****

UNIVERSIDAD NACIONAL DE LOMAS DE ZAMORA

FACULTAD DE INGENIERIA

Tecnicatura en Programación

Trabajo práctico N°2 (grupal)

**Materia:** Seguridad informática

**Alumnos:** Federico Tito, Maximiliano De Roo,   
Maximiliano Garrett, Roberto Frediani

**Profesor:** Bruno Diaz

**Trabajo practico N°2 – Investigación Grupal**

**Consigna:**

* **Investigar sobre:**
* Modelo OSI
* Números binarios, decimales y hexadecimales
* IRQ
* Puertos lógicos y Servicios vinculados
* Definición de IPV4, IPV6, rangos públicos y privados.

**Modelo OSI:**

Es una normativa formada por siete capas que define las diferentes fases por las que deben pasar los datos para viajar de un dispositivo a otro sobre una red de comunicaciones. En este estándar no se define una implementación de una arquitectura de red, sino que se establece un modelo sobre el cual comparar otras arquitecturas y protocolos.

El modelo OSI establece una arquitectura jerárquica estructurada en 7 capas, cada capa realiza servicios para la capa superior, a la que devuelve los resultados obtenidos, y a su vez demanda servicios a la capa inferior.

**Capa 1 (capa física):** Proporciona los medios de transportes para los bits, los medios básicos de transportes para representar los datos son tres: cable de cobre, fibra, inalámbrico. Para el medio de transporte de cable de cobre las señales son patrones de pulsos eléctricos, para la fibra las señales son patrones de luz, y para los medios inalámbricos son señales de ondas de radio.

Las tres funciones esenciales de la capa física son:

* Los componentes físicos: Son los dispositivos electrónicos de hardware, medios y conectores que trasmites y transportan las señales para representar los bits.
* La codificación de datos: La codificación es un método utilizado para convertir una corriente (stream) de bits en un código predefinido. Los códigos son grupos de bits para ofrecer un patrón predecible que pueda reconocer tanto el emisor como el receptor. Los patrones predecibles permiten distinguir los bits de datos de los de control para mejorar a la hora de detectar un error en los medios de transporte de estos bits.
* Señalización: La capa física debe generar señales inalámbricas, ópticas o eléctricas que se representan por el “1” y “0”. El método de representación de bits se denomina método de señalización. En esta capa física se debe definir qué tipo de señal representa un “1” “0” como un cambio en el nivel de una señal eléctrica o un impulso óptico u otro método de señalización.

La capa física también se ocupa de la configuración de:

1. línea punto a punto: La redes punto a punto se utiliza para comunicar únicamente a dos nodos que trabajan como socios, es decir, los dispositivos en red pueden tomar el rol tanto de emisor como receptor por ejemplo un dispositivo “A “hace una petición de información al dispositivo “B”, el dispositivo “A” hace de receptor y “B” de emisor o viceversa o sea que “B” hace una petición de información al dispositivo “A” ahora “B” hace de receptor ya que está esperando una respuesta y “A” de emisor ya que es el que va enviar esa información.
2. Punto a multipunto: Esta configuración ofrece varias rutas o caminos desde una única ubicación a varios lugares por ejemplo al enviar un correo es decir el que lo envía es el receptor y sale de una ubicación específica, a varios destinos o correos que son los receptores. o pude ser el caso de una teleconferencia donde uno solo es el que transmite (emisor) a varios o múltiples destinos que son los que están escuchando esa conferencia (receptor/es). Hay diferentes tipos de conexiones punto a multipunto como:

* Estrella: Un host conectados a varias terminales remotas.
* Bus: Un medio de comunicación común conectados a muchas estaciones remotas.
* Anillo: Todas las terminales conectadas a un mismo cable y si una falla hay problemas con todas.
* Malla: Es el tipo de conexión que se utiliza en las centrales telefónicas. Todas las terminales conectadas entre sí.

1. Comunicación serie: En este proceso se envían datos de un bit a la vez de forma secuencial sobre un bus o canal de comunicación.
2. Comunicación en paralelo: todos los bits se envían al mismo tiempo y por eso debe haber tantas líneas de comunicación como bits tenga la información a transmitir.

**Capa 2 (De enlace de datos):** Actúa de intermediario entre la capa de red y la capa física codificando las tramas recibidas desde la capa de red para luego ser transmitida desde la capa física. Esta capa organiza los unos y ceros en formatos lógicos para detectar errores en la capa física, establecer el método de acceso que las computadoras van a seguir para poder transmitir y recibir información, realizar la transferencia de datos a través del nivel físico, etc.

La capa de enlace también prepara los datos para ser colocados en el medio encapsulando el paquete de la capa 3 en una trama. Una trama tiene un encabezado y una información final que incluye las direcciones de enlace de datos de origen y de destino, calidad de servicio tipos de protocolos. La trama de esta capa incluye:

* Datos: Información del paquete desde la capa de red.
* Encabezado: Contiene información de control como direccionamiento (dirección MAC) y está ubicado al comienzo del PDU (unidad de dato de protocolo).
* Tráiler: Contiene información de control agregado al final del PDU.

**Subcapas de enlace de datos:**

* Control de enlace lógico: El control de enlace lógico (LLC) coloca información en la trama para identificar que protocolo de capa de red está siendo utilizada por la trama. Esta información permite que varios protocolos de la capa 3 (red) como la IP utilicen la misma interfaz de red y los mismos medios.
* Control de acceso al medio: El control de acceso al medio (MAC) proporciona a la capa de enlace de datos el direccionamiento y la delimitación de datos de acuerdo con los requisitos de señalización física del medio y al tipo de protocolo de enlace de dato en uso. Otras funciones que cumple son:
  + - Ensamblado de datos en tramas con campos de direccionamiento y detección de errores.
    - Desensamblado de tramas.
    - Control de acceso al medio de transmisión LAN.
    - Agregar la dirección MAC del nodo origen y el de destino

**Protocolos de la capa de enlace de datos:**

* PPP: Protocolo punto a punto que permite establecer una comunicación a nivel de la capa de enlace TCP/IP (protocolo de control de transmisión /protocolo de internet este hace posibles servicios como ftp, e-mail, etc. entre ordenadores que no están conectados a una misma red) entre dos computadoras. Generalmente se utiliza para establecer conexión a internet de un particular con su proveedor a través de un modem.
* Ethernet: Define el uso de los dos niveles inferiores (capa física y de enlace de datos), especifica sus normas de funcionamiento en una WLAN (Red de área local inalámbrica red informática formada por unidades ubicada en un espacio geográfico reducido y no usa medios físicos para su interconexión).
* HDLC: Es un estándar a nivel de enlace de dato que incluyen mecanismos para la detección y corrección de errores.
* Frame Relay: Es una técnica de comunicación mediante retransmisión de tramas para redes de circuitos virtuales. Consiste en una forma simplificada de comunicación de paquetes que transmiten una variedad de tamaños de tramas que es perfecto para la retransmisión de grandes cantidades de datos.

**Protocolo de ventana deslizante**

Este protocolo permite al emisor transmitir múltiples segmentos de información antes de comenzar la espera para que el receptor le confirme la recepción de los segmentos recibidos, esta confirmación se denomina ACK o acuse de recibo. Estos ACK son enviados del receptor al emisor ya que el receptor es el encargado de informar al emisor si ha recibido bien los segmentos enviados. En ese ACK también se envía el número de la siguiente trama que espera recibir. Con este ACK el emisor puede saber que envíos se realizaron con éxitos, los que fallaron y los que se esperan recibir.

**Control de flujo**

El control de flujo es necesario para al receptor de uno a varios emisores, realiza dos acciones muy importantes:

* Detección de errores: Se utiliza para detectar errores a la hora de enviar tramas al receptor e intentar solucionarlos y se realice mediante diferentes tipos de código como el CRC (verificación de redundancia cíclica que sirve para detectar errores usado frecuentemente en redes digitales), simple paridad (puede ser par, números de “1” par, o impar), paridad cruzada (paridad horizontal y vertical) y suma de verificación (tiene como propósito principal detectar cambios accidentales en una secuencia de datos).
* Corrección de errores: Surge a partir de la detección de errores Y necesita añadir a la información útil un número de bits redundantes (repetición de los bits) bastante superior al necesario para detectar y retransmitir. Para detectar los errores cada bit se repite tres veces y en caso de fallo se toma el bit que más se repite.

**Capa 3 (red):** Provee servicios para intercambiar secciones de datos individuales a través de la red entre dispositivos finales identificados. Para realizar este transporte de extremo a extremo se utiliza cuatro procesos básicos:

* Direccionamiento: Debe proveer un mecanismo para direccionar estos dispositivos finales. Si las secciones individuales de datos deben dirigirse a un dispositivo final, este dispositivo debe tener una dirección única. En una red IPv4, cuando se agrega esta dirección a un dispositivo, al dispositivo se lo denomina host.
* Encapsulación: Debe proveer encapsulación, los dispositivos no deben ser identificados solo con una dirección. Durante el proceso de encapsulación esta capa recibe la PDU (unidad de protocolo de dato se utiliza para el intercambio de entre unidades dentro de una capa del modelo OSI) de la capa 4 y agrega un encabezado de capa 3 para crear el PDU de la capa 3. Cuando nos referimos a la capa de red denominamos paquete a esta PDU. Cuando se crea un paquete el encabezado debe tener la información como la dirección del host la que se está enviando (dirección de destino), la dirección del host origen. Después de que la Capa de red completa el proceso de encapsulación, el paquete es enviado a la capa de enlace de datos que ha de prepararse para el transporte a través de los medios.
* Enrutamiento: Luego esta capa debe proveer los servicios para dirigir estos paquetes a su host destino. El host de origen y destino no siempre están conectados a una misma red. Cada paquete debe ser guiado a través de la red para llegar a su destino final. Los encargados de hacer que estos paquetes lleguen a destino son los routers y que su función es seleccionar una ruta y dirigirlos hacia su destino este proceso se denomina enrutamiento.
* Desencapsulamiento: El paquete llega al host destino y es procesado en la Capa 3. El host examina la dirección de destino para verificar que el paquete fue direccionado a ese dispositivo. Si la dirección es correcta, el paquete es desencapsulado por la capa de Red.

**Protocolo de capa de red**

Los protocolos implementados en la capa de red son:

* ARP (Protocolo de Resolución de direcciones): primer protocolo a nivel de red. Convierte dinámicamente las direcciones de internet en las direcciones de hardware exclusivas de las redes de área local. Es decir, recibe una dirección IP de destino y le envía una petición a todas las maquinas buscando la dirección MAC del terminal destino.
* ICMP (Protocolo de mensaje de control de internet): el segundo protocolo a nivel de red. Es una parte necesaria de cada implementación IP. Este maneja los mensajes de error y control para IP. Es decir, manda mensajes o notifica a los protocolos de la capa más cercana
* Versión 4 de protocolo de internet (IPV4): Este protocolo es uno de los protocolos fundamentales de internet, ya que es el que identifica los diferentes dispositivos conectado a la red. Una dirección IPV4 tiene 32 bits, por lo que tenemos casi 4.300 millones de direcciones únicas, aunque muchas de ellas están reservadas para tareas específicas, como por ejemplo los rangos de direccionamiento de IP privados (LAN) que no son enrutables a través de internet. Una dirección de IPV4 se representa en formato decimal separado por puntos. Los 32 bits están subdivididos en octetos de 8 bits cada uno.
* Versión 6 de protocolo de internet (IPV6): Este protocolo fue diseñada para para remplazar a IPV4 que tiene límites en el número de direcciones por lo que impide el crecimiento de la red. Funciona de manera similar a IPv4 en que proporciona las direcciones IP únicas y numéricas necesarias para que los dispositivos habilitados para Internet se comuniquen. Sin embargo, tiene una gran diferencia: utiliza direcciones de 128 bits que da 370 sextillones de direcciones.
* Las principales características de este protocolo son:
* Infraestructura de direcciones y enrutamiento eficaz.
* Mejora de compatibilidad para calidad de servicio.
* Multicast (envío de un mismo paquete a un grupo de receptores).
* Anycast (envío de paquetes a un receptor dentro de un grupo).
* Velocidad.
* Calidad de servicio.
* OSPF: Es un protocolo de red para encaminamiento jerárquico y calcula la ruta idónea entre dos nodos.

**Capa 4 (Transporte):** Permite la segmentación de datos y brinda el control necesario para reensamblar las partes dentro de los distintos Stream comunicación.   
Las responsabilidades que debe cumplir son:

* Seguimiento de la comunicación individual entre aplicaciones en los hosts de origen y destino.
* Segmentación de datos y gestión de cada segmento.
* Reensamble de segmentos.
* Identificación de las diferentes aplicaciones.

Las funciones principales especificadas por todos los protocolos de la capa de transporte incluyen:

* Segmentación y reensamblaje: La mayoría de las redes poseen una limitación en cuanto a la cantidad de datos que pude incluirse en una única PDU (Unidad de Datos de Protocolo) esta capa divide los datos en bloques de datos de un tamaño adecuado. En el destino la capa de transporte reensambla los datos antes de enviarlos a la aplicación.
* Multiplexación de conversaciones: Pueden existir varias aplicaciones o servicios ejecutándose en cada host de la red. A cada una de estas aplicaciones o servicios se le asigna una dirección conocida como puerto para que la capa de trasporte pueda identificar a que aplicación o servicio corresponden esos datos.

Además de utilizar la información contenida en los encabezados para las funciones básicas de segmentación de datos también esta capa provee de otros protocolos como:

* Entrega confiable
* Reconstrucción ordena de datos
* Control de flujo.

Un protocolo de la capa de transporte puede implementar un método para asegurar la entrega confiable de los datos, es decir, asegurar que cada sección de dato que envía el origen llegue a destino a través del seguimiento de datos transmitidos, acuse de recibo de los datos recibidos y retransmisión de datos sin acuse de recibo.

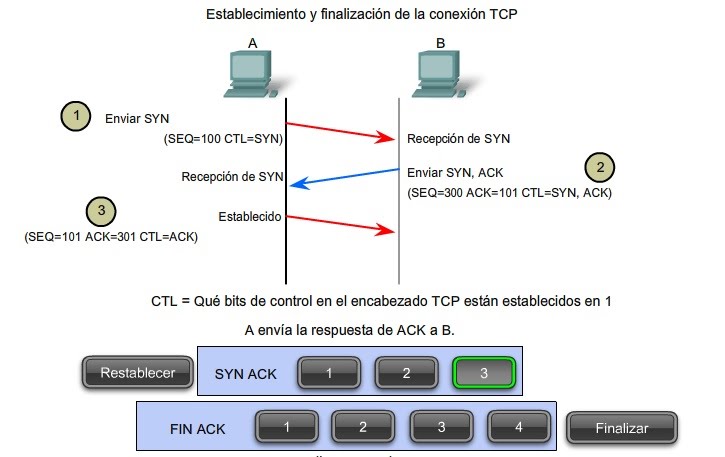
**Protocolo TCP y UDP**

Los dos protocolos más comunes de la capa de transporte son el de Control de Transmisión (TCP) y el protocolo de datagrama de usuario (UDP).

* UDP: Es un protocolo no orientado a conexión. Es decir, cuando una maquina “A” envía paquetes a una maquina “B” el flujo es unidireccional (una sola dirección). La transferencia de datos se realiza si haber realizado previamente una conexión con la máquina de destino “B” y el destinatario recibirá los datos sin enviar una confirmación al emisor “A”. es más rápida pero menos confiable ya que si se pierden datos no se volverá a transmitir ya que el receptor no envía ningún ACK, y el destinatario no conocerá al emisor de los datos excepto su IP.
* TCP: El protocolo TCP está orientado a conexión. Cuando una maquina “A” o emisor envía los datos a una maquina “B” o receptor, el receptor es informado de la llegada de datos y confirma su buena recepción con un ACK. También este protocolo garantiza que los datos serán entregados a su destino sin errores y en el mismo orden en que se transmitieron.

Esta capa también asigna un número de secuencia a los segmentos para que el host receptor los una en orden en que fueron transmitidos.

**protocolo TCP de enlace de tres vías**

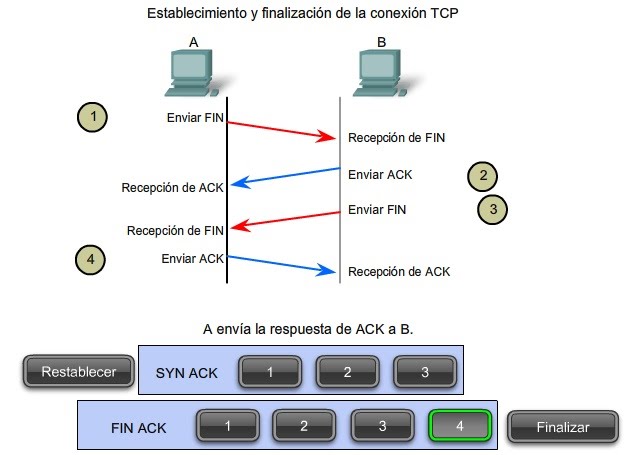


el protocolo de enlace de tres vías es un proceso por el cual se debe pasar antes de establecer y antes de finalizar una conexión y para establecerla se realiza una serie de pasos:

paso 1: el primer paso el cliente envía un segmento con el señalizador de control SYN (sincronizar números de secuencias) que con él se empieza a establecer una conexión a través de algún puerto en el servidor. Además el cliente, o receptor, en el campo de numero de secuencia (ISN) genera un número aleatorio para empezar a sincronizar con el servidor o sea rastrea el flujo de dato. El número que genera por ejemplo es el 100.

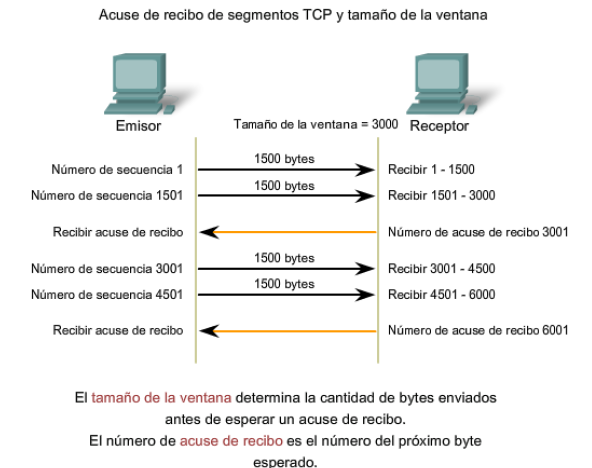
Paso 2: el servidor recibe el segmento TCP con el valor (SYN) y el número de secuencia. Si el puerto por el que se intenta realizar la conexión está disponible, el servidor envía un segmento TCP con el valor SYN y además el valor ACK. El ACK es el acuse de recibo o la confirmación de que llego el segmento de TCP, y en este se establece el número siguiente del que se estableció en el cliente que, si fue 100 en el anterior, en el servidor el acuse de recibo (ACK) establecerá el número 101. De esta manera el cliente entiende que el servidor recibió correctamente los datos, pero el servidor también genera un número de secuencia por ejemplo el 300.

Paso 3: el cliente recibe el segmento de TCP con los valores SYN y ACK entonces envía como respuesta otro segmento de TCP con el número de secuencia 101 y el ACK 301 y así termina el enlace de tres vías.



Cuando el cliente no tiene el cliente no tiene más datos para enviar entonces envía un segmento con el valor FIN, el servidor responde con otro segmento de valor ACK y otro con el valor FIN establecido y el cliente responde con otro valor ACK.

**tamaño de ventana**



El tamaño de ventana determina la cantidad de bytes que se pueden enviar para recibir un reconocimiento. El número de reconocimiento es el número del siguiente byte esperado o sea que el tamaño de ventana es la cantidad de byte que el dispositivo de destino de una sesión de TCP puede aceptar y procesar al mismo tiempo. El tamaño inicial de la ventana se acuerda durante el inicio entre emisor y receptor mediante protocolo de enlace de tres vías. Por ejemplo, en este caso el tamaño de ventana será de 3000 bytes y el TMS (tamaño máximo de segmento) será de 1500 byte. Entonces el emisor enviara una trama de 1500 bytes y a la vez envía otros 1500 bytes estableciendo un tamaño de venta de 3000 bytes que serán confirmados por el receptor y enviándole un ACK diciéndole lo siguiente que quiere que le envíe que será el byts 3001 o sea que pudo recibir los primeros 3000 bites y le manda el número siguiente que espera recibir, es decir 3001, para pedirle que envíe otros 3000 bytes. En caso de que ocurra una saturación después de que el emisor enviara 1500 bytes y otros 1500 el receptor le dirá a través de un ACK que baje el tamaño ya que supongamos que se perdió el segundo envío que es el byte 4501 entonces ese ACK contendrá el valor 4501 y el emisor enviara devuelta esa trama reduciendo el tamaño de ventana de 3000 bytes a 1500.

**Direccionamiento del puerto**

Los servidores basados en TCP y UDP mantienen un seguimiento de las varias aplicaciones que se comunican. Para diferenciar los segmentos para cada aplicación TCP o UDP cuenta con encabezados que pueden identificar estas aplicaciones.   
Estos identificadores se denominan puertos. Por ejemplo, cuando una aplicación de exploración web realiza una solicitud a un servidor web el explorador utiliza TCP y el número de puerto 80. Esto sucede porque el puerto TCP 80 es el puerto asignado a aplicaciones de servidores web por defecto.

**Capa 5 (Sesión):** Esta capa es la que se encarga de mantener y controlar el enlace establecido entre dos ordenadores que están transmitiendo datos. El servicio provisto por esta capa es la capacidad de asegurar que, dada una sesión establecida entre dos máquinas, la misma se pueda efectuar para las operaciones definidas de principio a fin, reanudándolas en caso de interrupción.

**Servicios que proporciona**

* Control de dialogo: Este pude ser simultaneo en los dos sentidos (full-duplex) o alternando en ambos sentidos (half-duplex).
* Agrupamiento: El flujo de datos se puede marcar para definir grupos de datos.
* Recuperación: La capa de sesión puede proporcionar un procedimiento de punto de comprobación, de forma que, si ocurre algún tipo de fallo entre puntos de comprobación, esta capa puede retrasmitir todos los datos desde el último punto de comprobación y no desde el principio.

**Esta capa proporciona tres protocolos**

* SQL: Es una esencialmente una instancia de SQL Server que se expone en internet, funcionando desde los servidores de Microsoft.
* RCP: Es un protocolo que permite a un programa de ordenador ejecutar código en otra máquina remota sin tener que preocuparse por las comunicaciones entre ambos. Este protocolo es un gran avance sobre los sockets.
* NETBIOS: Es una especificación de interfaz para acceso a servicios de red
* SCP: Básicamente idéntico al protocolo RCP, la única diferencia es que los datos son cifrados durante su transferencia.

**Capa 6 (presentación):** Tiene como función básica encargarse del formato en que se va a mostrar la información. Establece el contexto entre elementos de nivel de aplicación, determinando la sintaxis y la semántica para representar la información.

**Funcionamiento**

Su función principal es homogeneizar los formatos de representación de los datos entre equipos de la red, es decir, que, aunque distintos equipos puedan tener diferentes representaciones internas como de carácter, numéricos, sonido, o imágenes, los datos llegan de manera reconocible.

**Operaciones de la capa de presentación**

* Formateo de datos: Opera como traductor entre dos tipos diferentes de código por ejemplo el código ampliado de caracteres decimal codificado en binario (EBCDIC), que utilizado para representar los caracteres en pantalla y el código americano normalizado para el intercambio de información (ANCII), que tiene la misma función a la información para visualizarla o imprimirla.
* Cifrado de dato: Protege la información durante la transmisión.
* Compresión de datos: Usa algoritmos para reducir el tamaño de los archivos.
* Aplicar a los datos procesos criptográficos.
* Definir la estructura de los datos a transmitir.

**Protocolos de la capa de presentación**

* ASN.1: Forma normalizada de representar datos.
* MIME: Se usa para transportar los archivos adjuntos en protocolos como HTTP o SMTP.

**Estándares para la representación de imagen**.

* GIF: Un formato de imagen utilizado en los primeros tiempos de las comunicaciones
* TIF: Formato de archivo de imagen etiquetado un formato de imagen de alta resolución.
* JEPG: Formato grafico utilizado para fotografías e imágenes con buena calidad/compresión.

**Estándares que regulan la representación de sonido y películas**

* MID: Interfaz digital para instrumentos musicales para música digitalizada.
* MPEG: Estándar para la compresión y codificación de video.
* QUICK TIME: Estándar para el manejo de audio y video para los sistemas operativos de los MAC.

**Capa 7 (Aplicación):** Ofrece a las aplicaciones (de usuario o no) la posibilidad de acceder a los servicios de las demás capas y define los protocolos que utilizan las aplicaciones para intercambiar datos, como correo electrónico (POP y SMTP), gestores de bases de datos y protocolos de transferencia de archivos (FTP).

La capa de aplicación sirve como ventana a los usuarios y los procesos de las aplicaciones para acceder a servicios de red.

Esta capa contiene una variedad de funciones de uso común:

* Compartición de recursos y redireccionamiento de dispositivos
* Acceso remoto de archivos
* Acceso remoto a impresoras
* Comunicación entre procesos
* Administración de red
* Servicios de directorio
* Mensajería electrónica
* Terminales virtuales de red

**Los principales protocolos que se manejan son:**

* FTP: (file transfer protocol – protocolo transferencia de archivos)
* DNS: (DOMAIN NAME SERVICE – SERVICIO DE NOMBRES DE DOMINIO).
* DHCP: (protocolo configuración dinámica de anfitrión).
* HTTP: (protocolo de transferencia de hipertexto) para acceso a páginas de internet
* NAT: (traducción de dirección de red).
* POP: (post office protocol) para correo electrónico.
* TELNET: para acceder a equipos remotos.

**Sistema de numeración decimal:**

El sistema de numeración que utiliza­mos habitualmente es el decimal, que se compone de diez símbolos o dígi­tos (0, 1, 2, 3, 4, 5, 6, 7, 8 y 9) a los que otorga un valor dependiendo de la posición que ocupen en la cifra: unidades, decenas, centenas, millares, etc.

El valor de cada dígito está asociado al de una potencia de base 10, número que coincide con la cantidad de símbolos o dígitos del sistema decimal, y un exponente igual a la posición que ocupa el dígito menos uno, contando desde la de­recha.

En el sistema decimal el número 528, por ejemplo, significa:

5 centenas + 2 decenas + 8 unidades, es decir:

5\*102 + 2\*101 + 8\*100 o, lo que es lo mismo:

500 + 20 + 8 = 528

En el caso de números con decimales, la situación es análoga, aunque, en este caso, algunos exponentes de las potencias serán negativos, concreta­mente el de los dígitos colocados a la derecha del separador decimal. Por ejemplo, el número 8245,97 se calcularía como:

8 millares + 2 centenas + 4 decenas + 5 unidades + 9 décimos + 7 céntimos

8\*103 + 2\*102 + 4\*101 + 5\*100 + 9\*10-1 + 7\*10-2, es decir:

8000 + 200 + 40 + 5 + 0,9 + 0,07 = 8245,97

**Sistema de numeración binario.**

El sistema de numeración binario utiliza sólo dos dígitos, el cero (0) y el uno (1).

En una cifra binaria, cada dígito tiene distinto valor dependiendo de la posición que ocupe. El valor de cada posición es el de una potencia de base 2, elevada a un exponente igual a la posición del dígito menos uno. Se puede observar que, tal y como ocurría con el sistema decimal, la base de la potencia coincide con la cantidad de dígitos utilizados (2) para representar los números.

De acuerdo con estas reglas, el número binario 1011 tiene un valor que se calcula así:

1\*23 + 0\*22 + 1\*21 + 1\*20, es decir:

8 + 0 + 2 + 1 = 11

Para expresar que ambas cifras describen la misma cantidad lo escribimos así:

10112 = 1110

* **Conversión entre números decimales y binarios**

Convertir un número decimal al sistema binario es muy sencillo: basta con realizar divisiones sucesivas por 2 y escribir los restos obtenidos en cada división en orden inverso al que han sido obtenidos.

Por ejemplo, para convertir al sistema binario el número 7710 haremos una serie de divisiones que arrojarán los restos siguientes:

77 : 2 = 38 Resto: 1

38 : 2 = 19 Resto: 0

19 : 2 = 9 Resto: 1

9 : 2 = 4 Resto: 1

4 : 2 = 2 Resto: 0

2 : 2 = 1 Resto: 0

1 : 2 = 0 Resto: 1

* Tomando los restos en orden inverso obtenemos la cifra binaria:

7710 = 10011012

* **El tamaño de las cifras binarias**

La cantidad de dígitos necesarios para representar un número en el sistema binario es mayor que en el sistema decimal. En el ejemplo del párrafo anterior, para representar el número 77, que en el sistema decimal está compuesto tan sólo por dos dígitos, han hecho falta siete dígitos en binario.

Para representar números grandes harán falta muchos más dígitos. Por ejemplo, para representar números mayores de 255 se necesitarán más de ocho dígitos, porque 28 = 256 y podemos afirmar, por tanto, que 255 es el número más grande que puede representarse con ocho dígitos.

Como regla general, con n dígitos binarios pueden representarse un máximo de 2n, números. El número más grande que puede escribirse con n dígitos es una unidad menos, es decir, 2n – 1. Con cuatro bits, por ejemplo, pueden representarse un total de 16 números, porque 24 = 16 y el mayor de dichos números es el 15, porque 24-1 = 15.

* **Conversión de binario a decimal**

El proceso para convertir un número del sistema binario al decimal es aún más sencillo; basta con desarrollar el número, teniendo en cuenta el valor de cada dígito en su posición, que es el de una potencia de 2, cuyo exponente es 0 en el bit situado más a la derecha, y se incrementa en una unidad según vamos avanzando posiciones hacia la izquierda.

Por ejemplo, para convertir el número binario 10100112 a decimal, lo desarrollamos teniendo en cuenta el valor de cada bit:

1\*26 + 0\*25 + 1\*24 + 0\*23 + 0\*22 + 1\*21 + 1\*20 = 83

10100112 = 8310

**Sistema de numeración hexadecimal**

En el sistema hexadecimal los números se representan con dieciséis símbolos: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E y F. Se utilizan los caracteres A, B, C, D, E y F representando las cantidades decima­les 10, 11, 12, 13, 14 y 15 respectivamente, porque no hay dígitos mayores que 9 en el sistema decimal. El valor de cada uno de estos símbolos depende, como es lógico, de su posición, que se calcula mediante potencias de base 16. Calculemos, a modo de ejemplo, el valor del número hexadecimal 1A3F16:

1A3F16 = 1\*163 + A\*162 + 3\*161 + F\*160  
  
1\*4096 + 10\*256 + 3\*16 + 15\*1 = 6719  
  
1A3F16 = 671910

* **Conversión de números binarios a octales y viceversa**

Observa la tabla siguiente, con los siete primeros números expresados en los sistemas decimal, binario y octal:

| DECIMAL | BINARIO | OCTAL |
| --- | --- | --- |
| 0 | 000 | 0 |
| 1 | 001 | 1 |
| 2 | 010 | 2 |
| 3 | 011 | 3 |
| 4 | 100 | 4 |
| 5 | 101 | 5 |
| 6 | 110 | 6 |
| 7 | 111 | 7 |

Cada dígito de un número octal se representa con tres dígitos en el sistema binario. Por tanto, el modo de conver­tir un número entre estos sistemas de numeración equivale a "expandir" cada dígito octal a tres dígitos bi­narios, o en "contraer" grupos de tres caracteres binarios a su correspondiente dígito octal.  
  
Por ejemplo, para convertir el número binario 1010010112 a octal tomaremos grupos de tres bits y los sustituiremos por su equivalente octal:

1012 = 58

0012 = 18

0112 = 38

De ese modo: 1010010112 = 5138

La conversión de números octales a binarios se hace, siguiendo el mismo método, reemplazando cada dígito octal por los tres bits equivalentes. Por ejemplo, para convertir el número octal 7508 a binario, tomaremos el equivalente binario de cada uno de sus dígitos:

78 = 1112

58 = 1012

08 = 0002

Por tanto: 7508 = 1111010002

* **Conversión de números binarios a hexadecimales y viceversa**

Del mismo modo que hallamos la correspondencia entre números octales y binarios, podemos establecer una equivalencia directa entre cada dígito hexadecimal y cuatro dígitos binarios, como se ve en la siguiente tabla:

| DECIMAL | BINARIO | HEXADECIMAL |
| --- | --- | --- |
| 0 | 0000 | 0 |
| 1 | 0001 | 1 |
| 2 | 0010 | 2 |
| 3 | 0011 | 3 |
| 4 | 0100 | 4 |
| 5 | 0101 | 5 |
| 6 | 0110 | 6 |
| 7 | 0111 | 7 |
| 8 | 1000 | 8 |
| 9 | 1001 | 9 |
| 10 | 1010 | A |
| 11 | 1011 | B |
| 12 | 1100 | C |
| 13 | 1101 | D |
| 14 | 1110 | E |
| 15 | 1111 | F |

La conversión entre números hexadecimales y binarios se realiza "expandiendo" o "con­trayendo" cada dígito hexadecimal a cuatro dígitos binarios. Por ejemplo, para expresar en hexadecimal el número binario 1010011100112 bastará con tomar grupos de cuatro bits, empezando por la derecha, y reemplazarlos por su equivalente hexadecimal:

10102 = A16

01112 = 716

00112 = 316

Por tanto: 1010011100112 = A7316

En caso de que los dígitos binarios no formen grupos completos de cuatro dígitos, se deben añadir ceros a la izquierda hasta completar el último grupo. Por ejemplo:

1011102 = 001011102 = 2E16

La conversión de números hexadecimales a binarios se hace del mismo modo, reemplazando cada dígito hexadecimal por los cuatro bits equivalentes de la tabla. Para convertir a binario, por ejemplo, el número hexadecimal 1F616 hallaremos en la tabla las siguientes equivalencias:

116 = 00012

F16 = 11112

616 = 01102

Por tanto: 1F616 = 0001111101102

**IRQ:**

En el contexto de la informática, una interrupción (del inglés interrupt request, en español «petición de interrupción») es una señal recibida por el procesador de una computadora, para indicarle que debe «interrumpir» el curso de ejecución actual y pasar a ejecutar código específico para tratar esta situación.

Una interrupción es una suspensión temporal de la ejecución de un proceso, para pasar a ejecutar una subrutina de servicio de interrupción, la cual, por lo general, no forma parte del programa, sino que pertenece al sistema operativo o al BIOS. Una vez finalizada dicha subrutina, se reanuda la ejecución del programa.

Las interrupciones son generadas por los dispositivos periféricos habilitando una señal del CPU (llamada IRQ del inglés "interrupt request") para solicitar atención del mismo. Por ejemplo. cuando un disco duro completa una lectura solicita atención al igual que cada vez que se presiona una tecla o se mueve el ratón.

**La primera técnica que se empleó para esto fue el polling**, que consistía en que el propio procesador se encargara de sondear los dispositivos periféricos cada cierto tiempo para averiguar si tenía pendiente alguna comunicación para él. Este método presentaba el inconveniente de ser muy ineficiente, ya que el procesador consumía constantemente tiempo y recursos en realizar estas instrucciones de sondeo.

El mecanismo de interrupciones fue la solución que permitió al procesador desentenderse de esta problemática, y delegar en el dispositivo periférico la responsabilidad de comunicarse con él cuando lo necesitara. El procesador, en este caso, no sondea a ningún dispositivo, sino que queda a la espera de que estos le avisen (le "interrumpan") cuando tengan algo que comunicarle (ya sea un evento, una transferencia de información, una condición de error, etc.).

* **Funcionamiento del mecanismo de interrupciones:**

Todos los dispositivos que deseen comunicarse con el procesador por medio de interrupciones deben tener asignada una línea única capaz de avisar al CPU cuando le requiere para realizar una operación. Esta línea se denomina IRQ.

Las IRQ son líneas que llegan al controlador de interrupciones, un componente de hardware dedicado a la gestión de las interrupciones, y que puede estar integrado en el procesador principal o ser un circuito separado conectado al mismo. El controlador de interrupciones debe ser capaz de habilitar o inhibir las líneas de interrupción y establecer prioridades entre las mismas. Cuando varias líneas de petición de interrupción se activan a la vez, el controlador de interrupciones utilizará estas prioridades para escoger la interrupción sobre la que informará al procesador principal. También puede darse el caso de que una rutina de tratamiento de interrupción sea interrumpida para realizar otra rutina de tratamiento de una interrupción de mayor prioridad a la que se estaba ejecutando; aunque hay interrupciones que no se pueden deshabilitar (conocidas como interrupciones no enmascarables o NMI).

Un procesador principal que no tenga un controlador de interrupciones integrado, suele tener una única línea de interrupción llamada habitualmente INT. Esta línea es activada por el controlador de interrupciones cuando tiene una interrupción que servir. Al activarse esta línea, el procesador consulta los registros del controlador de interrupciones para averiguar cuál IRQ hay que atender. A partir del número del IRQ busca en la tabla de vectores de interrupción la dirección de la rutina a la que debe llamar para atender la petición del dispositivo asociado a dicha IRQ.

* **Procesamiento de una interrupción**
* Terminar la ejecución de la instrucción máquina en curso.
* Salvar el estado del procesador (valores de registros y flags) y el valor del contador de programa, IP, en la pila, de manera que, en la CPU, al terminar el proceso de interrupción, pueda seguir ejecutando el programa a partir de la última instrucción.

La CPU salta a la dirección donde está almacenada la rutina de servicio de interrupción (Interrupt Service Routine, o abreviado ISR) y ejecuta esa rutina que tiene como objetivo atender al dispositivo que generó la interrupción.

Una vez que la rutina de la interrupción termina, el procesador restaura el estado que había guardado en la pila en el paso 2 y retorna al programa que se estaba usando anteriormente.

* **Mecanismo y líneas de petición de interrupción**

El bus de control de la placa base dispone de líneas específicas para el sistema de interrupciones. Un PC típico dispone en su placa base de un controlador de interrupciones 8259 de Intel o de un circuito integrado análogo. Este dispositivo electrónico dispone de hasta 16 líneas IRQ, numeradas desde el 00 hasta el 15. En las nuevas placas base este circuito está integrado junto con el resto del chipset y permite hasta 24 interrupciones.

En el IBM PC y XT existían 8 líneas de petición de interrupción manejadas por el controlador de interrupciones Intel 8259. Estas líneas están numeradas del 0 al 7, las dos primeras están asignadas al timer tick del temporizador Intel 8253, y al teclado. Solo quedaban 6 líneas para otros dispositivos, que aparecen como tales en el bus de control (IRQ2 - IRQ7). A partir del modelo AT se añadieron otras 8 líneas, numeradas del 8 al 15, mediante un segundo controlador de interrupciones (PIC), aunque la tecnología empleada exigió colgarlo de la línea IRQ2 del primero, de forma que esta línea se dedica a atender las interrupciones del segundo controlador a través de la línea 9 de este último, y la línea 8 se dedicó al reloj de tiempo real, un dispositivo que no existía en los modelos XT.

Aunque internamente se manejan 16 líneas, no todas tienen contacto en los zócalos del bus externo (son las marcadas con asterisco en la tabla que sigue). La razón de esta ausencia en los zócalos de conexión es que son de asignación fija, y solo son usadas por ciertos dispositivos instalados en la propia placa base. En concreto la línea NMI está asignada al mecanismo de control de paridad de la memoria, la línea 0 está asignada al cronómetro del sistema y la línea 1 al chip que controla el teclado (dispositivos que pueden requerir atención urgente por parte del procesador). Es costumbre denominar IRQx a las que tienen prolongación en el bus.

Teóricamente las restantes líneas podrían ser asignadas a cualquier nuevo dispositivo, pero en la práctica algunas están reservadas a dispositivos estándar. Por ejemplo, IRQ3 está casi siempre asignado al puerto serie COM2 y el IRQ4 al COM1; IRQ6 al controlador estándar de disquetes y IRQ7 al puerto de impresora LPT1. La tabla 1 muestra las asignaciones clásicas para el XT y el AT.

En sistemas más modernos utilizan la arquitectura APIC de Intel con 24 líneas y 8 extra para enrutar las interrupciones PCI.

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre | Int (hex) | XT: Descripción | AT: Descripción |
| NMI | --- | Paridad\* | Paridad\* |
| IRQ0 | 08 | Temporizador\* | Temporizador\* |
| IRQ1 | 09 | Teclado\* | Teclado\* |
| IRQ2 | 0A | Reservado | Interrupciones 8 a 15 (PIC#2) |
| IRQ3 | 0B | Puertos serie COM2/COM4 | Puerto serie COM2/COM4 |
| IRQ4 | 0C | Puertos serie COM1/COM3 | Puertos serie COM1/COM3 |
| IRQ5 | 0D | Disco duro | Puerto Paralelo LPT2 |
| IRQ6 | 0E | Disquete | Disquete |
| IRQ7 | 0F | Puerto Paralelo LPT1 | Puerto Paralelo LPT1 |
| IRQ8 | 70 | No existe | Reloj de tiempo real\* |
| IRQ9 | 71 | No existe | Redirigido a IRQ2\* |
| IRQ10 | 72 | No existe | no asignado |
| IRQ11 | 73 | No existe | no asignado |
| IRQ12 | 74 | No existe | Ratón PS2 |
| IRQ13 | 75 | No existe | Coprocesador 80287\* |
| IRQ14 | 76 | No existe | Contr. disco IDE primario |
| IRQ15 | 77 | No existe | Contr. disco IDE secundario |
| IRQ16 | 78 | Existe | Contr. disco SATA primario |

Cuando se instala un dispositivo de entrada o de salida que puede necesitar muchísima atención del procesador, debe asignársele una IRQ adecuada. Dicho en otras palabras, cuando un dispositivo periférico requiera atención, debe enviar una señal en la línea IRQ especificada. Inicialmente esta asignación se efectuaba de forma manual y automática, por medio de puentes (jumpers) en la placa o dispositivo móvil, pero actualmente esta selección puede hacerse mediante software.

* **Tipos de interrupciones**

Atendiendo a la fuente que las produce, las interrupciones pueden clasificarse de la siguiente forma:

* **Interrupciones de hardware:** Son aquellas interrupciones que se producen como resultado de, por lo general, una operación de E/S. No son producidas por ninguna instrucción de un programa sino por las señales que emiten los dispositivos periféricos para indicarle al procesador que necesitan ser atendidos.  
  Cuando el microprocesador accede a un periférico (disco duro, puerto de comunicación...), puede transcurrir algún tiempo antes de que los datos sean obtenidos o transmitidos. La solución más simple es esperar hasta recibir los datos o hasta que se haya efectuado la transmisión (polling), pero esta solución bloquea todos los programas en ejecución, y eso no puede admitirse en un sistema multitarea. Por ello, en los sistemas modernos se prefiere un funcionamiento mediante interrupciones, ya que éstas permiten mejorar la productividad del procesador, de forma que este último puede ordenar una operación de entrada o salida y, en lugar de tener que realizar una espera activa, se puede dedicar a atender a otro proceso o aplicación hasta que el dispositivo esté de nuevo disponible, siendo dicho dispositivo el encargado de notificar al procesador mediante la línea de interrupción que ya está preparado para continuar o terminar la operación de entrada o salida.

* **Excepciones**: Son un tipo de interrupción sincrónica típicamente causada por una condición de error en un programa, como por ejemplo una división entre 0 o un acceso inválido a memoria en un proceso de usuario. Normalmente genera un cambio de contexto a modo supervisor para que el sistema operativo atienda el error. Así pues, las excepciones son un mecanismo de protección que permite garantizar la integridad de los datos almacenados tanto en el espacio de usuario como en el espacio kernel. Cuando el Sistema Operativo detecta una excepción intenta solucionarla, pero en caso de no poder simplemente notificará la condición de error a la aplicación/usuario y abortará la misma.
* **Interrupciones por software:** También denominadas llamadas al sistema, son aquellas generadas por un programa mientras este está ejecutándose. En general, actúan de la siguiente manera: Un programa en ejecución llega a una instrucción que requiere del sistema operativo para alguna tarea, por ejemplo, para leer un archivo en el disco duro (cuando un programa necesita un dato exterior, se detiene y pasa a cumplir con las tareas de recoger ese dato). En ese momento por tanto llama al sistema y se interrumpe virtualmente hasta recibir respuesta, en el ejemplo anterior hasta que no se haya leído el disco y el archivo esté en memoria principal. Durante esa espera las instrucciones que se ejecutarán no serán del programa, sino del sistema operativo. Una vez éste termine su rutina ordenará reanudar la ejecución del programa auto interrumpido en espera. Por último, la ejecución del programa se reanuda.
* **Determinación de la dirección de la rutina de servicio de interrupción**

Hay dos alternativas para determinar la dirección de la rutina de servicio de interrupción que se debe ejecutar al recibir una interrupción determinada:

* **Direcciones fijas:** Se hallan cableadas en el procesador y por tanto nunca pueden ser cambiadas. Esto implica que las RSI siempre estarán en una determinada posición de la memoria.
* **Direcciones variables (por interrupciones vectorizadas):** En este grupo se incluyen aquellas que presentan una dirección variable y que, por tanto, no se halla cableada en el procesador. De esta manera el dispositivo debe dar información acerca de la localización de la dirección de comienzo de la RSI asociada a dicho periférico.   
  Hay distintas metodologías de diseño para las interrupciones con direcciones variables.   
  En la actualidad, las alternativas que son implementadas de manera habitual son las siguientes:
* **Direccionamiento absoluto:** En este caso es el dispositivo o la interfaz del dispositivo la encargada de conocer la dirección de la RSI y de enviarla al procesador para que éste pueda localizar dicha subrutina y ejecutarla.
* **Direccionamiento relativo:** El dispositivo solo suministra parte de la dirección de comienzo y es el procesador el encargado de completarla (añadiendo bits o sumando una determinada cantidad, que siempre será fija). Esta alternativa tiene una ventaja sobre la anterior y es que permite especificar la dirección de comienzo con menos bits y por tanto simplifica el diseño. Ahora bien, tiene una desventaja principal y es que limita el número de dispositivos que podemos conectar y además ciertos bits de la dirección quedan fijados de forma permanente por la CPU lo que reduce la capacidad de reubicación de la RSI. Una alternativa que utilizan ciertos procesadores como el 8080 o el 8085 es que en vez de enviar solamente la dirección de comienzo de la RSI se envía también el código de la operación de salto (por ejemplo, CALL).
* **Direccionamiento indirecto:** También conocida como direccionamiento por interrupciones vectorizadas. Se mantiene una tabla de vectores de interrupción (direcciones de comienzo de las distintas RSI) y a cada interrupción se le asocia un número que será el índice por el cual se accederá a la tabla y se recuperará la información de la dirección de comienzo. Necesita señales de conformidad o handshaking para sincronizar al procesador con la interfaz, ya que esta última tiene que indicarle al procesador cuando va a enviarle el índice que necesita para buscar el vector de interrupción (INT) y el procesador deberá enviar otra señal para indicar que se ha reconocido la interrupción (INTA#).
* **Determinación de la fuente que genera la interrupción**

Hay distintas formas de identificar la fuente de una determinada interrupción. La primera alternativa que se consideró fue asignar una línea (un bit) para cada interrupción, lo cual suponía un gran costo en cuanto a la relación de número de dispositivos y número de bits usados y a menudo limitaba el número de dispositivos que se podían conectar. Por ello, se pensó con posterioridad en que en cada patilla de interrupción debería poder conectarse más de un dispositivo, debiendo implementar por tanto una metodología que permitiese identificar de forma unívoca de qué dispositivo se trataba. Para ello hay varias directrices:

**Polling:** El microprocesador comprueba de manera sistemática todos los dispositivos de manera que «busca» cuál de ellos fue el que solicitó la interrupción. Esto tiene una ventaja y es que es barato a nivel de coste hardware ya que el polling se implementa en software, no obstante, tiene otras desventajas que no podemos olvidar y es que suele ser lento porque tiene que comprobar en serie todos los dispositivos y establece una prioridad en los dispositivos (el orden de sondeo) y por tanto puede provocar inanición.

**Interrupciones vectorizadas:** Este concepto fue ya tratado en el apartado anterior. Como ventajas podemos destacar que suele ser rápido, pero implica un alto costo en el hardware.

**Hardware paralelo:** Se utiliza un registro de interrupción cuyos bits se controlan de forma independiente por las señales de petición de interrupción de cada periférico. Según la posición de cada bit en el registro, se establece la prioridad.

**Sistemas de prioridad:** El sistema operativo necesita un mecanismo para priorizar las interrupciones y tratar primero las más urgentes. Para ello, existen varias alternativas:

* **Interrupciones simultáneas:** No tienen por qué ocurrir de manera simultánea, sino que se refiere a que en un momento dado puede haber varias interrupciones activas.
* **Interrupciones anidadas:** Mientras se está procesando una determinada rutina de servicio de interrupción sucede otra señal de interrupción.   
  Existen dos métodos para tratar las interrupciones anidadas.

El primero se basa en inhabilitar las interrupciones mientras se está ejecutando una determinada RSI. Esto puede realizarlo el hardware de manera automática en algunos procesadores, pero en otros será el usuario el encargado de deshabilitarlas en caso de que no desee que ninguna otra interrupción pueda interrumpir el transcurso normal de la rutina de servicio de interrupción. No es aconsejable deshabilitar las interrupciones durante mucho tiempo ya que esto puede provocar errores y pérdida de información.

* **Inhibición de interrupciones**: Hay distintas alternativas de inhibición de interrupciones. Como ya hemos visto estas se pueden hacer de manera automática por el hardware en algunos casos mientras que en otros será el usuario el encargado de realizarlo por software y esto depende de la arquitectura del procesador que consideremos. Las distintas opciones son:
* **Deshabilitar todas las interrupciones:** Para esto basta con inhibir el bit del registro de flag dedicado a las interrupciones.
* **Deshabilitar al principio de la RSI y activarlas de nuevo al finalizar la misma**. Puede ser de manera automática o por el usuario.
* **Desactivar solo las interrupciones que tengan menor prioridad que la asociada a la RSI que se está ejecutando en ese momento.**
* **Deshabilitar de forma selectiva distintos niveles de prioridad de interrupción.** Para lo cual se emplean registros especiales denominados máscaras en el que cada uno de sus bits identifican a un nivel distinto y modificando su contenido se puede establecer que niveles están activos en ese momento. Se puede cambiar por el programador.

**Interrupciones simultáneas:** En este método tenemos dos alternativas, una de ellas es que exista algún hardware que tenga como entradas las señales de interrupción y de cómo salida la interrupción más prioritaria que está activa en ese momento. Otra alternativa es tener un método de identificación de prioridades distribuida y no generalizada como en el caso anterior, en este caso tenemos que destacar dos técnicas distintas que se pueden implementar en la práctica:

* **Daisy-chain (conexión en caden**a): Se puede conectar los distintos dispositivos en cadena, en orden decreciente de prioridad y por tanto la señal de reconocimiento de interrupción (INTA#) solo será pasada al siguiente dispositivo en caso de que el anterior (más prioritario) no haya solicitado los servicios del procesador. Sin embargo, algo importante es que las señales de interrupción que van al procesador están conectadas todas a un mismo cable, por tanto, deberemos utilizar alguna técnica especial para que no se produzca un cortocircuito. Para evitar precisamente que la pista se cortocircuite se utiliza la técnica del open-collecto o «colector abierto» y consiste en conectar el colector de un transistor a la pista común (un transistor por cada dispositivo) y por tanto estarán tantos colectores conectados como dispositivos tengamos (se entiende que son dispositivos que mandan petición de interrupción al procesador).
* **Híbrida:** mezcla las dos técnicas explicadas anteriormente.

**Puertos lógicos:**

El Puerto Lógico es una zona, o localización, de la memoria de un ordenador que se asocia con un puerto físico o con un canal de comunicación, y que proporciona un espacio para el almacenamiento temporal de la información que se va a transferir entre la localización de memoria y el canal de comunicación.  
  
Un puerto lógico es una salida de bits, que pueden ser 1 o 0, o sea, un puerto es el valor que se usa en el modelo de la capa de transporte para distinguir entre las múltiples aplicaciones que se pueden conectar al mismo host, o puesto. Entonces un puerto lógico de Internet es una interface de software que permitirá el ingreso y salida de data por aplicaciones que usan Internet.  
  
Los puertos se identifican por números desde 1 hasta 65.000 pudiendo llegar a más, siendo conocidos los puertos de 1 a 1024 como:

* HTTP puerto 80 transferencia de hipertexto por Internet
* FTP puerto 20 transferencia de data (mp3, documentos, etc)
* HTTPS puerto 443 transferencia segura
* SMTP puerto 25 correo electrónico

Del 1025 para arriba son desconocidos, algunos TROYANOS y otras aplicaciones inocuas.  
  
Los puertos lógicos son, al igual que los puertos físicos, necesarios para que nuestros programas puedan comunicarse con el exterior. La diferencia es que se enlazan virtualmente en nuestra conexión TCP con los programas, para tener una referencia, y que los otros programas puedan conectarse a los nuestros y traspasar información. Por ejemplo, podemos decir que el servidor web suele estar enlazado (escuchando) en el puerto 80, o que nuestro navegador, sale por el puerto 4000 para conectarse a este servidor. El servicio RPC (remote procedure call) escucha en los sistemas XP por el puerto 135, pero resulta que a ese puerto se conectaba el virus Blaster para infectar todos los Windows que encontraba a su paso.  
  
Muy pocas veces usaremos tantos puertos -65.000-. Podemos navegar, usar nuestro programa de mensajería instantánea, el cliente de IRC y mandar un correo y aun así estaremos usando sólo unas decenas. Además, están divididos según sus funciones.  
  
Los puertos de 1 a 1024 se llaman puertos reservados o conocidos. Tienen una función específica que mandan los estándares. La organización que se encarga de establecer los estándares es la IANA (Internet Assigned Numbers Authority), que se puede encontrar en www.iana.org. Por ejemplo, el 22 es para SSH (Secure SHell), del 135 al 139 para la NetBios. Los puertos que van de 1025 a 49151 no son estándar o puertos registrados, pero la IANA se encarga de asignarlos a distintas aplicaciones, que lo necesitan. El resto de puertos hasta el 65536 son los llamados efímeros, privados o dinamicos, porque son los clientes (el navegador, el cliente de correo, el cliente de FTP) los que lo eligen aleatoriamente para establecer desde ellos la conexión a los puertos servidores y, si la conexión cae, se liberan y pueden ser usados por cualquier otra aplicación o protocolo más tarde.  
  
Si aplicamos un escáner de puertos a nuestro propio PC podemos obtener gran cantidad de información. Conocer los puertos a la escucha nos proporciona información sobre los servicios que tenemos instalados como servidor. Conocer los puertos que “hablan” en nuestro ordenador, los efímeros, nos proporciona información sobre lo que está haciendo nuestra máquina, si se está comunicando con otra y a través de qué puerto. Esto puede ser un signo de que alguien nos está robando información. (Microsoft lo hace a menudo).

**Servicios vinculados a puertos lógicos fundamentales:**

* **Puerto 20** - FTP Data - Utilizado por servidores FTP (File Transfer Protocol) para la transmicion de datos en modo pasivo.
* **Puerto 21** - FTP - Tambien utilizado por servidores FTP. Su mala configuracion puede resultar en ataques (troyanos, hacking, etc..)
* **Puerto 22** - SSH - Puerto utilizado por Secure Shell (SSH), el cual es un protocolo y un programa que lo utiliza para acceder a maquinas remotas a través de una red. Además, funciona como una herramienta de transmisión de datos, desde ficheros sueltos hasta una sesión de FTP cifrado.
* **Puerto 23** - Telnet - Telnet es una herramienta que proporciona una ventana de comandos, los cuales permiten controlar un pc de forma remota. Es una de las formas más fáciles de entrar ilícitamente en un pc ajeno.
* **Puerto 25** - SMTP - Puerto utilizado por SMTP (Simple Mail Transfer Protocol), o en español Protocolo de transferencia simple de correo electrónico. Como deben suponer, es el protocolo, basado en texto, que permite transferir correo electrónico entre diferentes computadoras, PDA's, celulares, etc.
* **Puerto 53** - DNS - Este puerto lo utiliza el DNS (Domain Name System), esta base de datos distribuida o jerárquica, se encarga de traducir nombres de dominios a IP's.
* **Puerto 59** - DCC - Utilizado principalmente en programas de comunicación para transferir ficheros.
* **Puerto 79** - Finger - En este puerto se corre el servicio Finger, el cual ha sido uno de los mayores problemas de seguridad en Unix, ya que este proporciona información, muy detallada, de los usuarios de una máquina, estén o no logueados.
* **Puerto 80** - HTTP - Puerto que transmite el protocolo HTTP (Hypertext Transfer Protocol) que es el utilizado en cada transacción web (WWW).
* **Puerto 110** - POP3 - Puerto que utiliza el servicio POP3 (Post Office Protocol 3), que es el correo electrónico, offline.
* **Puerto 113** - IDENT - Servicio de identificación/autorización. Los servidores de internet, como POP, IMAP, SMTP, IRC consultan este puerto en respuesta a conexiones de clientes.
* **Puerto 135** - RPC - Remote Procedure Cell. Este servicio, es el encargado de administrar la comunicación con otro pc cuando un programa solicita ejecutar código en ese otro pc. De esta forma, el programador no tiene que preocuparse por esta conexión.
* **Puerto 139** - NetBIOS - Funciona el servicio NetBIOS que es el encargado de compartir ficheros de tu pc, por tu red interna. Con este puerto abierto peligras de que gente de todo el mundo, pueda ver y usar estos ficheros a través de internet.
* **Puerto 143** - IMAP - Se ejecuta el IMAP (Internet Message Access Protocol). Este es un servicio nuevo, por lo cual los servidores de internet que lo utilizan, no han tenido suficiente tiempo para madurar. Lo cual implica, que este sea una muy buena vía de acceso para los intrusos.
* **Puerto 443** - HTTPS - El Hypertext Transfer Protocol Secure, no es más que una versión segura, del ya mencionado HTTP.
* **Puerto 445** - MSFT DS - Server Message Block. A partir de Windows 2000, microsoft añadió la posibilidad de ejecutar SMB directamente sobre TCP/IP sin la capa extra de NBT.
* **Puerto 1080** - Socks - Aquí funciona el servicio Socks, el cual es un protocolo que permite a las aplicaciones cliente-servidor usar de manera transparente los servicios de un firewall de red.
* **Puerto 5000** - UPnP - El universal plug n' play define protocolos y procedimientos comunes para garantizar la Inter operatividad sobre PC's permitidos por red, aplicaciones y dispositivos inalámbricos.
* **Puerto 8080** - WebProxy - Este puerto lo pueden utilizar terceros para ocultar su verdadero IP a los servidores web.

**¿Qué es IPV4?**

IPV4 (Internet Protocol version 4), es la tecnología subyacente que nos permite conectar nuestros dispositivos a la web. Cada vez que un dispositivo accede a Internet (ya sea una PC, Mac, teléfono inteligente u otro dispositivo), se le asigna una dirección IP única y numérica.   
Para enviar datos de una computadora a otra a través de la web, se debe transferir un paquete de datos a través de la red que contiene las direcciones IP de ambos dispositivos.

Sin las direcciones IP, las computadoras no podrían comunicarse y enviar datos entre sí. Es esencial para la infraestructura de la web.

**¿Qué es IPv6?**

IPv6 (Internet Protocol versión 6), sucesor de IPv4. Funciona de manera similar a IPv4 en que proporciona las direcciones IP únicas y numéricas necesarias para que los dispositivos habilitados para Internet se comuniquen. Sin embargo, tiene una gran diferencia: utiliza direcciones de 128 bits.

**¿Qué diferencia hay entre ambas?**  
Como se indicó anteriormente, IPv6 utiliza direcciones de Internet de 128 bits. Por lo tanto, puede admitir 2^128 direcciones de Internet: 340,282,366,920,938,000,000,000,000,000,000,000,000 de ellos para ser exactos. Son muchas direcciones, tantas que requiere un sistema hexadecimal para mostrar las direcciones. En otras palabras, hay direcciones IPv6 más que suficientes para mantener Internet en funcionamiento durante mucho, mucho tiempo.

**¿Como es una dirección IP de IPV4?**  
  
Es un numero binario de 32 bits. Estos 32 bits se dividen en 4 byte.  
Cada uno de este byte se transforma a su notación decimal y se separa por puntos.

|  |  |  |  |
| --- | --- | --- | --- |
| 10000011 | 01101100 | 01111010 | 11001100 |
| 131 | **108** | **122** | **204** |

DIRECCION IP: **131.108.122.204**

**Identificador de red**

|  |  |
| --- | --- |
| 1000001101101100 | 0111101011001100 |
| RED | **HOST** |

La clase de dirección define la cantidad de bits de red y host.

|  |  |  |
| --- | --- | --- |
| CLASE A | Primer octeto está comprendido entre 0 – 127 ej. 127.1.1.1 | Primeros 8 bits representan a RED y últimos 24 bits a HOST |
| CLASE B | Primer octeto está comprendido entre 128 – 191 ej. 130.244.53.2 | Primeros 16 bits representan a RED y últimos 16 bits a HOST |
| CLASE C | Primer octeto está comprendido entre 192 – 223 ej. 192.168.100.50 | Primeros 24 bits representan a RED y últimos 8 bits a HOST |
| CLASE D | Primer octeto está comprendido entre 224 -239 |  |
| CLASE E | Primer octeto está comprendido entre 240 - 255 |  |

**Mascara y prefijos de RED:**

A close up of text on a white background

Description automatically generated

**¿Como es una dirección IP de IPv6?**

La función de la dirección IPv6 es exactamente la misma a su predecesor IPv4, pero dentro del protocolo IPv6.

Está compuesta por 8 segmentos de 2 bytes cada uno, que suman un total de 128 bits, el equivalente a unos 3.4×1038 hosts direccionables.

Su representación suele ser hexadecimal y para la separación de cada par de octetos se emplea el símbolo “:”. Un bloque abarca desde 0000 hasta FFFF. Algunas reglas acerca de la representación de direcciones IPv6 son:

* Los ceros iniciales, como en IPv4, se pueden obviar.  
   Ejemplo: 2001:0123:0004:00ab:0cde:3403:0001:0063 -> 2001:123:4:ab:cde:3403:1:63.
* Los bloques contiguos de ceros se pueden comprimir empleando “ :: ”. Esta operación sólo se puede hacer una vez.

Ejemplo: 2001:0:0:0:0:0:0:4 -> 2001::4.

Ejemplo no válido: 2001:0:0:0:2:0:0:1 -> 2001::2::1 (debería ser 2001::2:0:0:1 ó 2001:0:0:0:2::1).

**Paquetes IPV6**  
La cabecera se encuentra en los primeros 40 bytes del paquete, contiene las direcciones de origen y destino con 128 bits cada una, la versión 4 bits, la clase de tráfico 8 bits, etiqueta de flujo 20 bits, longitud del campo de datos 16 bits, cabecera siguiente 8 bits, y límite de saltos 8 bits.   
**Direcciones privadas:** llamadas privadas ya que estos rangos se utilizan para conectar varios computadores y/o dispositivos dentro de una LAN.

**Direcciones públicas:** llamadas publicas ya que estas son las direcciones que se utilizaban para viajar a través de internet, estas direcciones no se deben poner en una computadora interna ya que puede ocasionar conflictos al momento de comunicarse con el exterior.

**Tipos de comunicación:**

* **Unicast IPv4**: Tipo de comunicación uno a uno.   
  Un dispositivo de red se comunica con otro dispositivo de red. La dirección de la capa 3 utilizada para Unicast es IPv4 Clase A, Clase B, Clase C. La dirección de la capa 2 es una dirección MAC de unidifusión.
* **Multicast IPV4**: Tipo de comunicación uno a muchos.  
  Un dispositivo de red envía un paquete de datos IPv4 y se entrega a los dispositivos que están interesados ​​en ese tráfico. La dirección de la capa 3 utilizada para la multidifusión IPv4 corresponde a las direcciones IPv4 de clase D (comienza de 224 a 239). La dirección de la capa 2 para la multidifusión IPv4 comienza con "01: 00: 5e".
* **Broadcast IPv4**: Tipo de comunicación uno a todos.  
  Un dispositivo de red envía un paquete de datos IPv4 y se entregarán todos los dispositivos en ese segmento de LAN. El problema con el tráfico de difusión es que las transmisiones alteran todos los dispositivos en LAN y causan una pérdida de ancho de banda.  
    
  **En IPv6**, tenemos Unicast, Multicast y Anycast. El concepto de Unicast y Multicast es el mismo en IPv4 e IPv6, excepto los cambios en las direcciones IPv6 Layer 3 utilizadas para difusión y multidifusión y la dirección Layer 2 utilizada para multicast. La dirección de capa 2 utilizada para el tráfico de multidifusión de IPv6 comienza desde "33:33:" (en Ipv4, es "01: 00: 5e").
* **Anycast IPV6**   
  Este tipo de comunicación se utiliza para identificar una interfaz de un grupo de interfaces, que proporcionan el mismo servicio, pero cerca del cliente en la distancia de enrutamiento (podemos comparar la distancia de enrutamiento similar a la distancia geográfica).

**Índice**

* capa física:<https://moodle2.unid.edu.mx/dts_cursos_mdl/lic/TI/BN/AM/08/Capa_fisica_del_modelo_OSI.pdf>

<https://www.ecured.cu/Capa_f%C3%ADsica>

* capa de enlace de datos: <https://capaenlace.wordpress.com/2016/11/09/primera-entrada-del-blog/>

<https://eltallerdelbit.com/capa-2-osi/>

* capa de red: <https://eltallerdelbit.com/capa-de-red-capa-3-osi/>

<https://eltallerdelbit.com/capa-de-red-capa-3-osi/>

<http://blog.utp.edu.co/ee973/files/2012/04/capitulo05-capa-de-red.pdf>

<https://www.ecured.cu/Capa_de_red>

* capa de transporte: <https://eltallerdelbit.com/capa-4-osi-capa-de-transporte/>

<https://www.monografias.com/trabajos-pdf2/capa-transporte-modelo-osi/capa-transporte-modelo-osi.shtml>

* capa de sesión: <https://eltallerdelbit.com/capa-5-osi-capa-de-sesion/>

<http://itcelenes.mx.tripod.com/Unidad8CapadeSesion.html>

<https://www.ecured.cu/Capa_de_Sesi%C3%B3n>

<http://lisethmodeloosi.blogspot.com/p/capa-de-presentacion_22.html>

* capa de presentación: <https://www.ecured.cu/Capa_de_presentaci%C3%B3n>

<https://es.slideshare.net/robertovasconezbarragan7/capa-de-presentacin-modelo-osi>

<https://eltallerdelbit.com/capa-6-osi-capa-de-presentacion/>

<http://computecsena.blogspot.com/2012/05/capa-de-presentacion-modelo-osi.html>

* capa de aplicación: <https://eltallerdelbit.com/capa-7-osi-capa-de-aplicacion/>

<https://culturacion.com/especial-modelo-osi-capa-de-aplicacion/>

<http://www.ie.tec.ac.cr/acotoc/CISCO/R&S%20CCNA1/R&S_CCNA1_ITN_Chapter10_Capa%20de%20aplicacion.pdf>

* Información de tres vías:<http://rtelecomunicaciones.blogspot.com/2013/07/protocolo-tcp-de-enlace-de-tres-vias.html>
* imagen e información de tres vías:<https://sites.google.com/site/cursoenlineaccna1/unidad-2-capa-de-transporte-del-modelo-osi/2-2-protocolo-tcp-comunicacion-con-confiabilidad/2-2-3-establecimiento-y-finalizacion-de-la-conexion-tcp>
* imagen:<https://www.google.com/search?q=tama%C3%B1o+de+ventana+modelo+osi&tbm=isch&source=iu&ictx=1&fir=ct4V51A26ATeCM%253A%252CEAgU4axm5RFMnM%252C_&vet=1&usg=AI4_-kRvwjbint7UsHSdMG5RoZMHs1uyTw&sa=X&ved=2ahUKEwj2g-_j0qXiAhU8ILkGHek1C2wQ9QEwAnoECAkQCA#imgdii=2kQfEKYTpuLEpM:&imgrc=ct4V51A26ATeCM:&vet=1>
* información subcapa MAC: <http://espedatacomm.blogspot.com/2011/11/subcapas-de-la-capa-de-enlace.html>