CURSO DE ANALISIS DE TRAFICO

Por Alejandro Corletti (acorletti@hotmail.com)

1. INTRODUCCION:

1.1. Presentación de modelo de capas.

Son varios los protocolos que cooperan para gestionar las comunicaciones, cada uno de ellos cubre una o varias capas del modelo OSI (Open System interconnection), la realidad, es que para establecer la comunicación entre dos ETD se emplea más de un protocolo, es por esta razón que se suele hablar no de protocolos aislados, sino que al hacer mención de alguno de ellos, se sobre entiende que se está hablando de una ***PILA* de protocolos**, la cual abarca más de un nivel OSI, son ejemplo de ello X.25, TCP/IP, IPX/SPX, ISDN, etc.

Una forma de agruparlos es como se encuentran cotidianamente los siete niveles del modelo OSI en tres grupos que tienen cierta semejanza en sus funciones y/o servicios:

OSI	Generalizado
Aplicación	APLICACION
Presentación	
Sesión	
Transporte	TRANSPORTE
Red	RED
Enlace	
Físico	

La ISO (International Standard Organization), estableció hace 15 años este modelo OSI que hoy lleva la denominación ISO 7498 o más conocida como X.200 de ITU.

1.2. Modelo OSI y DARPA (TCP/IP).

El modelo OSI es, sin lugar a dudas el estándar mundial por excelencia, pero como todo esquema tan amplio presenta una gran desventaja, el enorme aparato burocrático que lo sustenta. Toda determinación, protocolo, definición o referencia que este proponga debe pasar por una serie de pasos, en algunos casos reuniendo personal de muchos países, que demoran excesivo tiempo para la alta exigencia que hoy impone Internet. Hoy al aparecer un nuevo dispositivo, protocolo, servicio, facilidad, etc. en Internet, el mercado si es útil, automáticamente lo demanda, como ejemplo de esto hay miles de casos (chat, IRC, SMS, WAP, etc). Si para estandarizar cualquiera de estos se tardara más de lo necesario, los fabricantes, se verían en la obligación de ofrecer sus productos al mercado, arriesgando que luego los estándares se ajusten a ello, o en caso contrario, los clientes finales sufrirían el haber adquirido productos que luego son incompatibles con otros. Hoy no se

puede dar el lujo de demorar en una red cuyas exigencias son cada vez más aceleradas e imprevisibles.

Para dar respuesta a esta nueva REVOLUCION TECNOLOGICA (Internet), aparecen una serie de recomendaciones ágiles, con diferentes estados de madurez, que inicialmente no son un estándar, pero rápidamente ofrecen una guía o recomendación de cómo se cree que es la forma más conveniente (según un pequeño grupo de especialistas) de llevar a cabo cualquier novedad de la red.

Se trata aquí de las RFC (Request For Commentaries), que proponen una mecánica veloz para que el usuario final no sufra de los inconvenientes anteriormente planteados, dando respuesta a las necesidades del mercado eficientemente.

Se produce aquí un punto de inflexión importante entre el estándar mundial y lo que se va proponiendo poco a poco a través de estas RFC, las cuales en muchos casos hacen referencia al modelo OSI y en muchos otros no, apareciendo un nuevo modelo de referencia que no ajusta exactamente con lo propuesto por OSI. Este modelo se lo conoce como Pila, stack o familia TCP/IP o también como modelo DARPA por la Agencia de Investigación de proyectos avanzados del DoD (Departamento de Defensa) de EEUU, que es quien inicialmente promueve este proyecto.

Este modelo que trata de simplificar el trabajo de las capas, y por no ser un estándar, se ve reflejado en la interpretación de los distintos autores como un modelo de cuatro o cinco capas, es más, existen filosóficos debates acerca de cómo debe ser interpretado.

En este texto, se va a tratar el mismo como un modelo de cinco capas, solamente por una cuestión práctica de cómo ajustan las mismas a los cuatro primeros niveles del modelo OSI, tratando de no entrar en la discusión Bisantina del mismo, y dejando en libertad al lector de formar su libre opinión sobre el mejor planteo que encuentre.

Si se representan ambos modelos, sin entrar en detalles de si las distintas capas coinciden exactamente o no (pues este es otro gran tema de discusión, que no será tratado en este texto), se pueden graficar más o menos como se presenta a continuación:

<table>
<tr><th>OSI</th><th>DARPA o TCP/IP</th></tr>
<tr><td>Aplicación</td><td rowspan="3">Aplication</td></tr>
<tr><td>Presentación</td></tr>
<tr><td>Sesión</td></tr>
<tr><td>Transporte</td><td>Transport</td></tr>
<tr><td>Red</td><td>Internetwork</td></tr>
<tr><td>Enlace</td><td>Medium Access</td></tr>
<tr><td>Físico</td><td>Phisical</td></tr>
</table>

1.3. Conceptos de: Primitivas, servicios y funciones, SAP, UDP y UDS.

1.3..1. Ente: Elemento activo que ejerce funciones o proporciona servicios a sus niveles adyacentes.. El ente puede ser Soft (Ej: Compresión de datos) o Hard (Ej: Microprocesador para armado de paquetes).

1.3.2. SAP (Service Access Point): Punto situado en la interfaz entre dos capas. En dicho punto estarán disponibles los Servicios requeridos y las Respuestas. Me significa explícitamente hacia que protocolo se dirige a través de esa interfaz. A través del SAP se puede multiplexar procesos, pues es el que me indica hacia que proceso se refiere un determinado Header.

1.3.3. Primitivas: Los mensajes entre entes se llevan a cabo a través de cuatro primitivas:

- Solicitud.
- Respuesta.
- Confirmación.
- Indicación.

1.3.4. SDU (Service Data Unit): Datos que se mantienen inalterados entre capas pares y se van transmitiendo en forma transparente a través de la red.

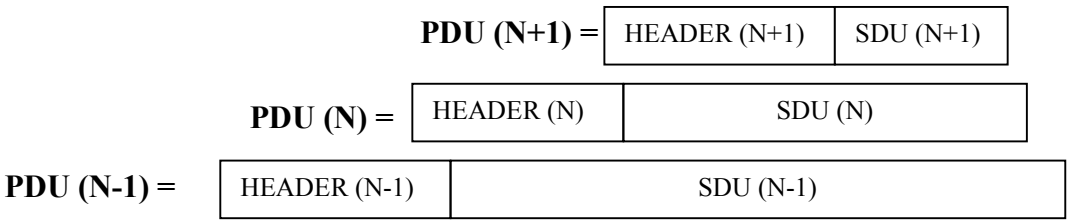
1.3.5. PDU (Protocol Data Unit): UDS más la información de control (Header) de ese nivel.

1.3.6. IDU (Interface Data Unit): Unidad de información que se transmite a través de cada SAP.

1.3.7. ICI (Information Control Interface): Información que el ente N+1 transfiere al ente N para coordinar su funcionamiento y queda en ese nivel (No pasa al siguiente).

Gráficos Resumen:

UDP = UDS + Header.



En cada capa se **"Encapsula"** el PDU recibido de la capa superior y se agrega un Header (En la capa 2 también una cola).

1.4. Funciones y/o servicios.

Sin entrar en detalles específicos de diferencias entre servicios y/o funciones, en este punto, se tratará de desarrollar cuáles son las tareas que se pretende que realice un esquema de comunicaciones para poder transmitir información en forma transparente a un usuario. Una vez analizadas estas tareas, se dividirán en un enfoque de niveles que es el que propone OSI, entrando en el detalle de cual de ellos desempeña cada una de las funciones y/o servicios.

1.4.1. Segmentación y reensamble:

Esta tarea trata de ajustar el tamaño de los datos a transferir al óptimo a colocar en el canal de comunicaciones. Este tamaño dependerá de varias causas:

- Determinados protocolos sólo aceptan un tamaño máximo o exacto de información (Ej ATM = 53 Bytes, Ethernet < 1526 Bytes, etc).
- Control de errores más eficiente.
- Equilibrado uso del canal (Evitar monopolios).
- Empleo de buffer más pequeños.
- DESVENTAJAS: Mayor información de control. Genera más interrupciones.

1.4.2. Encapsulamiento:

Se entiende por encapsulamiento al agregado de información de control a las unidades de datos y al tratamiento de ese bloque como un todo llamado UDP (Unidad de datos del Protocolo), el cual es entregado al nivel inferior como una "Caja Negra" pues es totalmente transparente para el nivel inferior todo lo que existe allí adentro, tomándolo completo como una Unidad de datos para ese nivel. Esta información de control que se adiciona, puede incluir alguno de los siguientes items:

- Dirección.
- Códigos de detección de errores.
- Información de control del protocolo.

1.4.3. Control de la conexión:

Esta tarea comprende todos los pasos necesarios para el establecimiento de la conexión, la transferencia de datos y el cierre de la conexión en los casos en que esta secuencia sea necesaria.

1.4.4. Entrega ordenada:

A medida que la información va descendiendo de nivel en el modelo, como así también cuando es colocada en el canal de comunicaciones y transferida a través del mismo, va sufriendo transformaciones yo viaja por caminos diferentes. Acorde al nivel responsable de estas transformaciones, existirán tareas que se encargarán por distintas técnicas, de entregar al nivel superior, las unidades de datos en la misma secuencia con que fue recibido en su nivel par en el ETD origen.

1.4.5. Control de flujo:

Esta actividad consiste en la posibilidad de regular la corriente de información a través del canal de comunicaciones. El control de flujo va desde la técnica más simple: Parada y espera, Hasta la de ventana deslizante, que permite tener hasta "n" tramas en el canal pendientes de ser entregadas o recibidas.

1.4.6. Control de errores:

El control de errores es la actividad que permite asegurar la confiabilidad de los datos en cada uno de los niveles pares de cada ETD. Como se tratará más adelante, el control de errores de un nivel, *no exime de ejecutar esta tarea a cualquier otro*, pues cada uno abarcará

determinados tramos dentro de la red, pudiendo ocurrir que el error no se produzca en el de su responsabilidad, ante lo cual no sería detectado, excepto que otra capa también lo esté haciendo. Para esta actividad se pueden emplear dos técnicas, FEC (Forward Error Control) y BEC (Backward Error Control).

1.4.7. Direccionamiento:

El concepto de direccionamiento es muy amplio, abarcando de alguna u otra forma, más de un nivel del modelo.

Si se hace una analogía con un envío postal, para un usuario final, la única dirección que le interesa, es la del domicilio postal al que desea enviar su carta. Detrás del mismo, existe todo un sistema diseñado y puesto en funcionamiento que permite que la carta que es depositada en un Buzón, sea colocada en una determinada bolsa (y no otra) cuyo código de identificación sólo conocen los empleados de las sucursales de correo: esta bolsa se dirigirá hacia un avión, ferrocarril, camión etc, cuya identificación de vuelo, andén, etc; sólo conocerá el nivel de empleados del ferrocarril, aeropuerto o transporte automotor, este proceso se puede desglosar hasta el mínimo detalle formando parte de un conjunto de direccionamiento absolutamente desconocido para un usuario final. No puede caber duda que quien diseñó el sistema de distribución de correo, conoce este detalle, y lo fue fraccionando por niveles de distribución para justamente lograr este efecto de transparencia.

Al referirse a un sistema de transferencia de datos ahora, es difícil luego de este ejemplo pensar que con una sola dirección el mismo funcionaría. Este planteo es el necesario para detallar todos los tipos de direccionamiento existentes, los cuales se pueden clasificar en cuatro categorías:

- Direccionamiento de nivel:

  Cada una de los distintos tipos de direcciones que se emplean en cada nivel, acorde al protocolo que se está empleando en ese nivel (Ej : X.25, Frame Relay, Ethernet, etc).

- Espacio de direcciones:

  Se puede tratar como: Local (“Mi Red”) o Global (Todos los ETD a los que se puede tener acceso fuera de la Red Local).

- Identificador de conexión:

  A que tipo de protocolo se está accediendo.

- Modo de direccionamiento:

  Se trata del tipo de destinatario del mensaje, este puede ser: Unicast – Multicast – Broadcast.

1.4.8. Multiplexado:

El concepto de multiplexado físico, a través de las distintas técnicas (TDM, PDM, FDM, etc) permite compartir un mismo canal físico por varios canales lógicos. Bajo este mismo concepto varias aplicaciones pueden estar ejecutándose durante una misma sesión (Ej: En una conexión a Internet, se puede estar consultando una página Web {HTTP}, enviando un correo {SMTP}, transfiriendo un archivo {FTP}, etc). Estos son ejemplos donde un mismo nivel permite operar con más de un nivel superior, entendiéndose como multiplexión lógica.

1.4.9. Servicios de transmisión:

Los distintos tipos de servicios de transmisión ofrecen las opciones de optimizar la relación costo/beneficio en el esquema de comunicaciones, por medio de este se puede establecer las siguientes opciones:

- Prioridades (Se basa en que ciertos mensajes necesitan ser transmitidos con menor demora que otros, como pueden ser los de control o servicios de red).
- Grado de Servicio (Distintas opciones de calidad de Servicio {QoS}).
- Seguridad (Permite implementar estrategias de seguridad, en cuanto a la confiabilidad de datos, descarte de tramas, recuperación, fallas, etc).

1.5.Presentación de la familia (pila) de protocolos TCP/IP.

En 1973 , los investigadores Vinton Cerf de la Universidad UCLA y Robert Kahn del MIT, elaboran la primera especificación del protocolo de comunicaciones TCP. Y es en 1983 cuando se abandona el protocolo de comunicaciones anterior NPC y se sustituye por el actual protocolo TCP/IP.
En 1987 la red dedicada a las news USENET , se integra en Internet . Usenet fue creada por tres estudiantes de Duke y Carolina del Norte en 1979 , Tom Truscott , Jim Ellis y Steve Bellovin . En cuanto al WWW (World Wide Web) , todo empezó en 1980 en el CERN (Consejo Europeo para la investigación Nuclear), Suiza. El investigador Tim Berners-Lee implementó una aplicación que establecía enlaces entre una serie de nodos y permitía ir avanzando por ellos. Diez años más tarde formó un equipo junto con Robert Cailliau y realizaron el primer prototipo sobre una máquina NEXT. La conexión se realizaba haciendo TELNET a una máquina , ejecutando en esta última el navegador.
En 1993 Berners-Lee crea junto a Eric Bina en el NCSA el primer navegador gráfico Mosaic , y un año más tarde funda la compañía Netscape Communications.
Esta breve introducción histórica, es la que va dando origen a los primeros protocolos de esta familia, a los cuales se van sumando muchos más que permiten al día de hoy implemantar todos los servicios que ofrece esta arquitectura.

1.6.Fuentes de información (RFC).

Como se mencionó anteriormente, la velocidad de avance de Internet, no soporta un burocrático sistema de estandarización como se venía haciendo con otras familias de protocolos, nace así la idea de las RFC (Request For Commentaries). Estas recomendaciones, no buscan estandarizar rigurosamente esta familia, sino que a medida que aparece una nueva funcionalidad, servicio, implementación, protocolo, etc. Inmediatamente se puede describir la mejor forma de llevarla a cabo mediante una RFC, la cual tiene diferentes "Estados de madurez" y rápidamente sienta un precedente. En la actualidad superan holgadamente las tres mil.

1.7. Breve descripción de protocolos que sustentan a TCP/IP (PPP, ISDN, ADSL, Ethernet, X.25, Frame Relay y ATM).

Como se verá más adelante, el protocolo IP puede ser implementado sobre una gran cantidad de protocolos que le permitan transportar (llevar) la totalidad de la información que este encapsula, es por esta razón que se trata aquí de dar una muy breve descripción de los más importantes de ellos.

a. PPP:

El Point to Point Protocol, es la implementación más simple del nivel de enlace para acceder a redes, por medio de líneas de telefonía conmutada, estableciendo como su nombre lo indica un enlace punto a punto con un nodo de acceso a la red, a través de un modem y por medio de los protocolos de la familia HDLC (High Level Data Link Connection), más específicamente LAP-M (Link Access Procedure - Modem).

b. ISDN:

ISDN es una tecnología de acceso en el rango de las telecomunicaciones y particularmente a los servicios de circuito virtual por conmutación de paquetes. ISDN pretende crear un sistema completo que permita abastecer cualquier servicio actual y futuro al usuario
Existen dos tipos de servicios ISDN: Básico o BRI (Basic Rate Interfaz) y PRI (Primary Rate Interfaz).
El BRI ofrece como servicio dos canales de 64 Kbps (Canal B: Bearer) y uno de 16 Kbps (Canal D: Delta) por eso es comúnmente llamado 2B + D, sumando un ancho de banda utilizable de 144 Kbps, si bien se debe tomar en cuenta que existen 48 Kbps empleados para separación de bandas y balanceo, que imponen un ancho de banda total de 192 Kbps siendo estos últimos transparentes y no utilizables para el usuario.
El cliente se encuentra representado por el CPE (Equipamiento del lado del Usuario), accediendo a una central telefónica llamada CO (Central Office), la cual es la encargada de la conmutación para lo cual emplea el sistema de señalización Nro 7 (SS 7) dentro de la red ISDN y el sistema de señalización DSS1 (Digital subscriber signalling) con el usuario por medio del canal D.
El PRI ofrece dos posibilidades, según la norma Europea se constituye con 30 B + D, posibilitando un ancho de banda disponible de 2,048 Mbps y según la norma de EEUU 23 B + D haciendo posible 1,544 Mbps. La unión de varios PRI puede hacerse bajo el esquema de ATM que de hecho constituye la base de B - ISDN (Broadband) o ISDN de banda ancha.
El ISDN mantiene las características de discado, es decir, se paga por su uso y en relación a las líneas dedicadas suele ser más económico hasta un máximo de 2 o 3 horas diarias de uso (que se corresponderían a unos 100 Mb diarios).

c. XDSL:

La DSL usa modernas técnicas de **procesamiento digital** para aprovechar la infraestructura de cobre instalada y crear lazos digitales remotos de alta velocidad en distancias de hasta 5.400 metros sin hacer conversiones de digital a analógico.
En un edificio grande o en un campus universitario, una DSLAM (DSL Access Multiplexer) se conecta a los cables telefónicos de cobre existentes que corren por las subidas del edificio hasta las computadoras de los usuarios.
Las PC del usuario se conectan a un **módem DSL** vía conexiones Ethernet estándar y el DSLAM, usado en lugar de conmutadores telefónicos de voz, transmite por sistema multiplex el tráfico de datos desde las líneas DSL a una interfaz ATM.
Esta transmisión de datos **punto a punto** en forma digital a elevada anchura de banda -hasta 7 Mbps o 8 Mbps- le da a la DSL una significativa ventaja sobre los sistemas ISDN y los módem de 56 Kbps. La transmisión analógica usa sólo una **pequeña porción** de la capacidad del alambre de cobre de transmitir información y por esta razón la velocidad máxima es de 56 Kbps. Aunque el ISDN es un buen sistema para transmisión a 64 Kbps - 2,048 Mbps- su tecnología no puede manejar las demandas de aplicaciones que requieren gran ancho de banda.

DSL crea conexiones más rápidas que ambos con **grandes canales de datos y mayores anchos de banda**. Estos grandes anchos de banda le permiten a la DSL manejar las demandas de aplicaciones que consumen mucho ancho de banda: videoconferencias en tiempo real, telemedicina y educación a distancia, por ejemplo.
Además del mayor ancho de banda, la DSL es en muchos aspectos una tecnología **más barata que la ISDN**.

Variantes de DSL

Las diferentes implementaciones de DSL sirven como canales de alta velocidad para conexiones remotas, pero tienen diferentes velocidades y características de operación.

- **ADSL (Asymmetric Digital Subscriber Line):** siendo esta variante es la más flexible, dado que proporciona numerosas velocidades ascendentes y descendentes, probablemente la ADSL llegará a ser la variante más popular en las pequeñas empresas y los usuarios en el hogar.
  Velocidad máxima ascendente: 2 Mbit/segundo.
  Velocidad máxima descendente: 64 Mbit/segundo.
  Distancia máxima: 5.400 m.

- **HDSL (High bit-rate DSL):** Esta es la más vieja de las variantes de las tecnologías DSL. Se usa para transmisión digital de banda ancha dentro de instalaciones de empresas y compañías telefónicas que requieren dos cables entrelazados y que usan líneas T1.
  Velocidad ascendente máxima: velocidad de T1.
  Velocidad descendente máxima: velocidad de T1.
  Distancia máxima: 3.600 m.

- **(ISDL) ISDN DSL:** Esta variante está más próxima a las velocidades de transferencia de datos de ISDN y puede ser activada en cualquier línea ISDN.
  Velocidad máxima ascendente: 128 kbits/s.
  Velocidad máxima descendente: 128 kbits/s.
  Distancia máxima: 5.400 m.

- **RADSL (Rate-Adaptive DSL):** Esta variante soporta software que automática y dinámicamente ajusta la velocidad a la cual pueden transmitirse las señales en la línea telefónica de determinado cliente.
  Velocidad máxima ascendente: 1 Mbit/s.
  Velocidad máxima descendente: 12 Mbit/s.
  Distancia máxima: 5.400 m.

- **SDSL (Single-Line DSL):** Esta variante es una modificación de HDSL.
  Velocidad máxima ascendente: 768 kbit/s
  Velocidad máxima descendente: 768 kbit/s.
  Distancia máxima: 3.000 m.

- **VDSL :** Es lo último en DSL y es una tecnología todavía en desarrollo.
  Velocidad máxima ascendente: 2,3 Mbit/s.
  Velocidad máxima descendente: 52 Mbit/s.
  Distancia máxima: 1.350 m.

d. **Ethernet:** Se tratará en detalle más adelante.

e. X.25:

En 1974, el CCITT emitió el primer borrador de X.25. Este original sería actualizado cada cuatro años para dar lugar en 1985 al "Libro Rojo" ampliando e incorporando nuevas opciones y servicios, que posteriormente siguieron siendo ajustadas.

El concepto fundamental de X.25 es el de **Red de Conmutación de paquetes,** siendo la precursora de este tipo de tecnologías. Por nacer muy temprano, su desarrollo fue pensado sobre redes de telecomunicaciones de la década del 70, teniendo en cuenta nodos de conmutación electromecánicos o híbridos, líneas exclusivamente de cobre, equipos terminales de datos de muy poca "Inteligencia" y baja velocidad de procesamiento. Basado en estos parámetros es que hace especial hincapié en la detección y control de errores, que como se puede esperar, se logra mediante una enorme redundancia en la transmisión. Para lograr este objetivo es que implementa un esquema de tres niveles asociados directamente a los equivalentes del modelo OSI.

En X.25 se definen los procedimientos que realizan el intercambio de datos entre los dispositivos de usuario y el nodo de ingreso a la red X.25 (no define lo que sucede dentro de la misma).

f. Frame Relay:

Es una de las técnicas de fast packet switching, llamada habitualmente conmutación de tramas. Es empleado fundamentalmente para el reemplazo de líneas punto a punto. Esta técnica opera sobre líneas de alta calidad, pues reduce sensiblemente la importancia que X.25 le da a la detección de errores, dejando esta tarea únicamente a los extremos. Por lo tanto, si las líneas son de baja calidad se deberá transmitir las tramas de extremo a extremo, bajando el rendimiento, incluso hasta ser peores que X.25 si el canal es muy malo.

Las estaciones terminales son responsables de:

- Cobertura de errores.
- Control de secuencia.
- Control de flujo.

Sus saracterísticas son:

- Alta velocidad y baja latencia.
- Basado en circuitos virtuales de nivel 2.
- Se reemplaza el término canal lógico por DLCI (Data Link Connection Identifier).
- Este DLCI identifica al circuito virtual en cualquier punto de la red (Igual que el canal lógico en X.25).
- Cada DLCI tiene significado local.
- La conmutación se produce a nivel trama.
- Orientado al tráfico por ráfagas.
- Comparte puertos.
- Permite el uso dinámico de ancho de banda.

g. ATM:

Primera solución capaz de eliminar la barrera entre LAN y WAN. Aplica el concepto de conmutación rápida de paquetes (llamados celdas).

Es un concepto, no un servicio que emplea nuevas técnicas de conmutación con muy bajo retardo. Se emplea un mínimo de retardo en cada nodo, dejando el control de errores y de flujo en los extremos. Asume que los enlaces son digitales, por lo tanto posee un muy bajo índice de errores. Integra voz, datos y en algunos casos vídeo.

Emplea la transmisión en banda ancha (Broadband).

¿Por qué B – ISDN?

La conmutación de circuitos se adapta perfectamente a los servicios Isocrónicos (Sincrónico pero continuo en su retardo, Ej: Voz).

La conmutación de paquetes se adapta perfectamente a la transferencia de datos.

ATM es una solución de compromiso: Una conmutación de paquetes que puede asegurar una entrega rápida y continua de voz e imágenes.

El ATM Forum se crea porque las normas ITU salen con demoras de cantidad de años, y la dinámica de los avances tecnológicos no puede soportar tanto tiempo. El ATM Forum fue fundado en octubre de 1991, es un consorcio internacional formado para acelerar el uso de los productos y servicios ATM a través de una rápida convergencia y demostración de las especificaciones. No es un instituto de estandarización sino que trabaja en colaboración con instituciones como ITU y ANSI.

- Nace en los laboratorios Bell a fines de los 80'.
- Las Unidad de transferencia de información es llamada celda la cual es de tamaño fijo (53 Byte) y son "relevadas" (Relay) entre cada nodo ATM por eso su concepto de Cell Relay.

  EEUU propone 64 Byte + 5 Byte (Necesita cancelar eco en TE).

  Europa propone 32 Byte + 4 Byte (Era baja la eficiencia de datos por celda).

  Se adopta : 64 + 32 = 96; 96 / 2 = **48 + 5** (Máx valor sin cancelar eco).
- Premisas de ATM: Red altamente confiable y de alta velocidad, nodos inteligentes para tratar errores.
- Reúne conceptos de conmutación de paquetes y de circuitos.
- Se lo denomina asíncrono por la discontinuidad entre celdas a nivel de usuario, pero a nivel físico es estrictamente sincrónico pues lo soporta SDH (Jerarquía Digital Sincrónica).
- Es Orientado a la Conexión, técnica que logra mediante el empleo de circuitos virtuales (VPI y VCI).
- Tecnología capaz de conmutar millones de unidades por segundo a través de los nodos introduciendo retardos muy bajos, para lograrlo:
  - Reduce las funciones en los nodos: Se le quita a estos toda la carga de procesamiento que no sea estrictamente imprescindible para el encaminamiento exitoso de la llamada.

- Delega funciones en los extremos: Confiando en la inteligencia que se posee hoy en los equipos terminales de datos, confía en estos muchas de las tareas.

1.8. Presentación de protocolos TCP, UDP e IP.

Dentro de esta pila de protocolos, como el modelo de referencia lo trata de mostrar, existen dos niveles bien marcados. Hasta el nivel cuatro (transporte) inclusive. "miran" hacia la red, por lo tanto todas las actividades que aquí se desarrollan tienen relación con el canal de comunicaciones y los nodos por los que pasa la información. Dentro de esta división, se encuentra el corazón de esta familia, se trata del protocolo IP de nivel 3 (red) y de los dos protocolos de nivel 4 (transporte) UDP y TCP. Sobre estos tres cae toda la responsabilidad de hacer llegar la información a través de la red, es por esta razón que se los trata de remarcar con esta presentación, y sub-clasificarlos de alguna forma respecto al resto.

1.9. Presentación de protocolos: FTP, Telnet, ARP y R_ARP, SMTP, POP3, IMAP, MIME, SNMP, HTTP, ICMP, IGMP, DNS, NetBIOS, SSL y TLS.

El resto de esta familia "miran" hacia el usuario. Se debe contemplar aquí las dos excepciones que son "ARP, R_ARP e ICMP-IGMP", que en realidad participan también en las tareas de red, pero como un complemento de la misma. Con estas excepciones salvadas, todo lo demás que se verá tiene como función principal cierta interacción con el usuario final para ofrecer servicios y/o funciones.

1.9. Protocolo Ipv6.

Ante varios problemas que fueron apareciendo durante la larga vida del protocolo IP versión 4, desde hace varios años se está estudiando, y en la actualidad ya implementando en laboratorio y troncales de Internet, una nueva versión del mismo llamada IP versión 6 (Ipv6) o IP Next Generation (IPNG), el cual ya ha llegado a un importante nivel de madurez, pero aún no se ha lanzado al mercado.
En virtud de la importancia que este revista, se tratará con más detalle en el punto 3.

2. PRINCIPIOS DE ANALISIS:

2.5. Trafico: Broadcast, multicast y dirigido:

El tráfico que se produce en una red se puede clasificar desde dos puntos de vista: Por su sentido y Por su forma de direccionamiento.

2.1.1. Por su sentido:

La dirección en que las señales fluyen en la línea de transmisión es un factor clave que afecta a todos los sistemas de comunicaciones de datos. Existen tres tipos de flujo de la información:

- Simplex:

  La transmisión entre dos equipos se realiza en un único sentido (por ejemplo la TV).

- Half-Duplex:

La transmisión se realiza en los dos sentidos, aunque no simultáneamente (por ejemplo los walkie talkies).

- Duplex:

Transmisión simultánea e independiente en ambos sentidos (por ejemplo el teléfono).

2.1.2. Por forma de direccionamiento:

- Unicast:

Se trata de una transmisión de un ETD a solo un ETD.

- Multicast:

Se trata de una transmisión de un ETD hacia un determinado grupo.

- Broadcast:

Es el tipo de transmisión de un ETD hacia absolutamente todos los ETD que escuchen la misma.

2.6.Sniffers y analizadores de protocolos:

¿Qué es un analizador de protocolos?

Un analizador de protocolos, captura conversaciones entre dos o más sistemas o dispositivos. No solamente captura el tráfico, sino que también lo analiza, decodifica e interpreta, brindando una representación de su escucha en lenguaje entendible; por medio de la cual se obtiene la información necesaria para el análisis de una red y las estadísticas que el analizador nos proporciona.

Esencialmente, un analizador de protocolos es una herramienta que provee información acerca del flujo de datos sobre una LAN, mostrando exactamente qué es lo que está sucediendo en ella, detectando anomalías, problemas o simplemente tráfico innecesario. Una vez que un problema es aislado, se pueden analizar las causas que lo producen y tomar las medidas para evitarlo.

Un analizador de protocolos debería proporcionar tres tipos de información sobre una LAN:

a. **Estadísticas** sobre tráfico de datos, estado de los dispositivos y líneas de errores en la LAN. Esta información ayuda a identificar tramas y condiciones generales que pueden señalar un problema inesperado o causar un bajo rendimiento en la red. Permite también determinar nuevas necesidades de Hardware para segmentar o crear subredes dentro de la LAN como podría ser el empleo de Switch o router y la ubicación y configuración correcta de los mismos.

b. **Captura de paquetes y decodificación** de los mismos en los distintos protocolos de cada nivel. Debería permitir también el filtrado correspondiente, que posibilite especificar en el mayor grado de detalle lo que se desea estudiar, dejando de lado la información innecesaria. Se suele filtrar por Dirección MAC, IP, Nombre NetBIOS, puertos, tipo de protocolo, secuencias de bit, etc.

c. **Representación de información histórica** en lapsos diarios, semanales, mensuales o en períodos establecidos por el usuario. Esta información provee una perspectiva histórica para cualquier nuevo problema o indica un problema potencial antes que este suceda.

Las estadísticas de estaciones de trabajo o servidores permiten identificar el tráfico generado por cada uno de ellos y el porcentaje de ancho de banda que consumen. Con esta información se puede determinar cuál es la que hace mayor uso del canal físico y cuáles son los recursos más usados. Por ejemplo si una estación genera un alto porcentaje de tráfico, esto puede estar indicando una falla en su tarjeta de red, permitiendo tomar las medidas correspondientes, basadas en observaciones reales de la red, y no por prueba y error.

Un concepto importante es que un analizador de protocolos no emplea SNMP; esta herramienta cuenta con la información específica que le permite identificar las diferentes secuencias de bit, y por medio de los Header establecidos para cada protocolo, los que responden a estos patrones los asocia a uno de ellos, "desarmando" las cadenas binarias en la información contenida en ellas. Por ejemplo SNMP no podría brindar información, sobre sesiones Telnet, TCP/IP, uso de ancho de banda, qué tipos de paquetes se emplean, etc. Un SNMP se debe considerar como un muy buen COMPLEMENTO de un analizador de protocolos.

Por último, en sus inicios, existía una notable diferencia entre los **sniffers** y estos dispositivos, pues los primeros, sólo se dedicaban a capturar el tráfico de la red y representarlo en su forma hexadecimal, sin desempeñar ninguna de las tareas recientemente descriptas que realiza un analizador de protocolos, un ejemplo aún vigente podría ser el "comando nmap" de Linux. En la actualidad, muchos de estos sniffers fueron incorporando más y más funcionalidades, pues una vez que está capturada la información, es muy simple realizar estadísticas, comparaciones, presentaciones gráficas de la misma, etc. Por lo tanto hoy, es muy confusa la denominación que se emplea para estos productos, pero siendo estrictos conceptualmente, un **sniffer** sólo captura tráfico y lo presenta de manera más o menos amigable (y nada más). **Un analizador de protocolos**, realiza esta tarea y a su vez procesa esta información para obtener todas las posibles necesidades de usuario con la misma.

2.7.Detección de sniffers.

Las técnicas de detección de sniffers que se emplean son varias y todas se basan en poder determinar si la interfaz de red se encuentra en modo promiscuo, lo cual es un claro síntoma que desea recibir todo el tráfico que pasa por ella, actividad no necesaria para ningún host que preste servicios en una red:

a. La más simple de estas es enviar un mensaje ARP a una dirección MAC falsa, si responde, es que se encuentra en modo promiscuo. La masa de los sniffers o analizadores de protocolos ya preveen esta técnica y simplemente anulan todo tipo de respuesta a nivel MAC.

b. Test específico del Sistema Operativo: Es muy similar al anterior, pero se envían mensajes ICMP de eco (Tipo 8), con la dirección MAC no existente en la red, se emplean también con mensajes que pueden ser Unicast, Multicast o Broadcast, y basado en el tipo de respuesta se determinará también qué sistema operativo posee el host (Linux: responde ante unicast [este es un bugs que la mayoría de los sistemas linux hoy tienen resuelto, pero existe un error en la pila TCP/IP de este S.O. con el cual responderán siempre a una dirección IP real, aunque la MAC sea falsa], BSD: responde ante multicast, NT: Lo hace analizando solo el primer octeto MAC

contra la dirección IP cuyo primer octeto sea Broadcast, independientemente del resto de la dirección MAC.

c. Test DNS: Esta técnica envía información acerca de un dirección IP y escucha por cualquier solicitud de resolución DNS desde un host hacia el servidor correspondiente.

d. Test de latencia del sistema: Este es el más complejo pero también el más eficiente. Se trata de enviar paquetes ICMP y medir los tiempos de respuesta. Si se incrementa el tráfico en la red, una interfaz en modo promiscuo irá tardando cada vez más tiempo en responder que el resto de las interfaces, pues ésta deberá procesar la totalidad de las tramas, mientras que el resto sólo lo hará con las tramas dirigidas a estas.

2.8. Introducción al Microsoft Network Monitor (Como herramienta de análisis y captura).

Como ya se mencionó con anterioridad, un analizador de protocolos es una herramienta que permite capturar, filtrar y analizar el tráfico de una red. Los datos capturados pueden ser guardados para un análisis posterior o analizados inmediatamente después de la captura. Esta herramienta puede ser una combinación de Hardware y Software, o simplemente Software como es el caso de Network Monitor de Microsoft. Este Software permite lo siguiente:

- Capturar tramas directamente desde la red.
- Mostrar y filtrar las tramas capturadas.
- Editar las tramas y transmitirlas por la red.
- Capturar tramas desde una computadora remota.
- Generar tráfico.

La versión simple del Network Monitor 1.2. está incluido con el Windows NT 4.0 y la versión completa con el Microsoft System Managment Server. Esta última puede ser instalada sobre Windows 3.1, Windows 95 y Windows NT Server y Workstation. En la actualidad ya existe la versión 2.0 que se emplea con la plataforma Windows 2000, y también viene la versión estándar con Windows 2000 y la Enterprise con el SMS 2.0.

Network Monitor consiste en dos componentes, Network Monitor application y Network Monitor Agent.

Network Monitor Application muestra estadísticas y datos de las capturas, permitiendo guardarlos.

Network Monitor Agent permite la captura de tráfico local o remoto. El cliente corriendo Network Monitor Agent captura y almacena en buffer el tráfico localmente; cuando el Network Monitor Application intenta mostrar el tráfico, el Agent se lo envía a través de la red para su estudio.

Es muy importante el concepto de MODO PROMISCUO, cuyo significado es que el adaptador de red permite el ingreso (“escucha”) absolutamente todas las tramas que pasan por el cable. En el manual de compatibilidades de Windows NT existe una lista de los adaptadores de red que operan de esta forma. Se debe tener en cuenta que un adaptador que trabaja en modo promiscuo significa que delega todo el trabajo en la CPU por lo tanto representa una sobrecarga de tareas al ETD que se

le instala, no siendo así en el que opera NO en modo promiscuo, que posee mecanismos de filtrado que liberan de las actividades de nivel 2 al ETD.

La pantalla del Network Monitor está dividida en cuatro partes:

a. Gráfica: Muestra la actividad que se produce en la red por medio de barras en porcentajes de tramas y byte por segundo, como así también Broadcast y Multicast por segundo.

b. Estadísticas de sesión: Muestra el resumen de tráfico entre dos host, y cual de ellos inicia broadcast y multicast.

c. Estadísticas totales: Resumen de la totalidad de la actividad en la red.

d. Estadísticas de estaciones: Muestra el resumen de la totalidad de tramas enviadas y recibidas por cada host.

2.4.1. Captura, filtrado y análisis de tramas.

El análisis de datos comienza con la vista de los datos capturados, esta pantalla muestra la totalidad de las tramas capturadas, las cuales pueden ser filtradas con anterioridad a la captura o luego de ella, para seleccionar las que se desee analizar.

La presentación de esta pantalla se divide en tres partes:

a. Panel resumen: Muestra la totalidad de las tramas presentando en columnas la siguiente información:

1) Trama: Número de trama capturada, en el orden que fue capturada.

2) Tiempo: Permite identificar el tiempo en el que inició la captura de esta trama o puede ser configurado para identificar la hora del día en que fue capturada.

3) Dirección MAC origen: Muestra la dirección de hardware del ETD que emitió la trama.

4) Dirección MAC destino: Muestra la dirección de hardware del ETD que recibió la trama.

5) Protocolo: El protocolo usado para transmitir la trama.

6) Descripción: Resumen del contenido de la trama.

7) Otra dirección origen: Identificador adicional de ETD que origina la trama (Por Ej: IP o IPX).

8) Otra dirección destino: Identificador adicional de ETD que recibe la trama (Por Ej: IP o IPX).

9) Tipo de dirección adicional: Especifica que tipo de dirección adicional se empleó.

b. Panel de detalle: Muestra todo el grado de detalle de la trama seleccionada en el panel anterior, desplegando la totalidad de los protocolos incluidos en esa trama.

c. Panel hexadecimal: Muestra en formato hexadecimal la totalidad de los Byte que fueron capturados en la trama seleccionada en el panel resumen.

2.9. Presentación de otras herramientas (Ethereal, Iris):

Estas dos herramientas se verán brevemente en forma práctica durante el curso, para contar con un espectro mayor de analizadores de protocolos y verificar que el empleo de los mismos es siempre el mismo, variando solamente el aspecto de su presenatción.

2.10. Captura, filtrado y análisis de tramas.

Como ya se mencionó, la gran diferencia entre un sniffer y un analizador de protocolos, pasa por los servicios que este último ofrece, los cuales en general van orientados hacia una mejor interpretación de la información capturada.

Para poder mejorar la visualización de la información , es muy importante "Pulir el bosque", es decir, **poder filtrar** lo que no se desea para clarificar la información que se está buscando. Todos los analizadores de protocolos poseen dos filtros:

- **Filtro de captura:** Permite seleccionar qué es lo que se desea ingresar a la memoria de la herramienta y qué no. Esta funcionalidad es de suma importancia, pues en redes de alto tráfico, es muy fácil que se desborde la memoria del PC donde se ejecuta el analizador de protocolos, y en el caso de desear capturar únicamente una determinada dirección, protocolo, puerto, etc. ¿De qué sirve almacenar el resto del tráfico?. Este filtro, permite registrar (Capturar) sólo lo que se desea, descartando el resto de la información que viaja por el cable.
- **Filtro de visualización:** En este caso, se trata de presentar una "mejor vista" de lo que ya ha sido capturado. Este filtro se emplea, una vez que se detuvo la captura, para poder elegir que se desea visualizar dentro de toda la información que ya se encuentra en memoria.

2.11. Presentación hexadecimal, binaria y decimal.

2.7.1. BIT: estado lógico equivalente a 1 o 0.

2.7.2. Byte u Octeto: Agrupación de 8 bit.

Esta definición es la que realmente se universalizó para el tratamiento de la información y la palabra octeto es hoy una de las bases de la transmisión de información, la razón de ser de esta convención radica en:

- La capacidad suficiente de codificación que posee un octeto, es decir 256 posibilidades diferentes.

  Si se plantea el conjunto de posibilidades este irá desde: 0000 0000, 0000 0001 , 0000 0010 , 0000 0011 , 0000 01000.....1111 1111.
  Ante lo cual permite hasta 256 códigos diferentes.

- El fácil pasaje entre el sistema decimal, hexadecimal y binario.

Suma Decimal	128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 256								
Peso Decimal	128	64	32	16	8	4	2	1	
Binario	b	b	B	b	b	b	b	b	
Peso hexadecimal	8	4	2	1	8	4	2	1	
Suma hexadecimal	8 + 4 + 2 + 1 = F				8 + 4 + 2 + 1 = F				FF
EJEMPLO									
Suma Decimal	128 + 32 + 2 + 1 = 163								
Peso Decimal	128	0	32	0	0	0	2	1	
Binario	1	0	1	0	0	0	1	1	
Peso hexadecimal	8	0	2	0	0	0	2	1	
Suma hexadecimal	8 + 2 = A				2 + 1 = 3				A3

Hexadecimal: Conjunto de16 símbolos (0, 1, 2, 3, 4 , 5 , 6, 7, 8, 9, A, B, C, D, E ,F).

2.7.3. CARACTER: Es la unidad de información a nivel alfabeto humano, representa cualquier símbolo del alfabeto usado como alfabeto normal. Se los clasificará en:

a. **Alfabéticos**: Letras en mayúsculas y minúsculas.

b. **Numéricos**: dígitos de 0 a 9.

c. **Especiales**: Puntuación, Paréntesis, operaciones aritméticas y lógicas, comerciales, etc.

d. **De operación y control**: Destinados al control de la transmisión de Información (Retorno de carro, nulo, SYN, ACK, DLE, EOT, SOH, etc)

2.7.4. Bloque, Mensaje, Paquete, Trama: Son distintas formas de agrupamiento de Byte, y se definen acorde a las distintas técnicas de transmisión de información o Protocolos de Comunicaciones.

3. ANALISIS DE TRAFICO POR PROTOCOLOS.

3.2. Análisis de tramas Ethernet (IEEE 802.3):

El funcionamiento de una red LAN a nivel dos (enlace) que opere por medio de CSMA/CD (Carrier sence multiple access/colition detect) se implementa por medio del protocolo Ethernet u 802.3 (la mínima diferencia entre ellas se verá en breve). Su funcionamiento es básicamente simple, si el canal está libre entonces se puede transmitir, caso contrario no. Como existe la posibilidad que un ETD escuche el canal, al estar este libre comience la transmisión, y antes de

llegar esta señal a cualquiera de los otros ETD de la LAN alguno de estos haga lo mismo, es que se analizan las colisiones. Una colisión se produce cuando dos ETD por tener el canal libre inician su transmisión, la cual no es otra cosa que un estado de tensión que oscila entre + 0,85Volt y - 0,85 Volt (o ausencia de ella) que se propaga por canal físico, al encontrarse dos señales dentro del mismo medio físico se produce una alteración en los niveles de tensión, la cual al llegar a cualquier ETD de la red se determina como una colisión. Los ETD que transmitieron pasan a un algoritmo de espera aleatorio (Llamado disminución exponencial binaria) e intentan transmitir nuevamente al cumplirse el plazo determinado por el algoritmo (son múltiplos de un valor muy especial que se llama tiempo de ranura), si durante 51,2 microsegundos (Tiempo de ranura) no se detecta ninguna colisión, este se ha APROPIADO del canal y se asegura que ningún otro ETD pueda transmitir, por lo cual continuará con el resto de su trama (tamaño máximo 1518 Byte) y luego entrará nuevamente en compulsa por el medio físico.

a. Formato de las direcciones MAC.

Las Direcciones MAC son reguladas por IEEE y están formadas por 6 octetos, representados como pares de números hexadecimales (hh-hh-hh-hh-hh-hh).

Los primeros tres octetos identifican al fabricante de la tarjeta. Estos tres octetos son asignados por un grupo de IEEE llamado **RAC** (Registration Authority Commitee) y pueden ser consultados en www.standars.iee.org, existe una metodología para solicitarlos y por ser 24 bit, se pueden asignar en el orden de 16.000.000 de valores. Estos tres primeros octetos se los denomina "**OUI** (Organizationally Unique Identifier) o "**company_id**", de estos 3 Byte, los dos primeros bit tienen un significado especial:

- bit 0: Individual (valor = 0), establece que este valor pertenece a una sola dirección MAC. Grupal (Valor = 1), forma parte de un conjunto de direcciones MAC.
- Bit 1: Universal (valor = 0), define que esta dirección es única en el mundo. Local (val or = 1) tiene significado solamente en el ámbito local.

Estos primeros 3 octetos, una vez asignados a una determinada empresa, se deja a criterio de la misma cómo asignará los valores de los 3 octetos siguientes denominados "**Extension identifier**", para que no puedan repetirse, pero IEEE-RAC no se responsabiliza ni establece ninguna pauta sobre los mismos. Es lógico pensar que un gran fabricante de tarjetas, complete la totalidad de los posibles números a emitir, IEEE-RAC establece que recién al haber completado el 90 % de las asignaciones podrá solicitar otro OUI para continuar fabricando (en la actualidad ya existen varias empresas en esta situación).

La concatenación de **"OUI + Extension identifier = EUI (Extended Unique Identifier",** conocido como "**EUI-48**", que es la verdadera denominación teórica de una dirección MAC.

Ejemplo de Representación gráfica de una EUI-48					
OUI (company_id)			**Extension Identifier**		
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
AC	DE	48	23	45	67
10101100	11011110	01001000	00100011	01000101	01100111
▲ bit más significativo					▲ bit menos significativo

b. Ethernet y 802.3:

El funcionamiento de este protocolo tiene sus orígenes en otro conocido como ALOHA (saludo de los Hawaianos, que es donde nació), al principio se creyó muy poco probable que está lógica de compulsa por un medio de comunicaciones fuera eficiente, pero en el muy corto plazo se descubrió que sí lo era. Digital, Intel y Xerox, se unen para ponerlo en funcionamiento sobre cable coaxil a 10 Mbps, y como inicialmente se lo empleó en enlaces satelitales que transmitían al "Ether", se lo llamó Ethernet (y se lo conocía como Ethernet DIX). En el año 1980 (80) y en el mes de febrero (2), IEEE toma cartas en el tema y crea el subcomité que estudiaría el tema de LAN y MAN y por la fecha en que entra en funcionamiento se lo llamó 802.x (x=distintas áreas), y será el responsable hasta la actualidad de regular el funcionamiento de estas redes.

Este grupo define todos los aspectos hoy conocidos como familia 802.x, de los cuales solamente en este texto se desea dejar claro algún aspecto de 802.3.

Lo más relevante aquí es que, si se recuerda el aspecto más importante del nivel de enlace (nivel 2) del modelo OSI, este "establece la comunicación con el nodo inmediatamente adyacente". En una topología LAN ¿Cuál es el nodo inmediatamente adyacente?. Ante esta cuestión IEEE, propone subdividir el nivel de enlace del modelo OSI en dos subniveles:

- **MAC** (Medium Acces Control): Responsable de todo lo referente al Hardware de red.
- **LLC** (Logical Link Control), 802.2 : Responsable de la comunicación con los protocolos superiores.

Modelo OSI (Ethernet)	IEE (802.x)
Enlace (nivel 2)	LLC
	MAC

La propuesta es muy coherente, pues facilita esta compleja actividad característica de las LAN. Pero desde ya, que esta propuesta no es reconocida por OSI, marcando una diferencia entre estos dos protocolos. Aparecen aquí estos dos estándares de mercado, que se recalca "NO SON IGUALES", si bien son muy parecidos. En el caso de CSMA/CD, que es el que interesa en este texto, **todo hardware y software de red soporta ambos protocolos y acorde a la trama que se trate aplica uno u otro**.

La diferencia más importante se encuentra en dos octetos del encabezado (que se tratarán a continuación). Cuando se trata de tramas IEEE 802.3, el encabezado MAC tendrá siempre encima de él el subnivel LLC, por esta razón no necesita definir a quién le debe entregar los datos, pues solo existe una opción (LLC); en esta situación los dos octetos referidos establecen la longitud del campo de datos y se llaman "**Length**". Cuando la trama es Ethernet (el nivel de enlace de OSI, no se encuentra subdividido) se debe aclarar a qué protocolo entregará los datos en el nivel 3 (Red), por ejemplo IPX, IP, etc. en este caso estos dos octetos se denominan "**Ethertype**", y se emplean justamente para definir qué tipo de protocolo se encuentra arriba de Ethernet.

La forma de distinguir de qué trama se trata es mediante el valor en hexadecimal de estos dos octetos: todo valor inferior a 0550h se corresponde a una trama IEEE-802.3; por encima de este se trata de una trama Ethernet.

c. Algoritmo de disminución exponencial binaria:

Como se mencionó en la introducción, al producirse una colisión, los ETD responsables de la misma dejan de transmitir (en realidad se envía una señal de atasco para avisar a todos los ETD de la red de este hecho). Automáticamente estos equipos generan un número aleatorio entre 0 y 1. Este número es motivado por el algoritmo de disminución exponencial binaria que propone generar un número aleatorio acorde a la siguiente fórmula:

$$\textbf{Nro Rand} = 2^{n} - 1$$

n = cantidad de colisiones detectadas en esta compulsa.

Al tratarse de la primera colisión: Nro Rand = $2^1 - 1 = 1$ => (Nro Random entre 0 y 1).

Este valor (0 ó 1) establece la cantidad de tiempos de ranura que esperará el ETD para volver a transmitir la trama que ocasionó la colisión, siendo el tiempo de ranura **51,2 µs**.

Si los dos ETD generan el mismo valor, colisionarán nuevamente, pero si obtienen valores diferentes, uno de los dos emitirá primero, y cuando pasen los 51,2 µs del segundo ETD y este desee transmitir, encontrará el canal ocupado y no podrá hacerlo (es decir que el primero ganó la compulsa).

Si hubiesen generado el mismo valor, es decir: los 2 ETD =1 ó los 2 ETD = 0, se producirá la segunda colisión, por lo tanto:

→ Nro Rand = $2^2 - 1 = 3$ => (Nro Random entre 0, 1, 2 ó 3)

Si ambos equipos obtuvieran el mismo valor, colisionarían nuevamente y entonces sería:

→ Nro Rand = $2^3 - 1 = 8$ => (Nro Random entre 0, 1, 2, 3 ,4, 5, 6, 7 u 8).

Si siguieran generando iguales números, esta situación se mantendría hasta:

→ Nro Rand = $2^{10} - 1 = 1023$ => (Nro Random entre 0 y 1023).

Si aún así esto continuara, se ejecutaría el mismo algoritmo con exponente = 10, durante seis veces más, y luego se comienza nuevamente.

Esto que parece muy poco probable, si bien es poco frecuente, no es tan así, pues se debe tener en cuenta que en una red donde existen varios ETD conectados a un mismo **dominio de colisión**, en cualquier momento de esta secuencia, puede entrar en juego otro ETD, caso en el cual, este último comenzaría a tratar el algoritmo como su primera colisión, y los anteriores seguirían con su rutina de disminución de probabilidades, y así también puede ingresar un cuarto, quinto, etc.

El último concepto que aún queda por tratar es el de **tiempo de ranura** (51,2 µs). Este valor nace de la definición misma de Ethernet, y aparece en los inicios de este protocolo, cuando estas redes se implementaban con topología Bus sobre cable coaxil grueso, con el cual se podían unir hasta cinco segmentos de 500m a través de cuatro repetidores regenerativos (y solo 3 de ellos cargados; se la conocía como norma 5-4-3). La distancia máxima que alcanzaba esta red era de 2.500m. Teniendo en cuenta el tiempo de latencia de los repetidores, una señal eléctrica tardaba en recorrer esta distancia ida y vuelta, aproximadamente este tiempo: 51,2 µs. El tema de fondo aquí radica en que si se tiene en cuenta dos ETD separados a esta distancia (el peor de los casos), suponiendo que uno de ellos comienza la transmisión, y un instante antes de llegar al segundo, este escucha el canal y por estar desocupado, comienza a transmitir; entonces se producirá

una colisión muy próxima al segundo ETD. El que inició la transmisión tomará consciencia de la colisión, cuando este estado anormal de tensión regrese a él, es decir, cuando haya recorrido los 2500m de ida y los 2500m de vuelta , que coincide con estos 51,2 µs. Si se supone que el segundo ETD no inició ninguna transmisión, al cabo de estos 51,2 µs ningún ETD de esta red podría transmitir, pues al escuchar el medio, lo encontraría ocupado. Esto se llama **Apropiarse del canal.**

Basado en estos conceptos es que se define que el tamaño mínimo de una trama Ethernet no puede ser menor de 64 Byte, pues **64 Byte = 512 bit** y **512 bit transmitidos a 10.000.000 de bit por segundo (10 Mbps) = 51,2 µs**. También se define que no podrá tener más de 1518 Byte, para evitar que el apropiado del canal sea eterno, evitando así monopolios del medio.

d. Armado de tramas:

Las tramas Ethernet son armadas en el subnivel MAC y responden a 14 octetos de encabezado y a 4 octetos de cola que es donde se realiza el CRC y entre estos campos van los datos.

Se debe tener en cuenta que para que todos los ETD de la red se sincronicen y sepan que se está por recibir una trama, antes de la misma se envían 7 octetos de preámbulo (10101010) y luego un octeto de inicio (10101011). Algunos autores lo consideran parte del encabezado Ethernet y otros no, en este texto no se considerarán parte del mismo.

El formato de una trama Ethernet es el que se detalla a continuación:

8	7	6	5	4	3	2	1
Dirección destino (6 Byte)							
Dirección Origen (6Byte)							
Tipo o longitud (2 Byte)							
Datos (Variable entre 46 y 1500 Byte)							
CRC (4 Byte)							

- Dirección destino: Especifica la dirección del host a alcanzar a nivel MAC.
- Dirección origen: Especifica la propia dirección a nivel MAC.
- Tipo o longitud: Si se trata del protocolo Ethernet el tipo de protocolo de nivel superior (Ethertype). Si es protocolo 802.3 especifica la longitud del campo de datos
- CRC: Control de redundancia cíclica, emplea el concepto de polinomio generador como divisor de la totalidad de la trama, el resto de esta operación

se enmascara con una secuencia determinada de bit y se envía en este campo. Se trata entonces de una división binaria, en la cual se emplea como polinomio generador justamente el CRC-32, que figura abajo, por lo tanto el resto de esta división SIEMPRE será una secuencia de bit de longitud inferior a 32 bits, que será lo que se incluye en este campo. Los formatos estandarizados de estos CRCs son los que se presentan a continuación:

- **CRC-12**: $X^{12} + X^{11} + X^3 + X^2 + X + 1$
- **CRC-16**: $X^{16} + X^{15} + X^2 + 1$
- **CRC CCITT V41**: $X^{16} + X^{12} + X^5 + 1$ (este código se utiliza en el procedimiento *HDLC*)
- **CRC-32 (Ethernet)**: = $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$
- **CRC ARPA**: $X^{24} + X^{23} + X^{17} + X^{16} + X^{15} + X^{13} + X^{11} + X^{10} + X^9 + X^8 + X^5 + X^3 + 1$

- Preámbulo: No está representado en la gráfica anterior, pues no es considerado como parte de la trama pero se trata de 7 byte que indican que comienza una trama y permite sincronizar relojes y 1 Byte de inicio.

e. Relación de Ethernet con Hub y Switch:

Las redes Ethernet, fueron diseñadas con topología "**Bus físico**", y posteriormente se incorpora la idea de interconectar varios ETD a un solo dispositivo, que en definitiva cumplía con las mismas funciones de un conector "T" del tipo BNC, pero en vez de bifurcar la señal por dos caminos (como lo hace una "T"), lo hace por "n" caminos. Esto da origen a los Hub, los cuales literalmente "Explotan" la señal recibida por cualquiera de sus puertos por todos los restantes (menos el que ingresó). La nueva forma que adoptan estas redes es más parecida a una **estrella**, pero como la lógica sigue siendo la misma, es decir, que la señal que emite un ETD sea escuchada por todos los conectados, es que también se la suele llamar "**Bus lógico**". Más adelante estos Hubs incorporan más funciones, de las cuales la más importante es la de regenerara la señal, actuando como un repetidor regenerativo de múltiples puertos.

Teniendo en cuenta lo tratado anteriormente sobre las colisiones, es muy lógico pensar que a medida que aumenta el número de ETD, aumentan las colisiones. Esta es una realidad en estas redes, si bien no depende directamente de la cantidad de host conectados, sino más bien del tipo de tráfico que estos generen. No se trata de una ley matemática ni de algo fácilmente calculable, estos datos se obtienen más bien mediante mediciones de tráfico de red (análisis de tráfico).

Lo que se debe tener en cuenta es que a medida que se necesitan más puestos de trabajo, se pueden agregar más Hubs e interconectarlos entre ellos, esta es una de las grandes ventajas que posee esta red: su flexibilidad. Cada nuevo puesto incorporado generará más y más colisiones, produciendo una baja paulatina en el rendimiento general de la red. Recordar aquí que la única función de un Hub es explotar la señal. Por lo tanto si se poseen n Hubs, y un ETD emite una trama, esta será explotada por todas las bocas de los n Hubs.

Al detectar esta baja performance (mediante análisis de tráfico), la primer medida es segmentar la red en dominios de colisión. Esta actividad la lleva a cabo un dispositivo que trabaja a nivel 2 del modelo OSI, denominado Switch, el cual, no se describirá en este texto, pero básicamente posee tablas dinámicas por cada uno de sus puertos, donde va almacenando las direcciones MAC fuente de cada trama que pasa por él, y a medida que va "aprendiendo" las mismas, comienza a poder conmutar una trama acorde a la puerta en la que la tiene almacenada. Con este principio, si dos ETD se encuentran en la misma puerta, y un Switch recibe una trama Ethernet con MAC origen y destino en ese segmento de red, no lo reenviará por ningún otro puerto; si recibiera una trama con destino en otro segmento de red, únicamente lo conmutaría por el puerto en el que tiene almacenado ese segmento. Como se puede apreciar, si dialogan dos ETD de un mismo segmento, esto no inhabilita a hacerlo a otros dos de otro segmento, cosa que no se podría lograr con Hubs pues estos explotarían la señal por todas sus bocas y el postulado rector de esta metodología es que si se escucha ruido en el canal no se puede transmitir.

El objetivo entonces de un Switch es armar diferentes dominios de colisión, posibilitando que más de un ETD pueda transmitir a la vez.

La gran salvedad que se debe analizar aquí es que si el Switch no posee elementos de Juicio para poder determinar que hacer con una trama, **opera exactamente igual que un Hub**, es decir, explota la señal por todas sus bocas. Esto es de vital importancia si se tiene en cuenta que una red mal diseñada (y en la gran mayoría de los casos, por falta de optimización de tráfico) genera una enorme cantidad de Broadcast a nivel MAC. Si se estudia este detalle, es fácil deducir ¿qué hará el Switch al recibir una trama con dirección destino Broadcast? → Lo explotará por todas sus bocas igual que un Hub. Por lo tanto en el análisis de tráfico es trascendente prestar atención a la generación de Broadcast que se produce en una red. Más adelante se seguirá analizando este tipo de tráfico (y también el Multicast), pues se produce también en otros niveles con igual impacto.

f. Spoof de direcciones MAC:

Cuando se trabaja en entornos LAN, el nivel de enlace toma como identificador de un ETD su dirección MAC, la cual por encontrarse impresa en la tarjeta de red (y en teoría ser única en el mundo), inicialmente debería ser difícil de falsificar. La realidad hace que no sea tan difícil, e incluso el propio SO Linux permite modificarla a través del comando ifconfig. Cuando la información es recibida en una red LAN, el primer identificador de direcciones que aparece es esta dirección, y en base a esta, el ETD decide si la entrega al nivel de red o no. Por ser la puerta de entrada a un host destino, se ha trabajado mucho por distintas opciones de engaño, cualquiera de ellas son lo que se denomino MAC spoofing, lo cual implica falsificar una dirección MAC.

g. Fast y Giga Ethernet:

Las redes día a día van exigiendo un mayor ancho de banda. En la actualidad las necesidades de voz e imágenes hacen que los 10Mbps de Ethernet sean insuficientes. Para dar solución a este problema se comienzan a estudiar nuevas opciones dando origen a Fast Ethernet.

Se plantearon inicialmente dos propuestas:

- Mantener el protocolo CSMA/CD en todos sus aspectos, pero aumentar en un factor 10 la velocidad de la red. Al mantener el tamaño de trama mínimo (64 bytes) se reduce en diez veces el tamaño máximo de la red, lo cual da un diámetro máximo de unos 400 metros. El uso de CSMA/CD supone la ya conocida pérdida de eficiencia debida a las colisiones.
- Aprovechar la revisión para crear un nuevo protocolo MAC sin colisiones más eficiente y con más funcionalidades (más parecido en cierto modo a Token Ring), pero manteniendo la misma estructura de trama de Ethernet.

La primera propuesta tenía la ventaja de acelerar el proceso de estandarización y el desarrollo de productos, mientras que la segunda era técnicamente superior. El subcomité 802.3 decidió finalmente adoptar la primera propuesta, que siguió su camino hasta convertirse en lo que hoy conocemos como **Fast Ethernet**, aprobado en junio de 1995 como el suplemento 802.3u a la norma ya existente.

Los objetivos fundamentales son:

- Mantener el CSMA/CD.
- Soportar los esquemas populares de cableado. (Ej. 10BaseT).
- Asegurar que la tecnología Fast Ethernet no requerirá cambios en los protocolos de las capas superiores, ni en el software que corre en las estaciones de trabajo LAN.

Para acelerar el proceso se utilizó para el nivel físico buena parte de las especificaciones ya desarrolladas por ANSI para FDDI. Los medios físicos soportados por Fast Ethernet son fibra óptica multimodo o monomodo, cable UTP categoría 3 y categoría 5 y cable STP (Shielded Twisted Pair).

Los partidarios de la segunda propuesta, considerando que sus ideas podían tener cierto interés, decidieron crear otro subcomité del IEEE, el **802.12**, que desarrolló la red conocida como **100VG-AnyLAN**. Durante cierto tiempo hubo competencia entre ambas redes por conseguir cota de mercado; hoy en día la balanza se decanta ampliamente hacia Fast Ethernet.

Giga Ethernet se aprueba por IEEE en el año 1998 como estándar **802.3Z** (zeta, por ser la última letra del alfabeto y pensar que será la última de esta familia). También se lo conoce hoy como 1000 Base-X.

Para su implementación sobre pares de cobre, se creó lanorma 802.3ab, que define el funcionamiento de este protocolo sobre cables UTP (Unshielded twisted pair) categorías 5, 5e o 6 y por supuesto para fibra óptica, de esta forma pasó a llamarse 1000 base-T.

En el 2002, IEEE ratificó una nueva evolución de este estándar para operar a 10 Gbps como **802.3ae**, hoy solo funciona sobre fibra óptica, pero ya existe la propuesta para cables de cobre. Mantiene aún la filosofía CSMA/CD (Carrier Sense Múltiple Access / Colision detection).

3.3. Análisis de datagramas (IP):

Se trata de un protocolo de nivel 3 no orientado a la conexión, permitiendo el intercambio de datos sin el establecimiento previo de la llamada. Una característica fundamental es que soporta las operaciones de fragmentación y defragmentación, por medio de las cuales un datagrama se subdivide y segmenta en paquetes más pequeños para ser introducidos a la red, y luego en destino se reconstruyen en su formato original para entregarlos al nivel superior. La otra operación que revista importancia es el ruteo, el cual implementa por medio de un esquema de direccionamiento que se trata a continuación. En la actualidad se emplea la versión 4 de este protocolo, pero está en estudio y muy próxima a implementarse la Versión 6 o Next Generation por razones que se tratarán al final de esta sección, pero en estos párrafos se explicará lo referido a la versión 4.

a. Direcciones IP (rfc 791):

Internet por medio del empleo del campo de direcciones de un datagrama, sólo puede identificar en cada uno de los bit dos elementos:

HOST (H).

NET (N).

Este campo de direcciones, está constituido por cuatro octetos, los cuales se pueden presentar en binario (bbbbbbbb.bbbbbbbb.bbbbbbbb.bbbbbbb), en hexadecimal (hh.hh.hh.hh) o en decimal (d.d.d.d). Es importante habituarse a la correspondencia entre binario y decimal para un ágil manejo de estas direcciones que como ya puede apreciarse, oscilarán entre 0/255.0/255.0/255.0/255 en sistema decimal. Dentro de este espectro en los cuatro octetos, existen varias direcciones RESERVADAS, las dos más comunes son:

- **00000000** (en binario) o 0 (en decimal): que especifica " La red donde me encuentro".
- **11111111** (en binario) o 255 (en decimal): que especifica un mensaje de "Broadcast".

Acorde al primer octeto, se pueden clasificar distintos tipo de redes**:**

0xxxxxxx Tipo A: Como el primer bit es 0, este tipo de redes solo podrán abarcar el rango de direcciones entre 0 y 127.

10xxxxxx Tipo B: Como el primer bit es 1 (ya pesa 128) y el segundo obligatoriamente 0, este tipo de redes solo podrán abarcar el rango de direcciones entre 128 + (0 a 63) a 192.

110xxxxx Tipo C Como los dos primeros bit son 11 (ya pesa 192) y el tercero obligatoriamente 0, este tipo de redes solo podrán abarcar el rango de direcciones entre 192 + (0 a 31) a 223.

1110xxxx Tipo D Como los tres primeros bit son 111 (ya pesa 224) y el cuarto obligatoriamente 0, este tipo de redes solo podrán abarcar el rango de direcciones entre 224 + (0 a 15) a 239. **Este tipo de direcciones están reservadas para empleo de multicast.**

11110xxx Tipo E Como los cuatro primeros bit son 1111 (ya pesa 240) y el quinto obligatoriamente 0, este tipo de redes solo podrán abarcar el rango de direcciones entre 240 + (0 a 7) a 247. **Este tipo de direcciones están reservadas para uso experimental por parte de los organismos de Internet**.

Al diferenciar estos tipo de redes, a su vez por medio de un concepto denominado MASCARA DE RED que se tratará más adelante, en particular las tipo A, B y C determinan ciertos límites entre Host y Net que se detallan a continuación:

Tipo A: (0 a 127), el primer octeto identifica a Net y los otros tres a Host. Por lo tanto existirán 127 posibles redes A y cada una de ellas podrá contener tres octetos de Host lo que equivale a 2^{24} = 16.777.214 Host, **(N.H.H.H).**

Tipo B: (128 a 191) Los dos primeros octetos identifican a Net y los otros dos a Host. Por lo tanto existirán 2^{14} Net = 16.384 posibles redes B y cada una de ellas podrá contener dos octetos de Host lo que equivale a 2^{16} = 65.534 Host, **(N.N.H.H).**

Tipo C: (192 a 223) Los tres primeros octetos identifican a Net y el último a Host. Por lo tanto existirán 2^{21} Net = 2.097.152 posibles redes C y cada una de ellas podrá contener un octeto de Host lo que equivale a 2^{8} = 254 Host, **(N.N.N.H).**

Las cantidades mencionadas numéricamente son las reales si bien pueden no coincidir con algunas potencias pues dentro de los rangos establecidos, también existen determinadas direcciones reservadas.

b. Máscara de Red y Subred:

Dentro de una red IP, se pueden crear distintas subredes, empleando el concepto de "Máscara". Estas subredes "piden prestado" bit de los campos identidad de Host, y por medio del empleo de AND lógico se solapan con el bit correspondiente a esa posición de la Máscara de subred, Si este último es un uno, se corresponderá a un bit que identifica a una dirección de red (N), caso contrario será una dirección de Host.

Ej:

	decimal	**Binario**
Dirección IP	193.66.66.240	11000001.01000010.01000010.11110000
Máscara	255.255.255.0	11111111.11111111.11111111.00000000
Identificación de Host o Net		NNNNNNNN. NNNNNNNN. NNNNNNNN. HHHHHHHH

En este ejemplo se identifica el Host número 240 perteneciente a la red tipo C número 193.66.66. Si se deseara poder crear dos subredes dentro de esta misma red, para segmentar distintos grupos lógicos de nivel 3, se debería cambiar uno de los bit correspondientes al cuarto octeto de la máscara, de manera tal que permitiera solaparse con su correspondiente bit de la dirección IP. Cualquiera de los ocho bit de este último octeto podría ser elegido, y técnicamente se crearían dos subredes, pero el mejor planteo para el diseño "entendible humanamente" es el de comenzar a enmascarar de izquierda a derecha, lo que permitirá identificar por el valor decimal de ese octeto a que subred se refiere (Pues el ser humano piensa en decimal y no en binario). Para aclarar este planteo se propone el siguiente ejemplo en la creación de dos subredes dentro de la misma red anteriormente descripta:

Ej:

	decimal	**Binario**
Dirección IP	193.66.66.240	11000001.01000010.01000010.11110000
Máscara caso 1	255.255.255.128	11111111.11111111.11111111.10000000
Identificación de Host o Net		NNNNNNNN. NNNNNNNN. NNNNNNN . NHHHHHHH

Máscara caso 2	255.255.255.8	11111111.11111111.11111111.00001000
Identificación de Host o Net		NNNNNNNN. NNNNNNNN. NNNNNNNN. HHHHNHHH

Para el **caso 1**, se crearon dos subredes (**Hasta ahora, pues aún falta un detalle más**), identificadas por el primer bit del cuarto octeto cuyo peso es de 128. Bajo este esquema lógico (humano), se identifican dos subredes cuyos límites están dados por el valor numérico (humano) del cuarto octeto, es decir que se plantea la subred 193.66.66.1 a 127 y la subred 193.66.66.128 a 254, por lo tanto todo Host que en su cuarto octeto tenga un valor menor a 128, pertenecerá unívocamente a la primera subred y caso contrario a la segunda.

Ej:

Dirección IP 193.66.66.24 ---------- Subred 1
193.66.66.200 ---------- Subred 2
193.66.66.129 ---------- Subred 2
193.66.66.4 ---------- Subred 1
193.66.66.167 ---------- Subred 2
193.66.66.211 ---------- Subred 2

Para el **caso 2**, también se crearon dos subredes, identificadas por el quinto bit del cuarto octeto cuyo peso es 8, técnicamente podría funcionar (al igual que el caso 1), pero la identificación de esas dos subredes sólo será pensando en binario, lo cual para ninguna PC o router será una limitación, pero si lo es en gran medida para cualquier ser humano.

Ej:

Dirección IP 193.66.66.24 ---------- Subred ?
193.66.66.200 ---------- Subred ?
193.66.66.129 ---------- Subred ?
193.66.66.4 ---------- Subred ?
193.66.66.167 ---------- Subred ?
193.66.66.211 ---------- Subred ?

Por último falta aclarar (**...**) un detalle más. En el **Caso 1** recientemente analizado (Dir IP : 193.66.66.240 – Máscara: 255.255.255.128), como se mencionó, se crearon dos subredes:

subred 193.66.66.1 a 127 (es decir su último octeto comienza con el primer bit = 0)

subred 193.66.66.128 a 254 (es decir su último octeto comienza con el primer bit = 1)

Recordando un concepto ya descripto, en un octeto la dirección de red no pueden ser todos unos (Broadcast), ni todos ceros (Mi red); en este caso el primer bit del último octeto será siempre “Todos unos” o “Todos ceros” pues es el único. Siguiendo este principio, NO SE PUEDEN REALIZAR DOS SUBREDES EMPLEANDO UN SOLO BIT, por lo tanto, si se deseara implementar dos subredes en este ejemplo, la máscara será 255.255.255.192, no pudiendo emplear el rango 00 ni 11 de los consecuentes primeros dos bit del octeto.

Si se desea implementar tres subredes, la máscara será 255.255.255.224, la que también permitiría implementar hasta seis subredes (Pues no se podrían asignar los rangos 000 y 111 de los tres primeros bit del último octeto). Siguiendo este razonamiento cabe esperar que para siete subredes, la máscara debería ser 255.255.255.239 lo que también permitiría hasta catorce, y así sucesivamente.

c. Classless Interdomain Routing (CIDR) (RFC: 1518/1519):

Ante el inesperado crecimiento de Internet, se produce una saturación del rango de direcciones clase B, dejando libres algunas direcciones clase A y C, y presentando la particular característica que muy pocas empresa (o casi ninguna), puede cubrir una clase A, y muchas necesitan más de una clase C. Ante este hecho, se van tomando una serie de medidas por medio de las cuales se ajusta la distribución, se restringe la asignación de direcciones a empresas que lo justifiquen con mucho grado de detalle, se distribuyen direcciones en tres zonas mundiales (RIPE, ARIN, y APNIC), pero esto comienza a provocar cada vez mayores tablas en los router con el consiguiente cuello de botella. Para presentar una solución a este problema (momentánea, pues la definitiva recién aparece con Ipv6), nace CIDR o también llamado "Supernetting". Este concepto permite combinar subredes que comparten más de una clase C o subdividir redes clase A o B. El concepto de Supernetting se atribuya a la diferencia con Subnetting. En este último, para crear subredes, se emplea mayor cantidad de bit en la máscara de red. En el caso de Supernetting, es justamente lo contrario (y de ahí su nombre), pues se emplearán menos bit de máscara de la que correspondería a su clase.

Estas direcciones de red deben compartir los bit de más alto orden de la máscara de red, sin respetar el concepto clásico de "clase". A continuación se presenta un ejemplo:

```
NET        192.168.5    (1100 0000.1010 1000.0000 0101.0000 0000)
NET        192.168.6    (1100 0000.1010 1000.0000 0110.0000 0000)
NET        192.168.7    (1100 0000.1010 1000.0000 0111.0000 0000)
MASK  255.255.252.0     (1111 1111.1111 1111.1111 1100.0000 0000)
```

En este ejemplo se aprecian tres rangos de direcciones de red que clásicamente se definirían como clase C, el empleo de CIDR, se pone de manifiesto a través de la máscara de red, la cual reduce dos bit, colocando en su tercer octeto el valor 252 en vez del clásico que debería ser 255. Si se analiza la combinación de bit de dirección con los de host, se trataría aquí de la red 192.168.4, y el broadcast de red debería ser 192.168.127.255 `(1100 0000.1010 1000.0000 0111.1111 1111)`. Las siguiente RFC hacen referencia a CIDR:

RFC 1467 - Difusión de CIDR en Internet

RFC 1517 - Condiciones de aplicabilidad de CIDR

RFC 1518 - Una arquitectura para la distribución de direcciones IP con CIDR

RFC 1519 - CIDR: asignación de direcciones y estrategia de agregación

RFC 1520 - Intercambiando información de encaminamiento a través de las fronteras de los proveedores en el entorno CIDR

d. Tablas de ruta:

Cada datagrama tiene sólo tres posibilidades:

- Ser pasado al nivel superior.
- Encaminarlo hacia alguna de las interfaces de red.
- Ser descartado.

Las tablas de rutas mantienen cuatro tipos de las mismas:

- host (Se trata de una ruta a una simple y específica dirección IP).
- Subnet (Ruta hacia una subred).
- Network (Ruta hacia toda una red).
- Default (Cuando ninguna de las anteriores coincide).

Ejemplo:

```
C:\>route print

Network Address  Netmask          Gateway Address  Interface       Metric
0.0.0.0          0.0.0.0           192.168.40.1    192.168.40.123  1
127.0.0.0        255.0.0.0         127.0.0.1       127.0.0.1       1
192.168.40.0     255.255.255.0     192.168.40.123  192.168.40.123  1
192.168.40.123   255.255.255.255   127.0.0.1       127.0.0.1       1
192.168.40.255   255.255.255.255   192.168.40.123  192.168.40.123  1
224.0.0.0        224.0.0.0         192.168.40.123  192.168.40.123  1
255.255.255.255  255.255.255.255   192.168.40.123  192.168.40.123  1
```

En este ejemplo de tabla de ruteo, se aprecia una dirección privada clase C con un host cuya dirección es 192.168.40.123 y contiene siete entradas

- La primera entrada (0.0.0.0) contiene la ruta por defecto.
- La segunda (127.0.0.0) es la dirección de loopback.
- La tercera (192.168.40.0) es la dirección de esta red.
- La cuarta (192.168.40.123) es la ruta para el local host, se debe prestar atención, que esta hace referencia luego al loopback, es decir que todo datagrama hacia esta dirección deberá ser tratado internamente.
- La quinta (192.168.40.255) especifica el broadcast de subred.
- La sexta especifica la dirección multicast.
- La última indica la dirección de broadcast.

Esta tabla de rutas en la mayoría de los casos es mantenida automáticamente, y muchas de estas direcciones las obtendrá al iniciar el host, de datos locales y a través de información obtenida desde servidores de la red, también pueden ser adicionadas en forma manual, y especificar que sean permanentes o transitorias.

Detección de direcciones IP duplicadas:

Al iniciarse un host, envía una solicitud ARP particular, pues lo hace solicitando la dirección MAC que se corresponde a su propia dirección IP, si alguien contesta este mensaje es que esta dirección IP ya está en uso.

Multihomed:

Este término se refiere al caso que un host se encuentre configurado con más de una dirección IP, esta configuración se puede presentar de tres maneras:

- Múltiples IP por NIC.
- Múltiples NIC.
- Múltiples IP y NIC.

e. IP Multicasting:

El IP Multicasting es la transmisión de un datagrama IP a un cierto grupo de cero a "n" host identificados por una única dirección IP. Los miembros de este grupo son dinámicos, es decir que pueden unirse y dejar grupos en cualquier momento, sin existir ninguna restricción de ubicación o cantidad de miembros. Un host puede ser miembro de más de un grupo y no necesita ser miembro de un grupo para enviar datagramas a él.

Existen grupos definidos como permanentes, que son los que tienen asignada una dirección IP "Bien conocida", esta permanencia se refiere a su dirección, no a sus miembros, los cuales pueden incorporarse y dejarlo dinámicamente en cualquier momento.

El empleo de IP multicast está controlado por los router que pueden o no coexistir con los Gateway

En el mapa de rutas de cada host, para soportar multicast, se agrega una línea adicional:

```
Network Address    Netmask    Gateway Address    Interface      Metric
  224.0.0.0        224.0.0.0    192.35.129.1    192.35.129.1     1
```

En este ejemplo, la dirección 192.35.129.1 es la propia dirección del host (no la del gateway de la red), es decir que si este host desea enviar cualquier mensaje multicast, lo hará a través de su interfaz de red y no lo encaminará al gateway por default.

Un tema de especial interés es que para poder enviar un datagrama, siempre debe ser resuelta la dirección MAC, este tema lo trata la RFC 1112 y se implementa de la siguiente manera: Ethernet identifica multicast colocando a uno el primer bit del primer octeto de su dirección MAC, es decir que este primer octeto adoptará el valor 01h, además IANA reserva (en acuerdo con IEEE que es quien asigna las direcciones MAC) el rango 00-00-5E-00-00-00 hasta 00-00-5E-7F-FF-FF, por lo tanto, los tres primeros octetos de la dirección MAC para multicast IP en redes Ethernet será siempre 01-00-5E. El grupo de direcciones IP es mapeado a una dirección multicast Ethernet, por reemplazo de los 23 bit de menor orden de la dirección IP a los 23 bit de menor orden de la dirección MAC, por ejemplo si un host con dirección MAC 01-22-A2-34-67-E1 deseara enviar un datagrama a la dirección multicast 224.0.0.3, la dirección MAC se formaría con los tres primeros octetos MAC multicast (01-00-5E), anexándole los últimos 23 bit de la dirección IP multicast (0.0.3 = 00-00-03 en hexadecimal), quedaría 01-00-5E-00-00-03. Un problema cierto que se plantea es que la dirección IP posee cuatro octetos, es decir 32 bit, en el caso de multicast, los cuatro primeros bit están impuestos y siempre son 1110, por lo tanto quedan disponibles veintiocho bit, de los cuales solo se mapean 23, es decir que se puede presentar el caso que dos direcciones multicast IP sean mapeadas a la misma dirección MAC, en este caso, será descartado el datagrama a nivel IP, al descubrir en destino que no va dirigido al grupo correspondiente.

Por defecto los datagramas IP Multicast son generados con TTL=1, y por convención, los router multicast emplean los valores de TTL para determinar cuán lejano deben encaminar los datagramas, estos valores de TTL están definidos y son los siguientes:

TTL = 0 se refieren al mismo host.
TTL = 1 se refieren a la misma subred. Los routers multicast al recibir un datagrama multicast con TTL=1, lo decrementan a cero, pero a diferencia de un datagrama

unicast, no envía ningún mensaje ICMP de TTL agotado, pues lo considera un evento normal.

TTL = 32 se refieren a la misma Site.
TTL = 64 se refieren a la misma región.
TTL = 128 se refieren al mismo continente.
TTL = 255 sin restricciones de ámbito.

La dirección multicast también ofrece información sobre las rutas, pues por ejemplo el rango comprendido entre 224.0.0.0 a 224.0.0.255 se emplea para un solo salto, es decir que ningún router multicast retransmitirá ningún datagrama con estas direcciones.
A continuación se presentan las direcciones multicast definidas por la RFC 1112:

224.0.0.0 Reserved
224.0.0.1 All Hosts on this Subnet
224.0.0.2 All Gateways on this Subnet (proposed)
224.0.0.3 Unassigned
224.0.0.4 DVMRP Routers(Distance Vector Multicast Routing Protocol, RFC 1075)
224.0.0.5 OSPFIGP OSPFIGP All Routers
224.0.0.6 OSPFIGP OSPFIGP Designated Routers
244.0.0.7-244.0.0.255 Unassigned
224.0.1.0 VMTP Managers Group (Versatile Message Transaction Protocol, RFC 1045).
224.0.1.1 NTP Network Time Protocol
224.0.1.2 SGI-Dogfight
224.0.1.3 Rwhod
224.0.1.4 VNP
244.0.1.5-244.0.1.255 Unassigned
224.0.2.1 "rwho" Group (BSD) (unofficial)
232.x.x.x VMTP transient groups

f. Fragmentación IP:

Como se verá en el Header de IP, uno de los puntos fuertes de este protocolo es la fragmentación, si bien en la actualidad no es muy empleado porque TCP puede determinar los tamaños de la información a transmitir por el canal para que justamente no sea necesario tener que fragmentar y defragmentar información en nodos intermedios. Igualmente este aspecto se expresa particularmente en este párrafo para marcar la importancia de esta tarea pues es de las técnicas más empleadas para evadir sistemas de seguridad en las redes TCP/IP.

g. Formato del datagrama IP:

<table>
<tr><td colspan="3">Versión</td><td colspan="3">Longitud de cabecera</td></tr>
<tr><td>precedencia</td><td>D</td><td>T</td><td>R</td><td colspan="2">Reservado</td></tr>
<tr><td colspan="6">Longitud total</td></tr>
<tr><td colspan="6">Identificador</td></tr>
<tr><td>Identificadores</td><td colspan="5">Desplazamiento de fragmentación</td></tr>
</table>

Tiempo de vida (TTL)
Protocolo
Checksum de cabecera
Dirección Fuente
Dirección Destino
Opciones y Relleno (Variable)
Datos (Variable)

Versión: 4 bits

Siempre vale lo mismo (0100). Este campo describe el formato de la cabecera utilizada. En la tabla se describe la versión 4.

Tamaño Cabecera: 4 bits

Longitud de la cabecera, en palabras de 32 bits. Su valor mínimo es de 5 para una cabecera correcta, y el máximo de 15.

Tipo de Servicio: 8 bits

Indica una serie de parámetros sobre la calidad de servicio deseada durante el tránsito por una red. Algunas redes ofrecen prioridades de servicios, considerando determinado tipo de paquetes "más importantes" que otros (en particular estas redes solo admiten los paquetes con prioridad alta en momentos de sobrecarga). Estos 8 bits se agrupan de la siguiente manera. Los 5 bits de menos peso son independientes e indican características del servicio:

Bit 0: sin uso, debe permanecer en 0.
Bit 1: 1 costo mínimo, 0 costo normal.
Bit 2: 1 máxima fiabilidad, 0 fiabilidad normal.
Bit 3: 1 maximo rendimiento, 0 rendimiento normal.
Bit 4: 1 mínimo retardo, 0 retardo normal.

Los 3 bits restantes están relacionados con la precedencia de los mensajes, un indicador ajunto que indica el nivel de urgencia basado en el sistema militar de precedencia (véase Message Precedence) de la CCEB, un organización de comunicaciones electrónicas militares formada por 5 naciones. La urgencia que estos estados representan aumenta a medida que el número formado por estos 3 bits lo hace, y responden a los siguientes nombres.

000: De rutina.

001: Prioritario.
010: Inmediato.
011: Relámpago.
100: Invalidación relámpago.
101: Procesando llamada crítica y de emergencia.
110: Control de trabajo de Internet.
111: Control de red.

Longitud Total: 16 bits

Es el tamaño total, en octetos, del datagrama, incluyendo el tamaño de la cabecera y el de los datos. El tamaño máximo de los datagramas usados normalmente es de 576 octetos (64 de cabeceras y 512 de datos). Una máquina no debería envíar datagramas mayores a no ser que tenga la certeza de que van a ser aceptados por la máquina destino.

En caso de fragmentación este campo contendrá el tamaño del fragmento, no el del datagrama original.

Identificador: 16 bits

Identificador único del datagrama. Se utilizará, en caso de que el datagrama deba ser fragmentado, para poder distinguir los fragmentos de un datagrama de los de otro. El originador del datagrama debe asegurar un valor único para la pareja origen-destino y el tipo de protocolo durante el tiempo que el datagrama pueda estar activo en la red.

Indicadores: 3 bits

Actualmente utilizado sólo para especificar valores relativos a la fragmentación de paquetes:

bit 0: Reservado; debe ser 0
bit 1: 0 = Divisible, 1 = No Divisible
bit 2: 0 = Último Fragmento, 1 = Fragmento Intermedio (le siguen más fragmentos)

La indicación de que un paquete es indivisible debe ser tenida en cuenta bajo cualquier circunstancia. Si el paquete necesitara ser fragmentado, no se enviará.

Posición de Fragmento: 13 bits

En paquetes fragmentados indica la posición, en unidades de 64 bits, que ocupa el paquete actual dentro del datagrama original. El primer paquete de una serie de fragmentos contendrá en este campo el valor 0.

Tiempo de Vida (TTL): 8 bits

Indica el máximo número de direccionadores que un paquete puede atravesar. Cada vez que algún nodo procesa este paquete disminuye su valor en, como mínimo, un direccionador. Cuando llegue a ser 0, el paquete no será reenviado.

Protocolo: 8 bits

Indica el protocolo de siguiente nivel utilizado en la parte de datos del datagrama. Vea Números de protocolo IP para comprender como interpretar este campo.

Suma de Control de Cabecera: 16 bits

Suma de Contol de cabecera. Se recalcula cada vez que algún nodo cambia alguno de sus campos (por ejemplo, el Tiempo de Vida). El método de cálculo (intencionadamente simple) consiste en sumar el complemento a 1 de cada palabra de 16 bits de la cabecera y hacer el complemento a 1 del valor resultante.

Dirección IP de origen: 32 bits

Dirección IP de destino: 32 bits

Opciones: Variable

Aunque no es obligatoria la utilización de este campo, cualquier nodo debe ser capaz de interpretarlo. Puede contener un número indeterminado de opciones, que tendrán dos posibles formatos:

Formato de opciones simple

Se determina con un sólo octeto indicando el Tipo de opción, el cual está dividido en 3 campos.

* Indicador de copia: 1 bit. En caso de fragmentación, la opción se copiará o no a cada nuevo fragmento según el valor de este campo:

0 = no se copia
1 = se copia.

* Clase de opción: 2 bits. Las posibles clases son:

0 = control
1 = reservada
2 = depuración y mediciones
3 = reservada.

* Número de opción: 5 bits. Identificador de la opción.

Formato de opciones compuesto

Un octeto para el Tipo de opción, otro para el Tamaño de opción, y uno o más octetos conformando los Datos de opción.

El Tamaño de opción incluye el octeto de Tipo de opción, el de Tamaño de opción y la suma de los octetos de datos.

La siguiente tabla muestra las opciones actualmente definidas:

Clase	Número	Tamaño	Descripción
0	0	-	Final de lista de opciones. Formato simple.
0	1	-	Ninguna operación (NOP). Formato simple.
0	2	11	Seguridad.
0	3	variable	Enrutado desde el Origen, abierto (Loose Source Routing).
0	9	variable	Enrutado desde el Origen, estricto (Strict Source Routing).
0	7	variable	Registro de Ruta (Record Route).
0	8	4	Identificador de flujo (Stream ID).
2	4	variable	Marca de tiempo (Internet Timestamping).

Final de Lista de Opciones:

Se usa al final de la lista de opciones, si ésta no coincide con el final de la cabecera IP.

Ninguna Operación (NOP):

Se puede usar para forzar la alineación de las opciones en palabras de 32 bits.

Seguridad:

Especifica niveles de seguridad que van desde "No Clasificado" hasta "Máximo Secreto", definidos por la Agencia de Seguridad de la Defensa (de EE.UU.).

Enrutado desde el Origen (abierto) y Registro de Ruta (LSSR):

Esta opción provee el mecanismo para que el originador de un datagrama pueda indicar el itinerario que ha de seguir a través de la red y para registrar el camino seguido.

Los Datos de Opción consisten en un puntero (un octeto) y una lista de direcciones IP (4 octetos cada una) que se han de alcanzar ("procesar"):

El puntero indica la posición de la siguiente dirección de la ruta, dentro de la Opción; así, su valor mínimo es de 4.

Cuando un nodo de Internet procesa la dirección de la lista apuntada por el puntero (es decir, se alcanza esa dirección) incrementa el puntero en 4, y redirige el paquete a la siguiente dirección. Si el puntero llega a ser mayor que el Tamaño de Opción significa que la información de ruta se ha procesado y registrado completamente y se redirigirá el paquete a su dirección de destino.

Si se alcanza la dirección de destino antes de haber procesado la lista de direcciones completa (el puntero es menor que el Tamaño de Opción) la siguiente dirección de la lista reemplaza a la dirección de destino del paquete y es a su vez reeemplazada por la dirección del nodo que está procesando el datagrama ("Ruta Registrada"), incrementando, además, el puntero en 4.

Utilizando este método de sustituir la dirección especificada en origen por la Ruta Registrada se asegura que el tamaño de la Opción (y de la cabecera IP) no varía durante su recorrido por la red.

Se considera que la ruta especificada por el originador es "abierta" porque cualquier nodo que procesa el paquete es libre de dirigirlo a la siguiente dirección siguiendo cualquier otra ruta intermedia.

Sólo puede usarse una vez en un datagrama, y, en caso de fragmentación, la opción se copiará a los paquetes resultantes.

Enrutado desde el Origen (estricto) y Registro de Ruta (SSRR):

Exactamente igual que LSSR, excepto en el tratamiento que los nodos harán de este datagrama. Al ser la ruta especificada "estricta", un nodo debe reenviar el paquete directamente a la siguiente dirección, es decir, no podrá redireccionarlo por otra red.

Registro de Ruta:

Mediante el uso de esta Opción se puede registrar el itinerario de un datagrama. Los Datos de Opción consisten en un puntero (un octeto) y un espacio relleno de ceros que contendrá la Ruta Registrada para el paquete.

Cuando un nodo recibe un paquete en el que está presente esta opción, escribirá su dirección IP en la posición indicada por el puntero, siempre que ésta sea menor que el Tamaño de Opción, e incrementará el puntero en 4.

Es preciso que el espacio reservado para la Ruta Registrada tenga una longitud múltiplo de 4; si al intentar grabar su dirección un nodo detecta que existe espacio libre pero es menor de 4 octetos, el paquete no se reenvía (se pierde) y se notifica el error, mediante ICMP, al originador del datagrama.

Esta Opción no se copia en caso de fragmentación, y sólo puede aparecer una vez en un paquete.

Relleno: Variable

Utilizado para asegurar que el tamaño, en bits, de la cabecera es un múltiplo de 32. El valor usado es el 0.

h. IP Spoof:

Esta técnica se emplea para falsificar un verdadera dirección IP a través de la colocación de una falsa, la cual puede hacer uso de una existente en la red o nueva. Presenta especial peligrosidad cuando los dispositivos de seguridad emplean el campo dirección IP para la implementación de las medidas de sus medidas, como pueden ser: control de accesos, permisos, ámbito interno o externo, validación o generación de alarmas, establecimiento de sesiones, reglas, etc.

3.4. TCP (Transport Control Protocol) (RFC 793 , 812, 813, 879, 896 y 1122).

Se trata del protocolo responsable de establecer y gestionar sesiones (conexiones lógicas) entre usuarios locales o remotos. Es también quien se encarga de la fiabilidad, control de flujo, secuenciamiento, aperturas y cierres. Es un protocolo orientado a la conexión, por lo tanto es el responsable de la transmisión de extremo a extremo. Emplea ACK, temporizadores, N(s) y N(r) con segmentos de longitud variable.

Una característica de su empleo es que el control de flujo lo realiza el receptor, el cual envía un valor de ventana (al emisor). El Transmisor puede enviar un número máximo de ventanas no mayor a ese valor, y al llegar al mismo interrumpe la transmisión hasta recibir los ACK correspondientes, que liberen posiciones en su cola circular de envío (N(s)).

El TCP permite multiplexar varias sesiones en una misma computadora por medio del concepto de socket (explicado en terminología).

a. Establecimiento y cierre de conexiones.

Al establecerse una sesión TCP, se produce un triple Handshake, el cual se establece por medio de los bit S y A (Característica de un protocolo orientado a la conexión), envió del primer segmento solicitando establecer una sesión y colocando su número de ventana en un cero relativo (Número generado pseudoaleatoriamente que establece el envío de la primer ventana) (bit A = 1), respuesta aceptando y enviando también su número de secuencia (bit S y A = 1) y por último establecimiento de la sesión TCP (bit A = 1); se inicia el cálculo del RTT (Round Trip Time), tiempo que le permite a TCP establecer el control de flujo calculando el promedio que tardan los segmentos enviados con sus correspondientes respuestas.

El cierre de una sesión TCP se produce al enviar el bit F = 1 ante lo cual se responderá con el bit A = 1 quedando finalizada la sesión

La ventana de recepción de TCP en redes Ethernet normalmente se configura a 8769 bytes, que equivale a seis segmentos de 1460 bytes, que sumados a los 20 byte del encabezado TCP, 20 de IP y 18 de Ethernet, hacen 1518 byte que es el tamaño máximo de la trama Ethernet. El tamaño de esta ventana se ajusta automáticamente acorde a una mecanismo de tiempos de recepción y está regulado por la RFC 1323.

b. Control de flujo

La RFC 1122 define los mecanismos de control de flujo de TCP, basados en los acknowledgments (ACK) de recepción. Este mecanismo permite al receptor de segmentos, regular el tamaño de ventana del emisor, para impedirle el envío de volúmenes de información que no esté en capacidad de procesar.

c. PMTU (Path Maximun Unit Discovery)

Este mecanismo está descripto por la RFC 1191 y permite determinar el MSS (Maximun Segmenet Size) de una determinada conexión.

El concepto está basado en el empleo del protocolo ICMP, por medio del cual, cuando se establece una conexión a través de una red no local, el bit Don't Fragment (DF) del encabezado IP es configurado a uno, es decir que no permite la fragmentación de ese datagrama. Si a lo largo de su trayectoria, este datagrama se encuentra con una red que

no soporta este tamaño (como sería el caso, por ejemplo, de un datagrama generado en una red Token ring, de 4000 Byte, que debe pasar por otra Ethernet), el router que lo recibe, al no poder fragmentar, descartará este datagrama, generando un mensaje ICMP de destino no alcanzable por fragmentación requerida y no permitida a la dirección origen del datagrama descartado, en el encabezado ICMP generalmente incluirá también el tamaño máximo permitido en esa red (dentro de un campo "no empleado" de 16 bit). El que originó el datagrama inicial, al recibir el mensaje ICMP, ajustará la MTU al tamaño indicado.

d. Retransmisión

TCP con cada segmento saliente inicia un timer de retransmisión (por defecto es 3 segundos al establecer la conexión, y se ajusta dinámicamente, RFC: 793), si ningún ACK es recibido al expirar este tiempo, entonces el segmento es reenviado, hasta llegar al *TcpMaxDataRetransmision*, que por defecto es cinco, luego de lo cual no vuelve a retransmitir ese segmento.

En el siguiente ejemplo se puede ver una secuencia de segmentos TCP, en los cuales luego del primero de ellos, se desconectó el host destino, por lo tanto no recibía el ACK de respuesta, en la primera columna el incremento fue duplicando su tiempo de envío entre cada segmento, y al llegar al último si no recibiera respuesta, abortaría la transmisión.

```
 delta   source ip     dest ip     prot  flags                    description
0.000   10.57.10.32   10.57.9.138   TCP ...A.., len: 1460, seq: 8043781, ack: 8153124, win: 8760
0.521   10.57.10.32   10.57.9.138   TCP ...A.., len: 1460, seq: 8043781, ack: 8153124, win: 8760
1.001   10.57.10.32   10.57.9.138   TCP .. A.., len: 1460, seq: 8043781, ack: 8153124, win: 8760
2.003   10.57.10.32   10.57.9.138   TCP ...A.., len: 1460, seq: 8043781, ack: 8153124, win: 8760
4.007   10.57.10.32   10.57.9.138   TCP ...A.., len: 1460, seq: 8043781, ack: 8153124, win: 8760
8.130   10.57.10.32   10.57.9.138   TCP ...A.., len: 1460, seq: 8043781, ack: 8153124, win: 8760
```

e. Velocidad de transferencia

TCP fue designado para optimizar su rendimiento sobre condiciones de enlace variables, dependiendo de varios factores:

- Velocidad del vínculo.
- Demora del vínculo.
- Tamaño de ventana.
- Congestión en los routers.

La capacidad de un vínculo está dada por lo que se conoce como el "producto ancho de banda/demora (Δf * RTT [Round Trip Time]). Si el enlace es de buena calidad, el tamaño de la ventana debería ser mayor o igual que la capacidad del mismo (65535 es el máximo), por el contrario, si el enlace posee mucho ruido o congestiones, el empleo de ventanas grandes no es conveniente, pues se aumentará la congestión o se reenviarán muchos paquetes.

f. Formato del segmento TCP.

<table>
<tr><td colspan="8">Puerto fuente</td></tr>
<tr><td colspan="8">Puerto destino</td></tr>
<tr><td colspan="8">Número de secuencia N(s)</td></tr>
<tr><td colspan="8">Número de aceptación N(r)</td></tr>
<tr><td colspan="3">Desplazamiento de datos</td><td colspan="5">Reservado</td></tr>
<tr><td colspan="2">Reservado</td><td>URG</td><td>ACK</td><td>PSH</td><td>RST</td><td>SYN</td><td>FIN</td></tr>
<tr><td colspan="8">Ventana</td></tr>
<tr><td colspan="8">Checksum</td></tr>
<tr><td colspan="8">Puntero de urgente</td></tr>
<tr><td colspan="8">Opciones y relleno (Variable)</td></tr>
<tr><td colspan="8">Datos (Variable)</td></tr>
</table>

- Puerto fuente y destino: (16), especifican los procesos de nivel superior que utilizan la conexión TCP.
- Número de secuencia y de aceptación: (32), indican la secuencia o posición de los octetos de datos dentro del módulo completo de transmisión. Concatenado a esto, dentro de este campo también va el número de ventana deslizante.
- Desplazamiento de datos: (4), cantidad de palabras de 32 bit que contiene la cabecera.
- Reservado: (6), no se permite su empleo, quedan reservados para uso futuro.
- URG: (1), indica si es un segmento urgente.
- ACK: (1), acknowledgement.
- PSH: (1), Entregar los datos al recibir este segmento.
- RST: (1), reset.
- SYN: (1), sincronismo.
- FIN: (1), último segmento.
- Ventana: (16), cantidad máxima de segmentos que puede enviar el transmisor.
- Checksum: (16), CRC de cabecera y datos.
- Puntero de urgente: (16), si el bit URG está puesto a 1, identifica la posición del primer octeto dónde los datos son urgentes. TCP no dice que hay que hacer con los datos urgentes, sólo los marca.
- Opciones: Son de la forma (Tipo-longitud-valor). Hoy sólo existen definidas 3. O = Fin de lista de opciones. 1 = No operación. 2 = Tamaño máximo de segmento (MSS).
- Relleno: completa a múltiplo de 32 bit de la cabecera.
- Datos: UDP de nivel superior.

Un detalle interesante de TCP es la inclusión de un "Pseudo encabezado", el cual se emplea al realizar el checksum (CRC). La mecánica del mismo es tomar datos del nivel de red (IP), en particular la dirección origen y destino, formar con estos otro encabezado adicional para realizar el cálculo teniendo en cuenta estos datos dentro del checksum de TCP. Estos datos adicionales, no son tenidos en cuenta en la longitud de TCP, ni tampoco se transmiten, pues luego estarán incluidos en el encabezado IP. El objetivo principal de este Pesudo encabezado es proporcionar a TCP la garantía de entrega en el destino correcto de su información, pues al llegar a destino el segmento TCP, nuevamente el host receptor tomará estos campos del nivel inferior y calculará su CRC, si el destino fuera incorrecto, este segmento sería descartado. Esta implementación rompe un poco el concepto de independencia de niveles, pero también sucede con otros protocolos y no es lo habitual.

Los protocolos más comunes que emplean TCP son los que se detallan a continuación:

Puerto	Protocolo
7	Eco
13	Fecha
20 y 21	FTP (File Transfer Protocol)
23	Telnet
25	SMTP (Single Mail Transfer Protocol)
80	HTTP (Hiper Text Transfer Protocol
137,138 y 139	NetBIOS

3.5. UDP (User datagram Protocol) (RFC 768).

Dentro de la pila de protocolos TCP/IP, existen dos opciones de protocolos de transporte. El TCP que se acaba de tratar y el UDP (que lamentablemente se abrevia igual que unidad de datos de protocolo, pero no se trata de esta sino de un protocolo de capa 4) el cual es un Protocolo NO ORIENTADO A LA CONEXIÓN, y se lo emplea con la masa de los protocolos de nivel superior cuando se opera sobre redes LAN. Está regulado por la **RFC 768**, y confía en la baja tasa de errores de una LAN y en los protocolos de nivel de aplicación, siendo por lo tanto un protocolo no confiable pero con la gran ventaja de la reducción de su cabecera a menos de 8 octetos.

a. Formato de la cabecera de UDP.

Puerto Origen
Puerto destino
Longitud
Checksum (Opcional)
Datos (Variable)

- Puerto origen y destino: (16), SAP de nivel de aplicación. El puerto origen es opcional, si no se emplea se colocan todos sus bit a cero.
- Longitud: (16), total de cabecera y datos.

- Checksum: (16), puede o no estar presente, si lo está es un CRC 16 de cabecera, datos y Pseudo cabecera cono TCP. Este campo también es opcional, y de no usarse también se rellena con ceros.
- Datos: (Variable), UDP (Unidad de datos de protocolo) de nivel superior.

Los protocolos más comunes que emplean UDP son los que se detallan a continuación:

Puerto	Protocolo
7	Eco
13	Fecha
53	DNS (Sistema de Nombres de Dominio)
67 y 68	BOOT cliente y Servidor
69	TFTP (Trivial File Transfer Protocol)
123	NTP (Network Time Protocol)
161 y 162	SNMP (Single Noetwork Monitor Protocol)

b. El peligro de los protocolos no orientados a la conexión:

Los protocolos no orientados a la conexión, como ya se vio, no establecen ni cierran la misma, sino que pasan directamente a la transferencia de datos, confiando que la información llegará al destino deseado.

Al generar este tipo de tráfico, **no se realiza ningún seguimiento del estado** de esa conexión, por eso también se los suele llamar sin estado. No existen números de secuencia, por lo tanto si se desea realizar un análisis de lo que está sucediendo se torna muy dificultoso. Por otro lado, **no se puede definir el "Sentido"** en el que se realiza la conexión, pues esta no existe, y sin ella tampoco se determinará quién desempeña el rol de cliente y quién el de servidor. Estos protocolos en particular son de especial interés en cuanto a las reglas de un firewall, justamente por las dos características que se acaban de mencionar.

3.6. ARP (Address Resolution Protocol) (RFC 826, 1293, 1390):

a. Funcionamiento

Para que se puede establecer la transferencia de datos entre dos ETD en la familia TCP/IP, estos deberán conocer obligatoriamente las direcciones IP y las de Hardware (MAC),del emisor y receptor; hasta que estas cuatro no se encuentren perfectamente identificadas, no se podrá iniciar ninguna transferencia de información. Bajo este esquema es fácil de pensar que si un ETD A desea envía información, conozca su Dirección MAC e IP, también es razonable que pueda conocer la dirección IP destino; el responsable de descubrir la dirección MAC faltante es el protocolo ARP. El mecanismo que emplea es el de mantener una tabla dinámica en memoria en cada ETD llamada caché ARP (La cual se puede analizar por medio del archivo ARP.exe), en la misma se van guardando todas las asociaciones de MAC-IP que escucha el ETD en la red. Al intentar transmitir información, analizará primero en su caché ARP si esta asociación existe, de no encontrarla generará un mensaje ARP.

b. Tipos de mensajes:

ARP trabaja por medio de una solicitud y una respuesta. La **Solicitud** es un broadcast de nivel 2 (FF.FF.FF.FF.FF.FF), el cual será escuchado por todas los ETD de la red con el formato que se graficará a continuación. Por ser Broadcast, todos los niveles 2 de todos los ETD de la red lo reconocerán como propio, entregando la UDP correspondiente al nivel 3 en todas los ETD. El único nivel 3 que lo tomará como suyo será el que identifique su propia dirección IP en esta cabecera quien responderá (**respuesta** ARP) colocando en la dirección MAC faltante la propia, pero ya no por medio de broadcast sino dirigida al ETD que generó la solicitud ARP, pues poseerá todos los datos necesarios. Al llegar a destino se completa toda la información necesaria para iniciar la transferencia de información, y se incluirá esta nueva asociación MAC-IP en la caché ARP.

Un caso obvio es cuando el ETD no pertenece a la propia red, por lo cual jamás será alcanzado por un broadcast de nivel 2 (pues un router lo filtraría). En esta situación, el router que sí escucha el broadcast (o puntualmente la solicitud ARP dirigida a él) reconoce que no se trata de una dirección de la red propia, y opera en forma similar a un ETD pero a través de tablas llamadas caché PROXY ARP, en las cuales mantiene asociaciones MAC-IP por cada puerto que posea, si en esta no se encuentra la dirección MAC buscada, generará un nuevo formato ARP por la puerta hacia la cual identifique la red IP correspondiente (este último paso puede variar acorde al tipo de protocolo que emplee el router), esto se repetirá a lo largo de toda una cadena de router hasta identificar la red IP correspondiente a la dirección buscada, donde se generará un broadcast que sí encontrará a la dirección IP deseada, la cual responderá la solicitud ARP, y por el camino inverso, y en forma dirigida llegarán la dirección MAC solicitada a destino, completando de esta forma las cuatro direcciones necesarias.

La última de las posibilidades existentes ocurre cuando un ETD no conoce su propia dirección IP, circunstancia que puede presentarse cuando bootea un ETD y solicita una asignación dinámica de dirección IP o también al inicializar un ETD que no poseen disco rígido. Ante este tipo de sucesos existe el Protocolo R_ARP (Reverse) (RFC 903), el cual genera un mensaje con formato semejante al ARP pero sin contener tampoco su propia dirección IP, la condición imprescindible para este protocolo es la existencia de un servidor R_ARP el cual recibirá este mensaje, resolviendo el direccionamiento IP del ETD que lo requiera. Los pasos de este protocolo son análogos a los del ARP. En el encabezado Ethernet, el campo identificador de protocolo de capa superior (SAP) llevará el valor 8035h que identifica R_ARP.

c. Formato del Header ARP.

Tipo de Hardware
Tipo de protocolo
Longitud de direcciones de Hardware
Longitud de direcciones de Protocolo
Código de operación
Dirección de Hardware del transmisor
Dirección IP del transmisor

Dirección de Hardware de receptor
Dirección IP de receptor

- Tipo de hardware: (16), interfaz de hardware empleado (Valor 1 para Ethernet).
- Tipo de protocolo: (16), identificador del protocolo que se emplea (Valor 0800 para IP).
- Longitud de dirección de Hardware y Protocolo: (8), limitan los campos posteriores a la cantidad de octetos que emplee cada uno de ellos.
- Código de operación: (16), (opcode), sólo se encuentran definidos 2 tipos, 1 = Solicitud, 2 = Respuesta (En realidad también están definidos 3 y 4 pero son para solicitud y respuesta de RARP) .
- Direcciones: (48) (32), especifican el tipo necesario para ser resuelto por ARP.

a. Ataque ARP:

Este ataque tiene sentido únicamente en redes LAN (No debe olvidarse que el 80 % de los ataques suceden en este entorno), se trata de una actividad verdaderamente peligrosa, pues redirecciona el tráfico hacia el equipo deseado. Su implementación es la siguiente:

- Se debe escuchar el tráfico ARP.
- Al detectar una solicitud ARP, se espera la respuesta correspondiente.
- Se capturan ambas.
- Se modifica el campo dirección MAC de la respuesta, colocando la dirección MAC de la máquina que desea recibir el tráfico IP, falsificando la verdadera MAC de la respuesta.
- Se emite la respuesta ARP falsa y ya está.

¿Qué se logra con esto?.

Si se supone que la solicitud ARP la emitió el host A y la respuesta ARP la emitió el host B, el resultado de estos mensajes es que el host A, al recibir la respuesta de B, almacena en su memoria caché ARP la dupla IP(B) ←→ MAC(B). Si a continuación de este diálogo, el host A recibe otra respuesta ARP que le asocia la IP(B) con una nueva MAC, supóngase MAC (X), el host A, automáticamente sobreesacribirá su memoria caché ARP con la nueva información recibida: IP(B) ←→ MAC(X). A partir de este momento cada vez que emita información hacia la dirección IP del host A, la dirección MAC que colocará será la MAC(X), ante lo cual, el nivel Etehernet del host A descartará esa información, la cual sí será procesada por el protocolo Ethernet del host X, el cual por ser el intruso, sabrá como procesarlo.

3.7. DNS (Domain Name System) (RFC 1706, 1591, 1034 y 1035):

a. TLD (genéricos y geográficos).

Este servicio es quien determina el formato de los nombres que se emplean en Internet y permite su asociación con la dirección IP correspondiente, queda establecido por la RFC 1591. Al nacer Internet, se implementó este servicio de nombres, a través de bases de datos de características planas, las cuales por el exponencial crecimiento de esta red, rápidamente obligó a estructurarse como un sistema jerárquico de archivos, residente en los servidores de nombres de dominio distribuidos en todo el mundo. El NIC (Network Information Center) lleva el registro mundial de todas las direcciones IP, y a que nombre se corresponde.

Dentro de esta jerarquía de nombres, el primer nivel se denomina Top Level Domain (TLD) y puede ser de dos formas, genéricos que se caracterizan por tres o cuatro caracteres, ellos son: com, mil, edu, net, gov, org y arpa o geográficos que identifica al País de dos caracteres, por ejemplo ar, us, uk, br, etc, este responde a los códigos internacionales de dos caracteres estandarizados por la norma ISO 3166 .

El 16 de noviembre del 2000 ICANN aprobó siete nuevos TLD genéricos, los cuales ya están todos en funcionamiento y son:

- aero: Para industrias de transporte aéreo (www.nic.aero).
- biz: Para empresas (www.nic.biz).
- coop: Para cooperativas (www.nic.coop).
- info: Profesionales de informática (www.nic.info).
- museum: Para museos (www.nic.museum).
- name: Para registrar nombres individuales (www.nic.name).
- pro: Para profesionales.

Todos ellos menos .pro ya se encuentran operacionales y comenzando a recibir registros.

La jerarquía se establece de derecha a izquierda, separada por puntos, avanzando en este orden de lo general a lo particular, llegando a identificar por ejemplo al usuario que pertenece al departamento de una empresa comercial de un determinado País (aperez.produccion.perezsa.com.ar), se estila el empleo del caracter @ si bien su uso no es obligatorio. Existen también niveles menores (regulados por la RFC 1480) que establecen subdominios como pueden ser localidades, colegios, librerías, agencias federales, etc.

Dentro de este sistema es que cuando se desea conectar con un determinado ETD, supongamos una Web, es probable que se conozca su nombre DNS pero casi seguro que no su dirección IP. En estos casos es que entra en juego el Sistema DNS, en el cual el primer Servidor DNS al cual el ETD se encuentra conectado, podrá conocer o no la dirección IP a la que se desea llegar, si la conoce, directamente la asocia por medio del sistema llamado “Solucionador de nombres”, caso contrario elevará su solicitud en forma jerárquica al servidor de nombres de dominio inmediatamente superior a este, el cual procederá en forma análoga, hasta llegar al que posea esta información. El conjunto de nombres administrados por un servidor se llama ZONA, y por cada una de ellas existe un servidor primario y uno secundario (El cual resulta evidente al configurar un ETD para ingresar a Internet, pues obligatoriamente el Internet Service Provider al cual se llama deberá haberlos notificado para configurar el ETD correspondiente).

Acorde a la solicitud del cliente, un servidor DNS opera en forma **recursiva** o **iterativa** **(**definidas a través de un Flag en la solicitud cliente), La primera de ellas se produce cuando un servidor no conoce la dirección IP solicitada, entonces envía la misma a su Root, el cual operará de la misma forma hasta resolver la solicitud. Una solicitud Iterativa, es aquella en la cual ante

la ausencia de respuesta, el servidor devuelve la misma al cliente pudiendo indicarle a que otro servidor recurrir o no, y es el cliente el responsable de ir iterando este pedido hasta encontrar solución.

b. Zonas:

Cada servidor de nombres tiene *autoridad* para cero o más zonas. Hay tres tipos de servidor de nombres:

primario

Un servidor de nombres primario carga de disco la información de una zona, y tiene autoridad sobre ella.

secundario

Un servidor de nombres secundario tiene autoridad sobre una zona, pero obtiene la información de esa zona de un servidor primario utilizando un proceso llamado *transferencia de zona*. Para permanecer sincronizado, los servidores de nombres secundarios consultan a los primarios regularmente (típicamente cada tres horas) y re ejecutan la transferencia de zona si el primario ha sido actualizado. Un servidor de nombres puede operar como primario o secundario para múltiples dominios, o como primario para unos y secundario para otros. Un servidor primario o secundario realiza todas las funciones de un servidor caché.

caché

Un servidor de nombres que no tiene autoridad para ninguna zona se denomina servidor caché. Obtiene todos sus datos de servidores primarios o secundarios. Requiere al menos un registro NS para apuntar a un servidor del que pueda obtener la información inicialmente.

c. Formato de su Header:

El formato de un mensaje DNS es el siguiente:

16 bit	16 bit
Identificación	Parámetros
Número de Solicitud	Número de respuesta
Número de autoridad	Numero adicional
Sección Solicitud	
Sección respuesta	
Sección autoridad	
Sección de información adicional	

- Identificación: Identifica al mensaje, se emplea para llevar la correspondencia entre solicitudes y respuestas.
- Parámetros:
  * bit 1 (Q): Valor 0 = solicitud, valor 1 = respuesta.
  * bit 1 a 4 (OpCode) Tipo de consulta: Valor 0 = estándar, valor 1 = Inversa, valor 2 = solicitud de estado de servidor, (existen valor 3 y 4 en desuso).
  * bit 5 (A): Seteado si la solicitud es autoritativa (Flag de Autoritativo).
  * bit 6 (T): Seteado si el mensaje es truncado, es más largo de lo que permite el canal.
  * bit 7 (R): Seteado si se requiere recursión (Flag de recursividad).

* bit 8 (V): Seteado si la recursión está disponible el servidor soporta recursión.
* bit 9 a 11 (B): Reservados para uso futuro, su valor debe ser cero.
* bit 12 a 15 (Rcode): (Tipo de respuesta) valor 0 = sin error, valor 1 = Error de formato en solicitud, valor 2 = falla en servidor, valor 3 = nombre inexistente, valor 4 = tipo de consulta no soportada, 5 = consulta rechazada.

- Numero de...: Lleva la cuenta del número de mensajes que se cursan en las secciones que le siguen en el formato.
- Sección solicitud: contiene las consultas deseadas, consta de tres sub-campos: Nombre de Dominio (longitud variable), Tipo de consulta (Host, mail, etc) y Clase de consulta (permite definir otros objetos no estándar en Internet).
- Sección respuesta, autoridad e información adicional: consisten en un conjunto de registros que describen nombres y sus mapeos correspondientes.

d. Conexiones UDP y TCP:

Los mensajes DNS se transmiten sobre UDP o sobre TCP, en ambos sobre el puerto 53, y la mecánica a seguir es la siguiente:

- Los mensajes transportados por UDP se restringen a 512 bytes. En el caso de TCP el mensaje va precedido de un campo de 2 bytes que indica la longitud total de la trama.
- Un "resolver" del DNS o un servidor que envía una consulta que no supone una transferencia de zona *debe* enviar una consulta UDP primero. Si la sección "answer" de la respuesta está truncada y el solicitante soporta TCP, debería intentarlo de nuevo usando TCP. Se prefiere UDP a TCP para las consultas porque UDP tiene un factor de carga mucho menor, y su uso es esencial para un servidor fuertemente cargado. El truncamiento de mensajes no suele ser un problema dados los contenidos actuales de la base de datos del DNS, ya que típicamente se pueden enviar en un datagrama 15 registros, pero esto podría cambiar a medida que se añaden nuevos tipos de registro al DNS.
- TCP debe usarse para actividades de transferencia de zonas debido a que el límite de 512 bytes de UDP siempre será inadecuado para una transferencia de zona.
- Los servidores de nombres deben soportar ambos tipos de transporte.

e. Inundación recursiva e iterativa:

La metodología de esta debilidad es la de aprovechar la potencia que tiene este protocolo para obtener información de algún nombre, para generar mayor cantidad de táfico en la red. Si se alcanzan volúmenes importantes, se habla de inundación, pues puede llegar a dejar fuera de servicio a una red o servidor, ocupándose únicamente de esta actividad. Todo aquel que haya desarrollado programas que implementen técnicas recursivas sabe bien la potencia que estas poseen (y seguramente los desbordamientos de memoria que sin lugar a dudas alguna vez le ocasionó por error). En el caso de la iteración, el problema es similar, pues se plantean casos en los cuales por falta de resolución se produce la "Iteración eterna", es decir que nunca dejará de soicitar un determinado nombre.

f. Vulnerabilidades de DNS.

Las vulnerabilidade sde este protocolo, pueden clasificarse en cuatro grandes familias:

- UDP: Entre los servidores se transfieren grandes volúmenes de infroamción a través del puerto UDP 53, el cual por ser no orientado a la conexión lo hace especialmente difícil de controlar y filtrar, aprovechándose esta debilidad.
- Obtención de Infromación: Los servidores DNS almacenan información importante, la cual es muy buscada y fácilmente obtenible por un intruso.
- Texto plano: Toda la infromación viaja en texto plano.
- Falta de autenticación: El protocolo no ofrece ninguna técnica de autenticación.

Las siguientes son algunas de lar RFC referidas a DNS:

- RFC 1032 - Guía de administrador de DNS
- RFC 1033 - Guía de las operaciones de administrador de DNS
- RFC 1034 - Nombres de dominio - Conceptos y servicios
- RFC 1035 - Nombres de dominio - Implementación y especificación
- RFC 1101 - Codificación DNS de nombres de red y de otros tipos
- RFC 1183 - Nuevas definiciones del DNS RR
- RFC 1706 - Registros de recursos DNS NSAP

3.8. ICMP (Internet Control Messaging Protocol) (RFC: 792).

Este protocolo es quizás uno de los más importantes de esta pila pues es el que se encarga de la supervisión y control de la red. Un datagrama viaja entre router a través de la red hasta alcanzar su destino, si ocurre algún error o para controlar esta travesía es que se generan estos mensajes. Este sistema de reportes, es tratado por la red como cualquier otro datagrama (Nivel 3), pero el Software de la capa 3 los interpreta de manera distinta. ICMP no especifica las acciones a tomar, solamente sugiere la misma.

Todas las cabeceras de ICMP comienzan con tres campos:

- TIPO: (8), especifica el mensaje ICMP.
- CODIGO: (8), Brinda un poco más de información sobre el error.
- CHECKSUM (16), CRC 16.

Los campos que continúan a estos tres, varían acorde al tipo de error, pero en la mayoría de ellos se encuentra incluido el encabezado del datagrama que generó el mensaje ICMP y también los 64 primeros octetos de este para dejar unívocamente establecido la identificación del error.

El estudio del funcionamiento del protocolo ICMP se puede entender básicamente desarrollando el significado del campo TIPO, el cual representa los distintos tipos de mensajes.

a. Tipos y códigos de los mensajes ICMP:

- 0 y 8: Eco de solicitud y de respuesta: No es ni más ni menos que el comando PING, que genera una solicitud y una respuesta (configurable), para determinar la continuidad del recorrido de un datagrama a lo largo de una red, su cantidad de saltos y el tiempo demorado.
- 3: Destino no alcanzable: Se genera cuando un datagrama no encuentra la dirección IP destino. También ocurre cuando el bit de no fragmentar de la cabecera IP esta puesto en 1, y la red destino no soporta bloques del tamaño de ese datagrama, por lo cual no podrá ser

entregado a esa red, causando la no llegada a destino. Dentro de este tipo es interesante tener en cuenta el campo Código, pues brinda información adicional sobre las causas por las cuales no se llega a destino, en particular si lo que se desea es obtener información sobre ese extremo (Extremadamente usado para ataques a redes). Los valores que toma son:

- 0: Red inalcanzable.
- 1: Host inalcanzable.
- 2: Protocolo inalcanzable.
- 3: Puerto inalcanzable.
- 4: Fragmentación requerida y bit de no fragmentar puesto a 1 en el datagrama origen.
- 5: Falla en la ruta.
- 6: Red desconocida.
- 7: Host desconocido.
- 8: Host origen aislado.
- 9: Acceso a la red administrativamente prohibido.
- 10: Acceso al Host administrativamente prohibido.
- 11: Red inalcanzable por tipo de servicio.
- 12: Host inalcanzable por tipo de servicio.

- 4: Fuente agotada: Sirve para regular el flujo de información. Implica un buffer lleno, causa por la cual sería conveniente que el Host transmisor dejara de hacerlo hasta que deje de recibir estos mensajes.
- 11: Tiempo de vida excedido: El campo TTL llegó a 0.
- 5: Se requiere redireccionamiento: Existe una ruta mejor.
- 12: Problemas con el parámetro: Error semántico o sintáctico en el encabezamiento IP.
- 13 y 14: Solicitud y respuesta de marcador de tiempo: Permite la sincronización de clock entre nodos, a través de la hora GMT (Greenwich Mean Time).
- 15 y 16: Solicitud y repuesta de información: Permite obtener información de un nodo. Este fue originariamente pensado para los protocolos BOOTP y R_ARP.
- 17 y 18: Solicitud y respuesta de máscara de dirección: Permite determinar las máscaras de las redes con que está conectada un nodo. Se emplea para el ruteo hacia esas redes.

3.9. IGMP (Internet Group Messaging Protocol) (RFC 1112).

Este protocolo es muy similar al anterior, pero está pensado para mensajes entre grupos de host empleando también los datagramas IP para transportar su información.

a. Multicast IP sobre Ethernet:

Los primeros cuatro bit de una dirección IP si se encuentran como 1110 identifican una dirección Tipo D o Dirección de multicast, los restantes 28 bit identificarán a que grupo se refiere, dentro

de este esquema se debería comenzar con 224.0.0.0 la cual no se emplea por ser reservada. La siguiente es 224.0.0.1 que define a todos los grupos de host y router que participan en multicast.

Una dirección multicast, se debe aclarar que jamás puede definir una dirección origen, siempre se referirá a una destino.

Dentro de un esquema Ethernet, para que una dirección de multicast de nivel IP pueda ser entregado a los equipos deseados, es imprescindible que el nivel 2 sepa identificarlos para que de alguna manera no los descarte en ese nivel. Para que esto suceda, una solución podría ser un Broadcast de nivel 2, ante lo cual todas los Host de esa LAN lo reconocerían como propio, lo desencapsularían y entregarían el datagrama al nivel 3 (IP). Esta opción desde ya desperdicia bastante el esquema de direccionamiento de Ethernet, pues lo ideal sería que sólo sea tenido en cuenta por los Host que integran el grupo de multicast de nivel IP. Para implementar este procedimiento Ethernet ubica los últimos tres octetos de la dirección IP en los mismos últimos tres de la dirección NIC o MAC de este nivel en una dirección grupal específica que es 01-00-5E-00-00-00.

Ej: Si el multicast IP fuera 224.0.0.1 la dirección Ethernet es 01-00-5E-00-00-01

Los Host que participan de un esquema de multicast IP, pueden desempeñar tres niveles diferentes:

- Nivel 0: El Host no puede recibir ni enviar IP Multicast.
- Nivel 1: El Host no puede recibir pero puede enviar IP Multicast.
- Nivel 2: El Host puede recibir y enviar IP Multicast.

b. Fases de IGMP:

IGMP posee dos fases:

- Fase 1: Cuando un Host se incorpora a un grupo de multicast enviando un mensaje a todos los Host del Grupo, anunciándose como miembro. Los router locales reciben este mensaje y lo propagan al resto de los miembros.
- Fase 2: Como los grupos son dinámicos, los router locales periódicamente sondean (Poll) los host para mantener el estado de los grupos.

c. Formato del mensaje IGMP:

4	4	8	16
Versión	Tipo	Reservado	CRC
Direcciones de Grupo			

- Versión: La versión actual es la 1.
- Tipo: Posee dos valores, Consulta de un router (1), y respuesta enviada por un host (2).
- Reservado: Debe estar puesto a cero.
- CRC: Control de error de los 8 octetos.
- Grupo de direcciones: Reporte de los miembros de un grupo multicast. Si es una consulta debe ir puesto a cero.

Existe un protocolo para transmitir información de ruteo entre grupos multicast que emplea el algoritmo de vector distancia y se llama DVMRP (Distance Vector Multicast Routing Protocol).
Empleos de multicast.

3.10. Telnet (Terminal remota)(RFC 854, 855 y 857):

a. Conceptos:

Este protocolo es el que hace posible el acceso a terminales remotas, operando las mismas como si fueran locales. Los comandos Telnet los usa el protocolo, no los usuarios debido a que el papel de Telnet es conectar al usuario, y que este se comunique en forma directa. Estos comandos se envían en un paquete llamado command que contiene 2 o 3 octetos, y son los que establecen la conexión. Una vez realizada la misma, habitualmente se solicitará un nombre de usuario y una contraseña, pues se está disponiendo el uso de los recursos de ese equipo. Es muy común su empleo para consultas BBS, a terminales en Internet y también para la administración remota de dispositivos de hardware como suelen ser Hub, Switch, Router, etc.

TELNET es un protocolo basado en tres ideas:

- El concepto de *NVT(Network Virtual Terminal) (NVT)*. Una NVT es un dispositivo imaginario que posee una estructura básica común a una amplia gama de terminales reales. Cada host mapea las características de su propia terminal sobre las de su correspondiente NVT, y asume todos los demás hosts harán lo mismo.
- Una perspectiva simétrica de las terminales y los procesos.
- Negociación de las opciones de la terminal. El protocolo TELNET usa el principio de opciones negociadas, ya que muchos host pueden desear suministrar servicios adicionales, más allá de los disponibles en la NVT. Se pueden negociar diversas opciones. El cliente y el servidor utilizan una serie de convenciones para establecer las características operacionales de su conexión TELNET a través de los mecanismos "DO, DON'T, WILL, WON'T"("hazlo, no lo hagas, lo harás, no lo harás")

b. Negociación:

Como se mencionó con anterioridad, este protocolo trabaja sobre TCP, es decir que primero se establece una sesión TCP y luego la conexión TELNET. Una vez realizada la misma, el cliente entra en una fase de negociación dinámica que determina las opciones de cada lado de la conexión, que justamente por ser dinámicas es que en cualquier momento pueden ser modificadas. Esta negociación se lleva a cabo por un conjunto de comandos TELNET, los cuales son precedidos por un caracter intérprete de comando (IAC) que es un octeto compuesto por todos unos (FF hex) y luego sigue el código de comando, y en el caso que este posea opción continuará un tercer octeto de opción negociada:

IAC	Command Code	Option Negotiated

c. Comandos y códigos:

A continuación se incluye la lista de códigos de comandos:

Comando	Dec	Hex	Descripción
End subNeg	240	FO	End of option subnegotiation command
No Operation	241	F1	No operation command

Data Mark	242	F2	End of urgent data stream.
Break	243	F3	Operator pressed the Break key or the Attention key.
Int process	244	F4	Interrupt current process
Abort output	245	F5	Cancel output from current process.
You there?	246	F6	Request acknowledgment
Erase char	247	F7	Request that operator erase the previous character.
Erase line	248	F8	Request that operator erase the previous line.
Go ahead!	249	F9	End of input for half-duplex connections.
SubNegotiate	250	FA	Begin option subnegotiation
Will Use	251	FB	Agreement to use the specified option
Won't Use	252	FC	Reject the proposed option.
Start use	253	FD	Request to start using specified option.
Stop Use	254	FE	Demand to stop using specified option
IAC	255	FF	Interpret as command.

En el caso que los comandos posean opciones negociables, las mismas son identificadas por una Option ID, la cual sigue inmediatamente después del comando. Las Option ID son las que se detallan a continuación:

Dec	Hex	Código de opción	Descripción
0	0	Binary Xmit	Allows transmission of binary data.
1	1	Echo Data	Causes server to echo back all keystrokes.
2	2	Reconnect	Reconnects to another TELNET host.
3	3	Suppress GA	Disables Go Ahead! command.
4	4	Message Sz	Conveys approximate message size.
5	5	Opt Status	Lists status of options.
6	6	Timing Mark	Marks a data stream position for reference.
7	7	R/C XmtEcho	Allows remote control of terminal printers.
8	8	Line Width	Sets output line width.
9	9	Page Length	Sets page length in lines.
10	A	CR Use	Determines handling of carriage returns.
11	B	Horiz Tabs	Sets horizontal tabs.
12	C	Hor Tab Use	Determines handling of horizontal tabs.
13	D	FF Use	Determines handling of form feeds.
14	E	Vert Tabs	Sets vertical tabs.
15	F	Ver Tab Use	Determines handling of vertical tabs.
16	10	Lf Use	Determines handling of line feeds.
17	11	Ext ASCII	Defines extended ASCII characters.
1	12	Logout	Allows for forced log-off.
19	13	Byte Macro	Defines byte macros.
20	14	Data Term	Allows subcommands for Data Entry to be sent.
21	15	SUPDUP	Allows use of SUPDUP display protocol.
22	16	SUPDUP Outp	Allows sending of SUPDUP output.
23	17	Send Locate	Allows terminal location to be sent.
24	18	Term Type	Allows exchange of terminal type information.
25	19	End Record	Allows use of the End of record code (0xEF).
26	1A	TACACS ID	User ID exchange used to avoid more than 1 log-in.
27	1B	Output Mark	Allows banner markings to be sent on output.
2	1C	Term Loc#	A numeric ID used to identify terminals.
29	1D	3270 Regime	Allows emulation of 3270 family terminals.
30	1E	X.3 PAD	Allows use of X.3 protocol emulation.
31	1F	Window Size	Conveys window size for emulation screen.
32	20	Term Speed	Conveys baud rate information.
33	21	Remote Flow	Provides flow control (XON, XOFF).
34	22	Linemode	Provides linemode bulk character transactions.
255	FF	Extended options list	Extended options list.

d. Vulnerabilidades:

Si bien posee otras de menor magnitud, la que jamás se debe olvidar es que absolutamente todo bajo este protocolo va como Texto Plano, es decir, que se lee con total libertad.

3.11. FTP (File Transfer Protocol) (RFC 959).

Este protocolo permite la transferencia remota de archivos sin establecer una sesión Telnet.

a. Establecimiento de la conexión y empleo de puerto de comando y puerto de datos.:

Una característica particular de su funcionamiento es que emplea dos puertos (dos canales TCP), el puerto 20 por medio del cual transfiere datos, llamado Data Transfer Process (DTP), y el puerto 21 por medio del cual transmite las instrucciones de comando llamado Protocol Interpreter (PI). Al igual que telnet, los comandos los usa el protocolo y no el usuario; estos comandos son secuencias en ASCII de cuatro caracteres (QUIT, PASS, PORT, DELE, LIST, ABORT, etc). Las conexiones FTP se inician de manera similar a Telnet con el nombre o dirección del Host destino (Ej: ftp 205.29.24.11), luego se debe registrar como usuario válido (en algunos se suele emplear la cuenta anonymous o guest) y generalmente como cortesía se emplea como contraseña la cuenta de correo electrónico, para permitirle al administrador llevar un registro de accesos. Luego se define un directorio de inicio, un modo de transferencia de datos (ASCII o binario), se inicia la transferencia y por último se detiene.

Las tramas de control FTP, son intercambios TELNET y contienen los comandos y opciones de negociación mencionadas en el punto anterior, sin embargo la mayoría de los mismos son simples textos en ASCII y pueden y pueden ser clasificados en comandos y mensajes FTP, los cuales se detallan a continuación:

b. Comandos:

Comando	Descripción
ABOR	Abort data connection process.
ACCT <account>	Account for system privileges.
ALLO <bytes>	Allocate bytes for file storage on server.
APPE <filename>	Append file to file of same name on server.
CDUP <dir path>	Change to parent directory on server.
CWD <dir path>	Change working directory on server.
DELE <filename>	Delete specified file on server.
HELP <command>	Return information on specified command.
LIST <name>	List information if name is a file or list files if name is a directory.
MODE <mode>	Transfer mode (S=stream, B=block, C=compressed).
MKD <directory>	Create specified directory on server.
NLST <directory>	List contents of specified directory.
NOOP	Cause no action other than acknowledgement from server.
PASS <password>	Password for system log-in.
PASV	Request server wait for data connection.
PORT <address>	IP address and two-byte system port ID.
PWD	Display current working directory.
QUIT	Log off from the FTP server.
REIN	Reinitialize connection to log-in status.
REST <offset>	Restart file transfer from given offset.

RETR <filename>	Retrieve (copy) file from server.
RMD <directory>	Remove specified directory on server.
RNFR <old path>	Rename from old path.
RNTO <new path>	Rename to new path.
SITE <params>	Site specific parameters provided by server.
SMNT <pathname>	Mount the specified file structure.
STAT <directory>	Return information on current process or directory.
STOR <filename>	Store (copy) file to server.
STOU <filename>	Store file to server name.
STRU <type>	Data structure (F=file, R=record, P=page).
SYST	Return operating system used by server.
TYPE <data type>	Data type (A=ASCII, E=EBCDIC, I=binary).
USER <username>	User name for system log-in.

c. Mensajes (Son las respuestas a los comandos):

Código	Descripción
110	Restart marker at MARK yyyy=mmmm (new file pointers).
120	Service ready in nnn minutes.
125	Data connection open, transfer starting.
150	Open connection.
200	OK.
202	Command not implemented.
211	(System status reply).
212	(Directory status reply).
213	(File status reply).
214	(Help message reply).
215	(System type reply).
220	Service ready.
221	Log off network.
225	Data connection open.
226	Close data connection.
227	Enter passive mode (IP address, port ID).
230	Log on network.
250	File action completed.
257	Path name created.
331	Password required.
332	Account name required.
350	File action pending.
421	Service shutting down.
425	Cannot open data connection.
426	Connection closed.
450	File unavailable.
451	Local error encountered.
452	Insufficient disk space.
500	Invalid command.
501	Bad parameter.
502	Command not implemented.
503	Bad command sequence.
504	Parameter invalid for command.
530	Not logged onto network.
532	Need account for storing files.
550	File unavailable.
551	Page type unknown.
552	Storage allocation exceeded.
553	File name not allowed.

d. T_FTP:

Existe un protocolo de transferencia de archivos diseñado para operar en modo No Orientado a la Conexión que es el TFTP (Trivial) (RFC: 783, 1350), el cual difiere del FTP en que no se registra en la máquina remota y que opera sobre UDP en lugar de TCP. Se define en el puerto número 69, y es común su empleo en ETD que no poseen disco rígido para cargar aplicaciones o programas fuente. Posee un conjunto de comandos y parámetros que se detallan a continuación:

Comando	Descripción
Read Request	Request to read a file.
Write Request	Request to write to a file.
File Data	Transfer of file data.
Data Acknowledge	Acknowledgement of file data.
Error	Error indication

Parámetro	Descripción
Filename	The name of the file, expressed in quotes, where the protocol is to perform the read or write operation.
Mode Datamode	The format of the file data that the protocol is to transfer.

e. Vulnerabilidades:

FTP, es uno de los primeros protocolos de la familia y por esta razón, nace en una época en la cual la seguridad no era un problema. Este origen lo hace particularmente vulnerable.

Toda la comunicación, al igual que Telnet, viaja en texto plano, desde la cuenta de usuario, la contraseña, hasta los comandos y los datos.

La mejor y más práctica alternativa en la actualidad es su empleo por medio de SSL/TLS, como se verá más adelante.

Como medidas a tomar en el servidor, siempre se debe:

- Revisar permanentemente la configuración del servidor y de ser posible, emplear software de verificación de archivos (tipo Tripware).
- No colocar contraseñas encriptadas en el archivo etc/passw en el área ftp anónimo.
- Prestar especial atención a la configuración anónimo.
- Actualizar permanentemente el servidor.
- Nunca colocar archivos del sistema en el directorio ~ftp/etc.
- Que nunca cincidan los nombres de cuentas del directorio ~/ftp/etc/passwd con el /etc/passwd.
- No activar TFTP si no es estrictamente necesario.

3.12. SMTP (Simple Mail Transfer Protocol) (RFC: 821, 822):

a. Funcionamiento:

Es el método definido por Internet para transferencia de correo electrónico. Emplea el puerto TCP 25. Trabaja por medio del empleo de colas o spooler donde va almacenando los mensajes recibidos en los servidores hasta que un usuario se conecte y transfiera su correspondencia, si esto no sucede en un determinado tiempo (Programable), los mensajes son descartados o

devueltos a su origen. Debe quedar perfectamente claro que su operatoria no es en tiempo real, sino que dependerá de la voluntad de sus corresponsales. Una característica particular es que todo el texto se transfiere en caracteres ASCII de 7 bit. Su conexión se produce por medio de tramas de comando y respuesta que incluyen instrucciones como mail, RCPT, OK, Texto, etc.

La RFC 821 especifica el protocolo empleado para la transferencia de correo, y la RFC 822 describe la sintaxis que deben seguir las cabeceras y su correspondiente interpretación, otra RFC que es conveniente tener en cuenta es la 974 que es la que define el estándar a seguir para el encaminamiento de correo a través de DNS.

b. Texto plano y extensiones:

Como se mencionó, el protocolo SMTP trabaja con texto plano (ASCII de 7 bit), lo cual en la actualidad no es suficiente para las aplicaciones que requieren imágenes, caracteres especiales, ficheros ejecutables, etc. Para este propósito es que se diseñaron las RFC 1521 y 1522 que definen MIME (Multipropose Internet Mail Extension), el cual transforma cadenas de 8 bit en grupos de siete que son los que viajarán por el canal de comunicaciones, y realizará el proceso inverso del lado receptor.

c. Mensajes (cabecera y contenido):

Cada mensaje tiene:

- Una cabecera (o sobre) con estructura RFC 822. La cabecera termina con una línea nula (una línea con sólo la secuencia <CRLF>).
- Contenido: Todo lo que hay tras la línea nula es el cuerpo del mensaje.

La cabecera es una lista de líneas de la forma:

field-name: field-value

Algunos campos habituales son:

To: Receptores primarios del mensaje.

Cc: Receptores Secundario("carbon-copy") del mensaje.

From: Identidad del emisor.

reply-to: El buzón al que se han de enviar las repuestas. Este campo lo añade el emisor.

return-path: Dirección y ruta hasta el emisor. Lo añade el sistema de transporte final que entrega el correo.

Subject: Resumen del mensaje. Suele proporcionarlo el usuario.

d. Comandos y códigos:

Todos los comandos, réplicas o datos intercambiados son líneas de texto, delimitadas por un <CRLF>. Todas las réplicas tienen un código numérico el comienzo de la línea. La secuencia de envío y recepción de mensajes es la siguiente:

1) El emisor SMTP establece una conexión TCP con el SMTP de destino y espera a que el servidor envíe un mensaje *"220 Service ready"* o *"421 Service not available"* cuando el destinatario es temporalmente incapaz de responder.

2) Se envía un HELO (abreviatura de "hello"), con el que el receptor se identificará devolviendo su nombre de dominio. El SMTP emisor puede usarlo para verificar si contactó con el SMTP de destino correcto.

   Si el emisor SMTP soporta las extensiones de SMTP definidas en el RFC 1651, puede sustituir el comando HELO por EHLO. Un receptor SMTP que no soporte las extensiones responderá con un mensaje *"500 Syntax error, command unrecognized"*. El emisor SMTP debería intentarlo de nuevo con HELO, o si no puede retransmitir el mensaje sin extensiones, enviar un mensaje QUIT.

   Si un receptor SMTP soporta las extensiones de servicio, responde con un mensaje multi-línea *250 OK* que incluye una lista de las extensiones de servicio que soporta.

3) El emisor inicia ahora una transacción enviando el comando MAIL al servidor. Este comando contiene la ruta de vuelta al emisor que se puede emplear para informar de errores. Nótese que una ruta puede ser más que el par *buzób@nombre de dominio del host*. Además, puede contener una lista de los hosts de encaminamiento. Si se acepta, el receptor replica con un *"250 OK"*.

4) El segundo paso del intercambio real de correo consiste en darle al servidor SMTP el destino del mensaje(puede haber más de un receptor). Esto se hace enviando uno o más comandos RCPT TO:<*forward-path*>. Cada uno de ellos recibirá una respuesta *"250 OK"* si el servidor conoce el destino, o un *"550 No such user here"* si no.

5) Cuando se envían todos los comandos rcpt, el emisor envía un comando *DATA* para notificar al receptor que a continuación se envían los contenidos del mensaje. El servidor replica con *"354 Start mail input, end with <CRLF>.<CRLF>"*. Nótese que se trata de la secuencia de terminación que el emisor debería usar para terminar los datos del mensaje.

6) El cliente envía los datos línea a línea, acabando con la línea <CRLF>. <CRLF> que el servidor reconoce con *"250 OK"* o el mensaje de error apropiado si cualquier cosa fue mal.

7) Ahora hay varias acciones posibles:

   - El emisor no tiene más mensajes que enviar; cerrará la conexión con un comando QUIT, que será respondido con *"221 Service closing transmission channel"*.
   - El emisor no tiene más mensajes que enviar, pero está preparado para recibir mensajes(si los hay) del otro extremo. Mandará el comando TURN. Los dos SMTPs intercambian sus papelees y el emisor que era antes receptor puede enviar ahora mensajes empezando por el paso 3 de arriba.
   - El emisor tiene otro mensaje que enviar, y simplemente vuelve al paso 3 para enviar un nuevo MAIL.

Una facilidad que ofrece SMTP es la unificación de mensajes con destino múltiple, los cuales son grabados como un sólo mensaje en los servidores, los que se encargan de distribuirlos a los n corresponsales. Se detallan a continuación los comandos y respuestas:

Comando	Descripción
DATA	Begins message composition.
EXPN <string>	Returns names on the specified mail list.
HELO <domain>	Returns identity of mail server.
HELP <command>	Returns information on the specified command.
MAIL FROM <host>	Initiates a mail session from host.

NOOP	Causes no action, except acknowledgement from server.
QUIT	Terminates the mail session.
RCPT TO <user>	Designates who receives mail.
RSET	Resets mail connection.
SAML FROM <host>	Sends mail to user terminal and mailbox.
SEND FROM <host>	Sends mail to user terminal.
SOML FROM <host>	Sends mail to user terminal or mailbox.
TURN	Switches role of receiver and sender.
VRFY <user>	Verifies the identity of a user.

Código de respuesta	Descripción de la respuesta
211	(Response to system status or help request).
214	(Response to help request).
220	Mail service ready.
221	Mail service closing connection.
250	Mail transfer completed.
251	User not local, forward to <path>.
354	Start mail message, end with <CRLF><CRLF>.
421	Mail service unavailable.
450	Mailbox unavailable.
451	Local error in processing command.
452	Insufficient system storage.
500	Unknown command.
501	Bad parameter.
502	Command not implemented.
503	Bad command sequence.
504	Parameter not implemented.
550	Mailbox not found.
551	User not local, try <path>.
552	Storage allocation exceeded.
553	Mailbox name not allowed.
554	Mail transaction failed.

e. Pasarelas SMTP:

Una pasarela SMTP es un host con dos conexiones a redes distintas. Las pasarelas SMTP se pueden implementar de forma que conecten distintos tipos de redes. Se puede prohibir el acceso a la pasarela a determinados nodos de la red, empleando la sentencia de configuración RESTRICT. Alternativamente, la seguridad se puede implementar con un fichero de autorización de accesos, que es una tabla en la que se especifican de quién y a quién se puede enviar correo por la pasarela.

f. Terminología:

e Algo de terminología referida al correo electrónico:

- Agente de usuario (UA, user agent): programa que se usa como interfaz de usuario para el correo electrónico (leer, componer, enviar, gestionar, etc.)
- Agente de transferencia de mensajes (MTA, message transfer agent): se encarga del encaminamiento y almacenamiento de los mensajes de correo hasta su destino final.
- Protocolo de acceso al correo electrónico: lo usa un UA para acceder a un MTA, y recoger el correo para un usuario. Ejemplo: POP, IMAP.

- Protocolo de envío de correo electrónico: lo usa un MTA para enviar correo a otro MTA (también puede usarlo un UA para enviarlo a un MTA). Ejemplo: SMTP.

3.13. POP (Post Office Protocol) (RFC:1082, 1725, 1734, 1939):

Este protocolo permite a un usuario conectarse a un sistema y entregar su correo usando su nombre de usuario y contraseña (muy usado en UNIX) a través del puerto TCP 110.

La metodología a seguir para descargar correo es la explicada en SMTP, lo cual implica que cuando un servidor recibe un mail, establece la sesión SMTP con este destino y entrega su mensaje, esto exigiría que el destino final se encuentre siempre encendido y disponga de los recursoso necesarios para desempeñar la tarea de cliente y Servidor SMTP. Ninguna de las dos características son exigibles a un PC personal (Recursos y no apagado). Un método intermedio es descargar la función de servidor SMTP de la estación de trabajo del usuario final, pero no la función de cliente. Es decir, el usuario envía correo directamente desde la estación, pero tiene un buzón en un servidor. El usuario debe conectar con el servidor para recoger su correo.

El POP describe cómo un programa que se ejecuta en una estación de trabajo final puede recibir correo almacenado en sistema servidor de correo. POP usa el término "maildrop" para referirse a un buzón gestionado por un servidor POP.

Existe una gran variedad de clientes de correo que emplean POP cuya última versión es la 3 (RFC 1725). Cuando estos clientes emplean POP3, tienen la opción de dejar sus mensajes a un servidor y verlos remotamente (modo *en línea*), o transferir sus mensajes a un sistema local y consultarlos (modo *fuera de línea*). Cada uno tiene sus ventajas y desventajas, en el modo *en línea*, independientemente de la ubicación física en la que se encuentre el usuario podrá consultar su mail pues este se encuentra almacenado en un servidor, se debe tener en cuenta que cada vez que se conecte al servidor le serán transferido la totalidad de los mensajes (si la conexión es lenta, puede demorar mucho); desde el punto de vista del Administrador, permite centralizar los backup de toda la información almacenada, pero tiene la desventaja que de no controlarse puede llenar fácilmente un disco rígido (inclusive este es un tipo de ataque de negación de servicio). El modo fuera de línea permite organizar en carpetas locales la información histórica acorde a la preferencia del cliente, y este al conectarse al servidor solamente bajará los mail nuevos, reduciendo el tiempo de conexión.

POP - 3 no soporta el uso de carpetas globales o compartidas, como tampoco el uso de listas globales de direcciones, por lo tanto no existe forma de ver la totalidad de las cuentas de una organización automáticamente (Lo debe organizar manualmente el Administrador).

Una mejora que aparece a POP son las extensiones MIME, ya mencionadas (Multimedia Internet Mail Extension), las cuales están estandarizadas por las RFC: 2045 a 2049 y su tarea principal es extender el contenido de los mensajes de correo para poder adjuntar datos de tipo genéricos. Define 5 tipos de cabeceras:

- MIME-version: La actual es 1.0
- Content-Description: Una descripción en texto plano del objeto del cuerpo, suele ser de utilidad cuando el objeto es no legible.
- Content-Id: Un valor unívoco especificando el contenido de esta parte del mensaje.
- Content-Transfer-Encoding: Indica cómo codificar y decodificar los datos.
- Content-Type: Indica con qué aplicación se tratarán los datos

También propone 8 tipos: Text, Image, Audio, Video, Application, Message, Model y Multipart. y varios subtipos: Text: html, plain o richtext; Image: gif, jpeg; ...

3.14. IMAP 4 (Internet Message Access Protocol Versión 4) (RFC: 1203,1730 a 1733, 2060):

Se trata de la evolución natural del POP 3, posee las mismas características del anterior y agrega algunas más que permiten escalar el servicio de correo a entornos de grupo.

Aparte de los modos en línea y fuera de línea, IMAP - 4 introduce un tercer modo que se llama *Desconexión*. En POP - 3, el cliente al conectarse y autenticarse automáticamente comenzaba a recibir la totalidad de los mail que se encontraban en el Servidor; en IMAP - 4 cuando un cliente se conecta y autentica en un servidor, este consulta sus *"banderas de estado"* para todos los mensajes existentes. Estas banderas permiten identificar a cada mensaje como: *Leído, borrado o respondido*, por lo tanto, puede ser configurado para bajar únicamente los marcados como *no leídos*, reduciendo sensiblemente el tiempo de conexión. Este modo facilita también cualquier anomalía que puede surgir durante la conexión pues el servidor IMAP - 4 entrega al **cliente sólo una copia de sus correos**, y mantiene el original hasta la sincronización completa de su caché, por lo tanto si se pierde una conexión durante una transferencia, al producirse la próxima conexión como primer medida se verá donde se abortó la previa par no reenviar todo, y en segundo lugar no se perderá ningún mensaje hasta que se sincronice el cliente y el servidor

Introduce también el concepto de *Vista Previa*, con lo cual el cliente puede revisar los encabezados de todos los mensajes y sobre estos decidir cuáles leer o borrar antes de bajarlos pues por ejemplo en conexiones dial-up es sumamente desagradable perder gran cantidad de tiempo en la recepción de avisos publicitarios forzados a ser recibidos por "no se sabe quien"

Introduce también el concepto de carpetas compartidas las cuales pueden ser consultadas por grupos de usuarios, y también el de pizarrones de noticias (semejante al protocolo NNTP: Network News Transfer Protocol), en los cuales los usuarios pueden "pinchar" sus carteles de novedades.

(POP-3, IMAP y MIME)

Vulnerabilidades del correo electrónico:

Las dos grandes vulnerabilidades que sufre el correo electrónico son referidas a su privacidad y su seguridad, dentro de ellas existen debilidades concretas que se tratan a continuación:

La privacidad es fácilmente vulnerable pues el correo viaja como texto plano, es decir, que si no se emplea algún algoritmo criptográfico, cualquiera puede tener acceso al mismo. En este tema, la mejor analogía es la del correo postal, en el cual a nadie se le ocurre enviar una carta sin el sobre.

La seguridad es atacada con dos técnicas puntuales: las bombas de correo (varias copias de un mismo mail a un solo destino) y el Spam (Correo no autorizado).

Las herramientas con que se cuenta para defenderse de estas vulnerabilidades son:

- S/MIME: Desarrollado por RSA el cual es una especificación para obtener autenticación por medio de firmas digitales. Se lo considera uno de los más seguros
- PGP: Pretty Good Privacy, el cual es un producto completo desarrollado por Phillip Zimmerman que ofrece que dentro de sus muchas funciones ofrece también autenticación, no repudio y criptografía siendo soportado por la mayoría de los clientes de correo.

- PEM: Privacy Enhanced Mail, el cual es una norma para permitir la transferencia de correo seguro. Con cualidades similares a PGP, siendo el estándar más reciente de los tres.

3.15. SNMP (Single Network Monitor Protocol).

Este es el protocolo que habilita las funciones que permiten administrar redes no uniformes. Esta regulado por la RFC 1155, 1156 y 1157, y básicamente separa dos grupos: Administradores y Agentes. Los Administradores (NMS: Network Managment Station) son los responsables de la administración del dominio ejecutando un determinado Software de monitoreo. Los agentes tienen a su vez un Software residente que responde a las solicitudes del administrador con la información guardada en sus bases de datos locales (MIB: Managment Information Base). Estas consultas en realidad pueden ejecutarse por dos métodos:

- Poll (Sondeo): La estación administradora sondea uno por uno a los agentes cada un determinado período de tiempo, y estos van informando si apareciera alguna novedad en su MIB desde el último sondeo.
- Interrupción: Los Agentes al aparecer alguna novedad en su MIB, envían un mensaje interrumpiendo los procesos del Administrador para notificar sus cambios.

Como puede deducirse cada uno de ellos tiene sus ventajas y desventajas; si una novedad apareciera inmediatamente después que un sondeo fue realizado a un agente, el Administrador tomaría conocimiento de este suceso recién en el próximo sondeo, lo cual por ejemplo en una red de Terapia Intensiva de un Hospital no sería muy saludable. Por el contrario, si se produjera alguna anomalía en el canal de comunicaciones en un sistema por interrupción, el Administrador nunca volvería a detectar novedades en un Agente que se encuentre sobre ese vínculo. Estos son algunos ejemplos, pero en virtud de la cantidad de posibilidades que existen es que se suelen implementar estrategias mixtas de monitoreo de red, que permitan superar estas contingencias.

Otro tema de especial interés en SNMP es la relación costo / beneficio de mantener la Administración absoluta de la red hasta los últimos recursos, lo cual genera un gran volumen de tráfico en la misma. Se suelen establecer límites sobre el nivel de importancia de los agentes a monitorear para reducir la carga que impone este protocolo.

a. Formato del encabezado:

Versión	Comunidad	PDU

- Versión: Indica el número de versión, los valores admitidos son 1, 2 y 3.
- Comunidad: Este nombre indica el grupo al cual pertenece el mensaje y es empleado para la autenticar al administrador antes que pueda ingresar al agente.
- PDU (Protocol Data Unit): Existen cinco tipos de PDU: GetRequest, GetNextRequest, GetResponse, SetRequest y Trap.

La PDU tiene a su vez un formato que es el siguiente:

PDU Type	Request ID	Error Status	Error Index	Objeto 1 Valor 1	Objeto 2 Valor 2	…. ….

- PDU type: Especifica el tipo de PDU, sus valores son:
  0 GetRequest.
  1 GetNextRequest.
  2 GetResponse.
  3 SetRequest.

- Request ID: Valor entero que controla la correspondencia entre agente y administrador.
- Error status: valor entero que indica operación normal o cinco tipos de error:
  0 noError.
  1 tooBig: El tamaño de la GetResponse PDU requerida, excede lo permitido.
  2 noSuchName: El nombre del objeto solicitado no tiene correspondencia con los nombres disponibles en la MIB.
  3 badValue: La SetRequest contiene un tipo inconsistente, longitud o valor para la variable.
  4 readOnly: No definido en la RFC1157.
  5 genErr: Otros errores, los cuales no están explícitamente definidos.
- Error index: Identifica la entrada en la lista que ocasionó el error.
- Object/value: Define el objeto con su valor correspondiente.

Existe también otro formato de PDU, que es el de Trap PDU, el cual tiene los siguientes campos:

PDU Type	Enterprise	Agent Address	Gen Trap	Spec Trap	Time Stamp	Objeto 1 Valor 1	Objeto 2 Valor 2	…. ….

- PDU Type: Valor 4.
- Enterprise: Identifica al administrador de la "empresa" que definió la trap.
- Agent Address: Dirección IP del agente.
- Generic Trap Type: Campo que describe el evento que está siendo reportado, los siguientes siete valores están definifdos:
  0 coldStart: La entidad ha sido reinicializada, indicando que la configuración pudo ser alterada.
  1 warmStart: La entidad ha sido reinicializada, pero la configuración no fue alterada.
  2 linkDown: El enlace ha fallado.
  3 linkUp: El enlace ha conectado.
  4 authenticationFailure: El agente ha recibido una autenticación SNMP indebida desde el administrador.
  5 egpNeighborLoss: Un EGP vecino esta caído.
  6 enterpriseSpecific: Un trap no genérico ha ocurrido, el cual es identificado por los campos Specific Trap Type y Enterprise.
- Specific Trap Type: Empleado para identificar un Trap no genérico.
- Timestamp: Representa el tiempo transcurrido entre la última reinicialización y la generación del presente trap.
- Combinación de la variable con su valor.

b. SNMP Versión 3:

En el mes de enero del año 1998 IETF propone un conjunto de RFC desde la 2271 a la 2275 las cuales definen un conjunto de medidas para implementar las tres grandes falencias que poseía el protocolo SNMP, estas son:

- Autenticación.
- Seguridad.

- Control de acceso.

Estos nuevos estándares propuestos son los que definen la nueva versión de este protocolo denominada versión 3. El propósito es definir una arquitectura modular que de flexibilidad hacia futuras expansiones.

Luego de un tiempo, en el mes de abril de 1999 aparecen ya como borrador estándar los mismos conceptos con algunas mejoras, dejando obsoletos los anteriores. Estas son las recomendaciones 2571 a la 2575, las cuales sientan definitivamente el funcionamiento de SNMPv3. Estas son:

2571 An Architecture for Describing SNMP Management Frameworks. B. Wijnen, D. Harrington, R. Presuhn. April 1999. (Format: TXT=139260 bytes) (Obsoletes RFC2271) (Status: DRAFT STANDARD)

2572 Message Processing and Dispatching for the Simple Network Management Protocol (SNMP). J. Case, D. Harrington, R. Presuhn, B. Wijnen. April 1999. (Format: TXT=96035 bytes) (Obsoletes RFC2272) (Status: DRAFT STANDARD)

2573 SNMP Applications. D. Levi, P. Meyer, B. Stewart. April 1999. (Format: TXT=150427 bytes) (Obsoletes RFC2273) (Status: DRAFT STANDARD)

2574 User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3). U. Blumenthal, B. Wijnen. April 1999. (Format: TXT=190755 bytes) (Obsoletes RFC2274) (Status: DRAFT STANDARD)

2575 View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP). B. Wijnen, R. Presuhn, K. McCloghrie. April 1999. (Format: TXT=79642 bytes) (Obsoletes RFC2275) (Status: DRAFT STANDARD)

En estas nuevas RFC aparecen una serie de conceptos que son los que se definen a continuación.

Entidad:

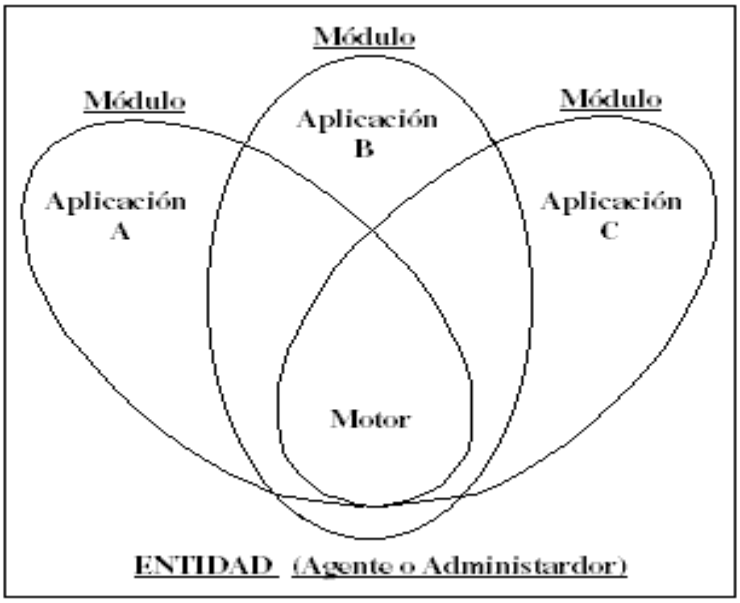
El concepto de entidad es una conjunto de módulos que interactúan entre sí, cada entidad implementa una porción de SNMP y puede actuar como los tradicionales nodo AGENTE, nodo GESTOR, o combinación de ambos.

Cada entidad incluye un MOTOR SNMP, siendo éste el encargado de implementar las funciones de:

- Envío de mensajes.
- Recepción de mensajes.
- Autenticación.
- Encriptado y desencriptado de mensajes.
- Control de acceso a los objetos administrados.

Estas funciones son provistas como servicios a una o más aplicaciones.

El conjunto de motor y aplicaciones son definidas como los módulos de esta entidad.



Gestor tradicional SNMP:

Un Gestor tradicional SNMP interactúa con los agentes SNMP a través del envío de comandos (get,get next y set) y recepción de respuestas. Este incluye 3 categorías de Aplicaciones:

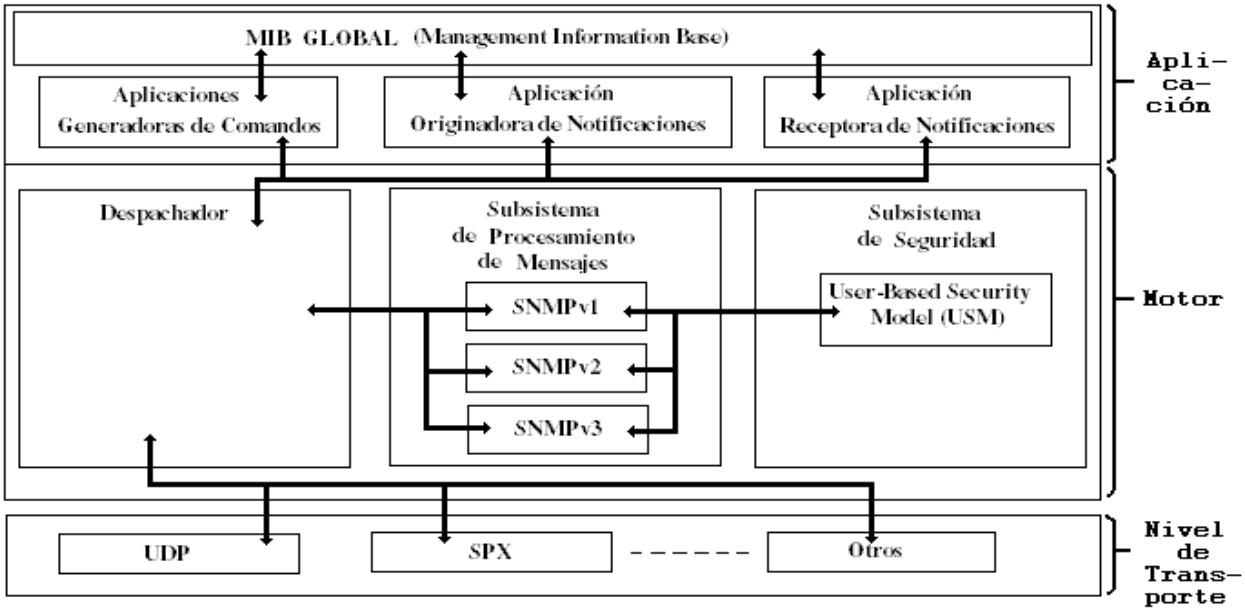
- Aplicaciones Generadoras de Comandos: Monitorean y controlan la administración de datos de un agente remoto.
- Aplicación Generadora de Notificaciones: Inicia mensajes asincrónicos.
- Aplicación Receptora de Notificaciones: Procesa mensajes entrantes asincrónicos.

Estas tres aplicaciones hacen uso de los servicios del motor SNMP.

Este motor debe contener:

1) Un Despachador: Encargado de administrar el tráfico. Para mensajes salientes, recibe las PDU (Unidad de datos de Protocolo) de las aplicaciones, determina el tipo de procesamiento requerido (Ej: SNMPv1, SNMPv2 o SNMPv3) y entrega estos datos al módulo de procesamiento de mensajes correspondiente. Para mensajes entrantes, acepta mensajes del nivel de transporte y lo deriva al módulo de procesamiento de mensajes correspondiente. Consecuentemente al recibir los mensajes procesados desde el módulo, los entregará hacia la aplicación apropiada o hacia el nivel de transporte según corresponda.
2) Un Subsistema de Procesamiento de Mensajes: Es el responsable del armado y desarmado de la PDU de este nivel. Recibe y entrega los mensajes del despachador. Si es necesario luego de armar la PDU (mensaje saliente) o antes de desarmarla (mensaje entrante), pasaría la misma al Subsistema de Seguridad

3) Un Subsistema de Seguridad: Es quien ejecuta las funciones de autenticación y encriptado. Recibe y entrega los mensajes al Subsistema de Procesamiento de Mensajes. Este subsistema soporta uno o más modelos distintos de seguridad llamado **User-Based Security Model (USM)** y está definido por la RFC- 2574.



Modelo de referencia de Administrador SNMP

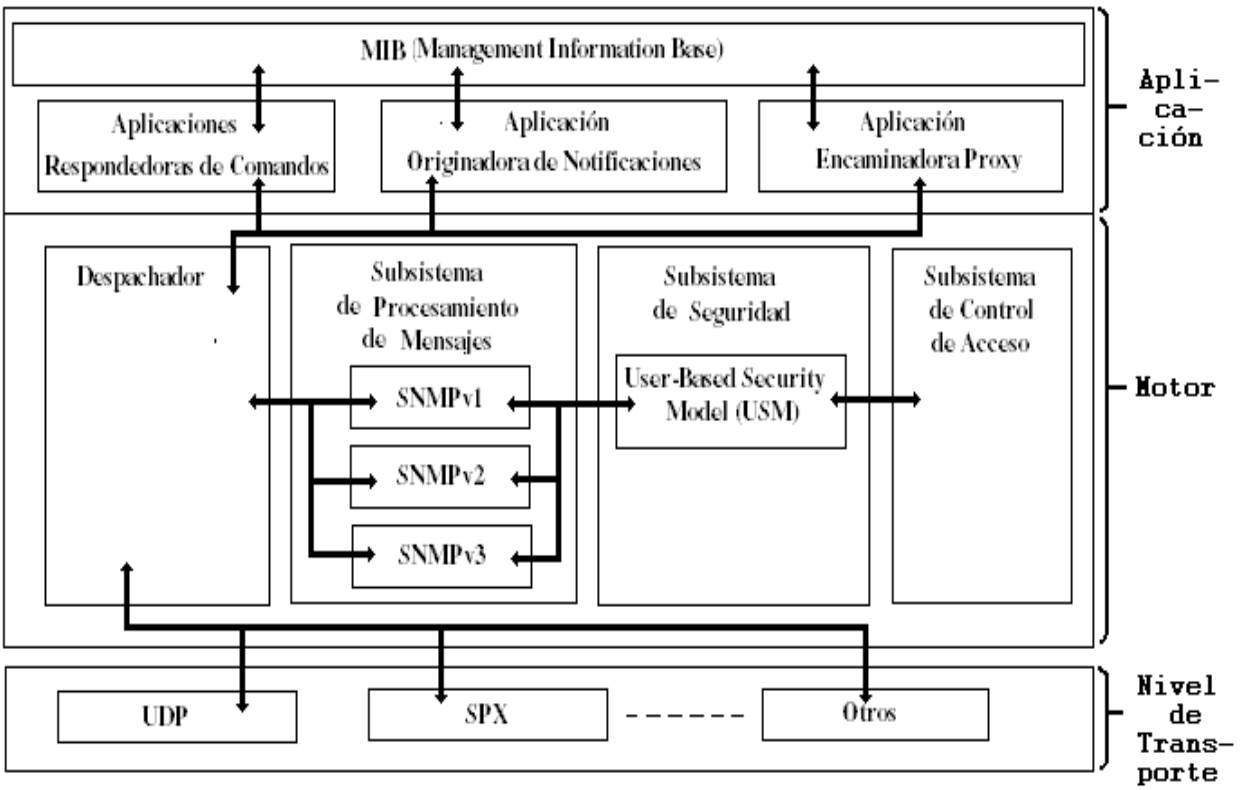
Agente Tradicional SNMP:

El agente contiene también 3 tipos de aplicaciones:

- Aplicaciones Respondedoras de Comandos: Provee acceso a los datos administrados.
- Aplicación Generadora de Notificaciones: Inicia mensajes asincrónicos.
- Aplicación Encaminadora Proxy: Encamina mensajes entre entidades.

El Agente tiene los mismos componentes que el Administrador e incluye uno más denominado:

- Subsistema de Control de Acceso: Es el encargado de proveer servicios de autorización para controlar el acceso a las MIBs.Un Agente soporta uno o más modelos de Control de Accesos diferentes llamado **View-Based Access Control Model (VACM)** y se encuentra definido por la RFC-2575.



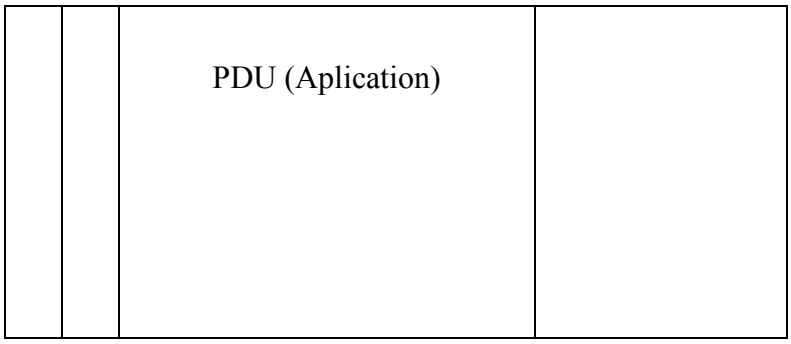
Modelo de referencia de Agente SNMP

Subsistema de procesamiento de mensajes:

Este modelo recibe PDU desde el despachador, tanto salientes como entrantes, las encapsula o desencapsula y acorde a la existencia de mecanismos de seguridad la entrega o no al USM para el tratamiento de los parámetros de seguridad (agregado, criptografiado o decriptografiado) y finalmente las devuelve al despachador para que este las entregue al nivel correspondiente.

Este modelo opera sobre los cinco primeros campos del encabezado SNMP, el cual se detalla a continuación:

A u t e n t i c a c :		Msg Version	Modelo de Procesamiento de Mensajes
		Msg ID	
		Msg Max Size	
		Msg FLAGS	
		Msg Security Model	
		Msg Autoritative Engine ID	Modelo de seguridad de usuario (USM)
		Msg Autoritative Engine Boot	
		Msg Autoritative Engine Time	
		Msg User Name	
		Msg Autentication Parameters	
		Msg Privacy Parameters	
	E n	Context Engine ID	Espacio de PDU
		Context Name	



- *Msg Version:* Corresponde el Nro 3.
- *Msg ID:* Identificador entre las dos entidades para coordinar solicitudes y respuestas
- *Msg Max Size:* Expresa el tamaño máximo en octetos que soporta el emisor.
- *Msg FLAGS:* Están definidos los tres primeros bit y significan lo siguiente:
  Bit 1 (bit de reporte): Si está en 1 entonces el receptor debe especificar bajo qué condiciones este puede causar un reporte. También es empleado en toda solicitud Get o Set.
  Bit 2 (Priv Flag): Indica que se emplea criptografía.
  Bit 3 (Aut Flag): Indica que se emplea autenticación.
- *Msg Security Model:* Indica qué modelo de seguridad se emplea (1 = SNMPv1, 2 = SNMPv2, 3 = SNMPv3).

Subsistema de seguridad:
El subsistema de seguridad ejecuta funciones de autenticación y encriptado, para las mismas define uno o más distintos *Modelos de seguridad de Usuario (USM)*. Específicamente la RFC-2574 establece que este modelo protege contra lo siguiente:

- Modificación de Información.
- Falsificación de entidad.
- Modificación de mensaje.
- Difusión de información.

También aclara que no protege contra ataques de negación de servicio ni análisis de tráfico.
Este modelo se emplea para proveer autenticación y privacidad, para esto define dos claves, una clave privada (PrivKey) y otra clave de autenticación (AutKey). El valor de estas claves no es accesible vía SNMP y se emplean de la siguiente forma:
Autenticación:

Se definen dos alternativas para esta tarea, HMAC-MD5-96 y HMAC-SHA-96.

La mecánica de esta función es que a través de una cadena de bit de entrada de cualquier longitud finita, generará un único resumen de salida de longitud fija. Que en el caso de esta norma es de 20 Byte para SHA o 16 Byte para MD5.

Esta función es llamada “One Way“ pues no es posible a través del resumen de salida obtener el texto de entrada, también resultará computacionalmente imposible obtener un valor de salida igual a través de otro valor de entrada, como así tampoco desde un valor de salida ya calculado, obtener otro valor de entrada diferente al verdadero.

La aplicación aquí propuesta toma los datos y la clave y produce un resumen:

- Resumen = H (clave, datos).

En cualquiera de los dos casos, se toman como válidos los primeros 96 bit, descartando el resto.

Esta especificación soporta el protocolo HMAC [RFC-2104] con la opción SHA1 (Hash Message Autentication-Secure Hash Standard Versión 1) [RFC-2404] y MD5 (Messge Digest Verión 5) [RFC-2403].

Criptografía:

Para esta actividad USM emplea el algoritmo DES (Data encryptation Standard) [ANSI X3.106] en el modo cifrado encadenado de bloques (CBC). La clave privada (PrivKey) antes mencionada de longitud 16 byte es empleada aquí dividiéndola en dos, los primeros 8 Byte, es decir 64 bit son empleados como clave para DES, el cual solo tendrá en cuenta 56, dejando 8 para control de paridad. Los últimos 8 Byte son empleados como Vector de Inicialización (IV) para comenzar con el cifrado en cadena.

Esta técnica CBC, se basa en tomar el primer bloque de texto plano, y realizar una operación XOR con un Vector de inicialización y luego de esta operación recién se pasará al cifrado de ese bloque. En el segundo bloque se realizará nuevamente la operación XOR, pero esta vez será el texto plano de este bloque con el bloque cifrado anteriormente, y luego se cifrará. Esta mecánica se irá realizando en los sucesivos bloques, es decir XOR con el bloque cifrado anterior y luego cifrado.
El descifrado se realiza en forma inversa.

- cifrado = E (clave, texto).
- D (clave, cifrado) = texto.

Campos del encabezado de USM:
Antes de tratar los campos de este modelo se debe tener en cuenta al concepto de autoritativo:

- Caso 1: Cuando un mensaje SNMP contiene datos que esperan una respuesta (Get, GetNext, Get Bulk, Set o Informes), entonces el receptor de ese mensaje es Autoritativo.
- Caso 2: Cuando un mensaje SNMP contiene datos que no imponen respuesta (Trap, Respuestas o Reportes), entonces el emisor de ese mensaje es Autoritativo.

Acorde a la gráfica anterior del encabezado SNMPv3, se puede apreciar que USM emplea los seis campos siguientes al Modelo de Procesamiento de Mensajes. Estos campos se detallan a continuación:

- *Msg Autoritative Engine ID:* Identificador de la entidad Autoritativa.
- *Msg Autoritative Engine Boot:* Este valor es un contador monótono creciente que identifica la cantidad de veces que la entidad autoritativa fue inicializada o reinicializada desde su configuración inicial.
- *Msg Autoritative Engine Time:* Este valor es un entero que describe el tiempo transcurrido en segundos desde el momento en que la entidad autoritativa incrementó el *Msg Autoritative Engine Boot* (es decir el tiempo desde la última vez que inició o reinició). Las entidades autoritativas llevan su tiempo exacto en segundos y las no autoritativas llevarán por cada entidad autoritativa con la que se comuniquen una apreciación del mismo, que servirá para compararlos en el momento oportuno (como se verá a continuación). Este valor son 32 bit, en el caso de no reinicializarse una entidad se irá acumulando y al llegar al valor máximo volverá a cero (En realidad como es un

valor de 32 bit, 2^{32} segundos son en el orden de 68 años, por lo tanto el sistema debería ser extremadamente sólido para no detenerse nunca en este lapso)

- *Msg User Name:* Nombre del usuario de este mensaje.
- *Msg Authentication Parameters:* Aquí es donde va el código de autenticación es decir el valor obtenido por HMAC. En el caso de no emplear autenticación es nulo.
- *Msg Privacy Parameters:* El valor aquí expuesto es el que se empleará para obtener el Vector de Inicializaciín (VI) para el algoritmo DES. En el caso de no emplear criptografía es nulo.

La secuencia de pasos a seguir con estos campos para la transmisión de un mensaje en este modelo es:

1) Como primera actividad se criptografían los datos en caso de implementar esta función.
2) Si se realizó el paso a. entonces se coloca en el campo *Msg Privacy Parameters* el valor correspondiente para generar el IV.
3) Si se emplea autenticación, la totalidad del mensaje se ingresa para obtener el resumen HMAC y su resultado es ubicado en el campo *Msg Authentication Parameters.*

En el caso de la recepción sería:

1) Realiza el cálculo de HMAC.
2) Compara el valor calculado con el correspondiente al campo campo *Msg Authentication Parameters*.
3) Si ambos valores son iguales, entonces toma el mensaje como auténtico y no ha sido alterado.
4) Verifica si el mismo está en un tiempo de ventana válido. Esta actividad se realiza de la siguiente forma:
   a) Toda entidad no autoritativa guarda tres parámetros en forma local de cada entidad autoritativa con la que se comunica, estos son:
      - El valor más reciente de *Msg Autoritative Engine Boot* recibido en la última comunicación.
      - El valor de tiempo estimado que debería tener la entidad autoritativa.
      - El último valor de tiempo recibido de la entidad autoritativa en el campo *Msg Autoritative Engine Time.*
   b) Al recibir un mensaje compara los campos del mensaje recibido con estos parámetros almacenados localmente.
   c) Las condiciones para que un mensaje sea considerado no auténtico son:
      - Diferencia de *Msg Autoritative Engine Boot.*
      - Diferencia en ± 150 segundos entre el valor calculado de *Msg Autoritative Engine Time* y el recibido en el mensaje.
   d) Si un mensaje es considerado no auténtico, una indicación de error es enviada al módulo respectivo.
5) Finalmente si está criptografiado, descifra el mismo.

Localización de claves:

Una clave localizada es un secreto compartido entre un usuario y un motor SNMP autoritativo. El problema del empleo de una sola clave por parte del usuario con todos los agentes es que si se descubriera la misma, sería vulnerable todo el sistema. Si el caso fuera lo contrario es decir que se deseara emplear una clave distinta para cada agente, entonces el usuario debería recordar todas las contraseñas lo cual en la práctica no es viable.

Para dar solución a estos problemas la RFC 2574 propone este proceso por el cual una clave única de usuario (o pueden ser dos: una para privacidad y otra para autenticación) es convertida

a múltiples claves únicas también, una para cada motor SNMP, este proceso es lo que se denomina **Localización de claves**. Las características fundamentales que propone este proceso son:

- Cada agente SNMP tiene su propia clave única para cada usuario autorizado a administrarlo, por lo tanto si la clave de uno de ellos es comprometida, no lo serán las del resto.
- La clave de un usuario es diferente en cada agente SSSNMP, por lo tanto si se compromete la clave de un agente, no comprometerá al resto ni a la clave del usuario.
- La administración de la red, puede realizarse en forma segura remotamente desde cualquier punto de la red.

Subsistema de Control de Accesos:

Este subsistema se ejecuta en los agentes tradicionales, permitiendo determinar quien está autorizado a acceder a la MIB de los mismos. El único modelo definido para esta actividad se denomina *Modelo de Control de Accesos basado en Vistas* (VACM: View-Based Access Control Model) y está definido en la RFC-2575.

VACM tiene dos características fundamentales:

- Determina si un acceso a la MIB local está permitido.
- Posee su propia MIB en la cual se definen las políticas de acceso y habilita la configuración remota.

Se definen cinco elementos que constituyen la VACM:

1) Grupos: Es un conjunto de cero o más duplas {Modelo de Seguridad, Nombre de seguridad} que definen los Objetos que pueden ser administrados por ese Nombre.
2) Nivel de seguridad: Define qué tareas serán permitidas (lectura, escritura o notificación) para cada grupo.
3) Contexto: Es un subconjunto de instancias de objetos en la MIB local. Permite agrupar objetos con distintas políticas de acceso.
4) Vistas de la MIB: Define conjuntos específicos de objetos administrados, los cuales se pueden agrupar en jerarquías de árboles y familias de manera tal que se pueda, por ejemplo, restringir su acceso a determinados grupos.
5) Política de acceso: VACM permite a un motor SNMP ser configurado para asegurar un conjunto de accesos correctos, los cuales dependen de los siguientes factores:
   - Los distintos usuarios pueden tener distintos privilegios de acceso.
   - El nivel de seguridad para una determinada solicitud.
   - El modelo de seguridad empleado para procesar las solicitudes de mensajes.
   - El contexto de la MIB.
   - La instancia al objeto para el cual fue solicitado el acceso.

3.16. DHCP Dynamic Host Configuration Protocol(RFC 1541, 1531, 1533 y 1534).

a. Evolución de los protocolos dinámicos (ARP, BOOTP):

Este protocolo es una evolución natural del BOOTP (BOOTstrap Protocol) que fue la primera implementación de protocolos de inicio para máquinas que no poseen disco rígido, las cuales al ser encendidas, deben primero hacerse presentes en la red y luego cargar el sistema operativo. Para automatizar este proceso IETF desarrollo este nuevo protocolo conocido como DHCP. Este último introduce dos grandes mejoras respecto al anterior:

- Primera: Permite a una máquina obtener toda la información necesaria en un solo mensaje.
- Segunda: Permite obtener una dirección IP rápida y dinámicamente.

b. Pasos de la asignación dinámica:

Todo cliente DHCP puede encontrarse en seis estados diferentes:

- Inicialización: Aún no posee ninguna dirección IP, para lo cual deberá contactar a todos los servidores DHCP en su red local. Para hacer esto generará un mensaje de descubrimiento DHCP.
- Selección: Espera una respuesta (Oferta DHCP) del primer servidor y lo elige.
- Solicitud: Ingresa a este estado cuando envía al servidor seleccionado un mensaje de solicitud DHCP.
- Enlace: Al recibir el ACK del servidor con la dirección solicitada, la cual ya queda asignada en forma definitiva por el lapso correspondiente.
- Renegociación: Al generar un mensaje de renegociación.
- Reenlace: Al generar un mensaje de reenlace.

Al ingresar una máquina al estado de enlace, inicia tres Timer de control de asignación, que en general son asignados por el servidor durante la asignación de direcciones:

- Renegociación: Por defecto es la mitad del intervalo de duración de la dirección asignada. Al expirar este tiempo el cliente debe renegociar su dirección IP.
- Reenlace: Por defecto expira al 87,5 % del tiempo de asignación. Al llegar a este tiempo, el cliente asume que el servidor se encuentra inactivo, y genera un mensaje Broadcast de reenlace
- Expiración: Expira si no recibe respuestas de ningún servidor y vence su tiempo de asignación.

Los protocolos DHCP y BOOTP son compatibles, de hecho un servidor DHCP puede ser programado para responder solicitudes BOOTP, sin embargo DHCP cambia el significado de dos campos en el Header del mensaje como se verá a continuación.

c. Formato del mensaje DHCP:

8	8	8	8
OP	HTYPE	HLEN	HOPS
TRANSACTION ID			
SECONDS		FLAGS	
CLIENT IP ADDRESS			
YOUR IP ADDRESS			
SERVER IP ADDRESS			
ROUTER IP ADDRESS			
CLIENT HARDWARE ADDRESS (16 octetos)			
SERVER HOST NAME (64 octetos)			
BOOT FILE NAME (128 octetos)			
OPTIONS (Variable)			

- OP: Toma valor (1) para solicitud y (2) para respuesta.
- HTYPE:
  (1) DHCPDISCOVER.

(2) DHCPOFFER.
(3) DHCPREQUEST.
(4) DHCPDECLINE.
(5) DHCPACK.
(6) DHCPNACK.
(7) DHCPRELEASE.

- HLEN: Especifica el tipo y longitud de la dirección de Hardware (Ej: Ethernet tiene tipo 1 y longitud 6 octetos).
- HOPS: El cliente coloca (0), si es necesario pasar a través de distintos router , el servidor BOOTP o DHCP lo incrementará.
- TRANSACTION ID: Contiene un número entero que permite llevar el control entre las solicitudes y respuestas.
- SECONDS: Determina el tiempo transcurrido desde que se inició la operación..
- FLAGS: Identifica por medio del primer bit si es un Broadcast, los restantes quince deben estar puestos a cero.
- CLIENT IP ADDRESS:
- YOUR IP ADDRESS:
- SERVER IP ADDRESS:
- ROUTER IP ADDRESS:
- CLIENT HARDWARE ADDRESS:
- SERVER HOST NAME:
- BOOT FILE NAME: Puede contener el tipo de booteo (Ej: UNIX)
- OPTIONS: Define máscara de subred, hora,etc

Hasta aquí se aprecia el aspecto general de DHCP, pero si se desea analizar en detalle su funcionamiento es necesaria remontarse a sus orígenes y tener en cuenta como nacen los protocolos de booteo (como se mencionó en la introducción).

Al inicializarse una máquina sin disco rígido, esta carga directamente de la ROM, una imagen a su memoria que le permite comenzar una secuencia de actividades. Como las direcciones IP no pueden ser impuestas por el fabricante pues justamente se trata de un esquema lógico, e inclusive en redes privadas estas pueden estar repetidas en cualquier lugar del mundo; la dirección IP debe ser solicitada a otra máquina en la red, e inclusive también necesitará una configuración particular para cada red, que también dependerá de otra máquina que podrá ser o no la misma que la anterior.

Una primera aproximación es el Protocolo R_ARP (Mencionado con anterioridad) que permite descubrir servidores y direcciones IP fuente y destino para este problema. La primera limitación está en la poca información que en sus campos se transmite (solo la IP cliente), la segunda es que es de muy bajo nivel, lo que exige a cualquier programador mucho más esfuerzo pues se debe llegar hasta el Hardware.

Para mejorar a R_ARP se diseñó BOOTP (Precursor de DHCP). La primera cualidad de este protocolo es que **emplea IP y UDP, causa por la cual tiene acceso al nivel de aplicación**, sin ninguna tarea adicional. El segundo detalle significativo es la cantidad de información que con solo dos mensajes (Solicitud y respuesta) puede transferir. Para graficar este concepto es conveniente analizar su formato (Similar a DHCP):

8 bit	8 bit	8 bit	8 bit
Operación	Tipo de Hard	Long Hard	Hops
Identificador de transacción			
Segundos		No definido	
Dirección IP cliente			
Su dirección IP			
Dirección IP del Server			

Dirección IP del Router
Dirección de Hardware cliente (16 octetos)
Nombre del Server (64 octetos)
Nombre del archivo de booteo (128 octetos)
Area específica del vendedor (64 octetos)

- Operación: Solicitud (1), Respuesta (2).
- Tipo y Longitud de Hardware: Especifica que tipo de hardware es empleado y su longitud (Ej: Ethernet : Tipo = 1 , Long = 6).
- Hops: El cliente coloca valor 0, si se permite el booteo a través de múltiples Router, el BOOTP Server, lo incrementa en 1.
- Identificador de transacción: Contiene un número entero generado pseudo-aleatoriamente, que permitirá reconocer respuestas con solicitudes.
- Segundos: Identifica el tiempo en el cual el cliente inició su operación de booteo.
- Direcciones y nombres: Estos campos le otorgan una gran flexibilidad a este protocolo, pues permiten ser llenados con la información que se posea, y en los campos que no se conozcan se colocarán ceros. Por lo tanto puede conocer o no su dirección, la del Server, el nombre del mismo, etc.
- Area específica del vendedor: Dentro de este campo se permite por medio de un formato establecido (un octeto de Tipo, un octeto de longitud y n octetos de valores):

Tipo	Longitud	Descripción
0	n	Relleno
1	4	Máscara de subred
2	4	Tiempo GMT
3	n	Direcciones IP de routers
4	n	Direcciones IP de Servidores de tiempo
6	n	Direcciones IP de DNS Server
7	n	Direcciones IP de de Log Server
10	n	Direcciones IP de Server de impresión
12	n	Nombres de cliente
13	2	Tamaño del archivo de booteo
128-254	n	Reservados para uso específico de esa Site
255	1	Fin de lista

Algunos de estos Tipos pueden obtenerse por medio de otros protocolos (Ej: ICMP, WINS,etc), pero los estándares recomiendan el empleo de estos campos en BOOTP para evitar tráfico de red.

Si se comparan con el formato DHCP se puede apreciar la similitud entre estos, pero en el detalle difieren.

Un detalle significativo de este protocolo es que no provee la información de booteo, le brinda toda la información de red necesaria, e inclusive en el campo *Nombre del archivo de booteo* puede especificar la dirección y el nombre de un servidor TFTP (por ejemplo), y con este dato completar el segundo paso que sería la obtención de los archivos de booteo. Este detalle aunque pase inadvertido es de suma importancia pues se puede desde optimizar el tráfico hasta poseer distintos

sistemas operativos de booteo (Ej: UNIX, Windows 2000, etc) en distintos servidores, y acorde a la solicitud BOOTP cliente, asignarle cada servidor.

Si bien el protocolo BOOTP fue un avance significativo, representa una configuración bastante estática, el administrador de red, crea un serie de parámetros para cada Host en el cliente y el servidor, los cuales permanecerán en los mismos por períodos de tiempo relativamente largos.

A fines de los 90'se hacen reales dos hechos que envejecen prematuramente este protocolo

- *Integración masiva de las distintas redes privadas a Internet.*
- *Empleo cotidiano de computadoras portátiles*.

La explosión de Internet hace que las asignaciones de direcciones IP sean escasas e imponen a muchas subredes rangos más pequeños que la cantidad de host que se poseen, obligando a reducir el tiempo de vida de las asignaciones IP dentro de una subred, para que de esta forma sean compartidas por más de un usuario. El empleo de Notebooks hace que las direcciones asignadas sean diferentes acorde a la subred en la cual se haga presente f'ísicamente. BOOTP no se adapta a estas situaciones pues realiza un mapeo estático. Bajo esta situación es que nace DHCP, tratado al principio.

d. Seguridad (¿Asignación dinámica o estática?):

Si bien el funcionamiento de la asignación dinámica, es un gran apoyo para los administradores de redes, y en realidad facilita notablemente esta tarea, eliminando a su vez las posibilidades de asignaciones repetidas de direcciones IP, lo cual era un problema bastante frecuente en redes grandes; nuevamente aparece esa peligrosa relación de facilidad Vs seguridad. Si se analiza en detalle la solicitud DHCP, se trata de un Broadcast, el cual será recibido por todos los servidores DHCP de la red (inclusive se puede programar cómo se desea operar con el mismo a través de los routers). Todos los servidores contestarán ofreciendo una dirección IP, y la que primero llegue al host solicitante , será la que este negocie en las dos tramas restantes.

Si se tiene en cuenta que en una red Ethernet, **no existe forma de asignar prioridades de acceso al canal**, nunca se podrá definir sectores de la red que sean clientes de ciertos servidores DHCP, como sí se puede hacer con servidores WINS o DNS (pues estos ya son parte de la configuración IP, estática o dinámica), pues al host aún no posee ninguna configuración IP. Este detalle desde el punto de vista de la seguridad, presenta dos grandes problemas:

1) Si se posee más de un servidor DHCP, los rangos de cada uno de ellos, serán adjudicados **aleatoriamente en los distintos host de la red**. Este aspecto no permite planificar grupos de direcciones IP que rápidamente identifiquen la ubicación física de un host en la red.

   Se desea hacer especial hincapié en este párrafo, pues en redes seguras (con IP estáticas bien asignadas), uno de las mayores ayudas que se tiene al realizar análisis de tráfico es que al capturar direcciones IP, inmediatamente se puede inferir de qué host se trata, si es parte de la red, dónde se encuentra el mismo, que funciones desempeña y por lo tanto qué derechos y obligaciones posee. Deténgase a pensar si esto es posible con asignaciones estáticas.

   Dentro de esta línea de pensamiento es que se debe tener muy en cuenta este punto al comenzar a diseñar una red (pues luego se hace muy dificultoso). Es de vital importancia desde el vamos dedicarle todo el tiempo necesario para planificar la estrategia de direccionamiento de la red completa (y si no se hizo desde el principio es una tarea que SI o SI se debe hacer). Un muy buen punto de partida es armar planos de cada subred, dimensionar las subredes, asignarle rangos de IP privadas acorde a la

magnitud de cada una de ellas, luego seguir avanzando de lo global a lo particular, es decir, dentro de cada subred continuar por regiones, edificios, pisos, grupos de trabajo, etc. hasta llegar a identificar al último host.

Ej: **10.65.130.67**

Podría ser interpretado como:

- **10.65** (1ro y 2do octeto): SubRed perteneciente a la **zona A** (Desde 10.65 hasta 10.127, podría ser zona B desde 10.128 hasta 10.191)
- **130** (3er octeto): **Edificio A2 - Piso 2** (Edificio A1 = Subred 1 hasta 127, Edificio A2 = Subred 129 hasta 254, dentro del mismo: piso 1 = 129, Piso 2 = 130, Piso 3= 131,.....).
- **67** (4to octeto): **Grup 1**:Podría asignarse desde 65 hasta 127 = grupo 1 y desde 129 a 191 grupo 2.

  Este ejemplo está basado en REDES REALES de alta seguridad, donde se necesita tener inmediata visualización de toda dirección IP que se monitorice, y por más que parezca difícil la lógica de asignación, cualquier operador que trabaje con una consola en un muy corto período de aprendizaje, al ver pasar cualquiera de estas direcciones **inmediatamente ubica de dónde se encuentra** pues si se sigue una buena lógica, es extremadamente fácil familiarizarse con la red**.**

Esta planificación no implica la obligación de realizar todo con direcciones estáticas, pero si la de hacerlo en los segmentos en los cuales las aplicaciones son críticas, excluyendo estos rangos de direcciones de los servidores DHCP que se hayan instalado.

2) Cualquier host que se conecte físicamente a la red obtendrá una dirección IP y a partir de aquí navegará con total libertad por la red LAN.

3.17. HTTP (HiperText Transfer Protocol) (RFC 1945):

a. Conceptos:

Este es el protocolo empleado entre clientes y servidores Web. La diferencia con los demás protocolos de nivel de aplicación es que este establece una sesión por cada información requerida (texto, sonido, gráficos, etc), esta finaliza al completarse la solicitud. Es normal la apertura de varias sesiones para bajar una sola página.

Desde la versión 1.0 en adelante incorpora MIME (Multimedia Internet Mail Extensions) para soportar la negociación de distintos tipos de datos.

El acceso a este protocolo es por medio del puerto 80 por defecto, pero es común en redes privadas el empleo de otro para incrementar las medidas de seguridad.

b. Solicitudes y respuestas:

Existen dos tipos de encabezado, el de solicitud y el de respuesta y son los siguientes:

Solicitud

Method	Request URI	HTTP version

- Method: Define el método a ser ejecutado sobre el recurso.
- Request URI (Uniform Resource Identifier): Recurso sobre el que se aplica la solicitud.
- HTTP Version: La versión a ser utilizada.

Respuesta

HTTP version	Status Code	Reason phrase

- HTTP Version: La versión a ser utilizada.
- Status code: Se trata de un entero de 3 dígitos que es el resultado de intentar entender y satisfacer la respuesta.
- Reason phrase: Descripción textual del status code.

c. CGI, ISAPI, NSAPI, Servlets y Cold Fusion:

Toda aplicación Web que posea cierto tipo de interacción con el cliente debe acceder a las funciones del servidor. En la actualidad existen dos interfaces que son las más difundidas en el mercado CGI (Common Gateway Interface) e ISAPI (Internet Server Application Programming Interface).

CGI: Es un método estándar de escribir programas para que funcionan en los servidores Web, a estos programas se los suele llamar "Scripts CGI", los cuales por lo general toman sus datos de entrada de formas HTML que les permiten luego ejecutar tareas particulares. Estos programas ofrecen una gran facilidad de desarrollo y como la interfaz con el usuario es HTML, se pueden acceder desde cualquier navegador. Cada llamada a ejecución de un scripts consume tiempo de CPU y recursos del servidor, por esta razón se debe prestar especial atención a la simultaneidad de las mismas.

ISAPI: En los casos donde prive la eficiencia, es una buena alternativa el empleo de esta interfaz, pues a diferencia de la anterior, las aplicaciones que emplean ISAPI, se compilan dentro de archivos DLL del servidor, siendo sensiblemente más eficientes. Estos archivos son el método nativo del ambiente Windows. LA desventaja aquí es que un colapso de DLL puede provocar serios problemas en el servidor.

NSAPI: Es una versión de ISAPI desarrollada por Nestcape, la cual también trabaja con sistemas Unix que soportan objetos compartidos.

Servlets: Se trata de componentes del lado del servidor, que son independientes de las plataformas pues se ejecutan en una máquina virtual jJava (JVM). Por ejecutarse dentro del servidor, no necesitan una interfaz gráfica de usuario, permitiendo una interacción completa entre los mismos (usuario y servidor). En el caso de los servlets de Java, estos ofrecen una solución para generar contenido dinámico, son objetos de programa que pueden cargarse en dinámicamente en los servidores Web, ampliannnndo su funcionalidad, desempeñándose mejor que las CGI. Estos Servlets a su vez son muy seguros y pueden emplearse sobre protocolos de seguridad como SSL.

Cold Fusion: Se trata de un producto que integra navegador, servidor y base de datos en importantes aplicaciones Web. Posee una perfecta integración con HTML, lo que lo convierte en una excelente opferta.

d. Vulnerabilidades:

- Uno de los principales agujeros de IIS es debido a la explotación de ISAPI, si los programas de ISAPI se ejecutan bajo la cuenta IUSR_MACHINENAME y se logra invertir la misma, se heredan los permisos de la misma, a partir de aquí se puede ejecutar cualquier tipo de programas, incluso las llamadas al sistema.
- Autenticación arbitraria de solicitudes remotas.
- Autenticación arbitraria de servidores Web.
- Falta de privacidad de solicitudes y respuestas.
- Abusos de características y recursos del servidor.
- Abuso sobre servidores que explotan sus errores y problemas de seguridad.
- Abuso sobre la información de registro (Robo de direcciones, nombres de dominio, archivos, etc).

3.18. NetBIOS over TCP/IP (RFC 1001 y 1002):

a. Puertos:

En el caso de las redes Microsoft Windows, es común el empleo del protocolo NetBIOS sobre TCP (NetBT), el cual emplea los siguientes puertos:

- UDP port 137 (name services)
- UDP port 138 (datagram services)
- TCP port 139 (session servicNes)

b. Ámbito:

Este protocolo se encuentra estandarizado por las RFC: 1001 y 1002. El acceso a este protocolo lo controla la Interfaz TDI (Transport Driver Interface). Se define una interfaz de software y una convención de nombres, por lo tanto siendo rigurosos, en realidad no se trata de un protocolo en si mismo. El concepto de NetBT nace de las primeras versiones de Microsoft, que implementaban un protocolo llamado NetBEUI, el cual es muy ágil, pero no ruteable, ante lo cual se sustenta en base a Broadcast, este funcionamiento es inaceptable ante redes que por su tamaño comienzan a emplear switch, y por supuesto no lo soportan los routers. Para mantener su estrategia de nombres es que en los entornos Windows de Microsoft se emplea hoy NetBT. Al salir a Internet, este sistema de nombres pierde sentido, pues es reemplazado por DNS, pero en los entornos locales resuelve los servicios de NT workstation, Server Service, Browser, messenger y netlogon. Es por esta razón que su ámbito está acotado a las redes LAN y no tiene significado en Internet.

c. Esquema de nombres:

Este sistema de nombres es plano, es decir que todos los nombres de una red deben ser únicos. Su longitud máxima es de 16 caracteres, quedando el último de ellos reservado para identificar el servicio, este último carácter puede tomar los valores que se detallan a continuación:

<computername>[00h]	Workstation Service
<computername>[03h]	Messenger Service

<computername>[06h]	RAS Server Service
<computername>[1Fh]	NetDDE Service
<computername>[20h]	Server Service
<computername>[21h]	RAS Client Service
<computername>[BEh]	Network Monitor Agent
<computername>[BFh]	Network Monitor Application
<username>[03]	Messenger Service
<domain_name>[1Dh]	Master Browser
<domain_name>[1Bh]	Domain Master Browser

Group Names

<domain_name>[00h]	Domain Name
<domain_name>[1Ch]	Domain Controllers
<domain_name>[1Eh]	Browser Service Elections

d. Protocolo nodo:

El método de resolución de estos nombres respecto de su dirección IP, depende de cómo sea configurado. El sistema de resolución se realiza a través del llamado protocolo "nodo"y ofrece las siguientes opciones:

- nodo-B: emplea Broadcast para resolver el nombre.
- nodo-P: emplea una comunicación Punto a punto.
- nodo-M: emplea primero nodo-B, y si no recibe respuesta usa nodo-P.
- nodo-H: emplea primero nodo-P, y si no recibe respuesta usa nodo-B (caso típico WINS).

Un servidor DNS puede ser configurado con un registro especial (Adicionado puntualmente como una zona DNS), que le instruye para que pase al servidor WINS todo nombre que no encuentre en su base de datos.

e. Vulnerabilidades:

Como se mencionó, este protocolo no tiene sentido en Internet, pues la resolución de nombres en este ámbito se realiza por medio del protocolo DNS, los tres puertos de este protocolo (137,138 y 139) siempre van a estar abiertos en las redes Microsoft, por lo tanto si se logra acceder a una red LAN de estos productos, se sabe cómo poder acceder a cualquier host. Las cuentas de usuario y contraseña de cualquier cliente de la familia Windows 9x, no están pensados para seguridad en red, por lo tanto son altamente violables. Si un intruso logra conectarse desde el exterior con cualquier host de la LAN, rápidamente podría obtener las listas de usuarios de la red, sus servicios, grupos y recursos compartidos, lo cual no es deseado por ningún administrador. En el 16to caracter se pone de manifiesto casi toda la organización de la LAN.

Por esta razón, JAMAS SE DEBE DEJAR PASAR POR UN ROUTER NI FIREWALL, los puertos de este protocolo, **se deben bloquear siempre en la frontera de toda LAN con**

Internet, pues no tiene sentido ninguna comunicación desde adentro hacia fuera o viceversa bajo este protocolo.

3.19. SSL y TLS:

El Secure Socket Layer, (SSL) es un protocolo diseñado originalmente por Netscape Development Corporation para garantizar la seguridad de las transacciones entre sus servidores y clientes en su versión 2.0. A partir de la versión 3.0 se convirtió un estándar utilizado no sólo para el WWW, sino para muchas otras aplicaciones utilizadas en Internet.

Siendo un protocolo que trabaja entre la capa de aplicación y la capa de transporte, permite que las aplicaciones existentes sean fácilmente adaptadas para hacer uso de este protocolo, proporcionando privacidad, integridad y autenticidad en la información transmitida.

Basado en el uso de la criprografía de claves públicas, utiliza los certificados X-509 para el manejo de estas claves y la infraestructura desarrollada en torno de estos certificados.

Actualmente es el estandar de comunicación segura en los navegadores web más importantes (protocolo HTTP), como Nestcape Navigator e Internet Explorer, y se espera que pronto se saquen versiones para otras otros protocolos de la capa de Aplicación (correo, FTP, etc.).

La identidad del servidor web seguro (y a veces también del usuario cliente) se consigue mediante el Certificado Digital correspondiente, del que se comprueba su validez antes de iniciar el intercambio de datos sensibles (Autenticación), mientras que de la seguridad de Integridad de los datos intercambiados se encarga la Firma Digital mediante funciones hash y la comprobación de resúmenes de todos los datos enviados y recibidos.

Desde el punto de vista de su implementación en los modelos de referencia OSI y TCP/IP, SSL se introduce como una "especie de nivel o capa adicional", situada entre la capa de Aplicación y la capa de Transporte, sustituyendo los sockets del sistema operativo, lo que hace que sea independiente de la aplicación que lo utilice.

Este protocolo también puede aplicar algoritmos de compresión a los datos a enviar y fragmentar los bloques de tamaño mayor a 2^{14} bytes, volviéndolos a reensamblar en el receptor.

La versión vieja (2.0) puede ser encontrada en:

http://home.netscape.com/newsref/std/SSL_old.html

a. De SSL a TLS:

A partir de la versión 3.0 de SSL toma participación IETF-TLS (Ineternet Engineering Task Force) Group y la convierte en un estándar de Internet bajo la denominación de TLS (Transport Layer Security) protocol en el año 1996. La información de detalle puede obtenerse en:

http://www.ietf.org/internet-drafts/ draft-ietf-tls-protocol-05.txt

Un tema de especial interés es que TLS fue diseñado especialmente para evitar el ataque de hombre del medio, es por esta razón que presenta mucha dificultad para pasar a través de proxies, pues considera a estos justamente como un ataque.

b. Versiones:

En la actualidad existen la versión SSL 3.0 que es la que se estandariza como TLS 1.0.

c. Handshake:

Un túnel TLS se inicia a través de una conexión normal, por ejemplo, en el caso de ser HTTP, primero se establece esta conexión en texto plano y a través del "Handshake" se crea la conexión segura por medio del intercambio de claves (con el método de Diffie - Hellman o Fortezza , certificados RSA o no, y resúmenes MD-5 o SHA) y la generación del secreto compartido, junto con el código de autenticación de mensaje (MAC).

Hay dos formas de realizar el handshake para iniciar una conexión TLS:

Handshake Completo: Se lleva a cabo el handshake completo para iniciar una conexión, lo cual puede incluir no autenticar las partes, autenticar el server, o autenticar el server y el cliente. Se elige el ciphersuite a usar y se intercambian las claves y secretos.

Handshake Abreviado: Se lleva a cabo el handshake abreviado para reanudar una conexión previa mantenida en el cache del server. Solo se verifican los parámetros del ciphersuite a usar.

Handshake Completo

Esta tabla muestra los diferentes mensajes en la conexión completa:

Cliente	Server
1) Client Hello	
	2) Server Hello
	3) Certificate
	4) Server Exchange
	5) Certificate Request
	6) Server Hello Done
7) Client Certificate	
8) Client Key Exchange	
9) Certificate Veify	
10) Change Cipher Spec	
11) Finished	
	12) Change Cipher Spec
	13) Finished
14) HTTP Request	
	15) HTTP Response
	16) Close Notify Alert
17) Close Notify Alert	

1) Client Hello

Una vez que el cliente (web browser) ingresa una URL **https://.....**, el browser llama al *parser* para que decodifique la URL ingresada. Este se da cuenta que el protocolo elegido es https, e inmediatamente crea un socket al host, al port 443 (el predefinido para HTTP/TLS cuando el protocolo de transporte es TCP).

Una vez creado el socket, el cliente manda un mensaje ClienteHello al server, en el cual van:

- La versión de SSL,TLS (generalmente 3,1).

- Un número Random que servirá luego para verificar integridad. Y crear el secreto compartido.
- Un session-id (inicialmente con valor 0).
- Los cifrados que soporta el cliente.
- Si empleará o no compresión.
- Genera un hash con todo esto para evitar modificaciones.

2) Server Hello

El server al recibir el mensaje anterior, genera este segundo mensaje con los siguientes datos:

- Versión de TLS que soporta el server.
- Número random generado por el server.
- La session-id que el server asigna a esta sesión.
- El algoritmo de cifrado que elige de los propuestos por el cliente.
- Y el algoritmo de compresión que empleará o no

3) Certificate

El servidor genera este mensaje con la lista de certificados que esté usando. Si depende de una CA, le enviará también el de ésta, para que el cliente pueda validarlo.

4) Server Key Exchange

Si no se emplean certificados, se genera este mensaje para transmitir su clave pública, caso contrario no se emite.

5) Certificate Request

Si el server debe autenticar al cliente (si el servicio que presta así lo requiere), genera este mensaje, en el cual se especifica la lista de CAs en los que confía este server, y los tipos de certificados requeridos, ordenados por preferencia del server.

6) Server Hello Done

Una vez generado el mensaje anterior, se arma este mensaje, que indica el fin del Hello del server. El server envía todos los mensajes juntos, desde el **SeverHello** hasta el **ServerHelloDone** en este momento y se queda esperando la respuesta del cliente.

Ahora, estos cuatro mensajes son mandados al cliente en un solo registro (o en varios si el Record Protocol tiene que fragmentarlo). No se genera el MAC, no se hace compresión, ni se encripta el registro, ya que ningún registro **change_cipher_spec** ha sido mandado aún. Los parámetros de seguridad (ciphersuite) elegidos pasan a ser el Estado Pendiente de TLS.

Todos los mensajes se pasan por la función HASH (incluido el **ClientHello** recibido) para poder verificar luego que no hayan sido falsificados.

7) Client Certificate

El cliente ahora procede a autenticar al server. Si el server puede ser autenticado el cliente prosigue con el handshake, de lo contrario se aborta el handshake.
Si se recibe el mensaje **Certificate Request**, el cual le dice al cliente que debe ser

autenticado, el cliente debe responder con una lista de certificados de cliente, para que el server lo pueda autenticar. Por lo tanto se genera un mensaje **Certificate**, similar al enviado por el server.

Los datos del mensaje son:

- Lista de certificados del cliente.

8) Client Key Exchange

Para comenzar a generar el secreto compartido, se genera este mensaje. Se trata aquí de 46 bytes de datos generados aleatoriamente (más 2 bytes de la versión de TLS que usa el cliente), que se envían encriptados con la clave pública del server, la cual se obtuvo del certificado del server.

Antes de mandar el mensaje client_key_exchange, el cliente calcula el secreto previo basado en la clave pública que recibió del server, y luego calcula el "key block", a partir del cual se derivan los MAC Secrets, las session keys y los IVs.

9) Certificate Verify

Para probar que los datos que se han recibido y enviado no han sido ni falsificados ni modificados, se genera este mensaje. Tanto el cliente como el server ingresan todos los mensajes del Handshake Protocol (los recibidos y los enviados desde el comienzo del handshake) en las funciones de hashing.

Para evitar que los valores generados por MD5 y SHA sean falsificados o modificados, se firma digitalmente con la clave privada del cliente (la cual él solo conoce).

Cuando el server reciba este mensaje, debe calcular las hashes MD5 y SHA tal cual lo hizo el cliente, y comparar el resultado calculado con lo que recibió en este mensaje. Si los valores coinciden, entonces se prosigue con el handshake, sino se aborta el handshake.

10) Change Cipher Spec

Este mensaje se utiliza para indicar al server que debe hacer de su estado pendiente (los parámetros de seguridad negociados en este handshake) su estado corriente. Esto significa que los siguientes registros que se envíen desde el cliente al server serán protegidos con el ciphersuite y claves negociados.

11) Finished

Este mensaje se genera para verificar con el server que los parámetros de seguridad fueron correctamente calculados por ambos. Este mensaje consta de 12 bytes generados a partir del secreto previo, y los hashes MD5 y SHA calculados hasta ahora.

En este punto, recién se envían al server todos los mensajes anteriores, desde **Client Certificate** hasta **Finished** inclusive.

El cliente se queda esperando la respuesta del server.

12) Change Cipher Spec

El server recibe la respuesta del cliente.

Primero recibe el mensaje **Certificate** del cliente en el cual viaja la cadena de certificados del cliente. El server autentica al cliente, si puede hacerlo continua con el próximo mensaje recibido del cliente, de lo contrario aborta el handshake.

Suponiendo que pudo autenticar al cliente, procesa el mensaje **ClientKeyExchange**, en el que viaja el secreto previo necesario para generar las claves de sesión, los secretos del MAC y los IVs.
El server desencripta el contenido del mensaje usando su clave privada.

Procesa el mensaje **CertificateVerify**, que contiene los hashes generados por el cliente.

Compara los hashes MD5 y SHA calculados por el server con los enviados por el cliente y si son iguales el server continua con el handshake, sino lo aborta.

Si los hashes coincidieron, el server procesa el mensaje **ChangeCipherSpec** enviado por el cliente, que tiene el efecto de cambiar el estado corriente (sin encriptación ni MAC) por el estado pendiente (encriptación). A partir de este momento, el server enviará y recibirá mensajes protegidos con el ciphersuite negociado.

El server procesa ahora el mensaje **Finished** que envió el cliente. Este mensaje viene protegido con el nuevo ciphersuite, y el cliente lo envía para verificar que los parámetros sean correctos. El server desencripta el mensaje con su clave de lectura, genera el MAC con su secreto MAC de lectura y compara con el del cliente para verificar que el mensaje no ha sido modificado, y genera los 12 bytes random de la misma forma que el cliente, si coinciden, entonces *este sentido* de la conexión ha sido verificado.

Ahora el server genera un mensaje **ChangeCipherSpec**, para que el cliente cambie el estado corriente de lectura con el estado pendiente de lectura, para poder usar el nuevo ciphersuite en este sentido de la conexión.

13) Finished

El server genera 12 bytes random de forma muy similar a como lo hizo el cliente en su mensaje Finished, usando el secreto compartido y las hashes calculadas hasta ahora.

Estos 12 bytes los encapsula en un mensaje **Finished**, genera el MAC con MD5, y encripta todo con la clave de escritura.

Luego envía estos dos últimos mensajes al cliente. En este momento la conexión ya está lista para transportar en forma segura datos de la capa de aplicación, salvo que el cliente envíe un mensaje de Alerta para abortar la conexión.

14) HTTP request

El cliente recibe el mensaje anterior del server, y cambia el estado de lectura corriente con el estado de lectura pendiente.

Ahora, el cliente recibe el mensaje **Finished** del server y chequea que la conexión en este sentido tenga los parámetros de seguridad correctos desencriptando el mensaje, comparando el MAC y chequeando los 12 bytes random generados por el server.

Si no se encontró ningún error, el cliente se dispone a hacer el pedido al server. En caso contrario se aborta la conexión enviando un **mensaje Alert** al server.

No habiendo errores en el handshake, el cliente (Web browser) se dispone ha hacer el pedido al servidor Web.

15) HTTP response

Se envía la información solicitada por el cliente.

El resto de la comunicación es de la misma forma para todos los datos del nivel de aplicación que se quieran transferir.

16) Close Notify Alert

Finalmente, el server manda un **Alert Close Notify** para indicarle al cliente que el server terminó. Por supuesto, este mensaje se envía protegido por el ciphersuite.

Dependiendo del protocolo de aplicación que esté usando la conexión, el servidor podría decidir esperar a que llegue el **Close Notify** del cliente para cerrar el socket TLS, o bién cerrarlo sin esperar el **Close nofity**.

17) Close Notify Alert

El cliente recibe este mensaje del server y responde con otro **Close Notify** para cerrar la sesión. Se envía protegido y se cierra el socket sobyacente.

Handshake Abreviado

En la tabla muestra los diferentes mensajes en la conexión abreviada. **Mensajes para la conexión con handshake abreviado,** el contenido de los mismos es de carácter similar al completo detallado anteriormente**:**

Cliente	Server
1)ClientHello	
	2) ServerHello
	3) ChangeCipherSpec
	4) Finished
5) ChangeCipherSpec	
6) Finished	
Datos de Aplicación <-----> <-----> Datos de Aplicación	

En una forma resumida, el handshake es como sigue:

El cliente envía un **ClientHello** usando el **session-ID** de la sesión a ser reanudada y otros 48 bytes random. El server luego chequea en su caché de sesiones para ver si encuentra esa sesión. Si se encuentra la sesión, y el server está dispuesto a reestablecer la conexión bajo el estado especificado de sesión, mandará un **ServerHello** con el mismo valor de **session-ID** y otros 48 bytes random. Se debe regenerar el secreto compartido y las claves de sesión, los secretos MAC y los IVs. En este punto tanto el cliente como el server deben enviar directamente mensajes **ChangeCipherSpec** y **Finished**. Una vez que el reestablecimiento está completo, el cliente y server empiezan a intercambiar datos de aplicación.

Si el server no puede encontrar el **session-ID** en su cache, el server genera una nuevo **session-ID** y tanto el cliente como el server TLS ejecutan un handshake completo.

d. Intercambio de claves, algoritmos de cifrado y resúmenes:

Sin la intención de exponer en este punto estas técnicas, se trata solamente de incorporar los nuevos aspectos que permite TLS, en virtud de nuevas legislaciones de EEUU, las cuales permiten la exportación de claves de 56 bits e intercambio e claves de hasta 1024 bits. Teniendo en cuenta esta liberación se incorporan las siguientes posibilidades:

Intercambio de Claves	Clave Intercambio	Cipher	Clave Cipher	Exportable	Hash
RSA	1024 bits	DES (CBC)	56 bits	SI	SHA

RSA	1024 bits	RC4	56 bits	SI	SHA
Diffie-Hellman (efimera, firma DSS)	1024 bits	DES (CBC)	56 bits	SI	SHA
Diffie-Hellman (efimera, firma DSS)	1024 bits	RC4	56 bits	SI	SHA
Diffie-Hellman (efimera, firma DSS)	sin límite	RC4	128 bits	NO	SHA

También existe documentación para el agregado a TLS del Elliptic Curve Cryptosystem (ECC). Que en la actualidad se considera como un algoritmo muy robusto y más veloz que RSA.
Para la implementación de Curvas elípticas se define toda la cipher suite, empleando los siguientes algoritmos de establecimiento de clave: ECES (Elliptic Curve Encryption Scheme), ECDSA (Elliptic Curve Digital Signature Algorithm), ECNRA (Elliptic Curve Nyberg-Rueppel Signature Scheme with Appendix), ECDH (Elliptic Curve Diffie-Hellman Key Agreement), ECMQV (Elliptic Curve Menezes-Qu-Vanstone Key Agreement).
También se contempla la incorporación de Kerberos. Las credenciales Kerberos se usan para llevar a cabo una autenticación mutua y para establecer un secreto compartido usado subsecuentemente para asegurar la comunicación cliente-servidor.

e. Puertos definidos:

Teóricamente TLS puede asegurar cualquier protocolo de la familia TCP/IP ejecutándose sobre todo puerto, si ambos lados conocen que en el otro extremo se está ejecutando TLS, sin embargo, en la práctica un grupo de puertos han sido reservados para cada uno de los protocolos comúnmente empleados en Internet, facilitando con esto la tarea a los firewalls. En el año 1998, IANA designó los siguientes puertos para SSL/TLS:

Keyword	Decimal	Description
Nsiiops	261/tcp	IIOP Name Service over TLS/SSL
Https	443/tcp	http protocol over TLS/SSL
Ddm-ssl	448/tcp	DDM-SSL
Smtps	465/tcp	smtp protocol over TLS/SSL
Nntps	563/tcp	nntp protocol over TLS/SSL
Sshell	614/tcp	SSLshell
Ldaps	636/tcp	ldap protocol over TLS/SSL
ftps-data	989/tcp	ftp protocol, data, over TLS/SSL
ftps	990/tcp	Ftp, control, over TLS/SSL
Telnets	992/tcp	telnet protocol over TLS/SSL
Imaps	993/tcp	imap4 protocol over TLS/SSL
Ircs	994/tcp	irc protocol over TLS/SSL
Pop3s	995/tcp	pop3 protocol over TLS/SSL

La lista de puertos y sus detalles puede ser encontrada en:
http://www.isi.edu/in-notes/iana/assignments/port-numbers

3.20. IP Versión 6 (IP Next generation):

a. Conceptos:

Desde hace tiempo ya se hacen evidentes algunas falencias que hoy tiene la actual versión del Protocolo IP (Versión 4). Algunas de ellas son la mala distribución que utiliza de sus cantidades de Host en cada una de sus redes (A, B y C); son muy pocas o ninguna las empresas que poseen los millones de Host que permite una dirección tipo A, hoy tampoco existen

2.100.000 empresas, por lo tanto, tanto en A como en C se desperdicia la asignación de direcciones, si bien hoy se asignan porciones de las mismas, este no fue el sentido con que fueron creadas. Otra debilidad es la no posibilidad asignaciones geográficas , lo que representa una enorme carga de tablas en los router exteriores, casi ya imposible de controlar. También se suman detalles de seguridad, criptogrfiado, Prioridades, longitud variable de cabecera, etc.

b. Características:

Las características principales de IPv6 son:

- Direccionamiento: 128 bit.
- Encaminamiento: Direccionamiento jerárquico.
- Prestaciones: Cabecera simple de 40 Byte, alineada de a 64 bit, y cualquier otra información se agrega como cabecera en extensión (opcional).
- Versatilidad: Formato flexible de opciones.
- Multimedia: Id de flujos.
- Multicast: Obligatorio.
- Seguridad: Soporte de autenticación y cifrado.
- Autoconfiguración: Tres métodos PnP.
- Movilidad: Surce routing, seguridad, detección de móviles, hand-off.
- Fragmentación: Únicamente de extremo a extremo, es decir que sólo el origen puede fragmentar. Para implementar esto, hace uso de PMTU (Path MTU, RFC 1191), que es el mecanismo empleado para determinar la máxima unidad de datos que contendrá un datagrama, una vez conocido este tamaño, armara todos los paquetes sin superar el mismo, por ende ningún router deberá fragmentarlo pues no será necesario.
- Tamaño de datagrama: Mantiene el mismo concepto que la versión 4 y propone un nuevo modelo de datagrama, llamado Jumbograma, el cual se define a través de una cabecera en extensión, y permite transmitir datagramas de hasta 4 Gbyte. La idea de esta nueva aplicación es permitir la transmisión de grandes volúmenes de datos entre servidores, los cuales no necesitan incrementar con tanta redundancia de cabecera, siendo el mejor representante de esto el empleo de cluster de servidores.

c. Header de IPv6:

<table>
<tr><td>Versión</td><td>Clase de tráf.</td><td colspan="2">Rótulo de flujo</td></tr>
<tr><td colspan="2">Longitud de carga útil</td><td>Sig. Cabecera</td><td>Límite Saltos</td></tr>
<tr><td colspan="4">Dirección fuente</td></tr>
<tr><td colspan="4">Dirección destino</td></tr>
<tr><td colspan="4">Posibles cabeceras de extensión</td></tr>
</table>

- Versión: (4), se mantiene el mismo tamaño para permitir distinguirlo del versión 4 y que puedan convivir durante algún lapso de tiempo.
- Clase de tráfico (4): (Video, audio, datos, voz, etc).Cuanto más alto sea su valor más importante es, los valores que puede adoptar son:

0 tráfico sin caracterizar
1 tráfico "filler"
2 transferencia de datos no atendida, como E-mail
3 reservado
4 transferencia de bloques de datos atendida, como FTP
5 reservado
6 tráfico interactivo, como TELNET
7 tráfico de control de Internet, como protocolos de encaminamiento

- Rótulo de flujo: (24), Todos los datagramas del mismo flujo (Ej: todos los datagramas de una misma FTP).
- Longitud de carga útil: (16): Cantidad de octetos de datos.
- Siguiente cabecera: (8), se permiten varias, todas ellas van después del campo Dirección destino y aquí se identifican cuales van.
- Límite de saltos: (8), para evitar lazos infinitos.
- Dirección origen y destino (128 c/u), aparece aquí aparte de Net y Host un nuevo identificador llamado Dirección de Agrupación, que identifica regiones topológicas.
- Posibles cabeceras de extensión (Extension Headers):
  Irán colocadas antes del campo de datos, cada cabecera tendrá un primer campo (8 bit) que indica la próxima cabecera (Next Header) que indica si existe otra cabecera de extensión o si esta es la última
  - Cabecera salto por salto (valor 0): Lleva información para analizar en cada router.
  - Cabecera extremo a extremo: Lleva información que solo se examinará en el destino.
  - Cabecera de enrutamiento (valor 43): Ruta fija.
  - Cabecera de fragmento (valor 44): Si existe fragmentación.
  - Cabecera de verificación de autenticidad(valor 51): Permite verificar autenticidad de origen.
  - Cabecera de confidencialidad: Los datos no deben ser leídos durante su paso por Internet.

d. Direccionamiento de IPv6:

- Direcciones de 128 bit (16 octetos) (más de 10^{38} direcciones posibles).
- A pesar de las restricciones de redes y reservadas aún quedan más de 1.500 direcciones por m^2 de la superficie de la tierra.
- Tres tipos de direcciones (unicast, anycast y multicast).
- No existen clases, similar al concepto de CIDR.

Notación general 3FFE:2213:AE56:54AD:34EF:9888:33EA:AA21
Los ceros contiguos se pueden eliminar, es decir los siguientes pares de cotetos se podrían representar como están indicados a su derecha:

:002E: → :2E:
:000A: → :A:
:6700: → :6700:
:0004:0000:0000:0000:000A: → :4::A: (Sólo una vez se pueden resumir las secuencias seguidas de ceros, con dos puntos seguidos ::).
0004:0000:0000:0000:000A:0000:0000:1243:00AD: → 4::A:0000:0000:1243:AD:

Las direcciones compatibles con Ipv4 se abrevian con un solo punto (en vez de doble), o cual indica que se trata de un solo octeto y no dos:

0:0:0:0:0:FFFF:201.200.32.129 → ::FFFF:201.200.32.129

e. Tipos de direcciones:

Se definen tres tipos de direcciones IPv6:

Compatibles con IPv4

Una dirección indicando un nodo IPv6 con una dirección que se puede mapear unívocamente al espacio IPv4. Tienen el prefijo IP 0:0:0:0:0:ffff. Por ejemplo, 0:0:0:0:0:FFFF:119.234.21.44

Mapeadas a IPv4

Una dirección IPv6 que indica un nodo sólo IPv4. Tienen el prefijo IP 0:0:0:0:0:0. Por ejemplo, 0:0:0:0:0:0:36.56.24.241.. Es importante darse cuenta de que las direcciones compatibles con IPv4 y las mapeadas a IPv4 utilizan el mismo espacio de direcciones. El prefijo sólo indica si el nodo soporta o no IPv6.

Sólo IPv6

Una dirección IPv6 que indica un nodo que soporta IPv6 donde los 32 bits inferiores no contienen necesariamente una dirección IPv4. Los 96 bits de orden superior son distintos de 0:0:0:0:0:FFFF o 0:0:0:0:0:0.

Direcciones especiales:

0:0:0:0:0:0:0:1 → Loopback.
0:0:0:0:0:0:0:0 → Dirección no especificada.

Concepto de prefijo: El prefijo es similar a la notación de cisco de máscara de red, es decir se anexa a continuación de la dirección IP, separado por una barra (/) en notación decimal el valor que identifica la cantidad de bit que están puestos a uno en la máscara de red (de izquierda a derecha):

0004:0000:0000:0000:000A:0000:0000:1243:00AD/48

RFC que hacen referencia a IPv6

1881 IPv6 Address Allocation Management.
1883 Internet Protocol, Version 6 (IPv6) Specification.
1884 IP Version 6 Addressing Architecture.
1887 An Architecture for IPv6 Unicast Address Allocation.
1897 IPv6 Testing Address Allocation.
1924 A Compact Representation of IPv6 Addresses.
1933 Transition Mechanisms for IPv6 Hosts and Routers.
1970 Neighbor Discovery for IP Version 6 (IPv6).
1971 IPv6 Stateless Address Autoconfiguration.
1972 A Method for the Transmission of IPv6 Packets over Ethernet Networks.
2073 An IPv6 Provider-Based Unicast Address Format.
2147 TCP and UDP over IPv6 Jumbograms.
2374 An IPv6 Aggregatable Global Unicast Address Format.
2375 IPv6 Multicast Address Assignments.
2460 Internet Protocol, Version 6 (IPv6) Specification.

2461 Neighbor Discovery for IP Version 6
2462 IPv6 Stateless Address Autoconfiguration.
2471 IPv6 Testing Address Allocation.
2473 Generic Packet Tunneling in IPv6 Specification.
2474 Definition of the Differentiated Services Field in the IPv4 and IPv6 Headers.
2491 IPv6 over Non-Broadcast Multiple Access (NBMA) networks
2526 Reserved IPv6 Subnet Anycast Addresses.
2675 IPv6 Jumbograms.
2732 Format for Literal IPv6 Addresses in URL's.
2893 Transition Mechanisms for IPv6 Hosts and Routers.
2928 Initial IPv6 Sub-. TLA ID Assignments.

ANEXO: RFC RELACIONADAS CON TCP/IP.

768	**User Datagram Protocol (UDP)**
783	**Trivial File Transfer Protocol (TFTP)**
791	**Internet Protocol (IP)**
792	**Internet Control Message Protocol (ICMP)**
793	**Transmission Control Protocol (TCP)**
816	**Fault Isolation and Recovery**
826	**Address Resolution Protocol (ARP)**
854	**Telnet Protocol (TELNET)**
862	**Echo Protocol (ECHO)**
863	**Discard Protocol (DISCARD)**
864	**Character Generator Protocol (CHARGEN)**
865	**Quote of the Day Protocol (QUOTE)**
867	**Daytime Protocol (DAYTIME)**
894	**IP over Ethernet**
919, 922	**IP Broadcast Datagrams (broadcasting with subnets)**
950	**Internet Standard Subnetting Procedure**
959	**File Transfer Protocol (FTP)**
1001, 1002	**NetBIOS Service Protocols**
1009	**Requirements for Internet Gateways**
1034, 1035	**Domain Name System (DNS)**
1042	**IP over Token Ring**
1055	**Transmission of IP over Serial Lines (IP-SLIP)**
1112	**Internet Gateway Multicast Protocol (IGMP)**
1122, 1123	**Host Requirements (communications and applications)**
1134	**Point-to-Point Protocol (PPP)**

1144	Compressing TCP/IP Headers for Low-Speed Serial Links
1157	Simple Network Management Protocol (SNMP)
1179	Line Printer Daemon Protocol
1188	IP over FDDI
1191	Path MTU Discovery
1201	IP over ARCNET
1231	IEEE 802.5 Token Ring MIB (MIB-II)
1332	PPP Internet Protocol Control Protocol (IPCP)
1334	PPP Authentication Protocols
1518	An Architecture for IP Address Allocation with CIDR
1519	Classless Inter-Domain Routing (CIDR): An Address Assignment and Aggregation Strategy
1533	DHCP Options and BOOTP Vendor Extensions[i]
1534	Interoperation Between DHCP and BOOTP
1541	Dynamic Host Configuration Protocol (DHCP)
1542	Clarifications and Extensions for the Bootstrap Protocol
1547	Requirements for Point-to-Point Protocol (PPP)
1548	Point-to-Point Protocol (PPP)
1549	PPP in High-level Data Link Control (HDLC) Framing
1552	PPP Internetwork Packet Exchange Control Protocol (IPXCP)

i