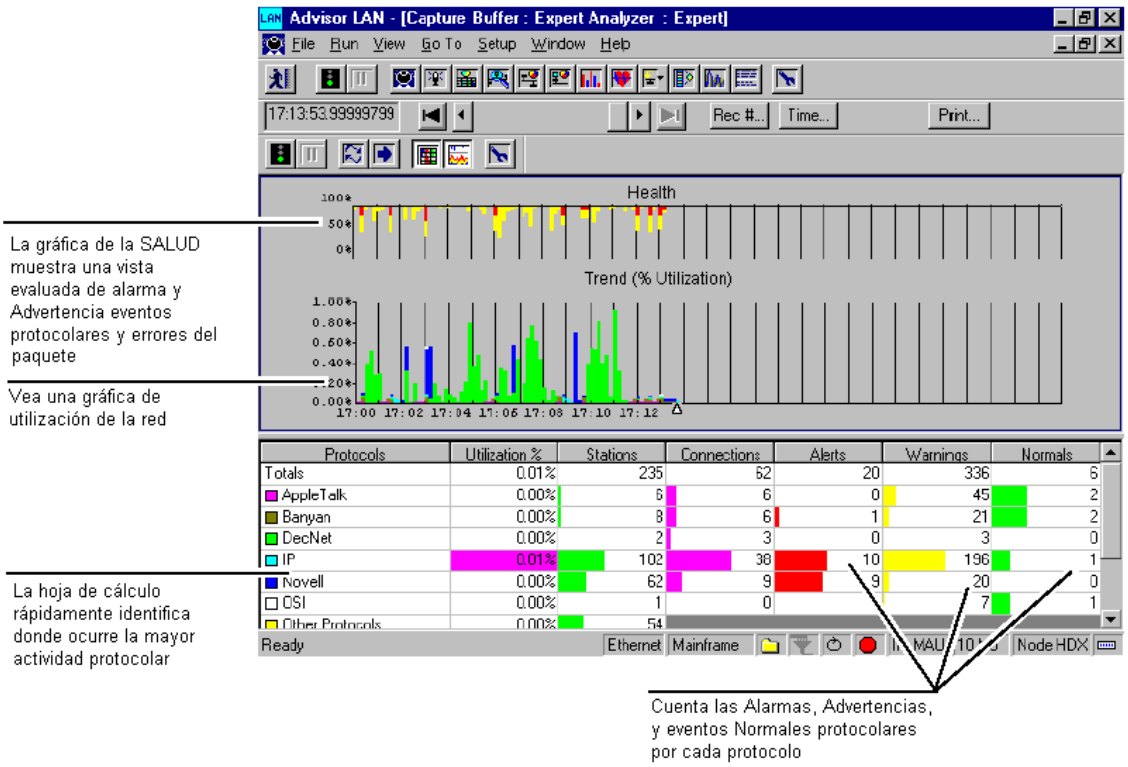


EMPEZANDO CON EL ANALIZAR ESPECIALISTA

(Vea la utilización de salud, y un vistazo a la actividad.)

La medición del Analizador Especialista es una manera excelente de ver la actuación para poner a punto su red. En la sección superior, el gráfico de Salud(Health) empieza con un gráfico del 100% y es decir su red esta libre de errores. Cuando un evento ocurre, un valor se substrae de la salud, más para los eventos alertas, advertir. Estos valores son configurables. El gráfico de tendencia(trend) despliega los valores de parámetros seccionados, resbalando en una ventana de 30 minutos. Cada protocolo mayor se muestra en un color diferente en el gráfico.

Debajo los gráficos de salud y tendencia esta una hoja de cálculo con información sobre su red. Las filas se etiquetan con las pilas protocolares activas descubiertas en la fuente de los datos. La hoja de cálculo muestra cuánta actividad protocolar está ocurriendo en cada uno. Forma esta medición, usted puede rápidamente profundizar para ver los detalles de apoyo para entender por qué una púa del gráfico o una cuenta grande ocurrió.

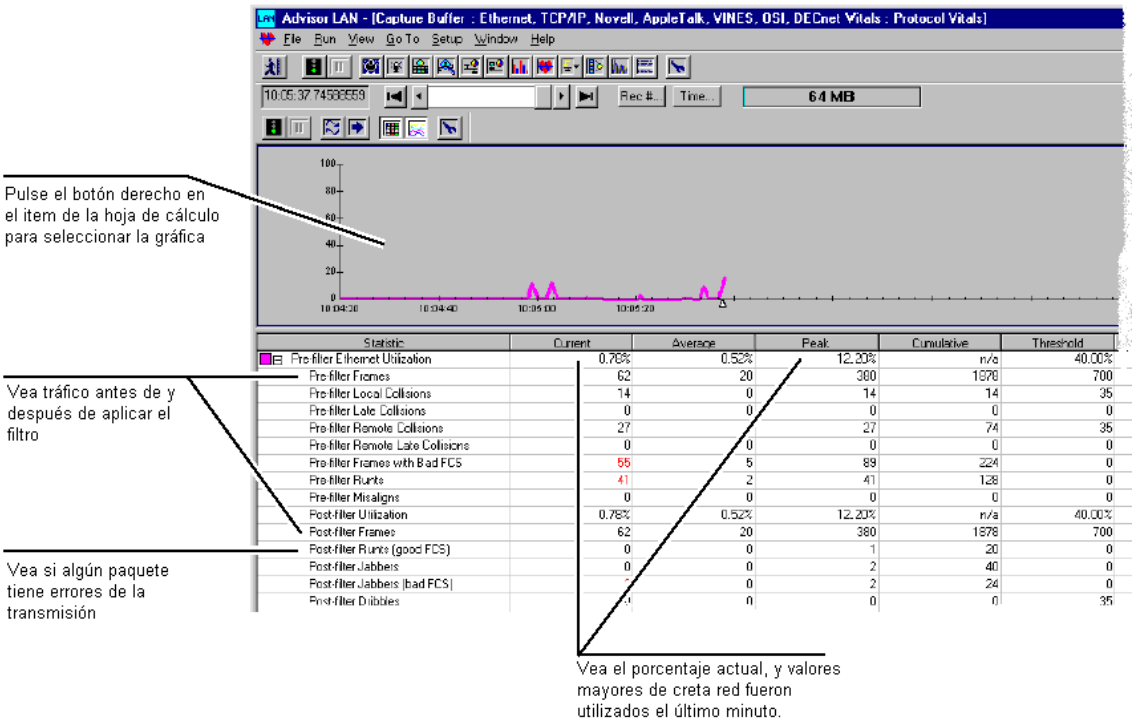


EXAMINANDO LA ACTIVIDAD DE LA CAPA FISICA

(¿Las estaciones pueden conectar y pueden comunicar?)

A menudo, la primera cosa que usted quiere saber es si sus estaciones de la red pueden conectar y comunicar. La medición Protocolar Vital lo muestra rápidamente si la capa física está operando.

Algunos de los contadores Vitales indican la actividad normal y deseable. Otros contadores pueden indicar una seria y potencial actividad perjudicial.



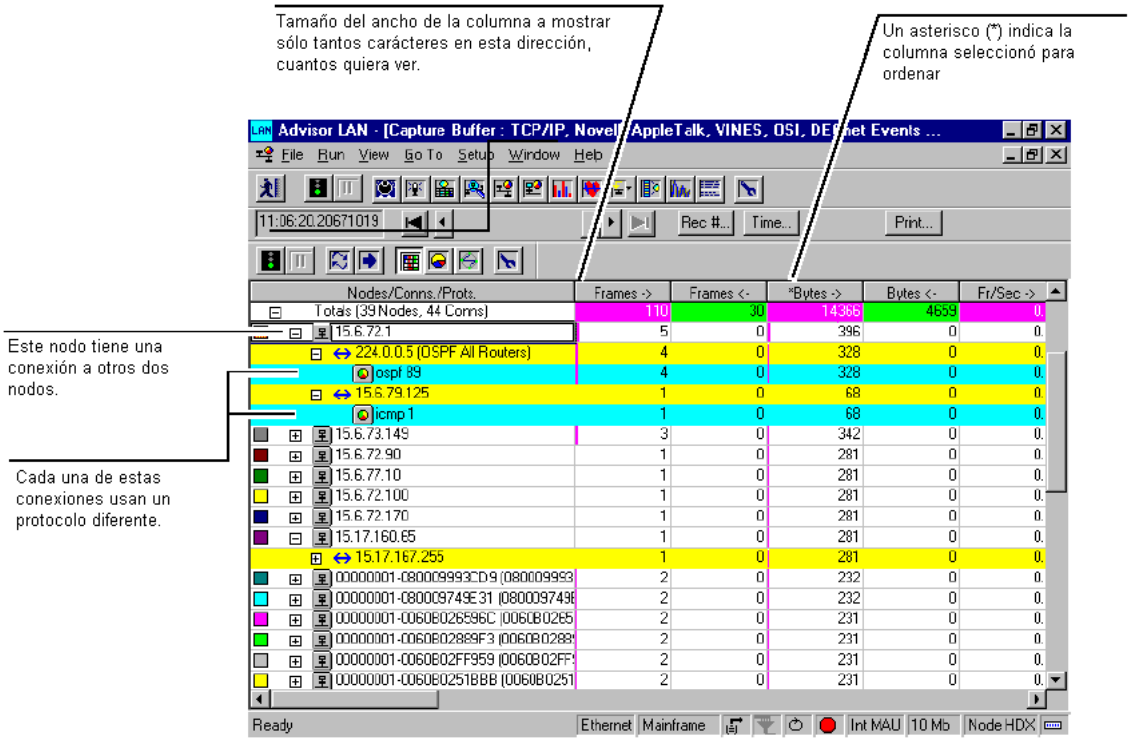
Statistic	Current	Average	Peak	Cumulative	Threshold
Pre-filter Ethernet Utilization	0.78%	0.52%	12.20%	n/a	40.00%
Pre-filter Frames	62	20	380	1878	700
Pre-filter Local Collisions	14	0	14	14	35
Pre-filter Late Collisions	0	0	0	0	0
Pre-filter Remote Collisions	27	0	27	74	35
Pre-filter Remote Late Collisions	0	0	0	0	0
Pre-filter Frames with Bad FCS	55	5	89	224	0
Pre-filter Runts	41	2	41	128	0
Pre-filter Misaligns	0	0	0	0	0
Post-filter Utilization	0.78%	0.52%	12.20%	n/a	40.00%
Post-filter Frames	62	20	380	1878	700
Post-filter Runts (good FCS)	0	0	1	20	0
Post-filter Jabbers	0	0	2	40	0
Post-filter Jabbers (bad FCS)	0	0	2	24	0
Post-filter Dribbles	0	0	0	0	35

EXAMINADO EL TRAFICO

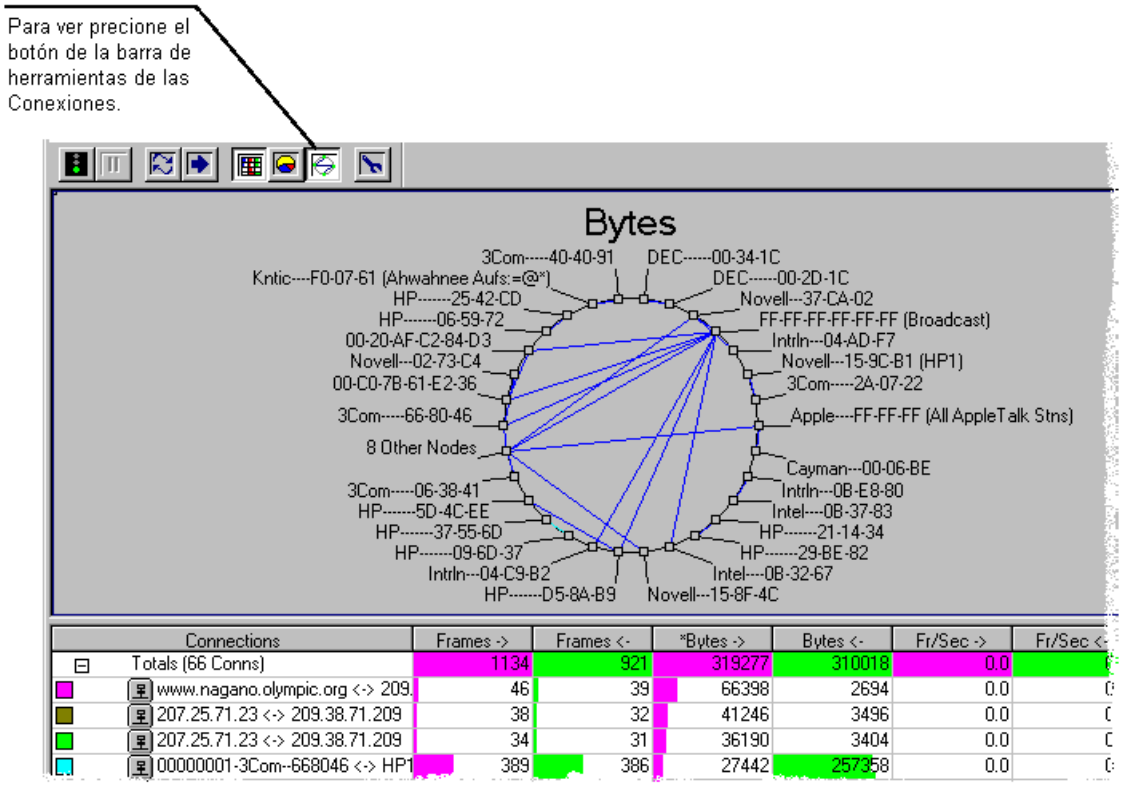
(¿Quién está enviando el la mayoría el tráfico?
¿Quién ellos están hablando a?
¿Qué protocolo ellos están usando?)

Cuando la utilización red es demasiado alta, y está causando la contestación de la red lenta, usted puede utilizar las mediciones Estadísticas de Conexión a ver rápidamente quién está usando la mayoría de ancho de banda de la red. Cuando usted sabe que quién los nodos esta dando problemas, usted puede tomar las decisiones sobre cómo mejorar sus comunicaciones de la red.

Usted puede escoger los parámetros diferentes para ordenar. Esto lo muestra donde la mayoría del tráfico está ocurriendo para las condiciones de tráfico diferentes.



También en la medida de Estadísticas de Conexión, usted puede usar la barra de herramientas para Mostrar las Conexiones en un diagrama de conexiones que gráfica.

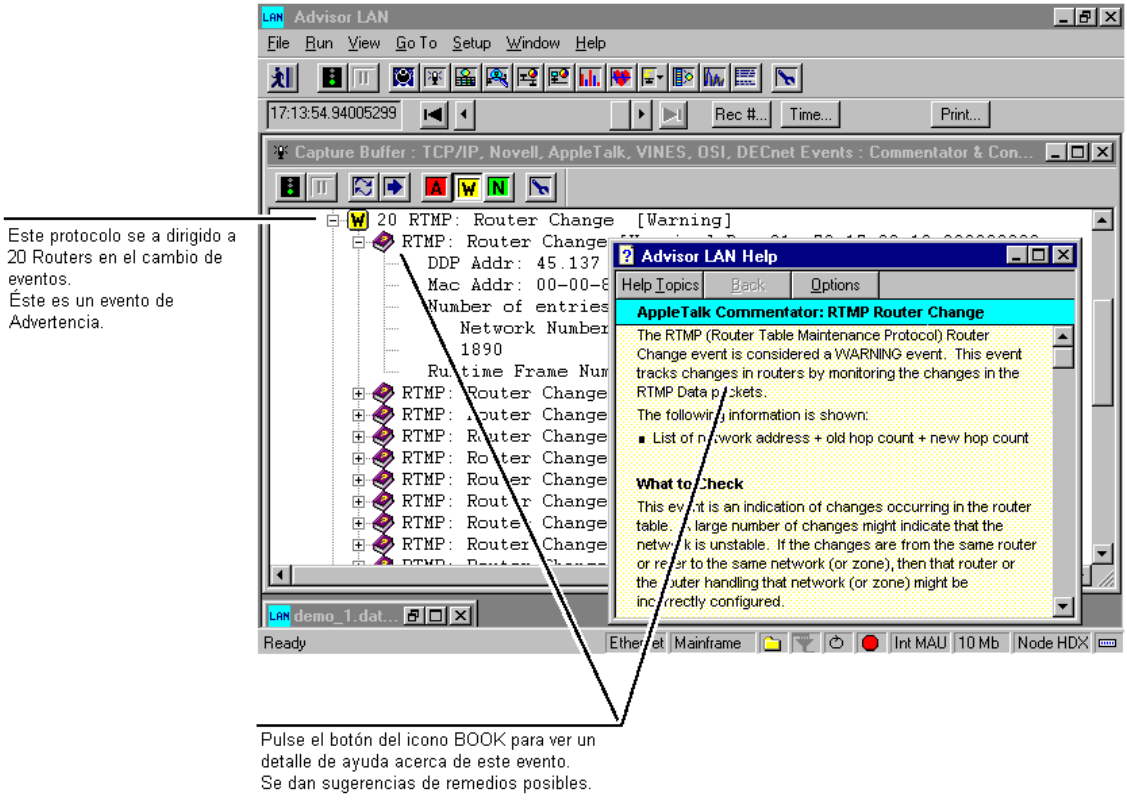


Connections	Frames ->	Frames <-	*Bytes ->	Bytes <-	Fr/Sec ->	Fr/Sec <-
Totals (66 Conns)	1134	921	319277	310018	0.0	0
www.nagano.olympic.org <-> 209.	46	39	66398	2694	0.0	0
207.25.71.23 <-> 209.38.71.209	38	32	41246	3496	0.0	0
207.25.71.23 <-> 209.38.71.209	34	31	36190	3404	0.0	0
00000001-3Com-668046 <-> HP1	389	386	27442	257358	0.0	0

EXAMINANDO ERRORES PROTOCOLARES

(¿Qué errores protocolares están ocurriendo es la red?)

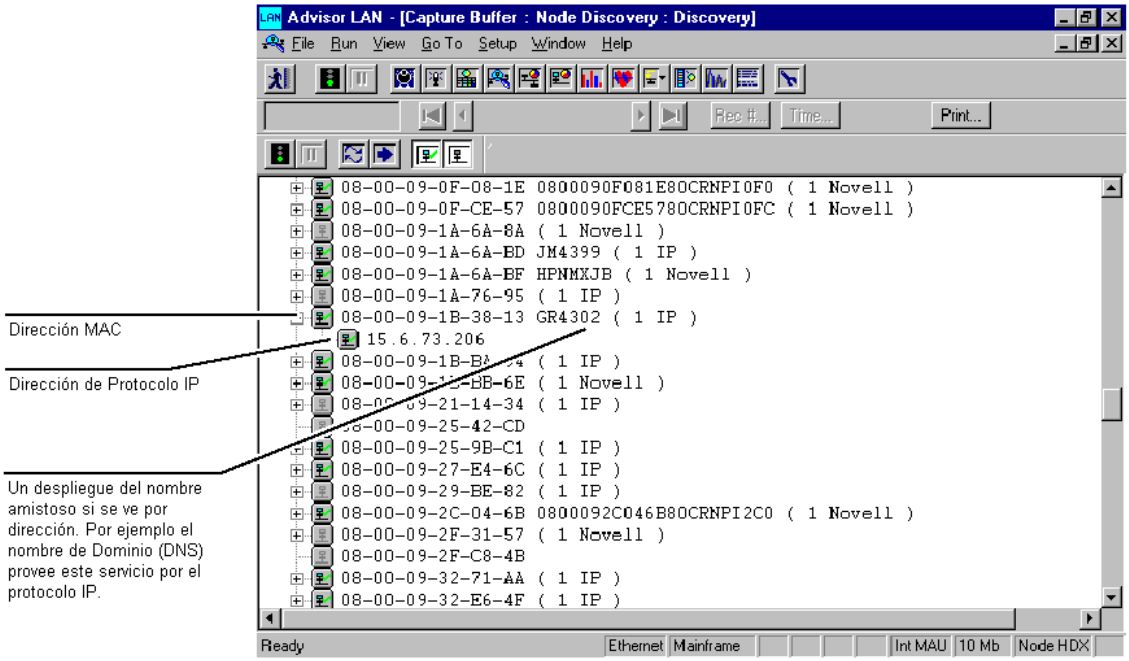
Con el Comentador de Medición, usted puede ver rápidamente que qué eventos protocolares están ocurriendo y organizan los eventos por gravedad.



DESCUBRIENDO QUE NODOS ESTAN EN LA RED

(¿Qué nodos inesperadamente esta en la red?)

Durante la ejecución de la medición, usted puede descubrir todos los nodos que están en su red. Usted puede ver nodos listados por dirección de MAC o dirección de capa de red. Usted también puede ver direcciones conocidas o bien sabe el nombre si es conocido.



DECODIFICACIÓN DE PAQUETES EN SU RED

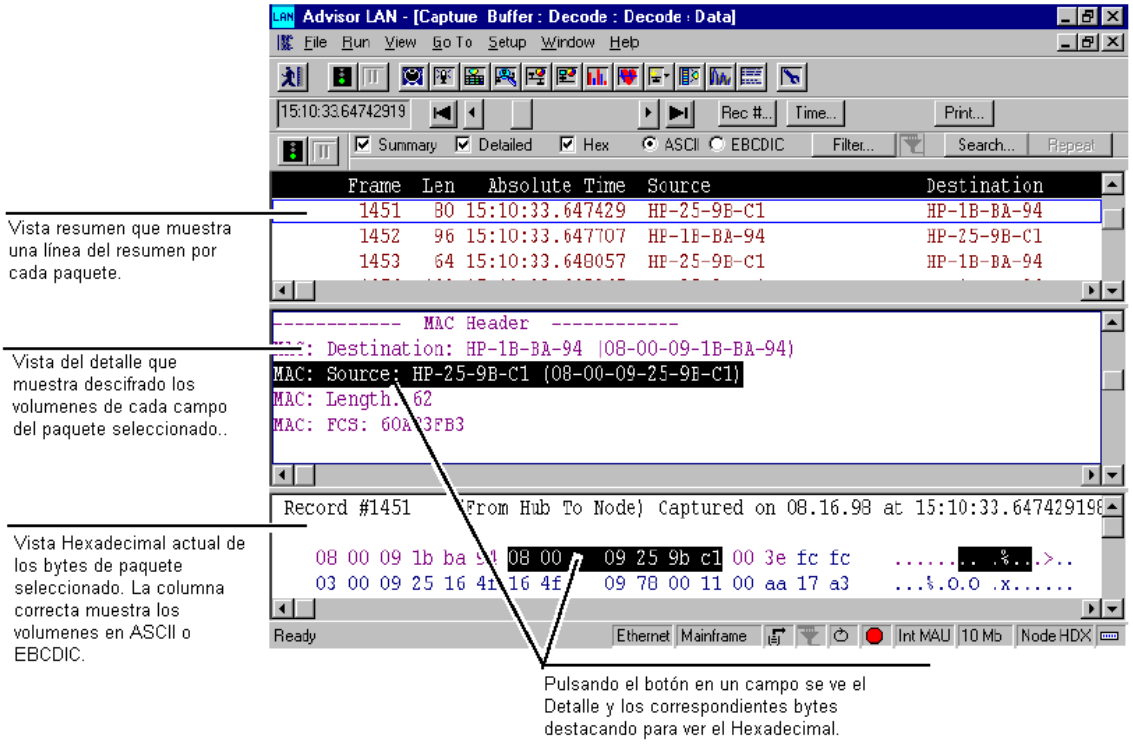
(¿Cuáles son los volúmenes reales de un paquete en mi red?)

El Decodificador de la medición interpreta los datos en un paquete según su protocolo para que usted puede examinar los volúmenes del paquete.

Se agregan remarcado y número del paquete cada paquete para ayudarle a examinar los datos.

Ejemplos de información, usted puede determinar como se ve el Descodificador:

- ¿Este nodo está en la red hace demandas y contestaciones correctamente?
- ¿Esta cronometrando los problemas existentes entre los paquetes en la red?
- ¿Los niveles más altos son de la pila protocolar que opera correctamente?



PUESTA EN MARCHA

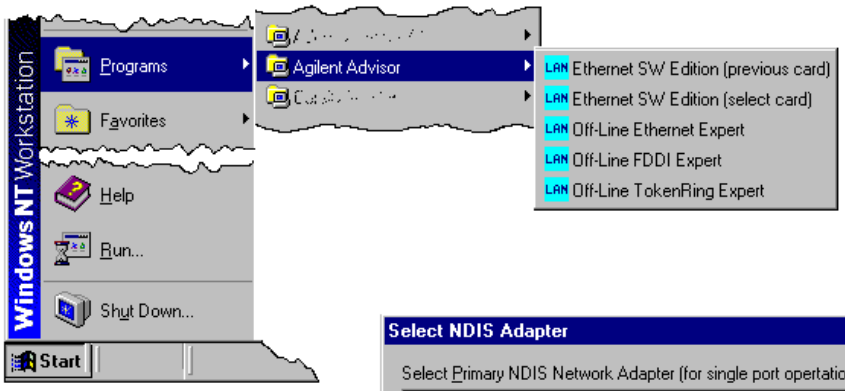
Este capítulo describe los pasos que usted acostumbra a la comprobación inicial con el AGILENT ADIVISOR SW EDITION.

Hay algunos pasos de optimización que usted realiza cada vez iniciando pruebas de su red. Otros pasos usted hace sólo una vez o sólo chequea previamente que todavía son válidos.

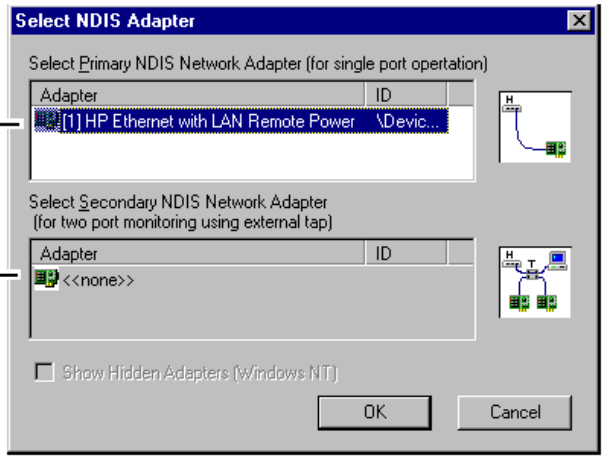
1. Instale una tarjeta de interfase de red (NIC) o una tarjeta de PCMCIA en la computadora personal usted está usando. Instale el software si necesario.

2. Use las instrucciones en el CD que contiene ADVISOR SOFTWARE EDITION CD para la instalación del software.

3. Inicio de la Aplicación ADVISOR SW EDITION



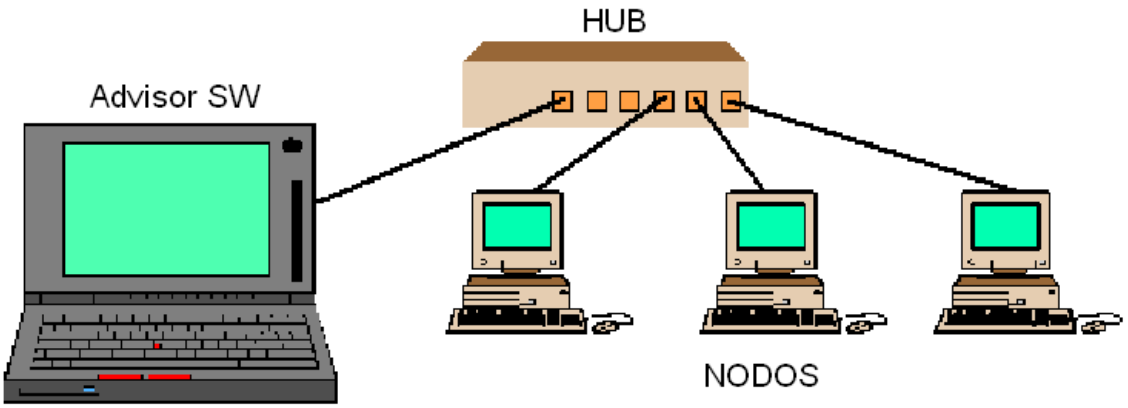
4. Seleccione la tarjeta que esta utilizando NIC o PCMCIA



En la primera sección es utilizada para conectar el ADVISOR SW EDITION como un nodo.

En la segunda sección es utilizada para conectar el ADVISOR SW EDITION en modo monitor de red.
Esta conexión usa en full Duplex Ethernet Tap, producto número J1990A.

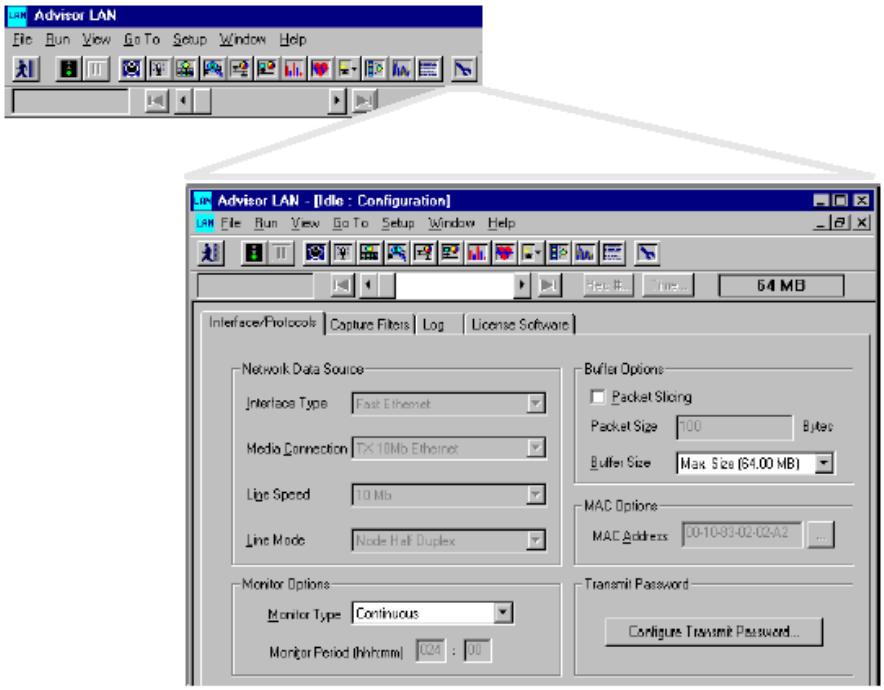
5. Conexión para la red.



6. Configure el ADVISOR SW EDITION PC.

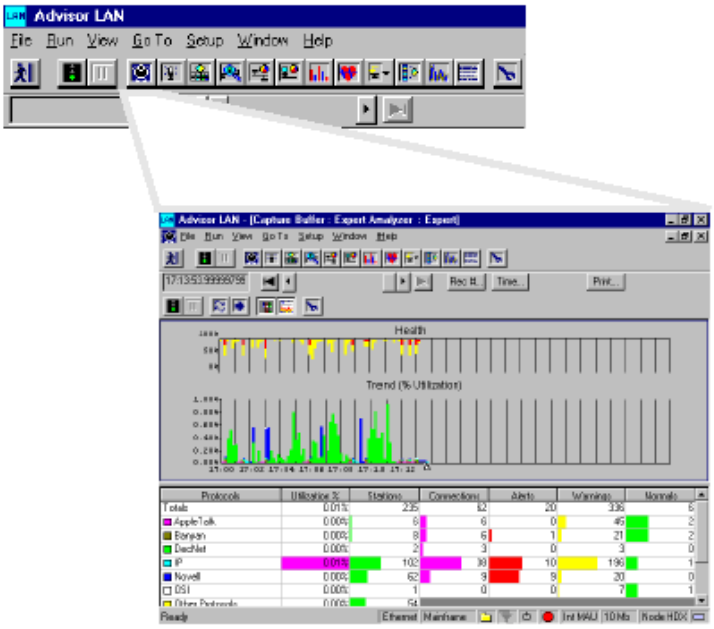
¿Qué pruebas de Puerto para la conexión?

¿Cómo usted quiere la captura buffer para operar?



7. Selección de medición.

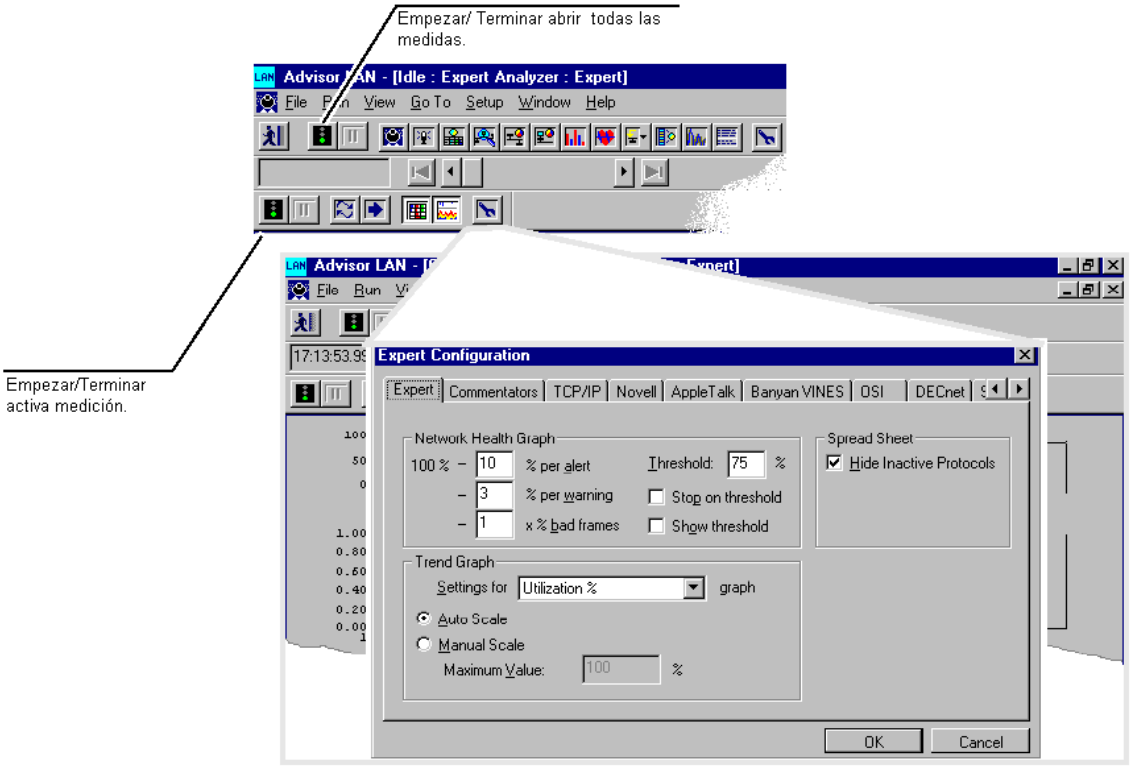
La medición del Analizador Experto esta listo para iniciar.



8. Configure la medición y empieza la correr.

¿Qué tiempo quiere por periodo de la muestra?

¿Usted quiere poner cualquier umbral?



INSTALACIÓN DE LA TARJETA NIC Y PCMCIA, Y EL SOFTWARE

Instalación de la tarjeta NIC y PCMCIA:

Usted puede tener una tarjeta de interfase de red NIC o PCMCIA, coloca la tarjeta en su PC para la interfase física específica que usted piensa conectar a.

PRECAUCION

Las descripciones generales del procedimiento para la tarjeta NIC y PCMCIA se proporcionan debajo. Para los detalles más completos, usted debe seguir las instrucciones del fabricante para la instalación.

GENERALIDADES DE LA INSTALACIÓN DE LA TARJETA DE INTERFASE A LA RED (NIC).

Éstos son los pasos generales por instalar a un NIC en su escritorio del PC.

1. Asegúrese el sistema ha apagado.
2. Quite el cordón de poder del sistema.
3. Abra la carcasa de la computadora para tener acceso a la tarjeta madre.
4. El hallazgo una hendidura vacía y quitar la tapa de metal.
5. La inserción de la nueva tarjeta.
6. Asegúrese que la tarjeta se sienta propiamente en la hendidura y entonces reemplaza el tornillo para sostener la tarjeta en el lugar.
7. Realice el montaje de la carcasa de la computadora y coloque todos los tornillos.
8. Encienda y se asegúrese que no hay ningún mensaje del error durante el inicio en la Prueba del Sistema.
9. La carga cualquier driver requerido (Si no ya se instaló).

GENERALIDADES DE LA INSTALACIÓN DE LA TARJETA PCMCIA.

Éstos son los pasos generales por instalar una tarjeta de PCMCIA un PC portátil.

1. Asegúrese que el portátil esta apago.
2. Inserte la tarjeta de la red en la hendidura a la PC.
3. Asegúrese que esta colocado del lado correcto de la tarjeta y se asiente propiamente.
4. Conecte el cable LAN a la tarjeta.
5. Encienda el poder.
6. Utilice el Panel de Control - Network para verificar que el adaptador está operando propiamente.

VERIFICACIÓN DE LA INSTALACIÓN DE LA TARJETA DE RED

Éstos son los pasos generales para verificar que una tarjeta de red se instalo propiamente.

1. Reiniciar el PC.

2. Si un Wizard se despliega al iniciar, usted se guiará durante la instalación de los drivers.
3. Seleccione el botón Inicio y seleccione Configuración – Panel de Control.
4. Para Windows 98:
    a. Seleccione el icono de Sistema.
    b. Seleccione Administración de Dispositivos.
    c. Extienda la línea de Adaptadores de Red
    d. Encontrar el adaptador apropiado de la lista.
    e. Seleccione el botón Propiedades para ver el estado del adaptador.

    Para Windows NT y 2000:
    a. Seleccione el icono de la Red.
    b. Seleccione la etiqueta de los Adaptadores para ver el estado del adaptador.

INSTALACIÓN DEL SOFTWARE.

Use las instrucciones que posee el CD del ADVISOR SW EDITION para instalar el ADVISOR SW EDITION SOFTWARE.

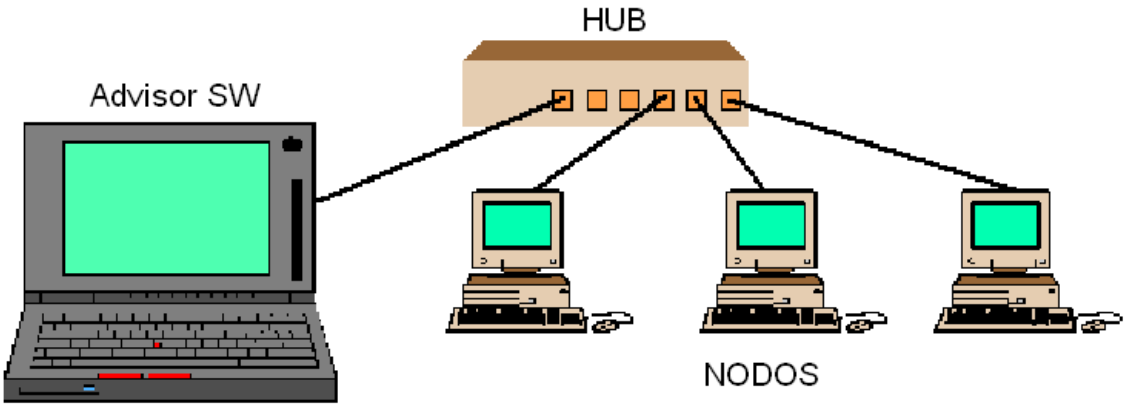
Si usted está instalando el software de la aplicación, siga las instrucciones proporcionadas con ese software.

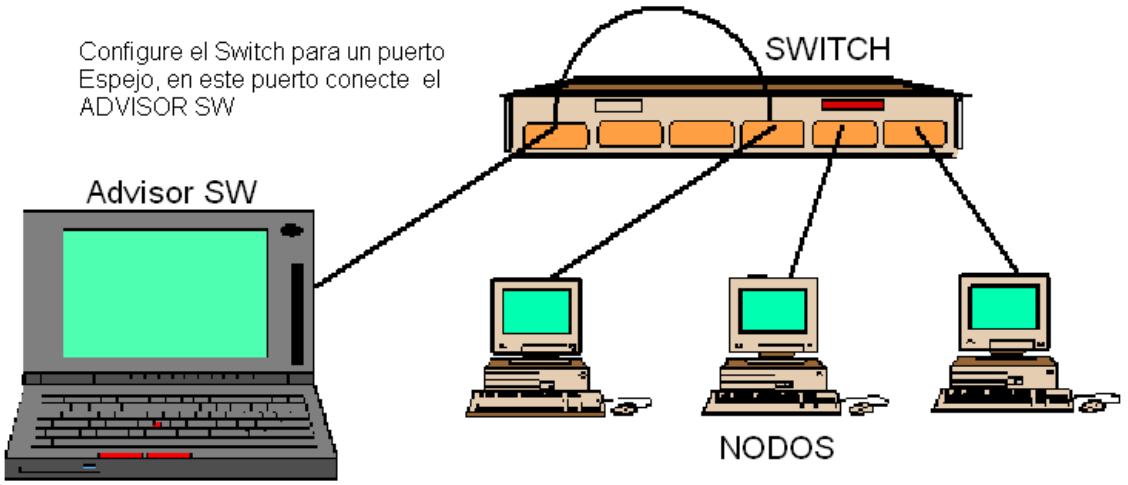
CONECTANDO A UNA RED

(Para Conectar como un Nodo)

CONECTE COMO UN NODO

Cuando usted conecta el ADVISOR SW EDITION como un nodo de un segmento una red de computadoras, el ADVISOR actúa como el nodo regular en la red. El ADVISOR puede ver el tráfico que ocurre en ese segmento de la red.



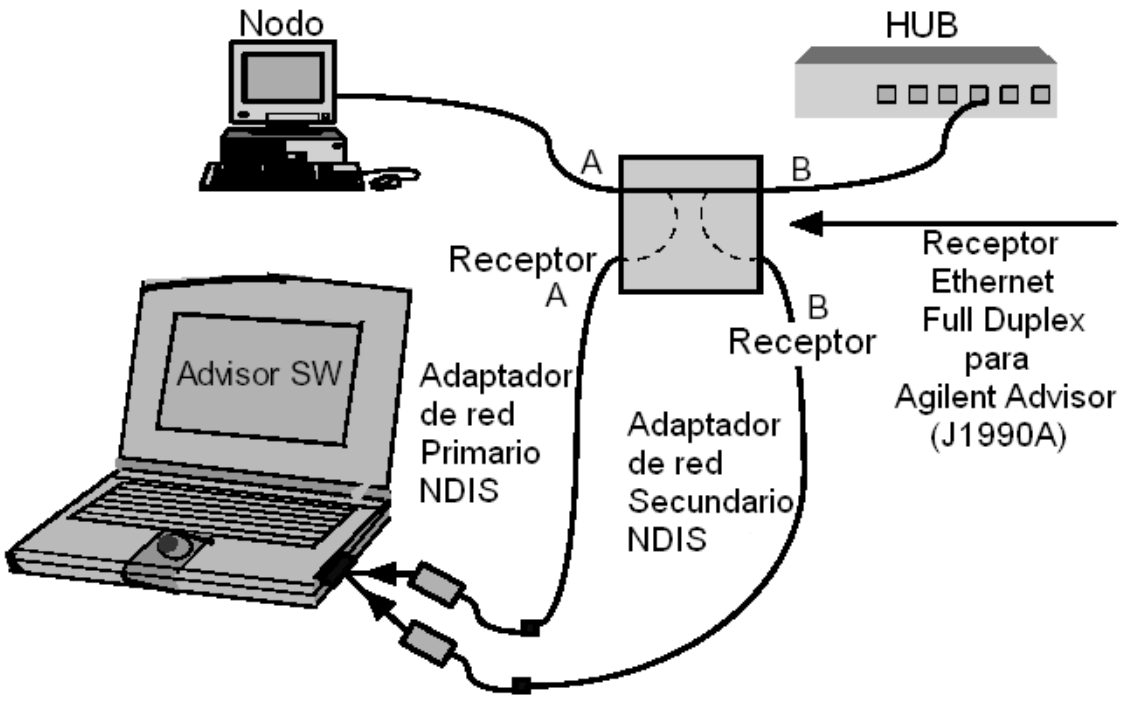


PARA EN MODO DE MONITOR DE TOQUE

(Conecte en Modo de Monitor de toque.)

Cuando usted conecta el ADVISOR SW EDITION en modo monitor de toque, el ADVISOR se inserta entre dos dispositivos de la red. El ADVISOR SW EDITION no actúa recíprocamente con el tráfico que ve, él sólo pasivamente despliegues que el tráfico que está pasando entre los dos dispositivos conectados a él. Cuando conectó en este modo de monitor de toque, ADVIDOR SW EDITION no puede transmitir.

TIP: ADVISOR SW EDITION no puede supervisar en el auto negociar el modo porque la negociación automática sólo puede ocurrir entre los dos dispositivos del extremo. En este modo, el ADVISOR SW EDITION está sólo supervisando y no puede participar en el proceso de auto negociación.

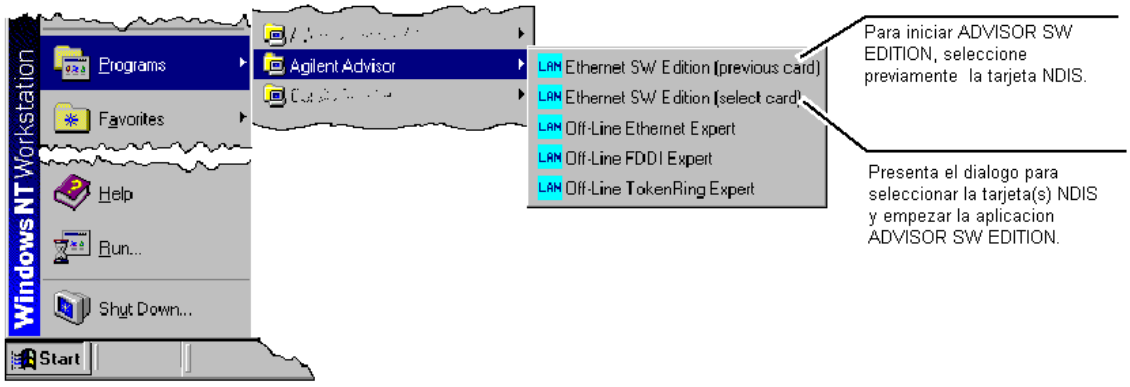


Supervisando entre un nodo y un HUB/SWITCH con dos tarjetas PCMCIA y receptor (Tap).

INICIANDO LA APLICACION

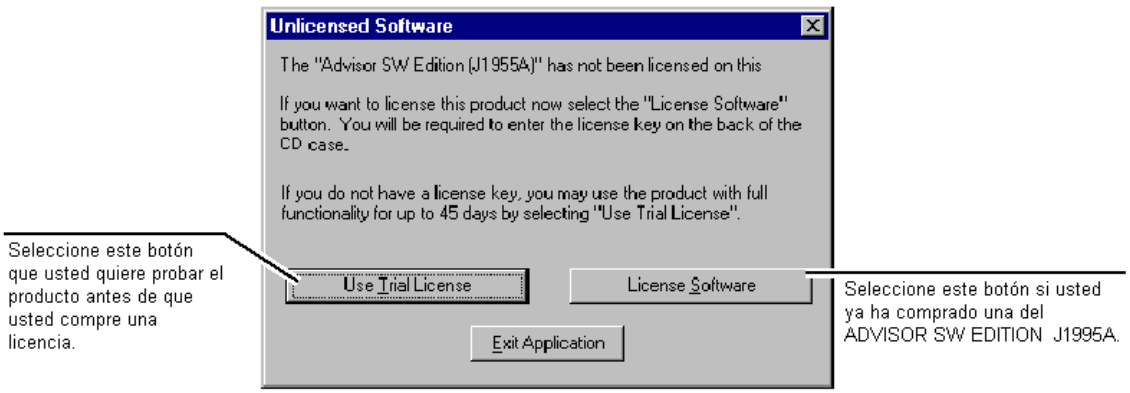
Iniciar el ADVISOR SW EDITION:

- Utilice el menú Inicio para abrir la aplicación ADVISOR SW EDITION.



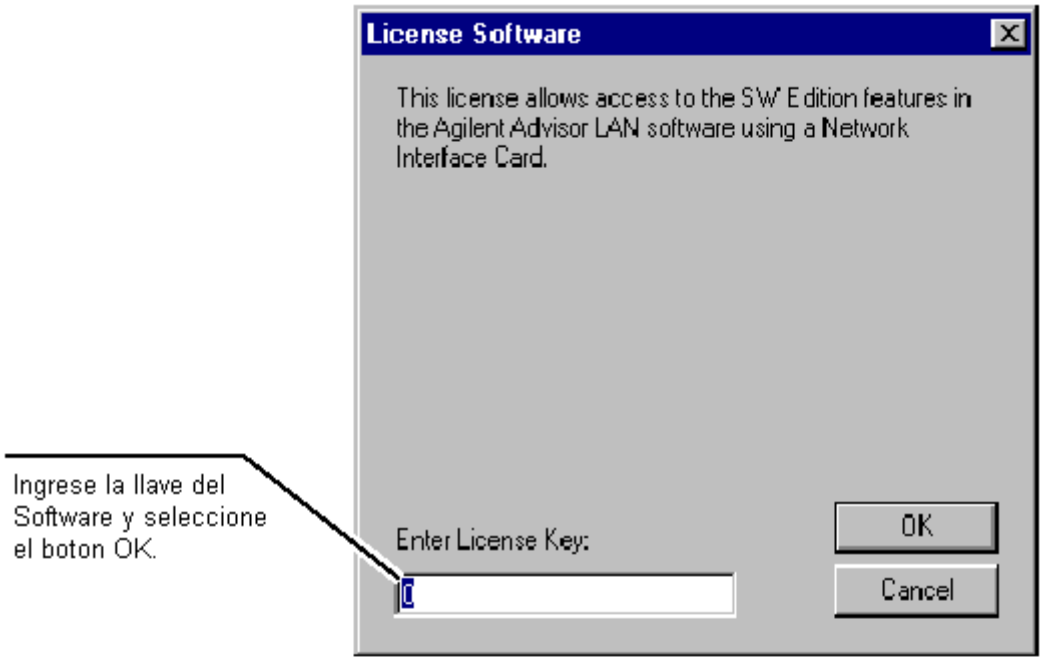
Selección el tipo de Software y la licencia a utilizar:

- El primer paso para iniciar el ADVISOR SW EDITION, en la caja dialogo seleccionar el tipo de licencia a utilizar.
- Si usted no ha adquirido el J1955A ADVISOR SW EDITION, debe utilizar la licencia temporal.
- Si usted adquirió el J1955A ADVISOR SW EDITION, debe ingresar ahora la Llave de licencia del software.



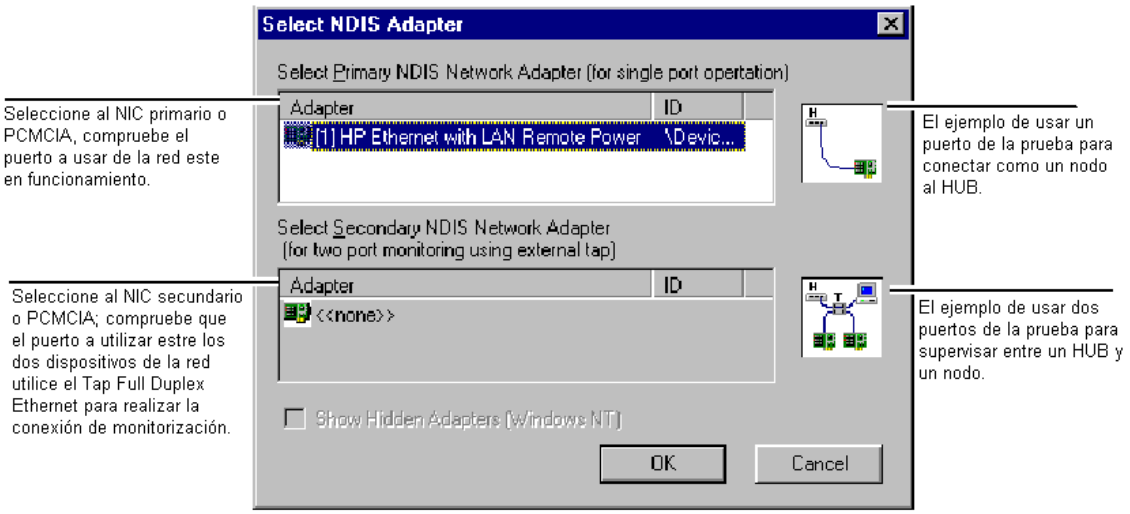
Ingreso de la Llave del Software si la tiene:

- Si seleccionó el botón LICENCE SOFTWARE, en la nueva caja de dialogo de ingresar la Llave de software que se encuentra con la documentación del J1955A ADVISOR SW EDITION.

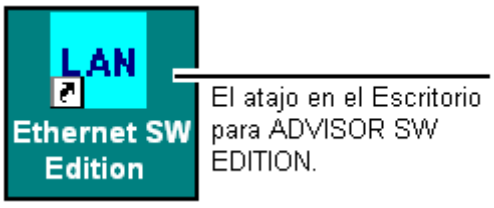


SELECCIÓN LA ESPECIFICACIÓN DEL DISPOSITIVO DE INTERFASE DE RED (Network Device Interface Specification NDIS), ADAPTADOR DEL PUERTO DE PRUEBA.

Cada vez que usted inicia el ADVISOR SW EDITION, usted puede seleccionar una tarjeta de NDIS o usted puede usar la tarjeta seleccionada previamente.



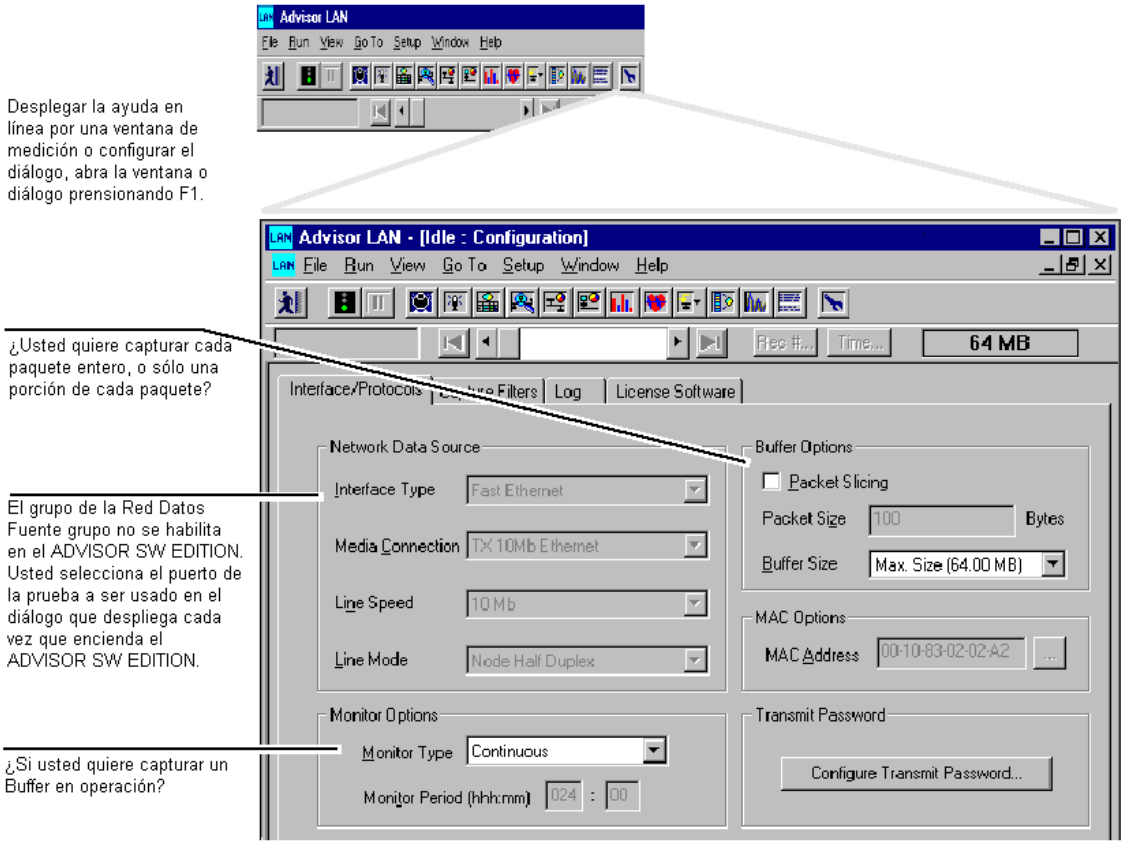
Si esta "selección" no se despliega, si usted puede tomar un atajo de Escritorio, o, si usted escoge "ETHERNET SW EDITION" en el menú INICIO.



CONFIGURANDO EL INSTRUMENTAL

Antes de que usted corra una medición, usted necesita configurar el ADVISOR SW EDITION.

Usted puede crear los Filtros / Contadores, para controlar qué paquetes se guarda en el buffer de captura. Y, usted puede almacenar la bitácora o resultados de la medición en un periodo largo de tiempo.

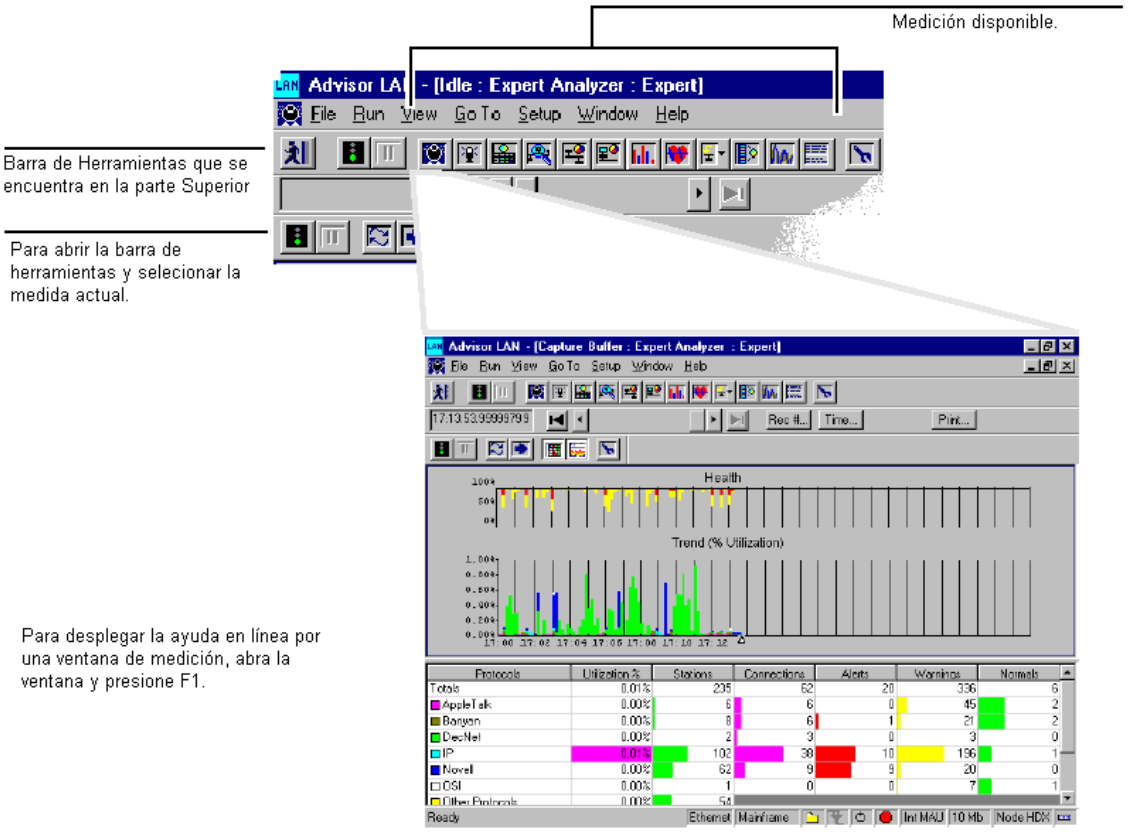


SELECCIONANDO UN MEDIDA

El ADVISOR SW EDITION tiene varios mediciones sobre que puede mostrar los parámetros diferentes cómo su red está operando. Usted puede abrir una medida con los botones en la barra de herramientas o con los artículos en el menú File – Open Measurement.

Los botones de la barra de herramientas son la manera más rápida de abrir una medida. Posicionando el cursor encima de un botón de la barra de herramientas para desplegar el nombre de la medida.

Más de una la medición puede estar abierta y corriendo en un momento.



CONFIGURANDO UN MEDIDA

Antes de correr una medición, usted puede personalizar una medida por seleccionar parámetros, para el control de los operadores de medición.

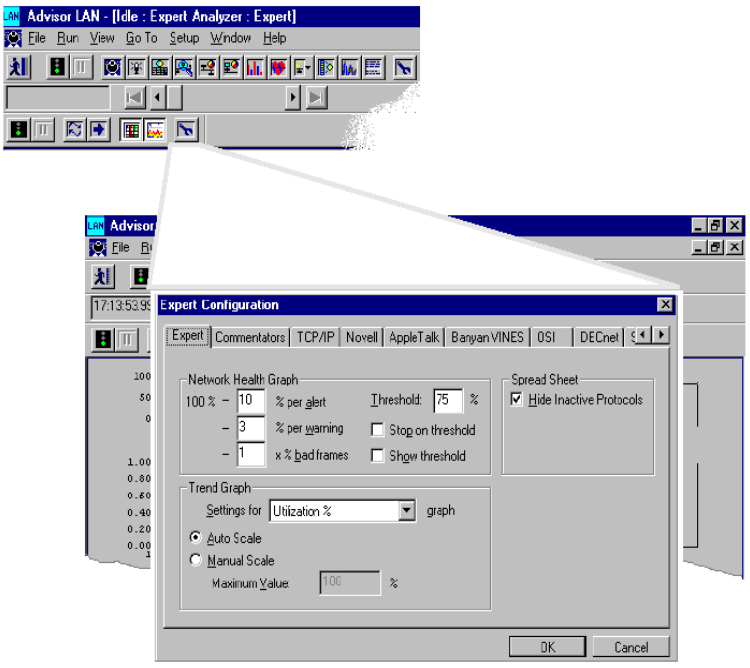
Si la medida del Analizador Especialista está abierto, el botón de la configuración para alguna medida no despliega.

¿Qué despliegue actualiza el intervalo que usted quiere usar?

¿Usted quiere poner los umbrales para determinar qué contar?

¿Usted quiere seleccionar qué mensajes protocolares para contar o ignorar?

Desplegar la ayuda en línea por una ventana de la medida o configurar el diálogo, abra la ventana

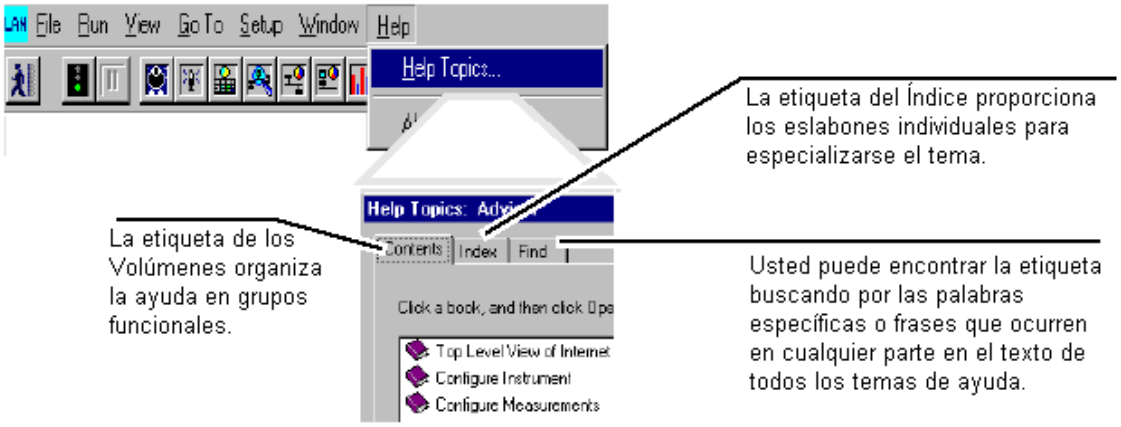


ENCONTRANDO MAS INFORMACION

(ADVISOR SW EDITION la ayuda en línea)

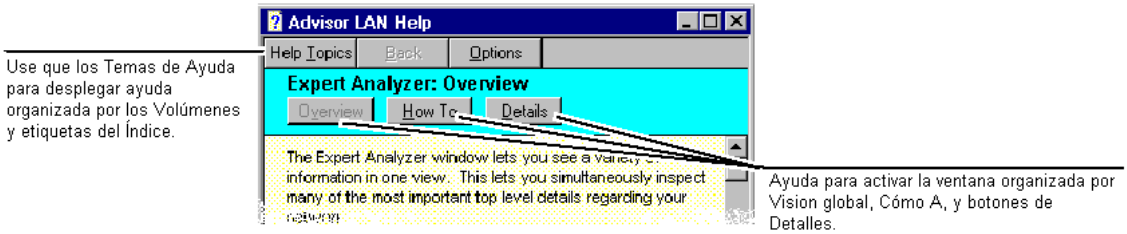
Se construye la ayuda en línea en la aplicación ADVISOR SW EDITION. Usted puede acceder la forma de ayuda en la barra del menú a la cima de la ventana de la aplicación o usando la tecla de F1.

El menú de Ayuda abre la ventana ayudan general del ADVISOR SW EDITION para que usted pueda escoger las tres maneras diferentes de encontrar la ayuda .



El contexto la Ayuda En línea Sensible:

- Usted puede encontrar la información rápidamente sobre la ventana de ADVISOR SW EDITION actualmente seleccionada presionando la tecla F1.



2.2.3 PRACTICA REALIZADA EN ESTE ANALIZADOR

Memoria técnica de una práctica realizada en la Universidad Politécnica de Madrid donde se empleo el AGILENT ADVISOR SW EDITION:

PRÁCTICA: Configuración Básica de Sistemas IP
Memoria

(Disponible en formato electrónico en: **www.dit.upm.es/david/master/lab-redes-ip/**)

1. Equipos utilizados

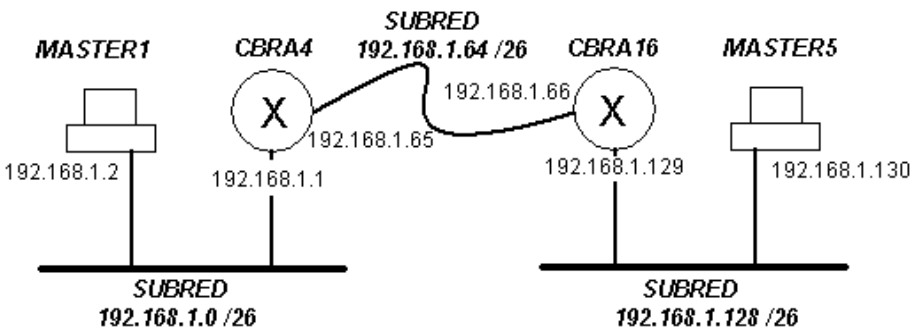
Escriba los números de los equipos utilizados:

- Router (CBRA1, CBRA2, etc): CBRA 16 – CBRA4
- PC (1, 2, etc): MASTER5 – MASTER1

2. Plan de Numeración

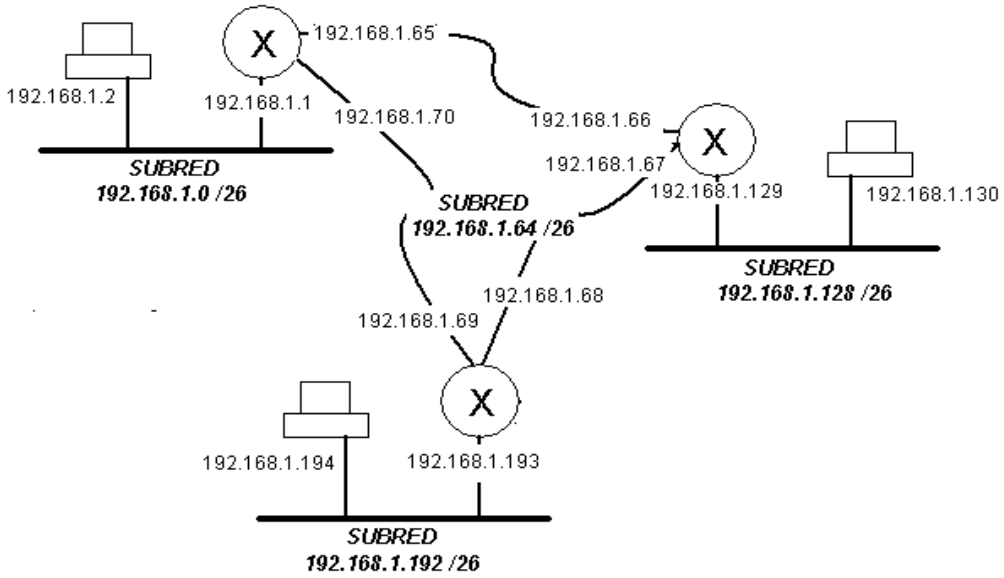
- Describa el esquema de subnetting utilizado, especificando la máscara elegida y los prefijos de subred, así como la primera y última dirección de cada subred:

192.168.1.0 /26	192.168.1.1 /26 – 192.168.1.62 /26
192.168.1.64 /26	192.168.1.65 /26 – 192.168.1.126 /26
192.168.1.128 /26	192.168.1.129 /26 – 192.168.1.190 /26



- Describa el esquema de subnetting modificado para la red de la Figura 2.

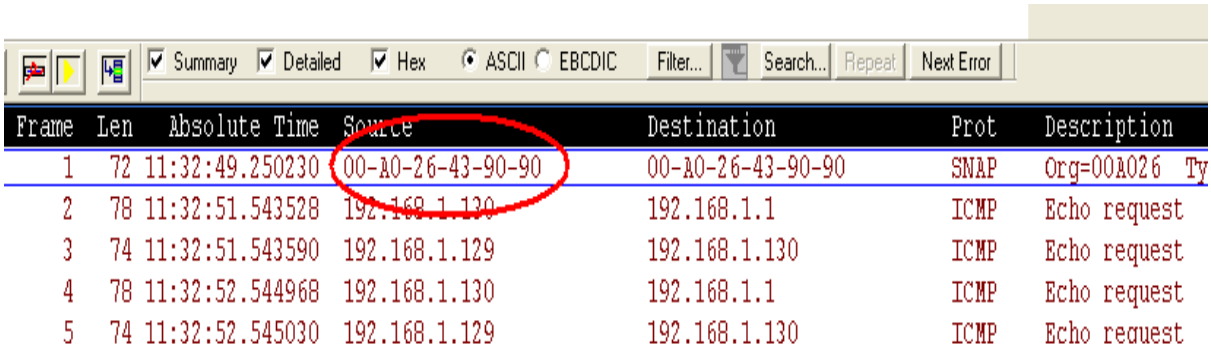
192.168.1.0 /26	192.168.1.1 /26 192.168.1.62 /26
192.168.1.64 /26	192.168.1.65 /26 192.168.1.126 /26
192.168.1.128 /26	192.168.1.129 /26 192.168.1.190 /26
192.168.1.192 /26	192.168.1.193 /26 192.168.1.254 /26



3. Configuración Básica del Router

- Especifique los parámetros de configuración IP del router. Incluya, además, la dirección MAC del mismo.

El ***IP*** 192.168.1.129 y la ***MAC*** es la 00-A0-26-43-90-90



Frame	Len	Absolute Time	Source	Destination	Prot	Description
1	72	11:32:49.250230	00-A0-26-43-90-90	00-A0-26-43-90-90	SNAP	Org=00A026 Ty
2	78	11:32:51.543528	192.168.1.130	192.168.1.1	ICMP	Echo request
3	74	11:32:51.543590	192.168.1.129	192.168.1.130	ICMP	Echo request
4	78	11:32:52.544968	192.168.1.130	192.168.1.1	ICMP	Echo request
5	74	11:32:52.545030	192.168.1.129	192.168.1.130	ICMP	Echo request

4. Configuración Básica del PC

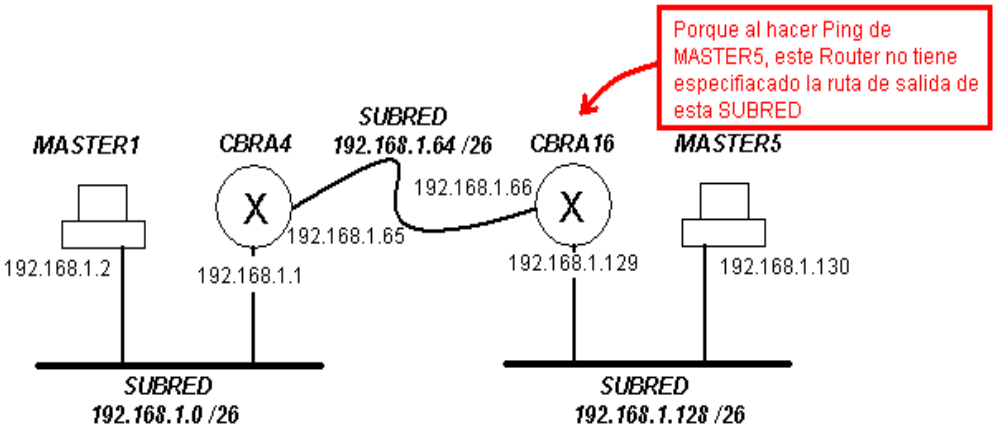
- Especifique los parámetros de configuración IP del PC. Incluya, además la dirección MAC del mismo.

El ***IP*** 192.168.1.130 y la ***MAC*** es la 00-40-95-32-31-1C

5. Configuración de las Tablas de Encaminamiento

- Describa con precisión porqué no funcionan los ping propuestos en el enunciado. Haga un esquema de la red y explíquelo sobre el mismo.

Por que no existe la ruta de encaminamiento, hacia el Router. Por tel motivo al hacer PING, solo lleva hasta el Router de la Red local.



- Especifique las entradas en las tablas de encaminamiento que ha tenido que añadir para garantizar la conectividad total:

```
S3 IP config>list all
Interface addresses
IP addresses for each interface:
  intf  0   192.168.1.129    255.255.255.192  NETWORK broadcast,    fill 0
  intf  1   192.168.1.66     255.255.255.192  NETWORK broadcast,    fill 0
  intf  2                          IP disabled on this interface

Routing

route to 192.168.1.0,255.255.255.192 via 192.168.1.65, cost 1
route to 192.168.1.128,255.255.255.192 via 192.168.1.66, cost 1

Protocols
Directed broadcasts: enabled
RIP: disabled
OSPF: disabled
Per-packet-multipath: disabled
Ip classless: disabled

Pool
First address: 192.168.0.0
Last address: 192.168.255.255
```

6. Configuración de un servidor de DHCP

```
S3 Config>p dhcp
-- DHCP Configuration --fig>
S7 Config
S3 DHCP config>enable server
Protocol not found
 DHCP Server: enabledConfig>
S3 DHCP config>server
S7 Config>list devi
S3 DHCP-Server config>list all   Ifc Type of interface
 ============================
 =    GLOBAL Parameters    =
 ============================
 =      HOST List    0     =
LAN1     0 Quicc Ethe
 ============================
   No Host defined
S3 DHCP-Server config>add subnet smaster5
 Subnet Address [0.0.0.0]? 192.168.1.128
S7 Config>ip
 Subnet Mask [0.0.0.0]? 255.255.255.192 ip
Internet protocol us
S7 Config>save
 Final IP Address [192.168.1.130]? 192.168.1.189

Saving configuration...OK
S3 DHCP-Server SUBNET [smaster5 0]>option routerrt
Are you sure to restart the system
 Router [0.0.0.0]? 192.168.1.129
Flash
S3 DHCP-Server SUBNET [smaster5 0]>
Initializing
```

3. CONCLUSIONES

Para tener una idea de los protocolos, y sus características, en la parte inicial de este trabajo monográfico se los describe sin llegar a profundizar, pues para el manejo del ADVISOR SW EDITION, se debe tener un conocimiento de los protocolos que transitan en una red.

Como conclusión de la realización de este trabajo tenemos el conocimiento de una nueva herramienta para la mejor administración de redes, y tener un mejor entendimiento de los protocolos que de transportan en una red.

Demás con un manejo adecuado de este Software también se puede diagnosticar los problemas que se presentan en una red, cosa que es muy difícil encontrar el origen de estos problemas cuando es una red muy agrande, pero aún cuando la red es el resultado la unión de varios tipos de red.

4. GLOSARIO

ACK (Affirmative Acknowledgment): respuesta afirmativa. Confirma o valida un bloque.
ADCCP (Advanced Data Communication Control Procedures). Desarrollado por la organización Americana de Normalización (ANSI) y conocido como Norma Federal Americana.
APDU (unidad de datos del protocolo de aplicación)
ANSI (American National Standard Institute)
BDLC (Burroughs Data Link Control) y UDLC (Univac Data Link Control). Utilizados por Burroughs y Univac respectivamente.
CCITT (Comité Consultatif International de Télégraphique et Téléphonique)
DEL (Data Link Escape): utilizado para cambiar el significado de los caracteres de control.
EIA (Electronic Industries Association)
ENQ (Enquiry): petición de repuesta de la otra estación. Se utiliza también para establecer el enlace.
EOT (End of Transmission): identifica el final de una secuencia completa de transmisión y normalmente implica la liberación del enlace.
ETX (End of Text): identifica el final de un bloque y el final del texto de un mensaje.
ETB (End of Transmisision Block): identifica el final de un bloque pero indica que siguen mas bloques en secuencia del mismo bloque.
HDLC (High-level Data Link Control).Constituye una familia de protocolos definida por la Organización Internacional de Normalización (ISO).
IDU (unidad de datos de la interfase)

ISO (International Organization for Standarization)
LAPB (Link Acces Procedure Balanced).Subconjunto de HDLC adoptado por el CCITT para el nivel de enlace de la interfase X.25 de acceso a las redes públicas de conmutación de paquetes.
NAK (Negative Acknowledgment): respuesta de rechazo. Indica que el bloque anterior se recibió con errores.
PDU (unidad de datos del protocolo)
SAP (punto de acceso al servicio)
SDLC (Synchronous Data Link Control). De IBM, similar a un subconjunto de HDLC aunque con variaciones adicionales.
SDU (unidad de datos del servicio)
SOH (Start of header): este carácter identifica el principio de una secuencia de caracteres que constituyen la cabecera de un mensaje.
SPDU (unidad de datos del protocolo de sesión)
STX (Start of Text): indica el principio de los datos del bloque.
SYN (Syncronous Idle): dos o más de estos caracteres proporcionan un medio para que el

receptor adquiera y mantenga la sincronización de carácter.

TPDU (unidad de datos del protocolo de transporte)

5. BIBLIOGRAFIA

TANENBAUM A. S.: “Redes de Computadoras”, Camille Trentacoste,vol 3, NJ: Prentice may, 2001

AGILENT TECHNOLOGIES.:”Advisor SW Edition”, Colorado Springs

WEB:

www.agilent.com

www.globalink.com

www.babylon.com