UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES ARAGÓN

“WEB SERVER EMBEBIDO CON MICROCONTROLADOR MC912NE64”

T E S I S P R O F E S I O N A L
QUE PARA OBTENER EL TÍTULO DE
INGENIERO EN COMPUTACIÓN
P R E S E N T A :

LUIS JACOBO HERNÁNDEZ ORTIZ

ASESOR: ING. ELEAZAR MARGARITO PINEDA DÍAZ

SAN JUAN DE ARAGÓN, EDO. DE MÉXICO, MARZO DE 2008

INTRODUCCIÓN.

El rápido avance en los últimos años de las telecomunicaciones, Hardware, Software, Firmware y la accesibilidad hacia muchas herramientas en estas áreas, nos brinda la oportunidad de aplicar tales desarrollos en áreas tan diversas como la imaginación lo permita.

Un caso específico es el presente trabajo en el cual se combina un medio de comunicación tan versátil, económico, universal y accesible como el Internet con microcontroladores de altas prestaciones, como los desarrollados por Freescale; además de lenguajes de programación de alto y bajo nivel, con el fin de satisfacer las crecientes necesidades de adquisición de datos de manera remota a través de un sistema electrónico de tipo embebido conocido como Web Server.

En el primer capítulo de este trabajo se muestra un panorama general de los sistemas embebidos, su estructura, sus características y su estado del arte; pero principalmente se aborda su complejo proceso de diseño.

En el segundo capítulo se analizan las más variadas formas que un sistema embebido puede ser conectado a una red de cómputo incluyendo el Internet, tales formas van desde utilizar una computadora como medio de conexión hasta los dispositivos que poseen una interfase de Ethernet incluida, en la cual se centra este trabajo.

El tercer capítulo trata de la Ethernet, tecnología base de las comunicaciones de redes de computadoras, su implementación física y lógica; así como de uno de los modelos de pilas de protocolo más famoso, el OSI.

El cuarto capítulo resalta las principales características de las capas de la principal pila de protocolos destinada a la conexión en Internet, el modelo TCP/IP.

En el quinto capítulo se documenta la forma como es implementado un Web Server, desde la selección del microcontrolador hasta el desarrollo del Hardware, del Software y del Firmware; así como la utilización de las herramientas que permiten que la fase de diseño sea más fácil y rápida, logrando como resultado un sistema eficaz.

1. **SISTEMAS EMBEBIDOS**.

OBJETIVO: Describir las características propias de los sistemas embebidos, las fases más importantes en su complejo proceso del diseño, su estado presente y su tendencia hacia el futuro.

1.1 SISTEMAS EMBEBIDOS.

Actualmente se vive en un mundo embebido, día a día se sobrevive con estos productos, provocando que la vida dependa en gran parte de la función propia de estos aparatos, pero pocas son las personas que se dan cuenta que dentro de cada sistema embebido hay un procesador y un programa ejecutándose que les permite funcionar.

Los sistemas embebidos se encuentran en una gran variedad de productos electrónicos tales como en la electrónica de consumo – teléfonos celulares, cámaras fotográficas y de video, video juegos, calculadoras, asistentes personales digitales; en la electrónica del hogar – hornos de microondas, teléfonos, alarmas, lavadoras, sistemas de iluminación, termostatos; electrónica de oficina – fax, copiadoras impresoras, scanner; en la electrónica de negocios – cajas registradoras, lectores de tarjetas, puntos de venta, cajeros automáticos; en la electrónica del automóvil – frenos, transmisión, fuel injection; en la electrónica de la industria – sistemas de adquisición y comunicación de datos, controladores, sistemas de seguridad y sin duda existe un sistema embebido en algún punto de una línea de producción. Se puede pensar que casi cualquier equipo que use electricidad, ya tiene o pronto tendrá un sistema embebido para su funcionamiento [1].

Esto no fue posible antes de 1971, año en que Intel introduce el primer microprocesador del mundo. Este circuito, el 4004, fue diseñado para usarse en una línea de calculadoras producidas por la compañía japonesa Busicom. En 1969 Busicom preguntó a Intel sobre un conjunto de circuitos integrados para cada uno de sus nuevos modelos de calculadoras. El 4004 fue la respuesta de Intel. Más que el habitual diseño de Hardware para cada calculadora, Intel propuso un circuito de propósito general que podía ser usado en toda su línea de calculadoras. Este circuito de propósito general fue diseñado para leer y ejecutar un conjunto de instrucciones de Software almacenadas en un circuito de memoria externo. La

idea de Intel fue que el Software que se le de a cada calculadora sea propio de sus características.

El microprocesador fue un éxito, y su uso se incremento durante la siguiente década. Rápidamente aparecieron aplicaciones embebidas incluyendo pruebas espaciales no tripuladas, luces de tráfico y sistemas de control aeroespacial. Silenciosamente, los sistemas embebidos han sobrevivido a la época de las computadoras personales o PC, trayendo microprocesadores a nuestras vidas personales y profesionales.

Es inevitable que el número de sistemas embebidos continué su rápido incremento, ya que día tras día surgen nuevos dispositivos embebidos que poseen enorme potencial de mercado.

Desde el punto de vista del diseño, los sistemas embebidos viven una constante evolución, el 4004 era un microprocesador de 4 bits en la actualidad se manejan microprocesadores de 64 bits y con una cantidad considerable de memoria y de periféricos.

Las herramientas de Software y Hardware para el diseño sistemas embebidos, además de representar todo un mercado, nos permiten planificar y optimizar recursos, además de que evolucionan al mismo paso que evolucionan los dispositivos embebidos. El diseño de un producto embebido requiere la combinación de bajo costo, bajo consumo eléctrico y alto rendimiento.

Aunque las técnicas de producción y características económicas son muy importantes, el proceso de diseño se ha vuelto más y más crítico ya que en gran parte de esto depende el éxito o un fracaso de un producto.

Todo diseño, no sólo electrónico, parte de una adecuada definición del problema, una amplia documentación, análisis del problema, una correcta propuesta de solución, planeación y cumplimiento de actividades; es decir, una metodología que de como resultado un producto que no exclusivamente satisface una necesidad en el mercado, si no también, el objetivo que toda empresa de electrónicos se pone como meta al invertir en él.

1.2 ¿QUÉ ES UN SISTEMA EMBEBIDO?.

No existe un concepto universal para sistema embebido, debido a que cada diseñador le da a sus sistemas un enfoque muy personal, pero uno de los más aceptados es el qué dice que un sistema embebido es una combinación de Hardware, Firmware, periféricos y tal vez algunas piezas electromecánicas o de algún otro tipo como sensores, diseñado para tener una función específica [2].

Por el contrario una computadora personal, que si bien está constituida también por una combinación de Hardware y Software más algunas piezas mecánicas, no está diseñada para un uso especifico, si no por el contrario, es posible darle una gran variedad de usos diferentes, debido a esto mucha gente le llama “Computadora de propósito general”.

Un sistema embebido es muchas veces un componente de un sistema mucho más grande, por ejemplo, en el sistema de frenos o el sistema de inyección de combustible de un automóvil, un microcontrolador conforma la parte embebida de dichos sistemas.

El Software y el Hardware que constituyen a un sistema embebido pueden ser sustituidos por un circuito integrado que realice única y específicamente esa función, pero se despreciaría otra de las características de los sistemas embebidos, la flexibilidad. Esto es que si se desea hacer alguna modificación, es más fácil cambiar algunas líneas de código al Software del

sistema embebido, llamado también como Firmware, que remplazar el circuito integrado y hasta algunas veces toda la tarjeta.

1.3 ESTRUCTURA DE UN SISTEMA EMBEBIDO.

Todo sistema embebido contiene un procesador, por ejemplo un microprocesador, un microcontrolador, un PLD (Dispositivo Lógico Programable), un FPGA (Arreglo de Compuertas Programables en Campo) o un DSP (Procesador Digital de Señales); el Firmware, un lugar en donde almacenarlo, es decir, memoria RAM y/o ROM que en muchos casos debido a las necesidades del sistema y características del dispositivo, está incluida dentro de él; y algún tipo de entrada y salida como puertos y periféricos [3]. La figura 1.1 muestra el esquema de un sistema embebido básico.

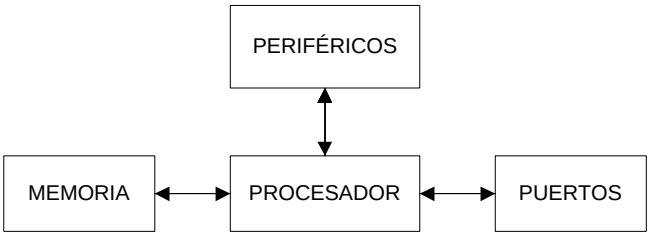


Figura 1.1 Sistema embebido básico.

Diferente al Software diseñado para computadoras de propósito general, el Firmware de los sistemas embebidos no puede correr en un sistema embebido diferente sin modificaciones significantes. Esto es principalmente porque la amplia variedad de diseño y de dispositivos de Hardware que existen y en apego a mantener bajo el costo del sistema, el Hardware en cada sistema embebido es adaptado específicamente para la aplicación. Como resultado, los circuitos innecesarios son eliminados y los recursos de Hardware son distribuidos hasta donde sea posible.

1.4 CARACTERÍSTICAS DE LOS SISTEMAS EMBEBIDOS.

Los sistemas embebidos poseen características que los diferencian de algún otro sistema electrónico, una PC por ejemplo. A continuación se analiza cada característica, haciendo énfasis en que no todos los sistemas embebidos poseen dichas características al mismo tiempo, definitivamente esto dependerá de la aplicación [4].

1.4.1 REALIZACIÓN DE TAREAS ESPECÍFICAS.

Una computadora personal ejecuta varios programas como procesadores de palabras, video juegos y cualquier otro programa que se le agregue. Por el contrario un sistema embebido usualmente ejecuta sólo un programa repetidamente. Por ejemplo, una calculadora siempre será una calculadora. Ésta característica fundamentalmente se debe a que un sistema embebido está constituido por un microprocesador dedicado, es decir, está programado para realizar sólo una o quizá algunas tareas específicas. Al hablar de cambio de tarea del sistema, se asocia directamente con la obsolescencia o con el rediseño.

1.4.2 AMPLIO SOPORTE DE FABRICANTES.

Los sistemas embebidos tienen soporte por una amplia gama de procesadores y arquitectura de procesadores. Actualmente existen en el mercado más de 50 fabricantes de procesadores embebidos, los cuales ofrecen más de 5000 diferentes dispositivos capaces de cubrir cualquier aplicación. Estos vendedores están en una diaria batalla para lograr que el procesador que ofrecen sea el ganador, es decir, el que uno elija para nuestra aplicación.

Por el contrario de una PC, sólo existen el mercado no más de cinco fabricantes que se disputan el mercado, entre las más importantes podemos encontrar a Intel y AMD.

1.4.3 EL COSTO SENSITIVO DE UN SISTEMA EMBEBIDO.

El costo de un sistema embebido es por lo regular bajo, pero esto se debe a muchos factores de diseño, por ejemplo, la elección del procesador adecuado y los dispositivos discretos pueden ser un factor, pero sí el diseño permite eliminar un circuito impreso, algún conector, utilizar una fuente de alimentación sencilla por usar un microcontrolador de alta integración en lugar de un microcontrolador con dispositivos periféricos separados se logrará una reducción de costos. Otro factor a considerar es la cantidad a producir; si la cantidad aumenta, algún punto de la producción se podrá abaratar; otro factor es el que derivado de la gran competencia, el mercado de semiconductores ofrece dispositivos económicos.

1.4.4 TIEMPO REAL DE LOS SISTEMAS EMBEBIDOS.

La tecnología actual nos permite diseñar sistemas embebidos de tiempo real, es decir, un sistema capaz de procesar una muestra de señal antes de que ingrese al sistema la siguiente muestra.

Un sistema embebido puede o no ser de tiempo de real dependiendo de los requerimientos específicos de la aplicación que se quiere implementar. De igual forma, el sistema será rápido sí y sólo sí se requiere de esta rapidez de acuerdo a la definición específica del sistema.

En el en el diseño de sistemas embebidos de tiempo real se deben considerar dos aspectos:

- Cuando el sistema tiene apremios por tiempo crítico, es decir, si una tarea a realizar por el sistema es de tiempo crítico, esta se debe de ubicar dentro de la ventana de tiempo o la función controlada por la tarea fallará.

- Cuando el sistema tiene apremios por tiempo sensitivo, es decir, si una tarea debiera tomar 3 segundos para realizarse pero toma, en promedio, 5 segundos, entonces tal vez el sistema será lento y no cumplirá con las especificaciones de la aplicación. Por ejemplo, si se trata de una impresora, imprimirá diez páginas por minuto en lugar de quince.

Para diseños complejos que requieran una buena administración de su tiempo, se recomienda el uso un sistema operativo de tiempo real (Real -Time Operating System, RTOS). Un RTOS le da la más alta prioridad a las tareas que necesiten para ejecutarse todo el tiempo necesario. Si alguna tarea falla al solicitar el tiempo suficiente para su ejecución, entonces es problema de programación del propio RTOS. Como los procesadores embebidos, los RTOS también tienen una alta variedad. Una cosa es segura, los errores de Software son más severos en los sistemas embebidos que en una PC.

1.4.5 POTENCIA DE LOS SISTEMAS EMBEBIDOS.

Casi todos los procesadores modernos son fabricados usando el proceso CMOS (Complementary Metal Oxide Silicon). La estructura de una compuerta simple de un dispositivo CMOS consiste en dos transistores MOS, uno de canal N y otro de canal P, conectados en configuración complementaria (de ahí el nombre), con el transistor canal N en la parte superior y el transistor canal P en la parte inferior.

Ambos transistores se comportan como interruptores perfectos. Cuando la salida es alta, o nivel lógico 1, el transistor canal P es apagado y el transistor canal N conecta la salida al voltaje de la fuente, la cual es la salida de la compuerta al resto del circuito.

Cuando el nivel lógico es cero, la situación es al revés, el transistor canal P conecta la salida a tierra mientras el transistor canal N es apagado. Esta configuración de circuito tiene una

propiedad interesante que lo hace atractivo desde un punto de vista de la energía disipada. Si el circuito está estático (sin cambios de estado), la pérdida de potencia es extremadamente baja. De hecho, debería ser cero si no fuera por una pequeña cantidad de corriente de fuga inherente en estos dispositivos, provocada por la temperatura ambiente o superior.

Cuando el circuito está cambiando constantemente como en un microprocesador, las cosas son diferentes. Mientras una compuerta cambia de niveles lógicos, hay un periodo de tiempo en donde los dos transistores están simultáneamente encendidos. Durante esta breve ventana, la corriente puede fluir de la fuente de voltaje a la línea de tierra a través de ambos dispositivos. El flujo de corriente significa potencia disipada y esto produce calor. Por lo tanto, entre mayor sea la velocidad de reloj de un procesador, más grande es el número de cambios de estados lógicos por segundo, esto significa más disipación de energía en forma de calor. Si se considera un microprocesador de 2 GHz con más de 10 millones de transistores, se comprobaría la relación casi lineal entre la velocidad y la disipación de potencia en los microprocesadores modernos. Justificable es entonces el, a veces voluminoso y complejo, sistema de enfriamiento de una PC.

1.4.6 LOS SISTEMAS EMBEBIDOS Y EL MEDIO.

Los sistemas embebidos están en cualquier lado. Los podemos encontrar en un automóvil, en un avión o en el espacio exterior; trabajando con condiciones climáticas fuera de lo normal. En la etapa de diseño de Hardware es donde se debe asegurar el funcionamiento del sistema bajo estas condiciones. Un ambiente extremo por lo regular significa mayor temperatura y humedad.

Muchos de los fabricantes de dispositivos poseen una versión que puede trabajar bajo estas condiciones, llamada de uso militar. Dichos dispositivos que son calificados para uso militar deben pasar una lista de requisitos y tener la documentación que lo apruebe, esta es la razón por la cual un dispositivo de uso militar puede llegar a incrementarse su valor miles de veces sobre el valor del de la versión de uso comercial.

1.4.7 HERRAMIENTAS EN EL DISEÑO DE SISTEMAS EMBEBIDOS.

Los sistemas embebidos son tan diferentes unos a otros en muchas formas. Es por eso que existen una gran variedad de herramientas que pueden ser usadas para crear, simular y probar tanto Software y sistemas embebidos.

El fabricante del dispositivo embebido en cuestión o a través de terceras partes, colocan en el mercado, de manera gratuita o a un costo, simuladores, emuladores, grabadores, debuggers, librerías e información que permiten realizar un diseño en menor tiempo, optimizar el sistema y muy probablemente disminuir costos.

1.4.8 CIRCUITO DE DEPURACIÓN EN LOS SISTEMAS EMBEBIDOS.

Actualmente la mayoría de microprocesadores embebidos tienen circuitería dedicada para la depuración. Quizás una de las más grandes diferencias entre los procesadores embebidos de hoy y aquellos de hace algunos años, es la casi obligatoria inclusión de la circuitería de depuración dentro del propio procesador.

1.5 EL DISEÑO DE SISTEMAS EMBEBIDOS.

El diseñador de sistemas embebidos se debe de asegurar en construir una implementación que cumpla con la funcionalidad deseada.

Un reto difícil es construir una implementación que optimice simultáneamente las características métricas del diseño. Una característica métrica es una característica medible en el diseño del sistema.

Las características métricas incluyen:

- **Costo unitario -** Es el costo monetario de manufactura de cada copia del sistema.

- **Tamaño -** Es el espacio físico requerido por el sistema y los componentes en el Hardware.

- **Costo de ingeniería no recurrente -** Es el costo monetario del diseño del sistema. Una vez que el sistema es diseñado, cualquier número de unidades puede ser manufacturada sin incurrir en algún costo adicional de diseño (de aquí el termino no recurrente).

- **Número de unidades -** La compensación entre el costo de producción y el costo de desarrollo es afectado la mayoría de veces por el número de unidades esperadas para ser producidas y vendidas. Usualmente es indeseable el desarrollo de sistemas para una producción de bajo volumen.

- **Rendimiento -** Es el tiempo de ejecución o rendimiento del sistema. Una manera muy común para medir el grado de rendimiento o potencia de procesamiento son los MIPS (Millones de Instrucciones Por Segundo). Un sistema es más poderoso en su procesamiento si ejecuta más MIPS. Sin embargo, otra característica importante necesaria a considerar es el ancho del registro, el cual tiene un rango de 8 a 64 bits. Las computadoras de propósito general o PC usan procesadores de 32 ó 64 bits exclusivamente, pero en los sistemas embebidos es común diseñar con procesadores de 8 y 16 bits.

- **Memoria -** Es la cantidad de memoria (ROM, RAM y Flash) requerida para almacenar el Software ejecutable y los datos manipulables. Aquí el diseñador de Hardware debe hacer la mejor estimación y estar preparado para incrementar o decrementar la cantidad actual, respecto al desarrollo del Software.

- **Potencia -** Es la cantidad de potencia consumida por el sistema, esto determina el tiempo de vida de la batería, o el requerimiento de un sistema de enfriamiento; mayor potencia significa más calor.

- **Flexibilidad -** Es la posibilidad de cambiar de funcionalidad del sistema sin afectar el costo de la ingeniería no recurrente. El Software es considerado muy flexible.

- **Confiabilidad -** Es que confiable en términos de funcionalidad es el sistema. Si el sistema es un juguete, no siempre tendrá que trabajar correctamente, pero si es un sistema que forma parte de un automóvil, es mejor asegurarse que el sistema realice su función correctamente, siempre y a tiempo.
- **Tiempo de vida -** Es el tiempo estimado en el cual el sistema tendrá una vida útil. Esto afecta en las decisiones de selección de componentes del Hardware, afectando obviamente el costo del sistema.

- **Tiempo de comercialización -** Es el tiempo requerido para diseñar, manufacturar y poner el sistema a la venta.

- **Seguridad -** Es que no exista alguna probabilidad de que el sistema diseñado cause daño.

- **Tiempo de pruebas y correcciones -** Es el tiempo necesario para construir una versión de trabajo del sistema, el cual puede ser más grande o más costoso que la implementación del sistema final, pero puede ser usado para verificar, corregir y refinar la funcionalidad del sistema.

Típicamente estas características métricas compiten una con otras, si una es mejorada se produce la degradación de otra. Por ejemplo, si se reduce el tamaño de una implementación, su rendimiento puede verse disminuido.

La mayoría de estas características métricas tienen un gran peso en el sistema embebido. En estos últimos años, el tiempo de comercialización ha llegado a ser especialmente exigente. Introducir un sistema embebido temprano al mercado puede marcar una gran

diferencia en los beneficios económicos, ya que las ventanas de tiempo son muy breves, casi siempre medidas en meses.

Para resolver mejor este desafió de optimización, el diseñador debe de estar familiarizado con una variedad de tecnologías de Hardware y Software para implementación de sistemas embebidos, y debe estar dispuesto a emigrar de una tecnología a otra con la finalidad de encontrar la mejor implementación.

Así pues, un diseñador no puede simplemente ser un experto en Hardware o un experto en Software, como es comúnmente en nuestros días, el diseñador debe ser un experto en ambas áreas.

1.5.1 CICLO DE VIDA DEL DISEÑO EMBEBIDO.

Distinto al diseño de una aplicación de Software para una plataforma estándar, el diseño de un sistema embebido implica que tanto el Software como el Hardware deben ser diseñados en paralelo. Aunque esto no siempre es así, hoy en día es una realidad para muchos diseños. Las profundas implicaciones de este proceso simultáneo del diseño influencian enormemente en cómo se diseñan los sistemas.

La figura 1.2 muestra una representación esquemática del ciclo de vida del diseño embebido [5].

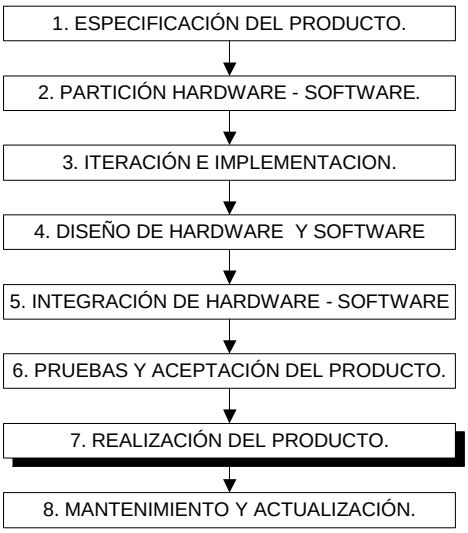


Figura 1.2 Ciclo de vida del diseño embebido.

El tiempo fluye de la parte superior a la inferior, avanzando a través de ocho fases:

- Especificación del producto.
- Partición Hardware - Software.
- Iteración e implementación.
- Diseño de Hardware y Software.
- Integración de Hardware - Software.
- Pruebas y aceptación del producto.
- Realización del producto.
- Mantenimiento y actualización.

El proceso del diseño embebido no es tan simple como lo representa la figura 1.2. Una considerable cantidad de iteraciones y optimizaciones ocurren en las fases y entre las fases. Se encontrarán defectos en los últimos pasos que casi obliguen regresar a la primera fase. Por ejemplo, cuando la prueba del producto revela un rendimiento deficiente que haga el diseño no competitivo, se tendrán que rescribir algoritmos, rediseñar Hardware, aumentar la velocidad del procesador, elegir un nuevo procesador, etc.

1.5.1.1 ESPECIFICACIÓN DEL PRODUCTO.

Esta fase de diseño es para muchos la más importante. Es en esta fase en donde se define el problema, es decir, ¿qué hará el sistema?, se justifica identificando la necesidad, En esta fase se debe establecer de una manera clara y precisa los objetivos, alcances, restricciones y límites que el sistema cumplirá; además de realizar un análisis de factibilidad precisando las características métricas como el costo, tiempo de vida, etc.

Una labor indispensable es la documentación, ésta debe incluir el estudio de los fundamentos de los problemas centrales del diseño, identificación y conocimiento de las variables que intervienen. Además de un análisis de alternativas existentes, enlistando sus características de funcionamiento, el costo, la forma, uso y tamaño.

El análisis del problema debe estar claramente separado en partes o módulos, de manera que estén bien especificados sus características deseables, sus límites y su forma de interactuar entre cada uno de ellos (por ejemplo, a través de protocolos o una señal de salida). Para cada módulo se debe identificar las diferentes alternativas de solución, valorar si se tiene o si se puede adquirir el conocimiento y la tecnología necesarios para solucionar el problema del módulo, bajo los requerimientos de solución ya establecidos.

Es de gran utilidad el elaborar un cuadro comparativo de las características, ventajas y desventajas de las soluciones posibles, además de documentar la manera en que se separó el problema, la identificación de los problemas centrales, las soluciones valoradas para cada uno de los módulos, identificación de los caminos de solución seleccionados y cuáles fueron sus criterios de selección.

Para la propuesta de solución seleccionada, se realiza un listado de las características específicas del problema que se va a atacar, y las formas en que se van a solucionar cada uno de los problemas centrales del diseño.

Por último, es la planeación de actividades, es decir, definir personal de trabajo, distribución de actividades, definición de etapas de trabajo, establecimiento de juntas para evaluación de avances y problemas, diagrama de tiempos, lista de componentes para pruebas, identificación de servicios externos, identificación de instrumentos o medidores para comparación de resultados, planeación de pruebas de funcionamiento, etc.

1.5.1.2 PARTICIÓN HARDWARE - SOFTWARE.

Puesto que un sistema embebido comprende tanto aspectos de Software como de Hardware, es en esta etapa donde la función del sistema es dividida para ser asignada al Hardware y al Software, es decir, se especifica de manera clara la función que realizará el Software y el Hardware, además de cómo y con qué recursos lo harán.

Se inicia con un análisis respecto al tamaño de los registros del dispositivo, es decir, uno de 8 bits, 16 bits, 32 bits ó hasta 64 bits; esto depende principalmente de la magnitud y precisión que se desee de los resultados que se obtienen al realizar una operación.
Posteriormente, se selecciona él o los dispositivos que realizarán las funciones principales del sistema, es decir, un PLD, un DSP, un FPGA, un microprocesador ó un microcontrolador.

Ya seleccionado que tipo de dispositivo se usará para el sistema, lo último en esta etapa es seleccionar exactamente el dispositivo, es decir, dentro de toda una familia de microcontroladores, por ejemplo, quien de ellos cumplirá exactamente con las necesidades del sistema respecto a tamaño, costo, puertos, velocidad ó periféricos.

Es también aquí en la que se analiza y se decide si un dispositivo puede sustituir parte del Software con el fin de que el dispositivo embebido realice otra función o viceversa, que el Software sustituya a un dispositivo con el fin de disminuir costos y espacio.

La decisión de particionar funciones de Hardware y Software es un problema complejo de optimización, es decir, pueden intervenir algunos de los siguientes factores en el diseño del sistema embebido:

- Costo sensitivo.
- Competitividad en el mercado.
- Desempeño.

Estos requerimientos hacen difícil crear un diseño óptimo para el producto embebido. La planeación de la partición depende de que procesador se use en el diseño. Como se ha hecho mención, se puede elegir de entre miles de microprocesadores, microcontroladores, FPGA, CPLD o DSP. La elección del dispositivo impacta en la decisión de la partición de Hardware – Software, lo cual impacta en las herramientas a utilizar, y así sucesivamente.

1.5.1.3 ITERACIÓN E IMPLEMENTACIÓN.

Esta fase, ubicada entre la partición y el diseño detallado de Hardware – Software, consiste en el inicio de la implementación del sistema.

El diseño en esta etapa es aún fluido debido a que el diseñador de Hardware y el diseñador de Software trabajan juntos, esto es con el objetivo de coordinarse en los puntos comunes del propio diseño por ejemplo, las terminales de un dispositivo hacia donde y que función ejecutarán. La iteración se presenta cada vez que sean presentadas diferentes puntos de

acuerdo entre los diseñadores de Hardware y Software hasta llegar al más adecuado para el diseño.

Después de esta etapa, las actividades individuales de Hardware y Software divergen teniendo como consecuencia una marcada división de su trabajo.

1.5.1.4 DISEÑO DETALLADO DE HARDWARE Y SOFTWARE.

Es esta la fase de diseño en la cual los diseñadores de Software y de Hardware trabajan en objetivos individuales para lograr un objetivo común.

Ambos diseñadores se apoyarán en diversas herramientas especializadas para realizar su trabajo, por ejemplo mientras el diseño de Hardware se realiza con herramientas de simulación como Pspice ó Multisim Workbench, con herramientas de diseño de PCB como Protel, OrCAD ó Eagle y con Software de diseño asistido como AutoCAD, 3D Studio ó Mechanical Desktop; el diseño del Software embebido se realiza con lenguajes de programación como ensamblador, C, C++, Basic, VHDL, etc., con tarjetas de evaluación las cuales contienen el dispositivo elegido para el diseño y la cual permite evaluar el funcionamiento del dispositivo con el programa que se este diseñando, con emuladores y depuradores que son también de gran utilidad y casi indispensable para el diseño ya que permiten la ejecución del programa línea por línea permitiendo la facilidad de ubicación de algún error en caso de que existiera; y desde luego el programador del dispositivo. Importante es mencionar que algunas herramientas, principalmente de Software, son proporcionadas gratuitamente o de manera limitada por el fabricante del dispositivo; mientras otras herramientas tienen algún costo y en muchos casos son fabricadas por terceras partes.

1.5.1.5 INTEGRACIÓN HARDWARE - SOFTWARE.

La integración Hardware – Software es la fase del desarrollo en la cual el trabajo realizado por los diseñadores de Hardware y Software se une para conformar el sistema al cual se quiere obtener.

El Software embebido o Firmware es programado en el Hardware diseñado, con esto se logrará un funcionamiento real de ambas partes.

Esta etapa tiene como herramienta principal el depurador o debug, el cual permite que el Software se pueda ejecutar línea a línea, logrando con esto observar los valores y cambios en los registros y en las variables de entrada y salida del sistema, pero principalmente detectar la ubicación exacta de algún error, si éste se llegara a presentar.

Es a partir de esta etapa en la que se esperan que los resultados sean lo más parecido a los simulados y a los planteados al principio de la proyecto, y es también en esta fase en la cual se inicia un ciclo de iteraciones y optimizaciones con el fin de llegar a los resultados esperados.

1.5.1.6 PRUEBA Y ACEPTACIÓN DEL PRODUCTO.

La prueba del producto toma un especial significado cuando su éxito o fracaso dependen del funcionamiento del sistema embebido.

Puede causar indiferencia el que ocasionalmente la impresora imprima mal una hoja ó que tengamos que reiniciar la computadora, pero si esta computadora forma parte de un sistema de alta seguridad, control de emergencia en una planta industrial o un manejador de base de datos, se estará en graves problemas. Por lo tanto el sistema de pruebas requeridas para un sistema embebido son mucho más estrictos para algunas aplicaciones que para otras, esto depende de donde será utilizado el producto.

Es recomendable que la serie de pruebas las realicen un grupo separado de técnicos o ingenieros, ya que pedirle al diseñador que pruebe su propio código o producto generalmente se obtienen malos resultados por realizar pruebas erráticas.

La figura 1.3 muestra como el costo a través del proceso de diseño es exponencial, debido a que entre más avanzada sea la etapa en donde fue localizado un error, más son las etapas que se tienen que retroceder para localizar el origen del error, provocando pérdidas económicas y tiempo. Por la tanto, el costo de los errores que se lleguen a presentar dentro del proceso de diseño, será diferente de a cuerdo a la fase en que se ubiquen.

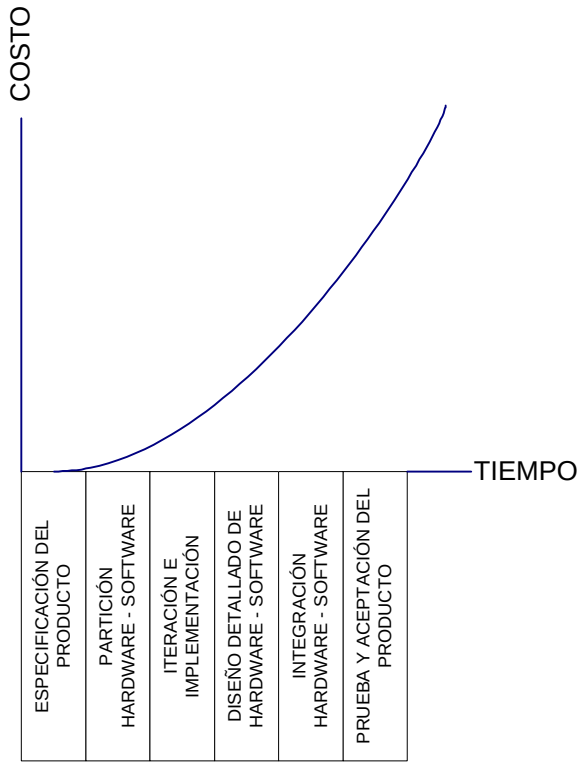


Figura 1.3 Relación costo – fase de diseño.

1.5.1.7 REALIZACIÓN DEL PRODUCTO.

Una vez que el prototipo ha superado todas las pruebas establecidas por el propio fabricante, se han corregido los errores y se han planteado estrategias de ventas, es tiempo de fabricar el producto para su colocación en el mercado, es decir, las fases que se refieren al diseño concluyeron. Es en esta fase en la cual se elaboran los manuales de usuario y los manuales técnicos (manual de servicio, diagramas, etc.).

1.5.1.8 MANTENIMIENTO Y ACTUALIZACION.

Una estrategia que algunas empresas del mercado de equipo electrónico tienen para mantener cautivo a sus consumidores es ofrecer, después de la venta del producto, una garantía que respalda por un tiempo limitado la calidad con la cual fue pensada durante el proceso de diseño. Pero posterior a esto es común que también se ofrezca un servicio de mantenimiento preventivo y correctivo; pero otras empresas van mucho más allá, ofrecen actualizaciones de Software y de Hardware. Un ejemplo muy claro es la industria de la telefonía celular, aún cuando la vida útil de un teléfono celular no es mayor de un año y medio, ponen a disposición del consumidor toda una gama de productos que van desde videojuegos, imágenes, sonidos, expansiones de memoria, accesorios y servicios.

1.6 EL ESTADO DEL ARTE.

La computación y la electrónica son dos ciencias que cuando alguna de ellas evoluciona, va implícita la evolución de la otra; son de las ciencias que más han influenciado en la humanidad, provocando que los últimos cincuenta años se hayan catalogado, por sus descubrimientos científicos y tecnológicos, como el periodo más vanguardista de la historia.

Todo esto se hace presente en nuestros días a manera que tanto el Hardware y Software para sistemas embebidos evolucionen rápidamente. Es posible encontrar en un solo dispositivo las características de velocidad, tamaño, economía, tamaño de palabra de procesamiento, bajo consumo de energía. Un ejemplo es el DSP, ya que puede realizar más de 100 MIPS, a un costo de $15.00 dólares y en menos de 1 cm^2 de área.

En lo referente al Software, es posible observar como, día a día se posicionan en el mercado aquellos que ofrecen al diseñador una interfaz accesible y que principalmente cuente con las suficientes herramientas para lograr diseños rápidos y eficaces.

Una opción por la cual se están inclinando los desarrolladores de sistemas embebidos es el tener en un solo Software el diseño embebido, diseño de PCB, simulación electrónica, simulación mecánica, simulación térmica, y diseño asistido. Como respuesta a esta necesidad empresas como Nacional Instruments ponen a disposición un Software que reúne muchas de estas herramientas.

Por último, un fenómeno mundial que en los últimos años se ha hecho presente, y que se espera continué por muchos más, es el abaratamiento de la tecnología, provocado principalmente por la amplia competencia y por la alta demanda del consumidor. Esto tiene como consecuencia que un diseñador independiente le cueste trabajo colocar en el mercado un diseño embebido propio, no así a una empresa con tiempo y calidad reconocido por el consumidor. Pero a pesar de esto, México, debido a su no consolidado desarrollo tecnológico, es un lugar en donde el diseño electrónico es todavía un área muy poco explorada y con grandes posibilidades de consolidar una empresa que pueda ofrecer calidad e innovación.

1.7 CONCLUSIONES CAPITULARES.

Los equipos y sistemas embebidos han surgido con la misma velocidad con que nacen nuevas necesidades, y no hay área donde al menos un problema o una necesidad sea cubierta por un sistema embebido.

El mercado de los dispositivos embebidos es tan amplio que permite cubrir, con su amplia gama de productos, desde un simple problema en el hogar hasta problemas tan complejos como los que se presentan en el área militar y espacial, en donde los sistemas son sometidos a condiciones extremas.

Un diseño embebido es un proceso complejo y que de acuerdo a las variables físicas y/o eléctricas que en él intervengan, su complejidad aumentará. Pero es el correcto planteamiento del problema y del producto en la fase de diseño, la necesidad a cubrir, la originalidad e innovación los que determinan que un sistema o producto sea exitoso.

2. CONECTIVIDAD DE SISTEMAS EMBEBIDOS A REDES LAN E INTERNET.

OBJETIVO: Describir las características de cada una de las formas que permiten conectar un sistema embebido a una red LAN e Internet, resaltando principalmente las ventajas y desventajas de cada método.

2.1 CONECTIVIDAD DE SISTEMAS EMBEBIDOS A REDES LAN E INTERNET.

En los inicios de la Internet, se pensaba que sólo sería utilizada para el intercambio de información entre computadoras conectadas a ella, esa idea ha cambiado.

El surgimiento de nuevas necesidades en áreas como la domótica, el control y monitoreo a distancia, entre otras, se han venido solucionando a través de sistemas embebidos, los cuales requieren de alguno de las medios de comunicación existentes.

Ethernet es una buena opción porque es de rendimiento competitivo, además de poderse implementar con un bajo costo. Ethernet es fácil de usar, es ampliamente disponible y tiene una estable y escalable infraestructura. Ethernet es descrita por el estándar IEEE 802.3.

Con Ethernet, dispositivos embebidos pueden ser conectados a Internet lo cual le permitirá acceder a otros dispositivos embebidos alrededor del mundo. Predicciones afirman que en próximos años el número de usuarios de computadora en la Internet se vera rebasado por los usuarios que no requieren una PC para conectarse. [6]

Hasta nuestros días, existen diferentes métodos para poder conectar un sistema electrónico a una red de cómputo, los hay desde los que requieren una PC para su funcionamiento hasta los que pueden trabajar de manera independiente y comunicarse entre ellos.

La conectividad de sistemas obedece a algún protocolo, como OSI ó TCP/IP, pueden ser implementados para un amplio rango de aplicaciones, incluyendo:

- Consulta o accesos a bases de datos.
- Servidores Web para dispositivos embebidos.
- Monitoreo remoto (diagnostico y recolección de datos).
- Control remoto de dispositivos en el campo.
- Uso de correo electrónico por dispositivos remotos.
- Reprogramación remota de memoria flash.

Los dispositivos de alta integración que actualmente ofrece el mercado de los semiconductores pueden manejar el control funcional de estas aplicaciones, más sin embargo es la adecuada selección del dispositivo y su correcta programación lo que permitirá un exitoso sistema embebido.

El método que se elija para conectar el sistema embebido con la red nos indica también el tipo de dispositivo que se elegirá entre la amplia variedad existente, ya que hay desde dispositivos con pocos recursos y prestaciones, en los cuales para conectarlos a la Internet se les deben implementar la pila de protocolo, hasta los que ya cuentan con un módulo Ethernet y que tan sólo requieren unos cuantos componentes externos para su funcionamiento.

2.2 MÉTODOS DE CONEXIÓN A UNA RED.

Dependiendo la necesidad, los recursos, los costos y los planes a futuro del sistema, existe una variedad de formas mediante las cuales es posible conectar a Internet un sistema embebido, estos métodos van desde el utilizar una computadora, con la cual se aprovechan los protocolos incluidos en el sistema operativo para conectarlo al sistema electrónico, hasta sistemas totalmente autónomos en los cuales el mismo microcontrolador incluye un módulo Ethernet.

2.2.1 LA COMPUTADORA COMO MEDIO DE CONEXIÓN.

Para esta forma de conexión, el sistema requiere de al menos dos computadoras, una que actúa como servidor (Server) y otra que actúa como monitor (Host) y a las cuales se les conecta vía puerto USB, serial ó paralelo el sistema embebido, es decir, se podrán enviar comandos y recibir información referente al estado del sistema electrónico.

Dado que no existe un límite de direcciones IP en el Internet, se pueden conectar N número de sistemas. La figura 2.1 nos muestra un sistema en el cual la computadora es la que realiza la función principal.

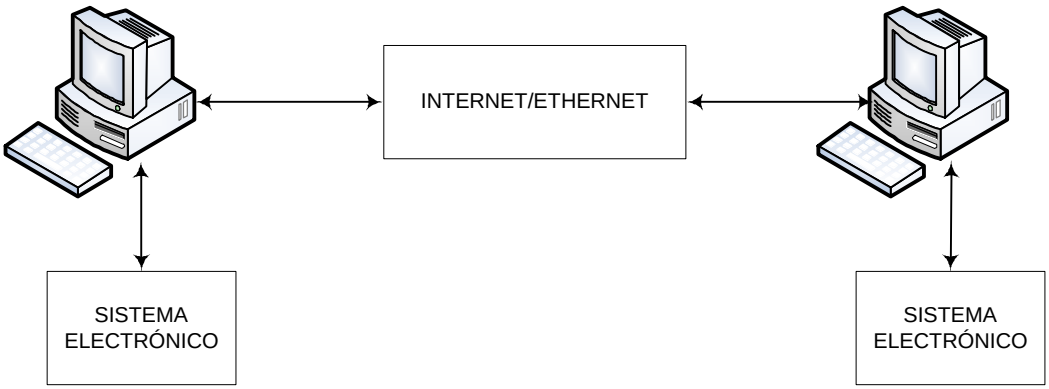


Figura. 2.1 La computadora como medio de conexión.

2.2.1.1 LA COMPUTADORA.

En esta forma de conexión, las computadoras tienen la función más importante del sistema, ya que son ellas las que realizan la comunicación Ethernet; de ellas podemos utilizar su capacidad de procesamiento, su velocidad, su capacidad de almacenamiento y principalmente su sistema operativo en el cual están implementados todos los protocolos que se requieren para una conexión a Internet.

2.2.1.2 EL SISTEMA ELECTRÓNICO.

Para este tipo de conexión, el sistema electrónico puede estar diseñado con casi cualquier tipo de microcontrolador, CPLD, FPGA, etc., dándole preferencia a aquellos que contengan algún modulo de comunicación serial, por ejemplo UART, USART ó USB; ya que será el medio por el cual se comunique el sistema electrónico con la computadora Server, además del número adecuado de puertos necesarios y propios para la aplicación. Dado que la

computadora es la encargada de la comunicación, la velocidad del dispositivo controlador es tan sólo importante para la adquisición y procesamiento de datos.

El diseño y complejidad del sistema electrónico variará de acuerdo a las necesidades, los recursos y al dispositivo controlador elegido.

2.2.1.3 EL SOFTWARE.

Este método, por su arquitectura, implica el uso de dos diferentes tipos de Software, el de la computadora Server y el de la computadora Host.

Para la computadora Server se utiliza un programa servidor, el cual permite conectar el sistema embebido por medio de uno de sus puertos a la PC, además de comunicar través de una dirección IP, vía Ethernet, a la computadora Host.

Para el caso de la computadora Host es relativamente más sencillo ya que se puede utilizar una herramienta de Windows llamada Telnet [7], por medio de la cual es posible enviar y recibir datos hacia la computadora Server y de esta manera monitorear y controlar el funcionamiento del sistema embebido de manera remota. También es posible implementar el Software, para el Server y el Host, con algún lenguaje de programación por ejemplo, C, C++ o Visual Basic

2.2.1.4 VENTAJAS Y DESVENTAJAS.

Existen tres claras desventajas de este método, la primera es que los sistemas electrónicos que se conecten carecen de autonomía debido a su total dependencia de una computadora; la segunda desventaja es el utilizar al menos dos computadoras para llevar a cabo la comunicación, eleva considerablemente el costo del sistema; y la tercera desventaja, su tamaño.

Como ventaja se puede observar que al utilizar una computadora es posible aprovechar sus ventajas de procesamiento, almacenamiento y velocidad para poder realizar tareas más complejas como por ejemplo bases de datos o páginas Web; otra ventaja, que permite diseñar tanto al sistema embebido como su Firmware de una forma relativamente sencilla, es que la computadora realiza el proceso de comunicación a través de su sistema operativo.

2.2.2 LA TARJETA DE RED COMO MEDIO DE CONEXIÓN.

Para esta forma de conexión se requiere de una tarjeta de red Ethernet (Network Interface Card, NIC) para realizar la conexión a una red LAN o a Internet.

El sistema embebido debe contar con el número de puertos adecuados, bus de datos, direcciones, lectura, escritura y reset; para poder conectar la tarjeta. Se pueden conectar n número de sistemas o Servers, cada uno de ellos con su respectiva e irrepetible dirección IP; además de una sola computadora (Host) que se encargará de monitorear y controlar a los n Servers que se conecten. [8]

La figura 2.2 nos muestra como en esta arquitectura es la tarjeta de red la responsable de la interfaz entre el sistema embebido y la red LAN e Internet.

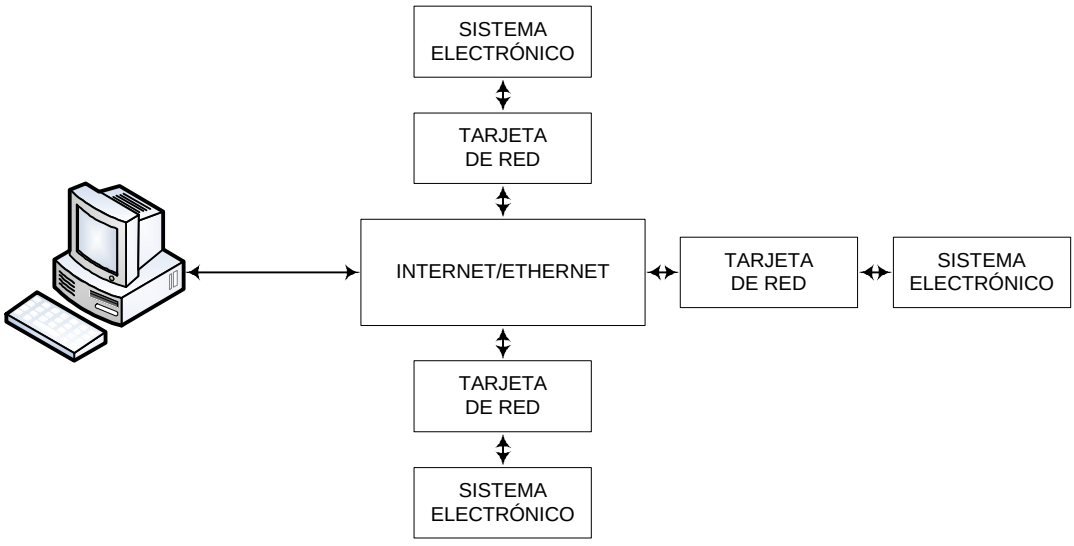


Figura 2.2 La tarjeta de red como medio de conexión.

2.2.2.1 LA TARJETA DE RED.

El dispositivo controlador del sistema embebido se comunica a la red por medio de lecturas y escrituras a registros internos de la tarjeta de red. Dicho dispositivo debe utilizar Acceso Directo a Memoria (Direct Memory Access, DMA) para acceder a la RAM de la tarjeta de red. Antes de transmitir o recibir datos a través de la red, la tarjeta debe ser inicializada o configurada. Este proceso de inicialización consiste en obtener o cambiar la dirección IP del controlador de acceso al medio (Media Access Controller, MAC). La dirección está ubicada en los primeros doce bytes de la primera página (página 0) de la RAM (por alguna razón, aparenta que cada byte es almacenado dos veces de manera consecutiva, cada segundo byte se debe ignorar para leer su dirección correcta, es decir, la dirección es almacenada sólo en seis bytes). Una dirección es asignada a cada tarjeta por su fabricante, donde los dos primeros bytes lo identifican, pero se puede asignar una diferente.

2.2.2.2 EL SISTEMA ELECTRÓNICO.

El sistema embebido puede estar basado en cualquier microcontrolador, PLD, FPGA, etc. que tenga como mínimo 16 terminales o puertos de los cuales ocho son para los datos, cinco para direccionar memoria y tres para lectura, escritura y reset del controlador de red incluido en la tarjeta de red, además de las terminales propias de la aplicación.

Como este método de conexión es totalmente autónomo, ya que no depende de otro sistema como en el método antes mencionado, que requiere el uso de una computadora, es recomendable un dispositivo controlador con una velocidad de procesamiento relativamente alta, ya que será este quien realice el proceso de la comunicación, es decir, él ejecutará la pila del protocolo de comunicación además de realizar la tarea específica del sistema, adquisición de datos por ejemplo.

La aplicación, los recursos y el dispositivo controlador determinarán el diseño y complejidad del sistema.

2.2.2.3 LA COMPUTADORA.

Para este caso, la computadora (Host) juega un papel secundario, es decir, en ella puede recaer el monitoreo (administra el estado de cada Server, recibiendo información de cada uno de ellos) y control (envía comandos a cada Server para que realicen una acción especifica); pero también se puede omitir su uso debido a la autonomía del sistema electrónico, la cual depende de la programación y necesidades de la aplicación.

El uso de una computadora también dependerá de la aplicación, es decir, se pueden aprovechar sus características ya antes mencionadas para poder obtener más y mejores resultados.

2.2.2.4 EL SOFTWARE.

El Software utilizado para este tipo de conexión puede estar implementado con un programas Server y TELNET, como el utilizado en el método anterior, pero también es posible utilizar programas de control y monitoreo como LabVIEW ó páginas Web elaboradas con alguna herramienta como HTML, Java ó alguna .NET, en las cuales los botones y ventanas nos permiten un fácil monitoreo y control del sistema remoto.

El Software, cuando se utilice computadora y la aplicación lo requiera, se puede conjugar su funcionamiento con un protocolo de comunicación destinado para aplicaciones industriales o Field bus, por ejemplo, MODBUS, CAN ó Estándares de Comandos Para Instrumentos Programables (Standard Commands for Programmable Instruments, SCIP), definido en el estándar IEEE 488.2.

2.2.2.5 VENTAJAS Y DESVENTAJAS.

En este método las ventajas son más significativas, nos permite realizar sistemas totalmente autónomos, no es indispensable para su funcionamiento una computadora, por lo tanto es económico. Como desventajas encontramos que el dispositivo controlador es el que realiza tanto el proceso de comunicación como el de la aplicación, por lo tanto se requiere que sea de características de alto desempeño como memoria, velocidad, disponibilidad de puertos y herramientas de programación accesibles las cuales permitan mantener la ventaja de la economía.

2.2.3 INTERFAZ DE CONTROL DE RED (NIC) COMO MEDIO DE CONEXIÓN.

Este modo de conexión es sólo una variación del método anterior, la diferencia radica en que aquí se conecta sólo el circuito que contiene la tarjeta de red, llamado Interfaz de Control de Red (Network Interface Controller, NIC). La forma de conexión del circuito controlador hacia el NIC es exactamente igual al explicado anteriormente.

El sistema, estará igualmente compuesto por n sistemas electrónicos (Servers) y de manera opcional una computadora (Host).

La figura 2.3 nos muestra la manera de conexión cuando se utiliza un NIC como interfaz de comunicación.

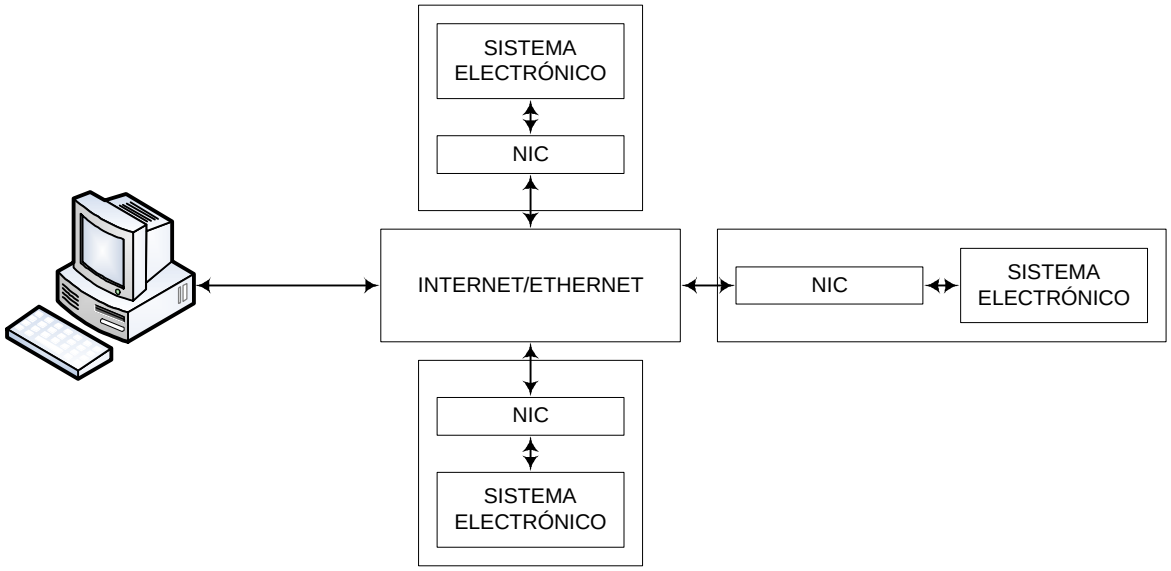


Figura 2.3 Arquitectura del sistema cuando se utiliza un NIC como medio.

2.2.3.1 INTERFAZ DE CONTROL DE RED (NIC).

Este es un dispositivo de muy alta escala de integración, diseñado para una fácil interfaz con una red LAN tipo CSMA/CD incluyendo Ethernet. El NIC implementa todas las funciones del layer de Control de Acceso al Medio (Media Access Control, MAC) para la transmisión y recepción de paquetes de acuerdo con el estándar IEEE 802.3. Cuenta con canales de Acceso Directo a Memoria (DMA) y FIFO que permite un diseño simple pero eficiente para el manejo de los paquetes.

2.2.3.2 SISTEMA ELECTRÓNICO.

El sistema electrónico cumple con las mismas características del método anterior, es decir, se puede implementar con cualquier microcontrolador, CPLD, FPGA, etc, que cuente con al menos 8 terminales para el puerto de datos de entrada y salida del NIC, 5 par direccionar y tres para las terminales de lectura, escritura y reset, además de los que se utilizarán para la propia aplicación.

Todos los componentes son alimentados con 5V, lo cual permite el diseño de un sistema aun más compacto, a diferencia que si fuera hecho con una tarjeta de red.

2.2.3.3 LA COMPUTADORA.

En este método, el uso de la computadora (Host) y el papel que esta desempeña en el sistema total es exactamente el mismo que el método anterior, es decir, no es indispensable su presencia a menos que la aplicación la requiera; con ella se puede monitorear y controlar a cada Server. Se sigue aprovechando la autonomía del sistema electrónico; con esto es posible, como ya se ha mencionado, realizar un sistema de control distribuido.

2.2.3.4 EL SOFTWARE.

El Software, al igual que el método anterior, no cambia. El criterio para su diseño sigue siendo el mismo. Se pueden utilizar desde sencillos programas como Telnet de Windows hasta una elaborada página Web y en la cual, con un algoritmo adecuado, se implemente un protocolo para comunicaciones industriales.

2.2.3.5 VENTAJAS Y DESVENTAJAS.

Al presente método de conexión ya es posible agregarle la característica, indispensable en estos últimos años, la miniaturización y el bajo consumo de energía eléctrica para realizar su trabajo. Sumándole las características anteriormente vistas como el de ser un sistema totalmente autónomo y económico, ya se está haciendo referencia a un sistema eficiente.
Las desventajas siguen siendo las mismas del método de conexión anterior.

2.2.4 EL FIRMWARE DEL SISTEMA EMBEBIDO COMO MEDIO DE CONEXIÓN.

Este método se basa fundamentalmente en el Software que se diseñe para el dispositivo de control, es decir, será en el Firmware en donde sean implementados en su totalidad los protocolos de Internet.

Un gran sistema operativo usualmente tiene protocolos de Internet ya integrados, mientras los sistemas embebidos aun no tienen provisto una pila de protocolo.
La diferencia física que podemos encontrar respecto a los dos métodos anteriores es que en este método es un sólo circuito, microcontrolador, CPLD ó FPGA; el que realice todas las funciones. El Firmware es el que realiza el papel más importante.

El sistema, estará compuesto por n sistemas electrónicos (Servers) y de manera opcional una computadora (Host). La figura 2.4 nos muestra el esquema de un sistema cuando es un sólo circuito y su Software los encargados de realizar una comunicación vía Internet.

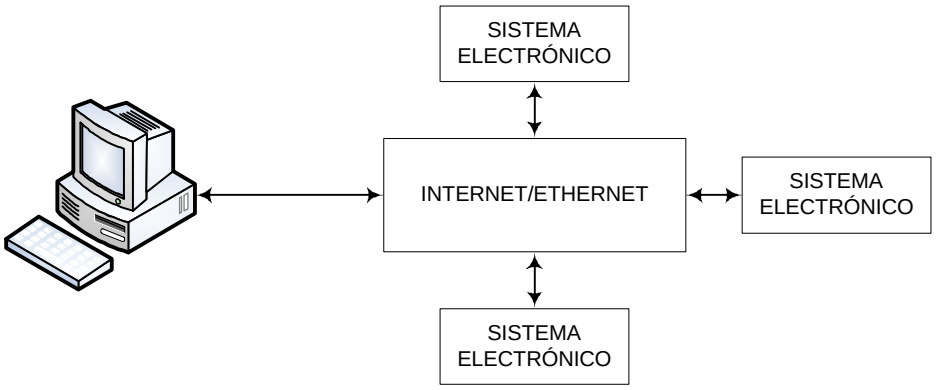


Figura 2.4 Arquitectura de un sistema cuando un sólo dispositivo realiza la comunicación.

2.2.4.1 EL DISPOSITIVO DE CONTROL.

El dispositivo de control que se seleccione, PLD, microcontrolador o FPGA; será el encargado de realizar el proceso de comunicación, en él será programado toda una serie de algoritmos que como resultado ofrezcan una pila de protocolo OSI ó TCP/IP.

Las características que debe poseer el dispositivo es velocidad, capacidad de memoria, preferentemente de 16 o más bits; para que el sistema sea práctico se recomienda que sea programable en sistema y con herramientas de programación económicas y accesibles.

2.2.4.2 LA COMPUTADORA.

Para este tipo de conexión la funcionalidad e importancia que posee la computadora no varia en lo absoluto respecto a los dos métodos anteriores, es decir, no es indispensable, es utilizada opcionalmente para control y monitoreo o cuando la aplicación la requiera.

2.2.4.3 EL SISTEMA ELECTRÓNICO.

La parte de Hardware es la que más cambios tiene en este método, ahora sólo es un dispositivo el que conforma al sistema. Definitivamente es ya todo un sistema compacto.

La elección del dispositivo es más rigurosa, ya que en el recae toda la labor del sistema en lo que respecta a comunicación, interfase de red y la propia aplicación. Por lo tanto se recomienda un dispositivo con amplias características de memoria, velocidad, puertos y de herramientas adecuadas que permitan conservar una filosofía de sistema compacto y económico.

2.2.4.4 EL SOFTWARE.

La programación para la computadora no cambia para este método respecto a los dos anteriores. El Firmware del dispositivo es el que cambia, ya que como se ha mencionado, es él que a través de sus algoritmos realizan la comunicación y la ejecución de acciones propias de la aplicación, es decir, para este método el Firmware es más complejo.

2.2.4.5 VENTAJAS Y DESVENTAJAS.

En esta forma de conexión, son las ventajas las que más sobresalen. Se puede destacar principalmente la miniaturización, la autonomía del sistema y el bajo consumo de corriente al tratarse sólo de un dispositivo.

La desventaja es seleccionar el dispositivo adecuado que pueda ejecutar el algoritmo del protocolo de comunicación, la interfase de red y la aplicación, es decir, su Firmware se vuelve complejo.

2.2.5 DISPOSITIVOS CON MAC Y TRANSCEIVER COMO MEDIO DE CONEXIÓN.

Una última generación de microcontroladores diseñados para cubrir las necesidades de conectividad en Internet de sistemas embebidos son aquellos a los cuales les son integrados módulos MAC (Media Access Controller, Controlador de Acceso al Medio), los cuales cumplen con la especificación de Ethernet 802.3, además de su módulo de EPHY (Ethernet Physical Transceiver, Transmisor-Emisor Físico de Ethernet) que permiten una fácil conexión física.[9]

De todos los métodos, este se puede clasificar como el más eficiente. Sus ventajas tanto en Hardware y en Firmware nos permiten realizar conexiones con pocos recursos y a un bajo

costo. Es posible implementar un nodo de red en tan sólo una pequeña tarjeta de tan sólo 4 x 4 cms.

La figura 2.5 muestra como la arquitectura del sistema con dispositivos que incluyen MAC y transceiver es la misma que si se utilizaran dispositivos en los cuales el Firmware es el que realiza, a través de algoritmos, los protocolos de comunicación.

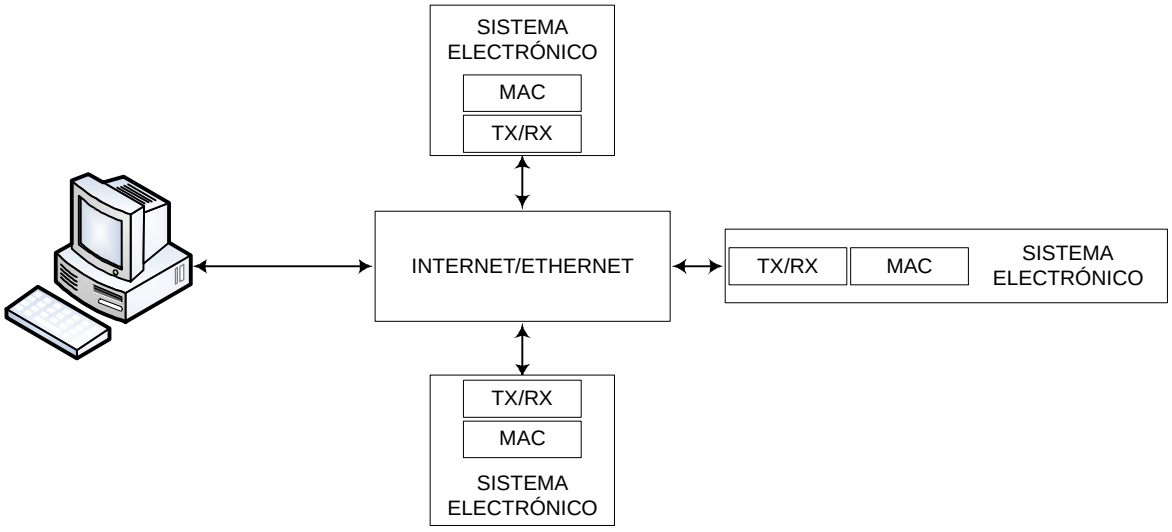


Figura 2.5 Arquitectura del sistema con dispositivos con MAC y transceiver incluidos.

2.2.5.1 EL DISPOSITIVOS DE CONTROL.

Es ya este tipo de microcontroladores los que permiten una conexión de una manera más práctica, sus módulos MAC y transceiver son configurados a través de algunos registros internos de RAM, es decir, el Firmware para la comunicación es considerablemente reducido debido a que los algoritmos de los protocolos de comunicación no son en su totalidad implementados en él.

Estos microcontroladores de alto desempeño además de contar con el módulo Ethernet, cuentan con las principales prestaciones de cualquier microcontrolador como son los convertidores analógico-digital, digital-analógico, comunicación serial, PWM, captura, interrupciones, timers, puertos, cantidades cada vez más grandes de memoria, etc. [10]

2.2.5.2 LA COMPUTADORA, EL SOFTWARE Y EL SISTEMA ELECTRÓNICO.

Para esta forma de conexión la computadora, el Software y el sistema electrónico no cambian respecto al método anterior; la computadora seguirá teniendo la función de control y/o monitoreo, no es indispensable su presencia a menos que la aplicación así lo requiera; su Software no sufre cambios, es implementado por algún lenguaje de programación para Internet como HTML, JAVA, o alguno de control como LabVIEW, etc., y el sistema electrónico físicamente tampoco tiene cambios, es implementado con un sólo componente.

2.2.5.3 VENTAJAS Y DESVENTAJAS.

En esta forma de conexión las ventajas son ampliamente superadas por las desventajas, además de la miniaturización y autonomía del sistema que ya se venían tratando, ahora se le suma el que el Firmware se realiza de una manera más simple, ya que es aprovechado el módulo de comunicación Ethernet. Es sin duda éste el método más adecuado para realizar el enlace de sistemas embebidos usando Internet como vía de comunicación.

2.3 CONCLUSIONES CAPITULARES.

Los avances tecnológicos, en electrónica y computación, nos ofrecen distintas formas de conectar un sistema embebido al mundo exterior; pero el mejor de los métodos para conectarlo a una red LAN o a Internet es aquel que se adapte más los objetivos del sistema, para esto se debe que tomar en cuenta el costo, tamaño, versatilidad, prestaciones y necesidades futuras, además de un profundo estudio de las ventajas y desventajas de los otros métodos.

Son también, el Hardware y Software, los que permiten implementar desde sencillos sistemas a base de comandos con Telnet, hasta sistemas complejos, por ejemplo un Web Server embebido, en el cual es posible la adquisición, graficación, modificación, análisis y bases de datos.

3. ETHERNET Y EL MODELO OSI.

OBJETIVO: Describir como uno de los modelos de referencia más populares, modelo OSI y sus protocolos en capas, además de la estandarización IEEE 802.3, la tecnología encargada de la interconexión física de computadoras en una red conocida como Ethernet, consigan que las redes de cómputo, además de permitir compartir recursos de hardware y software entre computadoras, sean un medio adecuado para interconectar sistemas embebidos.

3.1 ETHERNET Y EL MODELO OSI.

En 1972 comenzó el desarrollo de una tecnología de redes conocida como Ethernet Experimental. El sistema Ethernet desarrollado, conocido en ese entonces como red ALTO ALOHA, fue la primera red de área local (LAN) para computadoras personales. Esta red funcionó por primera vez en mayo de 1973 a una velocidad de 2.94Mb/s.

Las especificaciones formales de Ethernet de 10 Mb/s fueron desarrolladas en conjunto por las corporaciones Xerox, Digital (DEC) e Intel, y se publicó en el año 1980. Estas especificaciones son conocidas como el estándar DEC-Intel-Xerox (DIX), el libro azul de Ethernet. Este documento hizo de Ethernet experimental operando a 10 Mb/s un estándar abierto. La tecnología Ethernet fue adoptada para su estandarización por el comité de redes locales (LAN) de la IEEE como IEEE 802.3. El estándar IEEE 802.3 fue publicado por primera vez en 1985.

El estándar IEEE 802.3 provee un sistema tipo Ethernet basado, pero no idéntico, al estándar DIX original. El nombre correcto para esta tecnología es IEEE 802.3 CSMA/CD, pero casi siempre es referido como Ethernet, la cual fue adoptada por la organización internacional de estandarización (ISO), haciendo de el un estándar de redes internacional.

Ethernet continuó evolucionando en respuesta a los cambios en tecnología y necesidades de los usuarios. Desde 1985, el estándar IEEE 802.3 se actualizó para incluir nuevas tecnologías. Por ejemplo, el estándar 10BASE-T fue aprobado en 1990, el estándar 100BASE-T fue aprobado en 1995 y Gigabit Ethernet sobre fibra fue aprobado en 1998.

Ethernet es una tecnología de redes ampliamente aceptada con conexiones disponibles para PC, estaciones de trabajo científicas y de alta desempeño, mini computadoras y sistemas mainframe.

Ethernet provee detección de errores pero no corrección de los mismos.
Tampoco posee una unidad de control central, todos los mensajes son transmitidos a través de la red a cada dispositivo conectado. Cada dispositivo es responsable de reconocer su propia dirección y aceptar los mensajes dirigidos a ella.

Ethernet ha sido uno de los protagonistas en la evolución de las redes de cómputo.
Se entiende como red de cómputo a una colección interconectada de computadoras autónomas. Se dice que dos computadoras están interconectadas si son capaces de intercambiar información. La conexión puede ser de alambre de cobre, fibra óptica, microondas y satélites de comunicación. [11]

Si bien este concepto es demasiado general, nos sirve como punto de partida. La información que pueden intercambiar las computadoras de una red puede ser de lo más variada: correos electrónicos, vídeos, imágenes, música en formato MP3, registros de una base de datos, páginas Web, etc.

En la definición anterior se ha indicado el término computadora en un intento por simplificar. Sin embargo, las computadoras son sólo una parte de los distintos dispositivos electrónicos que pueden tener acceso a las redes, en particular a Internet. Otros dispositivos de acceso son los asistentes digitales personales (PDA), las televisiones (Web TV), PLC y tarjetas de adquisición de datos (DAQ).

Los dispositivos que forman las redes hablan entre sí mediante protocolos y para que se entiendan, deben de usar los mismos. Estos, son la base del intercambio de información entre dispositivos, que es uno de los principales objetivos de las redes.

Según el modelo OSI, se define como protocolo a aquel conjunto de reglas y formatos que gobiernan las comunicaciones entre dispositivos que ejecutan funciones a un mismo nivel en diferentes sistemas abiertos. [12]

Por tanto, se llama protocolo al conjunto de normas que se usan para estructurar los paquetes que contiene la información a transmitir.

Dado que se está hablando de redes digitales, la información y estructura de los protocolos siempre es binaria, es decir, está formada por unos y cero.

3.2 ETHERNET.

Ethernet define las características de cableado y señalización de nivel físico y los formatos de trama del nivel de enlace de datos del modelo OSI, es decir, el término Ethernet se refiere a la tecnología de productos de redes LAN (Redes de Área local) que cubren el estándar IEEE 802.3, el cual define lo que comúnmente es conocido como el protocolo CSMA/CD. [13]

Aunque otras tecnologías y protocolos han sido ofrecidos como un reemplazo, por ejemplo Token Ring 802.5, Fast Ethernet, FDDI, ATM y LocalTalk, Ethernet ha sobrevivido como la mayor tecnología LAN (85% de las computadoras en el mundo se conectan vía LAN) por que este protocolo tiene las siguientes características:

- Es fácil entender, implementar, manejar y mantener.
- Permite la implementación de una red a bajo costo.
- Posee una flexibilidad en topologías para la instalación de redes.
- Ha sido adoptado por un gran número de fabricantes.

3.2.1 ELEMENTOS UTILIZADOS EN UNA RED ETHERNET.

Las redes Ethernet consisten en nodos de red y el medio de interconexión. Los nodos de red se clasifican en dos grupos:

- **Equipo Terminal de Datos (DTE)** – Dispositivos que pueden ser la fuente o el destino de la trama de datos. Son dispositivos como computadoras personales, estaciones de trabajo, servidores de archivos o de impresión.

- **Equipo de comunicación de Datos (DCE)** – Dispositivos de red intermediarios que reciben y envían las tramas de datos a través de la red. Pueden ser dispositivos independientes como repetidores, conmutadores, ruteadores y concentradores, o unidades de interfase de comunicación como tarjetas de interfase de red (NIC) y módems.

El medio de interconexión actual de Ethernet es dos tipos de cable de cobre: el par trenzado sin blindaje UTP (Unshielded Twisted Pair) y el par trenzado con blindaje STP (Shielded Twisted Pair), además de algunos tipos de cable de fibra óptica. [14]

3.2.2 TOPOLOGÍAS DE REDES ETHERNET.

Las redes LAN toman muchas configuraciones topológicas, pero independientemente de su tamaño o complejidad, todas serán una combinación de sólo una de las tres estructuras básicas de interconexión.

La estructura más simple es la interconexión punto a punto, mostrada en la figura 3.1. Sólo dos unidades de la red son involucradas, y la conexión puede ser DTE a DTE, DTE a DCE o DCE a DCE. El cable en una interconexión punto a punto es conocido como una conexión de red o link. El máximo largo permitido para la conexión depende del tipo de cable y del método de transmisión que se use.

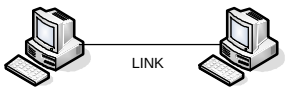


Figura 3.1 Interconexión Punto a Punto.

La red original Ethernet fue implementada con una estructura de bus coaxial, mostrada en la figura 3.2. El largo del segmento de bus fue limitado a 500 metros y hasta 100 estaciones pueden ser conectadas en el. Segmentos individuales pueden ser interconectados con repetidores, tanto múltiples como largas rutas no existen entre dos estaciones en la red y el número de DTE no excede los 1024.

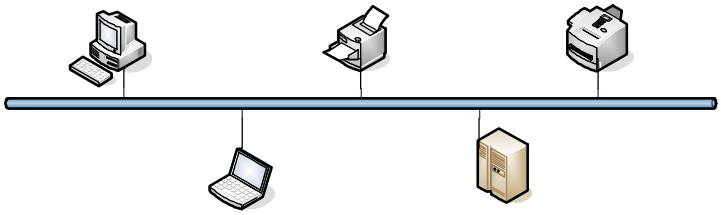


Figura 3.2 Interconexión Bus.

Desde principios de la década de 1990, la elección de configuración ha sido la topología tipo estrella, mostrada en la figura 3.3. La unidad central de la red es un hub o un conmutador (switch). Todas las conexiones en una red estrella son punto a punto implementadas con par trenzado o fibra óptica.

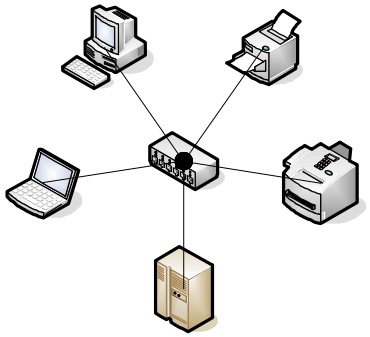


Figura 3.3 Interconexión Estrella.

3.2.3 DIFERENCIAS ENTRE ETHERNET Y IEEE 802.3.

Si bien IEEE 802.3 y Ethernet son similares, no son idénticos. Las diferencias entre ellos son lo suficientemente significantes como para hacerlos incompatibles entres sí.

Todas las versiones de Ethernet son similares en que comparten la misma arquitectura de acceso al medio múltiple con detección de errores, CSMA/CD. Sin embargo, el estándar IEEE 802.3 ha evolucionado en el tiempo de forma que ahora soporta múltiples medios en la capa física, incluyendo cable coaxial de 50 O y 75 O, cable par trenzado sin blindaje (UTP), cable par trenzado con blindaje (STP) y fibra óptica. Otras diferencias entre los dos incluyen la velocidad de transmisión, el método de señalamiento y la longitud máxima del cableado.

La diferencia más significativa entre la tecnología Ethernet original y el estándar IEEE 802.3 es la diferencia entre los formatos de sus tramas. Esta diferencia es lo suficientemente significativa como para hacer a las dos versiones incompatibles.

PREÁMBULO	DELIMITADOR DE INICIO DE TRAMA	DIRECCIÓN DE DESTINO	DIRECCIÓN DE ORIGEN	LONGITUD DE LA TRAMA	INFORMACIÓN	RELLENO (PAD)	SECUENCIA DE CHEQUEO DE TRAMA
7 BYTES	1 BYTE	2 O 6 BYTES	2 O 6 BYTES	2 BYTES	0 – 1500 BYTES	0 – n BYTES	4 BYTES

FORMATO DE LA TRAMA IEEE 802.3

PREÁMBULO	DIRECCIÓN DE DESTINO	DIRECCIÓN DE ORIGEN	TIPO DE TRAMA	INFORMACIÓN	RELLENO (PAD)	SECUENCIA DE CHEQUEO DE TRAMA
8 BYTES	6 BYTES	6 BYTES	2 BYTES	0 – 1500 BYTES	0 – n BYTES	4 BYTES

FORMATO DE LA TRAMA ETHERNET

Figura 3.4 Diferencias entre las tramas de IEEE 802.3 y Ethernet.

Una de las diferencias entre el formato de las dos tramas está en el preámbulo. El propósito del preámbulo es anunciar la trama y permitir a todos los receptores en la red sincronizarse a si mismos a la trama entrante. El preámbulo en Ethernet tiene una longitud de 8 bytes pero en IEEE 802.3 la longitud del mismo es de 7 bytes, en este último el octavo byte se convierte en el comienzo del delimitador de la trama.

La segunda diferencia entre el formato de las tramas es en el campo tipo de trama que se encuentra en la trama Ethernet. Un campo tipo es usado para especificar al protocolo que es transportado en la trama. Esto posibilita que muchos protocolos puedan ser transportados en la trama. El campo tipo fue reemplazado en el estándar IEEE 802.3 por un campo longitud de trama, el cual es utilizado para indicar el numero de bytes que se encuentran en el campo da datos.

La tercera diferencia entre los formatos de ambas tramas se encuentra en los campos de dirección, tanto de destino como de origen. Mientras que el formato de IEEE 802.3 permite el uso tanto de direcciones de 2 como de 6 bytes, el estándar Ethernet permite solo direcciones de 6 bytes.

El formato de trama que predomina actualmente en los ambientes Ethernet es el de IEEE 802.3, pero la tecnología de red continua siendo referenciada como Ethernet. [15]

3.2.4 LA TRAMA DE ETHERNET.

El estándar IEEE 802.3 define el formato de la trama de datos que es requerida por cualquier control de acceso al medio (MAC), más algunos formatos opcionales que son usados para ampliar la capacidad del protocolo. El formato de la trama de datos básica contiene siete campos:

? **Preámbulo –** Consiste de 7 bytes. Es un patrón alternado de unos y ceros que le indica al receptor que una trama esta iniciando, además de proporcionar, en la capa física receptora, una sincronía a cada una de las tramas de datos para indicar que parte de un total es dicha trama.

- **Delimitador de Inicio de Trama –** Consiste de 1 byte. Es un patrón alternado de unos y ceros, finalizando con dos unos consecutivos indicando que el siguiente bit es el bit más significativo del byte más significativo de la dirección destino.
- **Dirección Destino –** Consiste de 6 bytes. Este campo identifica cuál o cuáles receptores deberán recibir la trama. El bit mas significativo indica si la dirección es una dirección individual (indicada por un 0) o una dirección grupal (indicada por un 1). El siguiente bit indica si la dirección destino es globalmente administrada (indicada por un 0) o localmente administrada (indicada por un 1). Los restantes 46 bits son un valor asignado únicamente para identificar una estación o un grupo de estaciones en la red.
- **Dirección Origen –** consiste de 6 bytes – Este campo identifica a la estación emisora de la trama. La dirección destino es siempre una dirección individual, y su bit más significativo es siempre 0.
- **Longitud o Tipo –** Consiste de 2 bytes. Este campo puede indicar el número de bytes de datos que son contenidos en el campo de datos de la trama, o el tipo de trama si está es ensamblada usando un formato opcional. Si este campo es menor o igual a 1500, el número corresponderá al número de bytes en el campo de datos. Si el valor de este campo es mayor de 1536, la trama es un tipo de trama opcional, y el valor identificará el tipo particular de trama que será enviada o recibida.
- **Datos –** Es una secuencia de n bytes de cualquier valor, donde n es menor o igual que 1500. Si la longitud de este campo es menor que 46, se debe de extender agregando un relleno suficiente para alcanzar los 46 bytes.
- **Secuencia de Chequeo de Trama –** Consiste de 4 Bytes. Esta secuencia contiene un valor de 32 bits del chequeo de redundancia cíclica (CRC), el cual es calculado por el emisor y recalculado por el receptor para revisar tramas dañadas. Esta secuencia es generada con los campos de la Dirección Destino, Dirección Fuente, Longitud o Tipo y Datos.

3.2.5 TIPOS DE ETHERNET.

Existen una gran variedad de implementaciones de IEEE 802.3. Para distinguir entre ellas, se ha desarrollado una notación. Esta notación especifica tres características de la implementación:

- La tasa de transferencia de datos en Mb/s.
- El método de señalamiento utilizado.
- La máxima longitud del segmento de cable en cientos de metros del medio físico.

Algunos tipos de estas implementaciones de IEEE 802.3 son mostradas en la tabla 3.1:

	TIPO DE ETHERNET	**MEDIO FÍSICO**	**BANDA BASE**
ETHERNET	1BASE-5	PAR TRENZADO	1 Mb/s
	10BASE-5	COAXIAL 50 O	10 Mb/s
	10BASE-2	COAXIAL 50 O	10 Mb/s
	10BROAD-36	COAXIAL 75 O	10 Mb/s
	10BASE-T	PAR TRENZADO (UTP)	10 Mb/s
	10BASE-F	FIBRA ÓPTICA	10 Mb/s
FAST ETHERNET	100BASE-TX	PAR TRENZADO (UTP)	100 Mb/s
	100BASE-T4	PAR TRENZADO (UTP)	100 Mb/s
	100BASE-FX	FIBRA ÓPTICA	100 Mb/s
	100BASE-T2	PAR TRENZADO (UTP)	100 Mb/s
GIGABIT ETHERNET	1000BASE-SX	FIBRA ÓPTICA	1 Gb/s
	1000BASE-LX	FIBRA ÓPTICA	1 Gb/s
	1000BASE-CX	CABLE BLINDADO 150 O	1 Gb/s
	1000BASE-T	PAR TRENZADO (UTP)	1 Gb/s

Tabla 3.1 Tipos de implementaciones para IEEE 802.3.

3.3 JERARQUIA DE PROTOCOLOS.

Para reducir la complejidad de su diseño, muchas redes están organizadas como una serie de capas o niveles, cada una construida sobre la inferior. El número de capas y el nombre, el contenido y la función de cada una difieren de red a red. Sin embargo, en todas las redes el propósito de capa es ofrecer ciertos servicios a las capas superiores de modo que no tengan que ocuparse del detalle de la implementación real del servicio.

La capa n de una máquina lleva a cabo una conversación con la capa n de la otra. Las reglas y convenciones que se siguen en esta conversación se conoce colectivamente como protocolo de la capa n. Básicamente, un protocolo es un acuerdo entre las partes que se comunican sobre como va a proceder la comunicación.

En la figura 3.5 se muestra una red de cinco capas. Las entidades que comprenden las capas correspondientes en las diferentes máquinas se denominan pares. En otras palabras, son los pares los que se comunican usando el protocolo.

En realidad, los datos no se transfieren directamente de la capa n de una máquina a la capa n de otra. Más bien, cada capa pasa datos e información de control a la capa que está inmediatamente debajo de ella, hasta llegar a la capa más baja. Bajo la capa 1 está el medio físico a través del cual ocurre la comunicación real.

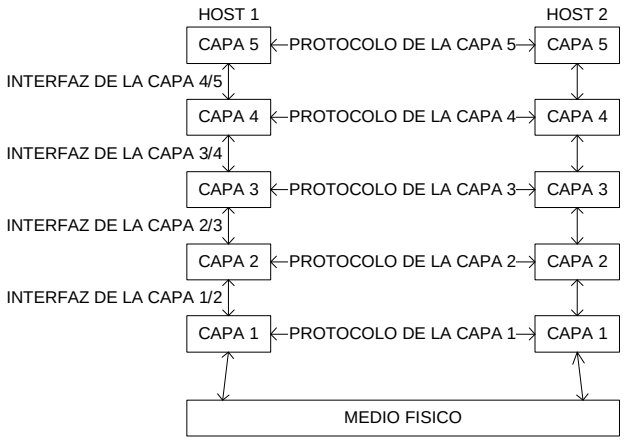


Figura 3.5 Capas, Protocolos e interfaces en una red de cinco capas.

Entre cada par de capas adyacentes hay una interfaz. La interfaz define cuáles operaciones y servicios primitivos ofrece la capa inferior a la superior.

Un conjunto de capas y protocolos recibe el nombre de arquitectura de red. La especificación de una arquitectura debe contener información suficiente para que un programador pueda escribir el programa o construir el hardware para cada capa de manera que cada una obedezca en forma correcta el protocolo apropiado. Ni los detalles de la implementación ni la especificación de las interfaces forman parte de la arquitectura porque se encuentran ocultas dentro de las máquinas y no son visibles desde fuera. Ni siquiera es necesario que las interfaces en todas las máquinas de una red sean iguales, siempre que cada máquina pueda usar correctamente todos los protocolos. La lista de protocolos empleados por cierto sistema, con un protocolo por capa, se llama pila de protocolos. [16]

3.3.1 INTERFACES Y SERVICIOS.

La función de cada capa es proporcionar servicios a la capa que está encima de ella.
Los elementos activos de cada capa generalmente se llaman entidades. Una entidad puede ser de software (como un proceso), o de hardware (como un circuito integrado de entrada/salida). Las entidades de la capa n implementan un servicio que usa la capa n+1. En este caso la capa n se llama proveedor del servicio y la capa n+1 es el usuario del servicio. La capa n puede usar los servicios de la capa n-1 con el fin de proveer su propio servicio; puede ofrecer varias clases de servicio, por ejemplo: comunicación rápida cara y comunicación lenta barata.

Los servicios están disponibles en los SAP (Service Access Points, Puntos de Acceso al Servicio). Los SAP de la capa n son los lugares en que la capa n+1 puede tener acceso a los servicios ofrecidos. Cada SAP tiene una dirección que lo identifica de manera única. Por ejemplo, los SAP del sistema telefónico son los enchufes en los que se pueden conectar los teléfonos modulares, y las direcciones de los SAP son los números telefónicos de estas tomas. Para llamar a alguien necesitamos saber la dirección de SAP de quien debe de recibir

la llamada. De manera similar, en el sistema postal las direcciones de calle y número de apartado postal. Para mandar una carta debemos conocer la dirección de SAP del destinatario.

Para que dos capas intercambien información, tiene que haber un acuerdo sobre el conjunto de reglas relativas a la interfaz. En la interfaz típica, la entidad de la capa n+1 pasa una IDU (Interface Data Unit, Unidad de Datos de la Interfaz) a la entidad de la capa n a traves del SAP como lo muestra la figura 3.6. La IDU consiste en una SDU (Service Data Unit. Unidad de Datos de Servicio) y cierta información de control. La SDU es la información que se pasa mediante la red a la entidad par y después hasta la capa n+1. La información de control es necesaria para ayudar a la capa inferior a efectuar su trabajo (por ejemplo, la cantidad de bytes en la SDU) pero no forma parte de los datos mismos.

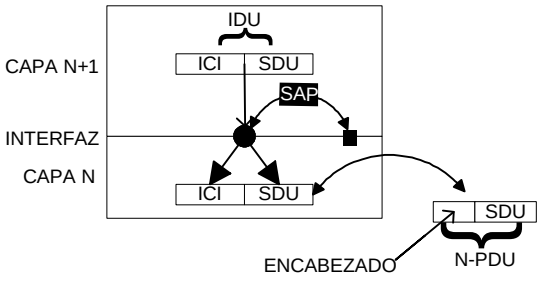


Figura 3.6 Relación entre capas en una interfaz.

Para que se transfiera la SDU, la entidad de la capa n puede tener que fragmentarla en varios pedazos, a cada uno de los cuales se les da un encabezado y se envía como una PDU (Protocol Data Unit, Unidad de Datos de Protocolo) independiente, que podría ser un paquete. Las entidades pares usan los encabezados de las PDU para acarrear su protocolo de par. Los encabezados indican cuáles PDU contienen datos y cuáles contienen información de control, proveen números de secuencias y cuentas, y otras cosas.

3.3.2 SERVICIOS ORIENTADOS A CONEXIÓN Y SIN CONEXIÓN.

Las capas pueden ofrecer dos tipos diferentes de servicio a las capas que se encuentran sobre ellas, los orientados a la conexión y los que carecen de conexión.

El servicio orientado a la conexión encuentra su modelo en el sistema telefónico. Para conversar con alguien, descolgamos el teléfono, marcamos el número, hablamos y después colgamos. De manera similar, para usar un servicio de red orientado a la conexión, el usuario del servicio establece primero una conexión, la usa y después la libera. El aspecto esencial de una conexión es que actúa como un tubo: el emisor empuja objetos (bits) por un extremo y el receptor los saca en el mismo orden por el otro extremo.

En contraste, el servicio sin conexión toma su modelo del sistema postal. Cada mensaje (carta) lleva la dirección completa de destino, y cada uno se encamina a través del sistema en forma independiente de todos los demás. Normalmente, cuando se envían dos mensajes al mismo destino, el primero que se envió será el primero en llegar. Sin embargo, es posible que el primero que se envió se retrase tanto que el segundo llegue primero. Con un servicio orientado a la conexión, esto es imposible. [17]

3.4 EL MODELO DE REFERENCIA OSI.

En sus inicios, el desarrollo de redes sucedió con desorden en muchos sentidos. A principios de la década de 1980 se produjo un enorme crecimiento en la cantidad y el tamaño de las redes. A medida que las empresas tomaron conciencia de las ventajas de usar esta tecnología de interconexión, las redes se agregaban o expandían a casi la misma velocidad a la que se introducían las nuevas tecnologías de red.

Para mediados de la década de 1980, estas empresas comenzaron a sufrir las consecuencias de la rápida expansión. De la misma forma en que las personas que no hablan un mismo idioma tienen dificultades para comunicarse, las redes que utilizaban diferentes especificaciones e implementaciones tenían dificultades para intercambiar información. El mismo problema surgía con las empresas que desarrollaban tecnologías privadas o propietarias. Esto significa que una sola empresa o un pequeño grupo de empresas controlan todo uso de la tecnología. Las tecnologías que respetaban reglas propietarias en forma estricta no podían comunicarse con tecnologías que usaban reglas propietarias diferentes. Para enfrentar el problema de incompatibilidad de redes, la Organización Internacional para la Estandarización (ISO, por sus siglas en inglés) investigó modelos de redes como la de Digital Equipment Corporation (DECnet), la Arquitectura de Sistemas de Red (SNA) y TCP/IP a fin de encontrar un conjunto de reglas aplicables de forma general a todas las redes. Con base en esta investigación, la ISO desarrolló un modelo de red que ayuda a los fabricantes a crear redes que sean compatibles con otras redes.

El modelo se llama modelo de referencia OSI (Open Systems Interconection, Interconexión de sistemas abiertos) de la ISO puesto que se ocupa de la conexión de sistemas abiertos, esto es, sistemas que están abiertos a la comunicación con otros sistemas. La figura 3.7 muestra la estructura de este modelo.

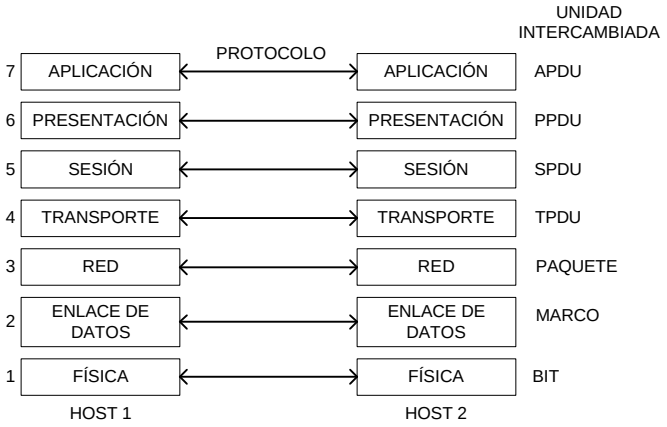


Figura 3.7 Estructura del Modelo OSI.

El modelo OSI tiene siete capas. Los principios que se aplicaron para llegar a las siete capas son los siguientes:

- Cada capa resuelve sólo una parte del problema de la comunicación con una función bien definida.
- Se debe crear una capa siempre que se necesite un nivel diferente de abstracción.
- La función de cada capa se debe elegir pensando en la definición de protocolos estandarizados internacionalmente.
- Los límites de las capas deben elegirse a modo de minimizar el flujo de información a través de interfaces.
- La cantidad de capas debe ser suficiente para no tener que agrupar funciones distintas en la misma capa y lo bastante pequeña para que la arquitectura no se vuelva inmanejable.

Note que el modelo OSI en sí no es una arquitectura de red por que no especifica los servicios y protocolos exactos que se han de usar en cada capa; sólo dice lo que debe hacer cada capa. Sin embargo, la ISO también ha elaborado estándares para todas las capas, aunque no sean parte del modelo de referencia mismo. [18]

3.4.1 LA CAPA FÍSICA.

La capa física corresponde al nivel uno del modelo OSI. Aquí las consideraciones de diseño tienen mucho que ver con las interfaces mecánicas, eléctrica y de procedimientos, y con el medio de transmisión físico que esta bajo la capa física. Es estrictamente necesaria su presencia en cualquier modelo.

La capa física tiene que ver con la transmisión de bits por un canal de comunicación. Las consideraciones de diseño tienen que ver con la acción de asegurarse de que cuando un lado envié un bit uno, se reciba en el otro lado como un bit uno, no como un bit cero. Las preguntas típicas aquí son: cuantos volts deberán usarse para representar un uno y cuantos para un cero; cuantos microsegundos dura un bit; si la transmisión se puede efectuar simultáneamente en ambas direcciones o no; cómo se establece la conexión inicial y cómo se interrumpe cuando ambos lados han terminado; y cuantas puntas tiene el conector de la red y para que sirve cada una.

3.4.2 LA CAPA DE ENLACE DE DATOS.

La tarea principal de la capa de enlace de datos es tomar un medio de transmisión en bruto y transformarlo en una línea que parezca libre de errores de transmisión no detectados a la capa de red. Esta tarea la cumple al hacer que el emisor divida los datos de entrada en marcos de datos (unos cientos o miles de bytes, normalmente), que transmita los marcos en forma secuencial y procese los marcos de acuse de recibo que devuelve el receptor. Puesto que la capa física solamente acepta y transmite una corriente de bits sin preocuparse por su significado o su estructura, corresponde a la capa de enlace de datos crear y reconocer los límites de los marcos. Esto se puede lograr añadiendo patrones especiales de bits al principio y al final del marco. Si estos patrones de bits ocurrieran en los datos por accidente, se debe tener cuidado especial para asegurar que estos patrones no se interpreten incorrectamente como delimitadores de marcos.

Una ráfaga de ruido en la línea puede destruir por completo un marco. En este caso, el software de la capa de enlace de datos de la maquina fuente puede retransmitir el marco. Sin embargo, las transmisiones repetidas del mismo marco introducen la posibilidad de duplicar marcos. Se podría enviar un marco duplicado si se perdiera el marco del acuse de

recibo que el receptor devuelve al emisor. Corresponde a esta capa resolver el problema provocado por los marcos dañados, perdidos y duplicados. La capa de enlace de datos puede ofrecer varias clases de servicio distintas a la capa de la red, cada una con diferente calidad y precio.

Otra consideración que surge en la capa de enlace de datos (y también de la mayor parte de las capas más altas) es cómo evitar que un transmisor veloz sature de datos a un receptor lento. Se debe emplear algún mecanismo de regulación de tráfico para que el transmisor sepa cuánto espacio de almacenamiento temporal (buffer) tiene el receptor en ese momento. Con frecuencia esta regulación de flujo y el manejo de errores están integrados.

Si se puede usar la línea para transmitir datos en ambas direcciones, esto introduce una nueva complicación que el software de la capa de enlace de datos debe considerar. El problema es que los marcos de acuse de recibo para el tráfico de A a B compiten por el uso de la línea con marcos de datos para el tráfico de B a A. Ya se inventó la solución inteligente, plataformas transportadoras.

Las redes de difusión tienen una consideración adicional en la capa de enlace de datos: cómo controlar el acceso al canal compartido. Una subcapa especial de la capa de enlace de datos se encarga de este problema, la subcapa de acceso al medio.

3.4.3 LA CAPA DE RED.

La capa de red se ocupa de controlar el funcionamiento de la subred. Una consideración clave de diseño es determinar cómo se encaminan los paquetes de la fuente a su destino. Las rutas se pueden basar en tablas estáticas que se alambran en la red y rara vez cambian. También se pueden determinar al inicio de cada conversación, por ejemplo en una sesión de

terminal. Por último, pueden ser altamente dinámicas, determinándose de nuevo con cada paquete para reflejar la carga actual de la red.

Si en la subred se encuentran presentes demasiados paquetes a la vez, se estorbaran mutuamente, formando cuellos de botella. El control de tal congestión pertenece también a la capa de red.

En vista de que los operadores de la subred podrían esperar remuneraciones por su labor, con frecuencia hay una función de contabilidad integrada a la capa de red. Cuando menos, el software debe contar cuántos paquetes o caracteres o bits envía cada cliente para producir información de facturación. Cuando un paquete cruza una frontera nacional, con tarifas diferentes de cada lado, la contabilidad se puede complicar.

Cuando un paquete debe viajar de una red a otra para alcanzar su destino, pueden surgir muchos problemas. El tipo de direcciones que usa la segunda red pueden ser diferentas del de la primera; puede ser que la segunda no acepte en absoluto el paquete por ser demasiado grande; los protocolos pueden diferir y otras cosas. La capa de red debe resolver todos estos problemas para lograr que se interconecten redes heterogéneas.

En las redes de difusión el problema del ruteo es simple y la capa de red con frecuencia es delgada o incluso inexistente.

3.4.4 LA CAPA DE TRANSPORTE.

La función básica de la capa de transporte es aceptar datos de la capa de sesión, dividirlos en unidades más pequeñas si es necesario, pasarlos a la capa de red y asegurar que todos los pedazos lleguen correctamente al otro extremo. Además, todo esto se debe hacer de manera eficiente en forma que aísle a las capas superiores de los cambios inevitables en la tecnología del hardware.

En condiciones normales, la capa de transporte crea una conexión de red distinta para cada conexión de transporte que requiera la capa de sesión. Sin embargo, si la conexión de transporte requiere un volumen de transmisión alto, la capa de transporte podría crear múltiples conexiones de red, dividiendo los datos entre las conexiones para aumentar el volumen. Por otro lado, si es costoso crear o mantener una conexión de red, la capa de transporte puede multiplexar varias conexiones de transporte en la misma conexión de red para reducir el costo. En todos los casos, la capa de transporte debe lograr que la multiplexión sea transparente para la capa de sesión.

La capa de transporte determina también qué tipo de servicio proporcionará a la capa de sesión y, finalmente, a los usuarios de la red. El tipo mas popular de conexión de transporte es un canal de punto a punto libre de errores que entrega mensajes o bytes en el orden en que se enviaron. Sin embargo, otras posibles clases de servicio de transporte son el transporte de mensajes aislados sin garantía respecto al orden de entrega y la difusión de mensajes a múltiples destinos. El tipo de servicio se determina al establecer la sesión.

La capa de transporte es una verdadera capa de extremo a extremo, del origen al destino. En otras palabras, un programa en la máquina fuente sostiene una conversación con un programa similar en la máquina destino, haciendo uso de los encabezados de mensaje y de los mensajes de control. En las capas bajas, los protocolos se usan entre cada maquina y sus vecinas inmediatas, y no entre las máquinas de origen y destino, que pueden estar separadas por muchos enrutadores.

La diferencia entre las capas 1 a la 3, que están encadenadas, y las capas 4 a la 7, que son de extremo a extremo. Muchos nodos están multiprogramados, lo que implica que múltiples conexiones entran y salen de cada nodo. En este caso se necesita una manera de saber cuál mensaje pertenece a cuál conexión. El encabezado de transporte, es una opción para colocar esta información.

Además de multiplexar varias corrientes de mensajes por un canal, la capa de transporte debe cuidar de establecer y liberar conexiones a través de la red. Esto requiere alguna clase de mecanismo de asignación de nombres, de modo que un proceso en una máquina pueda describir con quién quiere conversar. También debe haber un mecanismo para regular el flujo de información, a fin de que un nodo rápido no pueda saturar a uno lento. Tal mecanismo se llama control de flujo y desempeña un papel clave en la capa de transporte (también en otras capas). El control de flujo entre nodos es distinto del control de flujo entre enrutadores.

3.4.5 LA CAPA DE SESIÓN.

La capa de sesión permite a los usuarios de máquinas diferentes establecer sesiones entre ellos. Una sesión permite el transporte ordinario de datos, como lo hace la capa de transporte, pero también proporciona servicios mejorados que son útiles en algunas aplicaciones. Se podría usar una sesión para que el usuario se conecte a un sistema remoto de tiempo compartido o para transferir un archivo entre dos máquinas.

Uno de los servicios de la capa de sesión es manejar el control del dialogo. Las sesiones pueden permitir que el tráfico vaya en ambas direcciones al mismo tiempo, o sólo en una dirección a la vez. Si el tráfico puede ir únicamente en un sentido a la vez (en analogía con una sola vía de ferrocarril), la capa de sesión puede ayudar a llevar el control de los turnos.

Un servicio de sesión relacionado es el manejo de fichas. Para algunos protocolos es esencial que ambos lados no intenten la misma operación al mismo tiempo. A fin de controlar estas actividades, la capa de sesión proporciona fichas que se pueden intercambiar. Solamente el lado que posea la ficha podrá efectuar la operación crítica.

Otro servicio de sesión es la sincronización. Considere los problemas que pueden ocurrir cuando se trata de efectuar una transferencia de archivos de 2 horas de duración entre dos máquinas que tienen un tiempo medio entre rupturas de 1 hora. Cada transferencia, después de abortar, tendría que empezar de nuevo desde el principio y probablemente fallaría también la siguiente vez. Para eliminar este problema, la capa de sesión ofrece una forma de insertar puntos de verificación en la corriente de datos, de modo que después de cada interrupción sólo se deban repetir los datos que se transfirieron después del último punto de verificación.

3.4.6 LA CAPA DE PRESENTACIÓN.

La capa de presentación realiza ciertas funciones que se piden con suficiente frecuencia para justificar la búsqueda de una solución general, en lugar de dejar que cada usuario resuelva los problemas. En particular, y a diferencia de todas las capas inferiores que se interesan sólo en mover bits de manera confiable de acá para allá, la capa de presentación se ocupa de la sintaxis y la semántica de la información que se transmite.

Un ejemplo típico de servicio de presentación es la codificación de datos en una forma estándar acordada. La mayor parte de los programas de usuario no intercambian cadenas de bits al azar; intercambian cosas como nombres de personas, fechas, cantidades de dinero y cuentas. Estos elementos se representan como cadenas de caracteres, enteros, cantidades de punto flotante y estructura de datos compuesta de varios elementos más simples. Las diferentes computadoras tienen códigos diferentes para representar cadenas de caracteres (por ejemplo, ASCII y Unicote), enteros (por ejemplo, en complemento a uno y en complemento a dos), y demás. Con el fin de hacer posible la comunicación entre computadoras con representaciones diferentes, las estructuras de datos por intercambiar se pueden definir en forma abstracta, junto con un código estándar que se use en el cable. La

capa de presentación maneja estas estructuras de datos abstractas y las convierte de la representación que se usa dentro de la computadora a la representación estándar en la red y viceversa.

3.4.7 LA CAPA DE APLICACIÓN.

La capa de aplicación contiene varios protocolos que se necesitan con frecuencia. Por ejemplo, existen cientos de tipos de terminales incompatibles con el mundo. Considere la situación de un editor de pantalla completa que debe de trabajar en una red con muchos tipos diferentes de terminal, cada uno con formatos diferentes de pantalla, secuencias de escape para insertar y eliminar texto, mover cursor, etcétera.

Una forma de resolver este problema es definir una terminal virtual de red abstracta que los editores y otros programas pueden manejar. Para cada tipo de terminal, se debe escribir un programa para establecer la correspondencia entre las funciones de la terminal virtual de red y las de la terminal real. Por ejemplo, cuando el editor mueva el cursor de la terminal virtual a la esquina superior izquierda de la pantalla, este software debe emitir la secuencia apropiada de órdenes a la terminal real para poner su cursor en ese lugar. Todo el software de terminal virtual está en la capa de aplicación.

Otra función de la capa de aplicación es la transferencia de archivos. Los diferentes sistemas de archivos tienen convenciones diferentes para nombrar los archivos, formas diferentes de representar líneas de texto, etc. La transferencia de un archivo entre dos sistemas diferentes requiere la resolución de estás y otras incompatibilidades. Este trabajo también pertenece a la capa de aplicación, lo mismo que el correo electrónico, la carga remota de trabajos, la búsqueda en directorios y otros recursos de uso general y especial.

3.4.8 TRANSMISIÓN DE DATOS EN EL MODELO OSI.

La figura 3.8 muestra un ejemplo de cómo se pueden transmitir datos empleando el modelo OSI. El proceso remitente tiene algunos datos que quiere enviar al proceso receptor, así que entrega los datos a la capa de aplicación, la cual añade entonces al principio el encabezado de aplicación AH (que puede ser nulo) y entrega el elemento resultante a la capa de presentación.

La capa de presentación puede transformar este elemento de diferentes maneras y posiblemente añadir al principio un encabezado, entregando el resultado a la capa de sesión. Es importante darse cuenta que la capa de presentación no sabe cuál porción de los datos es entregados a ella por la capa de aplicación es la AH, si existe, y cuáles son en verdad los datos del usuario.

Este proceso se repite hasta que los datos alcanzan la capa física, donde son transmitidos realmente a la máquina receptora. En esa máquina se retiran los distintos encabezados, uno por uno, conforme el mensaje se propaga hacia arriba por las capas hasta que por fin llega al proceso receptor.

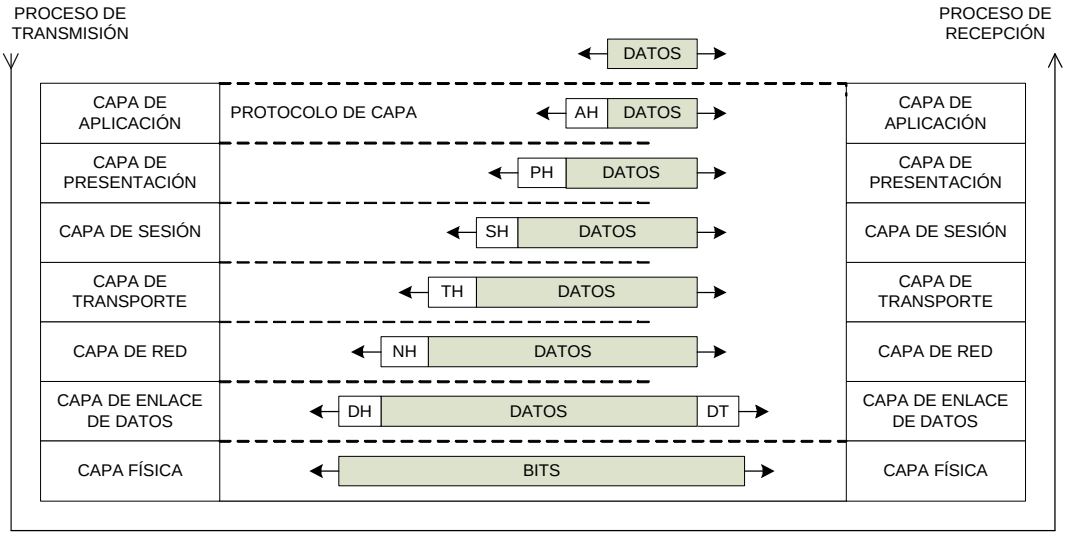


Figura 3.8 Ejemplo de transmisión de datos en el modelo OSI.

La clave de todo este proceso es que aunque la transmisión real de los datos es vertical, cada capa se programa como si fuera horizontal. Por ejemplo, cuando la capa de transporte emisora recibe un mensaje de la capa de sesión, le añade un encabezado de transporte y lo envía a la capa de transporte receptora. Desde su punto de vista, el hecho de que en realidad debe dirigir el mensaje a la capa de red de su propia máquina es un tecnicismo sin importancia. A manera de analogía, cuando un diplomático que habla español se dirige a las Naciones Unidas, piensa que se dirige a los demás diplomáticos de la asamblea. El hecho de que en realidad sólo hable con su traductor se ve como un detalle técnico. [19]

3.4.9 LA RELACIÓN LÓGICA DE IEEE 802.3 Y EL MODELO DE REFERENCIA OSI.

La figura 3.9 muestra las capas lógicas del modelo IEEE 802.3 y su relación al modelo de referencia OSI. Como con todos los protocolos IEEE 802, la capa de enlace de datos del modelo de referencia OSI es dividida dentro de dos subcapas IEEE 802, la subcapa Control de Acceso al Medio (MAC, Media Access Control) y la subcapa MAC – client. La capa física de IEEE 802.3 corresponde a la capa física ISO.

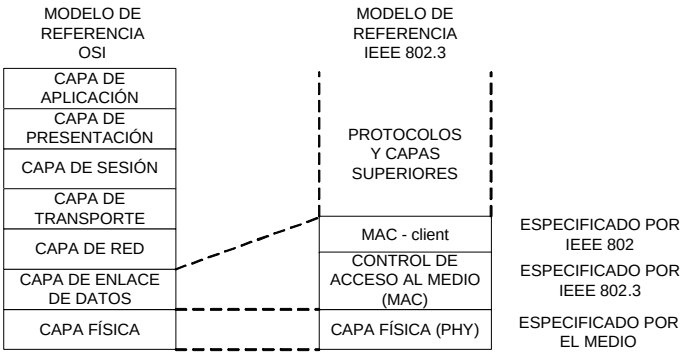


Figura 3.9 Relación lógica de Ethernet con el modelo de referencia OSI.

La subcapa MAC – client puede corresponder a uno de los siguientes casos:

- **Control Lógico de Enlace** (LLC, Logical Link Control), si la unidad es un DTE. Esta subcapa proporciona la interfase entre la MAC Ethernet y las capas superiores en la pila de protocolos. Esta subcapa esta definida por IEEE 802.2 estándar.

- **Entidad Puente**, si la unidad es una DCE. Las entidades puentes proporcionan interfaces LAN – LAN entre redes LAN que usan el mismo protocolo (por ejemplo, Ethernet a Ethernet) y también entre diferentes protocolos (por ejemplo, Ethernet a Token Ring). Las entidades puentes están definidas por IEEE 802.1 estándar.

Debido a que las especificaciones para Control Lógico de Enlace y entidades puente son comunes para todos los protocolos LAN IEEE 802, la compatibilidad de redes se convierte en la responsabilidad primaria para el protocolo de la red en particular. La figura 3.10 muestra diferentes requerimientos de compatibilidad impuestos por la subcapa MAC y la capa física para la comunicación básica de datos sobre un enlace Ethernet.

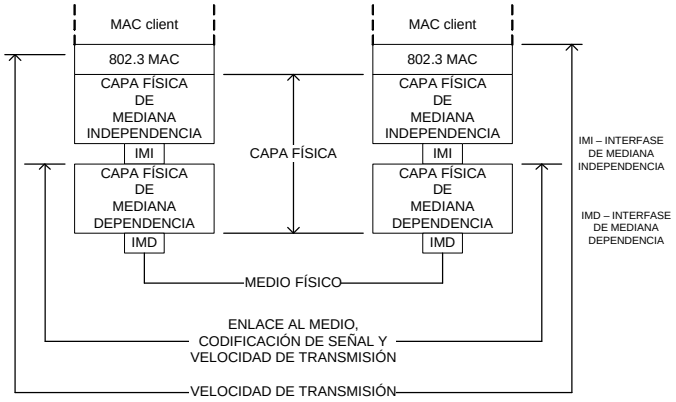


Figura 3.10 Requerimientos de compatibilidad en capas MAC y física para comunicación de datos básica.

La capa MAC controla el acceso del nodo a la red y es específico al protocolo individual. Todas las MAC IEEE 802.3 deben conocer el mismo conjunto básico de requerimientos lógicos, sin tener en cuenta si estos incluyen uno o más de definidas extensiones opcionales del protocolo. El único requerimiento para la comunicación básica (comunicación que no requiere extensiones opcionales del protocolo) entre dos nodos de red es que ambas MAC puedan soportar la misma velocidad de transmisión.

La capa física IEEE 802.3 especifica la velocidad de transmisión de datos, la señal codificada y el tipo de interconexión al medio de los dos nodos. Ethernet Gigabit, por ejemplo, esta definido para operar sobre par trenzado o fibra óptica, pero cada tipo de cable o procedimiento de codificación de señal requiere una diferente implementación de capa física.

3.5 CONCLUSIONES CAPITULARES.

Es sin duda los años de investigación en el Ethernet y la implementación de los distintos protocolos lo que han permitido la gran evolución en las redes de cómputo. Características como la velocidad de transmisión, el ancho de banda y el medio de transmisión físico es el reflejo de esta revolución, la cual parece no terminar debido al constante nacimiento de nuevas aplicaciones y necesidades.

Estos cambios son también gracias a organismos como ISO, los cuales se encargaron de estandarizar protocolos y tecnologías con la finalidad de facilitar la interconexión de computadoras o sistemas computarizados.

Ethernet es una de las tecnologías que más se ha hecho presente en las distintas áreas del quehacer humano, desde una oficina hasta un hospital desde una universidad hasta el hogar.

4. EL PROTOCOLO TCP/IP.

OBJETIVO: Fundamentar la teoría que permita comprender la forma como uno de los protocolos más importantes para realizar la interconexión de sistemas de cómputo a nivel Internet, es también aplicado para interconectar otros tipos de sistemas, entre ellos los embebidos.

4.1 EL MODELO DE REFERENCIA TCP/IP.

Sobre la década de los 60, la DARPA (Defense Advanced Research Projects Agency), perteneciente al Departamento de Defensa de los Estados Unidos (DoD), inició un programa de investigación de tecnologías de comunicación entre redes de diferentes características, pero no fue hasta la década de los 70 cuando se comenzaron las primeras aplicaciones. El proyecto se basaba en la transmisión de paquetes de información, y tenia por objetivo la interconexión de redes. De este proyecto surgieron dos redes: Una de investigación, ARPANET, y una de uso exclusivamente militar, MILNET.

A finales de 1969 cuatro hosts fueron conectados a ARPANET, la cual fue creciendo rápidamente durante los años siguientes, pero fue a partir de 1972 cuando se comenzó a investigar la forma de que los paquetes de información puedan moverse a través de varias redes de diferentes tipos y no necesariamente compatibles. De esta manera se consiguen enlazar redes independientes consiguiendo que puedan comunicarse de forma transparente los ordenadores de todas ellas. Este proyecto recibió el nombre de "Internetting", y para referirse al sistema de redes funcionando conjuntamente y formando una red mayor se utilizó el nombre de Internet. La red continuó extendiéndose por todo el país con gran rapidez, conectando a universidades e instituciones de investigación y educación, organizaciones gubernamentales o no gubernamentales, y redes privadas y comerciales. De esta manera continuó su desarrollo durante los años 80 extendiéndose internacionalmente, pero ha sido en los 90 cuando Internet se ha convertido en un nuevo y revolucionario medio de comunicación a escala mundial. Los nuevos medios desarrollados para hacer el acceso a Internet mucho más sencillo y agradable para cualquier usuario han influido notablemente en esta expansión, convirtiendo a Internet en la gran red mundial.

Para comunicar las redes, se desarrollaron varios protocolos: El protocolo de Internet y el protocolo de control de transmisión. Posteriormente estos protocolos se englobaron en el conjunto de protocolos TCP/IP o modelo de referencia TCP/IP: Protocolo de Control de Transmisión (TCP) y Protocolo de Internet (IP), que fueron los dos primeros en definirse, y que son los más utilizados de la familia.

Existen tantos protocolos en este conjunto que llegan a ser más de 100 diferentes, entre ellos se encuentra el popular HTTP (HyperText Transfer Protocol, Protocolo de Transferencia de Hipertexto), que es el que se utiliza para acceder a las páginas Web, además de otros como el ARP (Address Resolution Protocol, Protocolo de Resolución de Direcciones), el FTP (File Transfer Protocol, Protocolo de Transferencia de Archivos) y el SMTP (Simple Mail Transfer Protocol, Protocolo Sencillo de Transferencia de Correo) y el POP (Post Office Protocol, Protocolo de Oficina Postal) para correo electrónico, TELNET para acceder a equipos remotos, entre otros.

Algunos de los motivos de su popularidad son:

- Independencia del fabricante.
- Soporta múltiples tecnologías (Ethernet, Token Ring, X.25...).
- Independencia del sistema operativo (Unix, Linux, Windows)
- Proporcionan un esquema común de direccionamiento que permite a un dispositivo con TCP/IP localizar a cualquier otro en cualquier punto de la red.
- Son estándares de protocolos abiertos y gratuitos.
- Puede funcionar en máquinas de cualquier tamaño y tipo (Macintosh o Intel).
- Estándar de EEUU desde 1983.

La arquitectura de un sistema en TCP/IP tiene una serie de metas:

- La independencia de la tecnología usada en la conexión a bajo nivel y la arquitectura del ordenador.
- Conectividad universal a través de la red.
- Reconocimientos de extremo a extremo.
- Protocolos estandarizados.

El protocolo TCP/IP tiene que estar a un nivel superior del tipo de red empleado y funcionar de forma transparente en cualquier tipo de red. Y a un nivel inferior de los programas de aplicación (páginas WEB, correo electrónico…) particulares de cada sistema operativo. Todo esto nos sugiere el siguiente modelo de referencia, figura 4.1, también basado en capas. [20]

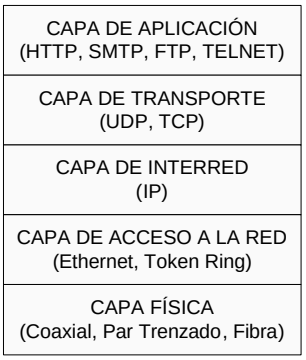


Figura 4.1. Estructura del Modelo de Referencia TCP/IP.

Cabe aclarar que muchos autores estructuran el modelo de referencia TCP/IP, en la cual la capa física esta incluida en la capa de acceso a la red, como se muestra en la figura 4.2.

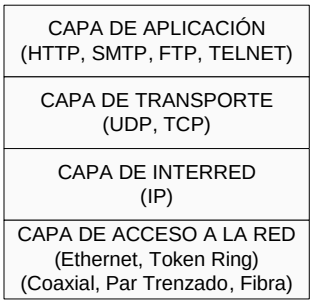


Figura 4.2 Variante de la Estructura del Modelo de Referencia TCP/IP.

4.1.1 LA CAPA DE ACCESO A LA RED.

El modelo de referencia TCP/IP realmente no dice mucho de lo que aquí sucede, fuera de indicar que el nodo se ha de conectar a la red haciendo uso de algún protocolo de modo que pueda enviar por ella paquetes de IP.

Como el propósito de esta capa es transportar una cadena de bits en bruto de una máquina a otra, se pueden usar varios medios físicos para la transmisión real; cada uno con su propio nicho en términos de ancho de banda, retardo, costo y facilidad de instalación y mantenimiento. A grandes rasgos, los medios se agrupan en medios guiados, como el cable de cobre y la fibra óptica, y medios no guiados, como la radio y los láseres a través del aire. Internet es un conjunto de tecnologías que permiten interconectar a redes muy distintas entre sí, por ejemplo, Ethernet con Token Ring.

Cada PC tiene una dirección física, llamado MAC (Media Access Control), la cual está determinada por la tarjeta de red. Cada tarjeta de red tiene un número único de 48 bits que la identifica de todas las demás, los tres primeros bytes identifican a cada fabricante. Las direcciones y los números para cada fabricante son administrados por la el IEEE, lo que permite una duplicación de dirección.

En realidad esta dirección física se encuentra en el circuito NIC de la tarjeta de red, el cual es un circuito con memoria ROM que permite sólo una única escritura, evitando así que se pueda cambiar la dirección.

4.1.2 LA CAPA DE INTERRED.

El Departamento de Defensa quería que las conexiones permanecieran intactas mientras las máquinas de origen y destino estuvieran funcionando, aún si alguna de las máquinas o de las líneas de transmisión en el trayecto dejara de funcionar en forma repentina. Es más, se necesitaba una arquitectura flexible, pues se tenía la visión de aplicaciones futuras con requerimientos divergentes, abarcando desde la transferencia de archivos hasta la transmisión de eventos en tiempo real.

Todos estos requerimientos condujeron a la elección de una red de conmutación de paquetes basadas en una capa de interred carente de conexiones. Esta capa, llamada capa de interred, es el eje que mantiene unida toda la arquitectura. La misión de esta capa es permitir que los nodos inyecten paquetes en cualquier red y los hagan viajar de forma independiente a su destino (que podría estar en una red diferente).

Los paquetes pueden llegar incluso en un orden diferente a aquel en que se enviaron, en cuyo caso corresponde a las capas superiores reacomodarlos, si se desea la entrega ordenada. Nótese que aquí se usa interred en un sentido genérico, aunque esta capa esté presente en la Internet.

La capa de interred define un formato de paquete y protocolo oficial llamado IP (Internet Protocol, Protocolo de Interred). El trabajo de la capa de interred es entregar paquetes IP a donde se supone que deben ir. Aquí la consideración más importante es claramente el ruteo de los paquetes, y también evitar la congestión. Por lo anterior es razonable decir que la capa de interred TCP/IP es muy parecida en funcionalidad a la capa de red OSI.

4.1.2.1 DIRECCIONES IP.

Una dirección física se utiliza para interconectar a las computadoras que pertenecen a la misma red. Para identificar globalmente a una computadora dentro de un conjunto de redes TCP/IP se utilizan las direcciones IP. Observando una dirección IP se sabe si pertenece a nuestra red o a otra distinta ya que todas las direcciones IP de la misma red comienzan con los mismos números. La tabla 4.1 muestra un ejemplo de la relación entre las direcciones IP y la red a la que pertenecen.

DIRECCIÓN FÍSICA	DIRECCIÓN IP	RED
00-60-52-0B-B7-7D	192.168.0.10	RED 1
00-E0-4C-AB-9A-FF	192.168.0.1	
A3-BB-05-17-29-D0	10.10.0.1	RED 2
00-E0-4C-33-79-AF	10.10.0.7	
B2-42-52-12-37-BE	10.10.0.2	
00-E0-89-AB-12-92	200.3.107.1	RED 3
A3-BB-08-10-DA-DB	200.3.107.73	
B2-AB-31-07-12-93	200.3.107.200	

Tabla 4.1 Relación entre las direcciones físicas e IP con su respectiva red.

La dirección IP es el identificador de cada computadora, la cual debe ser distinta a todas las demás direcciones que estén vigentes en ese momento en el conjunto de redes visibles por la propia computadora. En el caso de Internet, no puede haber dos computadoras con 2 direcciones IP (públicas) iguales. Pero sí podríamos tener dos computadoras con la misma dirección IP siempre y cuando pertenezcan a redes independientes entre sí (sin ningún camino posible que las comunique).

Las direcciones IP se clasifican en:

- **Direcciones IP públicas.** Son visibles en todo Internet. Una computadora con una IP pública es accesible (visible) desde cualquier otra computadora conectado a Internet. Para conectarse a Internet es necesario tener una dirección IP pública.

- **Direcciones IP privadas (reservadas).** Son visibles únicamente por otros hosts de su propia red o de otras redes privadas interconectadas por routers. Se utilizan en las empresas para los puestos de trabajo. Las computadoras con direcciones IP privadas pueden salir a Internet por medio de un router (o *proxy*) que tenga una IP pública. Sin embargo, desde Internet no se puede acceder a computadoras con direcciones IP privadas.

A su vez, las direcciones IP pueden ser:

- **Direcciones IP estáticas (fijas).** Una computadora que se conecte a la red con dirección IP estática siempre lo hará con una misma IP. Las direcciones IP públicas estáticas son las que utilizan los servidores de Internet con objeto de que estén siempre localizables por los usuarios de Internet.

- **Direcciones IP dinámicas.** Una computadora que se conecte a la red mediante dirección IP dinámica, cada vez lo hará con una dirección IP distinta. Las direcciones IP públicas dinámicas son las que se utilizan en las conexiones a Internet mediante un módem. Los proveedores de Internet utilizan direcciones IP dinámicas debido a que tienen más clientes que direcciones IP (es muy improbable que todos se conecten a la vez).

Las direcciones IP están formadas por 4 bytes (32 bits). Se suelen representar de la forma *a.b.c.d* donde cada una de estas letras es un número comprendido entre el 0 y el 255. Por ejemplo la dirección 129.42.18.99.

Dependiendo del número de computadoras que se necesiten para cada red, las direcciones de Internet se han dividido en las clases primarias A, B y C. La clase D está formada por direcciones que identifican no a una computadora, sino a un grupo de ellos. Las direcciones de clase E no se pueden utilizar (están reservadas). La tabla 4.2 nos muestra las diferentes clases que existen de las direcciones IP.

CLASE	0	1	2	3	4	8	16	24	31
A	0	RED				COMPUTADORA			
B	1	0	RED				COMPUTADORA		
C	1	1	0	RED				COMPUTADORA	
D	1	1	1	0	MULTIDIFUSIÓN				
E	1	1	1	1	RESERVADO				

Tabla 4.2 Clases de las direcciones IP.

De acuerdo a la clase de la dirección IP, es posible saber el número de redes y computadoras, rango de direcciones y la máscara de subred para cada tipo de clase. La tabla 4.3 nos indica esta relación.

CLASE	NÚMERO DE REDES	NÚMERO DE COMPUTADORAS POR RED	RANGO DE DIRECCIONES	MÁSCARA DE SUBRED
A	128	16777214	0.0.0.0 – 127.0.0.0	255.0.0.0
B	16384	65534	128.0.0.0 – 191.255.0.0	255.255.0.0
C	2097152	254	192.0.0.0 – 223.255.255.0	255.255.255.0
D	-	-	224.0.0.0 – 239.255.255.255	-
E	-	-	240.0.0.0 – 55.255.255.255	-

Tabla 4.3 Número de redes, computadoras direcciones y máscaras de acuerdo a la clase de dirección IP.

No todas las direcciones comprendidas entre la 0.0.0.0 y la 223.255.255.255 son válidas para una computadora, algunas de ellas tienen significados especiales. Las principales direcciones especiales se resumen en la tabla 4.4. Su interpretación depende de la computadora desde la cual se utiliza.

BITS DE RED	BITS DE COMPUTADORA	SIGNIFICADO	EJEMPLO
TODOS 0		PROPIA COMPUTADORA	0.0.0.0
TODOS 0	COMPUTADORA	COMPUTADORA INDICADA DENTRO DE LA PROPIA RED	0.0.0.10
RED	TODOS 0	RED INDICADA	192.168.1.0
TODOS 1		DIFUSION A LA PROPIA RED	255.255.255.255
RED	TODOS 1	DIFUSION A LA RED INDICADA	192.168.1.255
127	CUALQUIER VALOR DE COMPUTADORA	LOOKBACK (PROPIA COMPUTADORA)	127.0.0.1

Tabla 4.4 Direcciones IP especiales.

Intranet es una red privada que utiliza los protocolos TCP/IP. Puede tener salida a Internet o no. En el caso de tener salida a Internet, el direccionamiento IP permite que las computadoras con direcciones IP privadas puedan salir a Internet pero impide el acceso a las computadoras internas desde Internet. Dentro de una intranet se pueden configurar todos los servicios típicos de Internet (Web, correo, mensajería instantánea, etc.) mediante la

instalación de los correspondientes servidores. La idea es que las intranets sean como Internet en miniatura o lo que es lo mismo, Internet es una intranet pública gigantesca.

La tabla 4.5 muestra las direcciones de redes se encuentran reservadas para su uso en redes privadas. Una dirección IP que pertenezca a una de estas redes se dice que es una dirección IP privada. [21]

CLASE	RANGO DE DIRECCIONES
A	10.0.0.0
B	172.16.0.0 – 172.31.0.0
C	192.168.0.0 – 192.168.255.0

Tabla 4.5 Direcciones reservadas para redes privadas.

4.1.2.2 MÁSCARA DE SUBRED.

Una máscara de subred es aquella dirección que enmascarando nuestra dirección IP, nos indica si otra dirección IP pertenece a nuestra subred o no. La tabla 3.5 muestra las máscaras de subred correspondientes a cada clase.

Si expresamos la máscara de subred de clase A en notación binaria, tenemos:

11111111.00000000.00000000.00000000

Los unos indican los bits de la dirección correspondientes a la red y los ceros, los correspondientes a la computadora. Según la máscara anterior, el primer byte es la red y los tres siguientes bytes, la computadora. Por ejemplo, la dirección de clase A 35.120.73.5 pertenece a la red 35.0.0.0.

Supongamos una subred con máscara 255.255.0.0, en la que tenemos un ordenador con dirección 148.120.33.110. Si expresamos esta dirección y la de la máscara de subred en binario, tenemos:

148.120.33.110 10010100.01111000.00100001.01101110 (dirección de una máquina)
255.255.0.0 11111111.11111111.00000000.00000000 (dirección de su máscara de red)
148.120.0.0 10010100.01111000.00000000.00000000 (dirección de su subred)

Al hacer el producto binario (AND lógico) de las dos primeras direcciones obtenemos la tercera. Si hacemos lo mismo con otro ordenador, por ejemplo el 148.120.33.89, obtenemos la misma dirección de subred. Esto significa que ambas máquinas se encuentran en la misma subred.

148.120.33.89 10010100.01111000.00100001.01011001 (dirección de una máquina)
255.255.0.0 11111111.11111111.00000000.00000000 (dirección de su máscara de red)
148.120.0.0 10010100.01111000.00000000.00000000 (dirección de su subred)

En cambio, si tomamos la 148.115.89.3, observamos que no pertenece a la misma subred que las anteriores.

148.115.89.3 10010100.01110011.01011001.00000011 (dirección de una máquina)
255.255.0.0 11111111.11111111.00000000.00000000 (dirección de su máscara de red)
148.115.0.0 10010100.01110011.00000000.00000000 (dirección de su subred)

En una red de redes TCP/IP no puede haber computadoras aisladas todas pertenecen a alguna red, todas tienen una dirección IP y una máscara de subred (si no se especifica se toma la máscara que corresponda a su clase). Mediante esta máscara una computadora sabe si otra computadora se encuentra en su misma subred o en otra distinta. Si pertenece a su misma subred, el mensaje se entregará directamente. En cambio, si las computadoras están configuradas en redes distintas, el mensaje se enviará a la puerta de salida o router de la red de la computadora origen. Este router pasará el mensaje al siguiente de la cadena y así sucesivamente hasta que se alcance la red de la computadora destino y se complete la entrega del mensaje. [22]

4.1.2.3 EL PROTOCOLO IP.

En la capa de red, la Internet puede verse como un conjunto de subredes, o sistemas autónomos interconectados. No hay una estructura concreta, pero existen varios backbone principales. Éstos se construyen a partir de líneas de muy alto ancho de banda y enrutadores rápidos. Conectados a los backbone hay redes regionales (de nivel medio), y conectadas a estas redes regionales están las LAN de muchas universidades, compañías y proveedores de servicio de Internet.

La parte que mantiene unida a Internet es el protocolo de capa de red, IP (Internet Protocol, Protocolo Internet). A diferencia de la mayoría de los protocolos de capa de red, éste se diseñó desde el principio con la interconexión de redes en mente. Su trabajo es proporcionar un medio de mejor esfuerzo para el transporte de datagramas del origen al destino, sin importar si estas computadoras están en la misma red, o si hay otras redes entre ellas.

IP es el principal protocolo de la capa de red. Este protocolo define la unidad básica de transferencia de datos entre el origen y el destino, atravesando toda la red de redes. Además, el software IP es el encargado de elegir la ruta más adecuada por la que los datos serán enviados. Se trata de un sistema de entrega de paquetes (llamados datagramas IP) que tiene las siguientes características:

Es no orientado a conexión debido a que cada uno de los paquetes puede seguir rutas distintas entre el origen y el destino. Entonces pueden llegar duplicados o desordenados.

Es no fiable porque los paquetes pueden perderse, dañarse o llegar retrasados.

En la comunicación en Internet, la capa de transporte toma corriente de datos y las divide en datagramas. En teoría, los datagramas pueden ser de hasta 64 Kbytes cada uno, pero en la práctica por lo general son de unos 1500 bytes. Cada datagrama se transmite a través de Internet, posiblemente fragmentándose en unidades más pequeñas en el camino. Cuando todas las piezas llegan finalmente a la computadora destino, son reensambladas por la capa de red, dejando el datagrama original. Este datagrama entonces es entregado a la capa de transporte.

El datagrama IP es la unidad básica de transferencia de datos entre el origen y el destino. Viaja en el campo de datos de las tramas físicas de las distintas redes que va atravesando. Cada vez que un datagrama tiene que atravesar un router, el datagrama saldrá de la trama física de la red que abandona y se acomodará en el campo de datos de una trama física de la siguiente red. Este mecanismo permite que un mismo datagrama IP pueda atravesar redes distintas: enlaces punto a punto, redes ATM, redes Ethernet, redes Token Ring, etc. El propio datagrama IP tiene también un campo de datos, será aquí donde viajen los paquetes de las capas superiores. [23]

Un datagrama IP consiste en una parte de cabecera y una parte de texto. La cabecera tiene una parte fija de de 20 bytes y una parte opcional de longitud variable. El formato de la cabecera se muestra en la figura 4.3. Se transmite en orden de izquierda a derecha, comenzando por el bit de orden mayor del campo de versión.

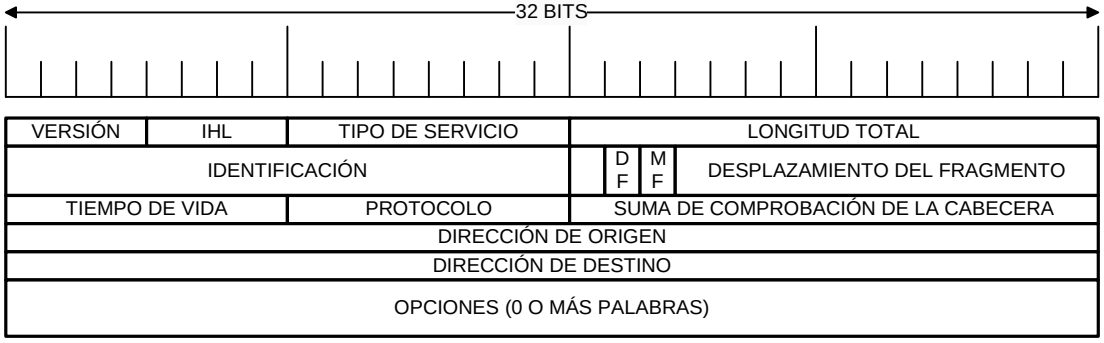


Figura 4.3 Formato de la cabecera de IP.

- **Versión** (4 bits). Indica la versión del protocolo IP. La Internet Enginnering Task Force (IETF) creó el proyecto IPng (Internet Protocol the Next Generation), también llamado IPv6. Esta nueva versión del Protocolo de Internet (IP) sustituirá progresivamente a la actual IPv4, ya que brinda mejores características entre las que destacan: espacio de direcciones prácticamente infinito; posibilidad de autoconfiguración de computadoras y ruteadores; soporte para seguridad, computación móvil, calidad de servicio; un mejor diseño para el transporte de tráfico multimedia en tiempo real, aplicaciones anycast y multicast; así como la posibilidad de transición gradual de IPv4 a IPv6.

  IPv4 soporta 4.294.967.296 (2^{32}) direcciones de red diferentes, un número inadecuado para dar una dirección a cada persona del planeta, y mucho menos para cada coche, teléfono, o PDA; mientras que IPv6 soporta 340.282.366.920.938.463.463.374.607.431.768.211.456 (2^{128} o 340 septillones) direcciones. Se espera que IPv4 se siga soportando hasta por lo menos el 2025, dado que hay muchos dispositivos heredados que no se migrarán a IPv6 nunca y que seguirán siendo utilizados por mucho tiempo.
- **IHL** (4 bits). Longitud de la cabecera expresada en múltiplos de 32 bits. Dado que la longitud de la cabecera no es constante, se incluye este campo. El valor mínimo es de 5, correspondiente a 160 bits = 20 bytes, cifra que aplica cuando no hay opciones. El valor máximo de este campo es de 15, lo que limita la cabecera a 60 bytes y, por tanto, el campo de opciones a 40 bytes.
- **Tipo de servicio** (8 bits) permite a la computadora indicar a la subred el tipo de servicio que quiere. Son posibles varias combinaciones de confiabilidad y velocidad. Para voz digitalizada, la entrega rápida le gana a la entrega precisa. Para la transferencia de archivos, es más importante la transmisión libre de errores que la transmisión rápida. En la práctica, los enrutadores actuales ignoran por completo a este campo. Este campo esta dividido en:
  - **Prioridad** (3 bits). Un valor de 0 indica baja prioridad y un valor de 7, prioridad máxima.

  Los siguientes tres bits indican cómo se prefiere que se transmita el mensaje, es decir, son sugerencias a los enrutadores que se encuentren a su paso los cuales pueden tomarlas en cuenta o no.
  - **Bit D** (Delay, retardo). Solicita retardos cortos (enviar rápido).
  - **Bit T** (Throughput, rendimiento). Solicita un alto rendimiento (enviar mucho en el menor tiempo posible).

- **Bit R** (Reliability, confiabilidad). Solicita que se minimice la probabilidad de que el datagrama se pierda o resulte dañado (enviar bien).

Los siguientes dos bits no tienen uso.

- **Longitud total** (16 bits). Indica la longitud total del datagrama expresada en bytes. Como el campo tiene 16 bits, la máxima longitud posible de un datagrama será de 65535 bytes. Actualmente este límite es tolerable, pero con las redes futuras de gigabits se requerirán datagramas más grandes.
- **Identificación** (16 bits). Número de secuencia que junto a la dirección origen, dirección destino y el protocolo utilizado identifica de manera única un datagrama en toda la red. Si se trata de un datagrama fragmentado, llevará la misma identificación que el resto de fragmentos.
- **Banderas** (3 bits) o indicadores. Sólo 2 bits de los 3 bits disponibles están actualmente utilizados. El bit de Más fragmentos (**MF**) indica que no es el último datagrama, todos los fragmentos excepto el último tienen establecido este bit. Y el bit de No fragmentar **(NF)** prohíbe la fragmentación del datagrama. Si este bit está activado y en una determinada red se requiere fragmentar el datagrama, éste no se podrá transmitir y se descartará.
- **Desplazamiento de fragmentación** (13 bits). Indica el lugar en el cual se insertará el fragmento actual dentro del datagrama completo, medido en unidades de 64 bits. Por esta razón los campos de datos de todos los fragmentos menos el último tienen una longitud múltiplo de 64 bits. Si el paquete no está fragmentado, este campo tiene el valor de cero. Dado que se proporcionan 13 bits, puede haber un máximo de 8192 fragmentos por datagrama, dando una longitud máxima de datagrama de 65536 bytes, uno más que el campo de longitud total.
- **Tiempo de vida** (8 bits) o TTL. Es un contador que sirve para limitar la vida de un paquete. Se supone que este contador cuenta el tiempo en segundos, permitiendo una vida máxima de 255 segundos; debe disminuir en cada salto y se supone que disminuye muchas veces al encolarse durante un tiempo grande en un enrutador. En la práctica, simplemente cuenta los saltos. Cuando el contador llegue a cero, el datagrama se descarta y se devuelve a la computadora origen un mensaje ICMP de tipo "tiempo excedido" para informar de la incidencia.
- **Protocolo** (8 bits). Una vez que la capa de red ha ensamblado un datagrama completo, necesita saber que hacer con él. Este indica el protocolo utilizado en el campo de datos: 1 para ICMP, 2 para IGMP, 6 para TCP y 17 para UDP.

- **CRC cabecera** (16 bits). Contiene la suma de comprobación de errores sólo para la cabecera del datagrama. La verificación de errores de los datos corresponde a las capas superiores.
- **Dirección origen** (32 bits). Contiene la dirección IP del origen.
- **Dirección destino** (32 bits). Contiene la dirección IP del destino.
- **Opciones IP**. Este campo no es obligatorio y especifica las distintas opciones solicitadas por el usuario que envía los datos (generalmente para pruebas de red y depuración).
- **Relleno.** Si las opciones IP (en caso de existir) no ocupan un múltiplo de 32 bits, se completa con bits adicionales hasta alcanzar el siguiente múltiplo de 32 bits (recuérdese que la longitud de la cabecera tiene que ser múltiplo de 32 bits).

4.1.2.3 FRAGMENTACIÓN.

Las tramas físicas tienen un campo de datos y que es aquí donde se transportan los datagramas IP. Sin embargo, este campo de datos no puede tener una longitud indefinida debido a que está limitado por el diseño de la red. El MTU de una red es la mayor cantidad de datos que puede transportar su trama física. El MTU de las redes Ethernet es 1500 bytes y el de las redes Token Ring, 8192 bytes. Esto significa que una red Ethernet nunca podrá transportar un datagrama de más de 1500 bytes sin fragmentarlo.

Un enrutador (router) fragmenta un datagrama en varios si el siguiente tramo de la red por el que tiene que viajar el datagrama tiene un MTU inferior a la longitud del datagrama. El siguiente ejemplo, con la figura 4.4, muestra cómo se produce la fragmentación de un datagrama.

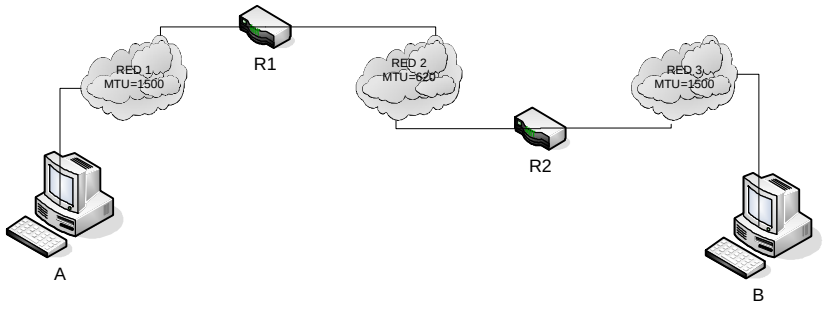


Figura 4.4 Envío de datagrama de A a B usando fragmentación.

Supongamos que la computadora A envía un datagrama de 1400 bytes de datos (1420 bytes en total) a la computadora B. El datagrama no tiene ningún problema en atravesar la red 1 ya que 1420 < 1500. Sin embargo, no es capaz de atravesar la red 2 (1420 = 620). El enrutador R1 fragmenta el datagrama en el menor número de fragmentos posibles que sean capaces de atravesar la red 2. Cada uno de estos fragmentos es un nuevo datagrama con la misma Identificación pero distinta información en el campo de Desplazamiento de fragmentación y el bit de Más fragmentos (MF). Veamos el resultado de la fragmentación:

Fragmento 1: Longitud total = 620 bytes; Desplazamiento = 0; MF=1 (contiene los primeros 600 bytes de los datos del datagrama original).
Fragmento 2: Longitud total = 620 bytes; Desplazamiento = 600; MF=1 (contiene los siguientes 600 bytes de los datos del datagrama original).
Fragmento 3: Longitud total = 220 bytes; Desplazamiento = 1200; MF=0 (contiene los últimos 200 bytes de los datos del datagrama original).

El enrutador R2 recibirá los 3 datagramas IP (fragmentos) y los enviará a la red 3 sin reensamblarlos. Cuando la computadora B reciba los fragmentos, recompondrá el datagrama original. Los enrutadores intermedios no reensamblan los fragmentos debido a que esto supondría una carga de trabajo adicional, a parte de memorias temporales. Nótese que el ordenador destino puede recibir los fragmentos cambiados de orden pero esto no supondrá ningún problema para el reensamblado del datagrama original puesto que cada fragmento guarda suficiente información.

Si el datagrama del ejemplo hubiera tenido su bit No fragmentar (NF) a 1, no hubiera conseguido atravesar el enrutador R1 y, por tanto, no tendría forma de llegar hasta el host B. El enrutador R1 descartaría el datagrama. [24]

4.1.2.5 EL PROTOCOLO ARP.

Dentro de una misma red, las máquinas se comunican enviándose tramas físicas. Las tramas Ethernet contienen campos para las direcciones físicas de origen y destino (6 bytes cada una). El problema que se nos plantea es cómo podemos conocer la dirección física de la máquina destino. El único dato que se indica en los datagramas es la dirección IP de destino. ¿Cómo se pueden entregar entonces estos datagramas? Necesitamos obtener la dirección física de la computadora a partir de su dirección IP. Esta es justamente la misión del protocolo ARP (Address Resolution Protocol, Protocolo de Resolución de Direcciones).

Por ejemplo si en la tabla 4.6 la computadora A envía un datagrama con origen 192.168.0.10 y destino 10.10.0.7 (B). Como la computadora B se encuentra en una red distinta la computadora A, el datagrama tiene que atravesar el router 192.168.0.1 (R1). Se necesita conocer la dirección física de R1. Es entonces cuando entra en funcionamiento el protocolo ARP: A envía un mensaje ARP a todas las máquinas de su red preguntando "¿Cuál es la dirección física de la máquina con dirección IP 192.168.0.1?". La máquina con dirección 192.168.0.1 (R1) advierte que la pregunta está dirigida a ella y responde a A con su dirección física (00-E0-4C-AB-9A-FF). Entonces A envía una trama física con origen 00-60-52-0B-B7-7D y destino 00-E0-4C-AB-9A-FF conteniendo el datagrama (origen 192.168.0.10 y destino 10.10.0.7). Al otro lado del enrutador R2 se repite de nuevo el proceso para conocer la dirección física de B y entregar finalmente el datagrama a B. El mismo datagrama ha viajado en dos tramas físicas distintas, una para la red 1 y otra para la red 2.

COMPUTADORA/ENRUTADOR	DIRECCIÓN FÍSICA	DIRECCIÓN IP	RED
A	00-60-52-0B-B7-7D	192.168.0.10	RED 1
RI	00-E0-4C-AB-9A-FF	192.168.0.1	
	A3-BB-05-17-29-D0	10.10.0.1	RED 2
B	00-E0-4C-33-79-AF	10.10.0.7	
R2	B2-42-52-12-37-BE	10.10.0.2	
	00-E0-89-AB-12-92	200.3.107.1	RED 3
C	A3-BB-08-10-DA-DB	200.3.107.73	
D	B2-AB-31-07-12-93	200.3.107.200	

Tabla 4.6 Relación de direcciones físicas, direcciones IP, enrutador y la red que corresponde.

Observemos que las preguntas ARP son de difusión (se envían a todas las máquinas). Estas preguntas llevan además la dirección IP y dirección física de la máquina que pregunta. La respuesta se envía directamente a la máquina que formuló la pregunta.

4.1.2.6 EL PROTOCOLO ICMP.

Debido a que el protocolo IP no es fiable, los datagramas pueden perderse o llegar defectuosos a su destino. El protocolo ICMP (Internet Control Message Protocol, protocolo de mensajes de control y error) se encarga de informar al origen si se ha producido algún error durante la entrega de su mensaje. Pero no sólo se encarga de notificar los errores, sino que también transporta distintos mensajes de control.

El protocolo ICMP únicamente informa de incidencias en la red pero no toma ninguna decisión. Esto será responsabilidad de las capas superiores. Los mensajes ICMP viajan en el campo de datos de un datagrama IP, como se puede apreciar en la figura 4.5.

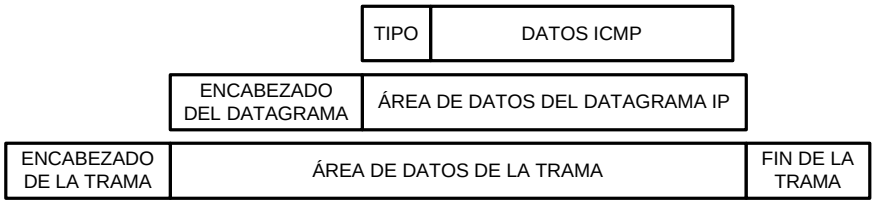


Figura 4.5 Mensajes ICMP dentro del área del datagrama IP.

Puede darse el caso de que un mensaje ICMP se pierda o se dañe. Si esto llega a ocurrir no se creará un nuevo mensaje ICMP sino que el primero se descartará sin más. Los mensajes ICMP comienzan con un campo de 8 bits que contiene el tipo de mensaje, según se muestra en la tabla 4.7. El resto de campos son distintos para cada tipo de mensaje ICMP.

TIPO	MENSAJE ICMP
0	Respuesta de eco (*Echo Reply*)
3	Destino inaccesible (*Destination Unreachable*)
4	Disminución del tráfico desde el origen (*Source Quench*)
5	Redireccionar (cambio de ruta) (*Redirect*)
8	Solicitud de eco (*Echo*)
11	Tiempo excedido para un datagrama (*Time Exceeded*)
12	Problema de Parámetros (*Parameter Problem*)
13	Solicitud de marca de tiempo (*Timestamp*)
14	Respuesta de marca de tiempo (*Timestamp Reply*)
15	Solicitud de información (obsoleto) (*Information Request*)
16	Respuesta de información (obsoleto) (*Information Reply*)
17	Solicitud de máscara (*Addressmask*)
18	Respuesta de máscara (*Addressmask Reply*)

Tabla 4.7 Mensajes ICMP.

4.1.2.6.1 PING.

Se trata de una utilidad, definido dentro del protocolo ICMP, que comprueba el estado de la conexión con uno o varios equipos remotos por medio de los paquetes de solicitud de eco y de respuesta de eco para determinar si un sistema IP específico es accesible en una red. Es útil para diagnosticar los errores en redes o enrutadores IP.

Muchas veces se utiliza para medir la latencia o tiempo que tardan en comunicarse dos puntos remotos, y por ello, se utiliza entre los aficionados a los juegos en red el término PING para referirse al lag o latencia de su conexión.

Existe otro tipo: Ping ATM. Este tipo de ping se utiliza en las redes ATM (como puede ser una simple ADSL instalada en casa) y, en este caso, las tramas se transmiten son ATM (nivel 2 del modelo OSI). Este tipo de paquetes se envían para probar si los enlaces ATM están correctamente definidos.

El comando PING se utiliza con la línea de comandos: ping + IP de la otra PC. Por ejemplo:

```
C:\>ping 192.168.0.1

Haciendo ping a 192.168.0.1 con 32 bytes de datos:

Respuesta desde 192.168.0.1: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.0.1: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.0.1: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.0.1: bytes=32 tiempo<1m TTL=128

Estadísticas de ping para 192.168.0.1:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),

Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 0ms, Máximo = 0ms, Media = 0ms
```

4.1.2.7 ENCAMINAMIENTO.

Una *red de redes* está formada por redes interconectadas mediante routers o encaminadores. Cuando enviamos un datagrama desde una computadora hasta otra, éste tiene que ser capaz de encontrar la ruta más adecuada para llegar a su destino. Esto es lo que se conoce como encaminamiento.

Los routers (enrutadores o encaminadores) son los encargados de elegir las mejores rutas. Estas máquinas pueden ser computadoras con varias direcciones IP o bien, aparatos específicos. Los routers deben conocer, al menos parcialmente, la estructura de la red que les permita encaminar de forma correcta cada mensaje hacia su destino. Esta información se almacena en las llamadas tablas de encaminamiento. [25]

Observemos que debido al sistema de direccionamiento IP esta misión no es tan complicada. Lo único que necesitamos almacenar en las tablas son los prefijos de las direcciones (que nos indican la red). Por ejemplo, si el destino es la máquina 149.33.19.4 con máscara 255.255.0.0, nos basta con conocer el encaminamiento de la red 149.33.0.0 ya que todas las que empiecen por 149.33 se enviarán hacia el mismo sitio.

4.1.3 CAPA DE TRANSPORTE.

La capa de transporte está sobre la capa de interred en el modelo TCP/IP. Esta capa se diseñó para permitir que las entidades pares en los nodos de origen y destino lleven a cabo una conversión, lo mismo que en la capa de transporte OSI.

La capa de red transfiere datagramas entre dos computadoras a través de la red utilizando como identificadores las direcciones IP. La capa de transporte añade la noción de puerto para distinguir entre los muchos destinos dentro de una misma computadora. No es suficiente con indicar la dirección IP del destino, además hay que especificar la aplicación que recogerá el mensaje. Cada aplicación que esté esperando un mensaje utiliza un número de puerto distinto; más concretamente, la aplicación está a la espera de un mensaje en un puerto determinado (escuchando un puerto).

Pero no sólo se utilizan los puertos para la recepción de mensajes, también para el envío: todos los mensajes que envíe una computadora debe hacerlo a través de uno de sus puertos. La figura 4.6 representa una transmisión entre la computadora 194.35.133.5 y la 135.22.8.165. El primero utiliza su puerto 1256 y el segundo, el 80.

Figura 4.6. Transmisión entre dos computadoras tomando en cuenta dirección IP y puertos.

La capa de transporte transmite mensajes entre las aplicaciones de dos ordenadores. Por ejemplo, entre nuestro navegador de páginas Web y un servidor de páginas Web, o entre nuestro programa de correo electrónico y un servidor de correo.

4.1.3.1 PUERTOS.

Una computadora puede estar conectada con distintos servidores a la vez; por ejemplo, con un servidor de noticias y un servidor de correo. Para distinguir las distintas conexiones dentro de una misma computadora se utilizan los puertos.

Un puerto es un número de 16 bits, por lo que existen 65536 puertos en cada computadora. Las aplicaciones utilizan estos puertos para recibir y transmitir mensajes. Los números de puerto de las aplicaciones cliente son asignados dinámicamente y generalmente son superiores al 1024. Cuando una aplicación cliente quiere comunicarse con un servidor, busca un número de puerto libre y lo utiliza. En cambio, las aplicaciones servidoras utilizan unos números de puerto prefijados: son los llamados puertos well-known (bien conocidos). La tabla 4.8 resume los puertos más importantes.

NOMBRE	PUERTO/PROTOCOLO	DESCRIPCIÓN
FTP	21/TCP	FILE TRANSFER [CONTROL]
TELNET	23/TCP	TELNET
SMTP	25/TCP	SIMPLE MAIL TRANSFER
DOMAIN	53/TCP/UDP	DOMAIN NAME SERVER
WWW-HTTP	80/TCP	WORLD WIDE WEB HTTP
POP3	110/TCP	POST OFFICE PROTOCOL – VERSION 3
NNTP	119/TCP	NETWORK NEWS TRANSFER PROTOCOL
NETBIOS-SSN	139/TCP/UDP	NETBIOS SESSION SERVICE

Tabla 4.8 Puertos más importantes.

Dos pares dirección IP-Puerto es conocido como conexión. No puede haber dos conexiones iguales en un mismo instante en toda la red. Aunque bien es posible que una misma computadora tenga dos conexiones distintas y simultáneas utilizando un mismo puerto. El protocolo TCP utiliza el concepto de conexión para identificar las transmisiones. En la figura 4.7 se han creado tres conexiones. Las dos primeras son al mismo servidor Web (puerto 80) y la tercera a un servidor de FTP (puerto 21).

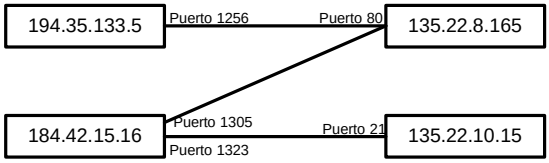


Figura 4.7 Conexiones IP-Puertos.

Para que se pueda crear una conexión, el extremo del servidor debe hacer una *apertura pasiva* del puerto (escuchar su puerto y quedar a la espera de conexiones) y el cliente, una *apertura activa* en el puerto del servidor (conectarse con el puerto de un determinado servidor).

4.1.3.2 PROTOCOLO UDP.

El UDP (User Datagram Protocol, Protocolo de Datagrama de Usuario), al igual que IP, es un protocolo no orientado a conexión y no confiable; para aplicaciones que no necesitan la asignación de secuencia ni el control de flujo del TCP y que desean utilizar los suyos propios, es decir, proporciona una comunicación muy sencilla entre las aplicaciones de dos computadoras. Este protocolo también se usa ampliamente para consultas de petición y respuestas de una sola ocasión, del tipo cliente-servidor, y en aplicaciones en las que la entrega pronta es más importante que la entrega precisa, como las transmisiones de voz o video.

UDP utiliza el protocolo IP para transportar sus mensajes, como lo muestra la figura 4.8, no añade ninguna mejora en la calidad de la transferencia; aunque sí incorpora los puertos origen y destino en su formato de mensaje. Las aplicaciones (y no el protocolo UDP) deberán programarse teniendo en cuenta que la información puede no llegar de forma correcta. [26]

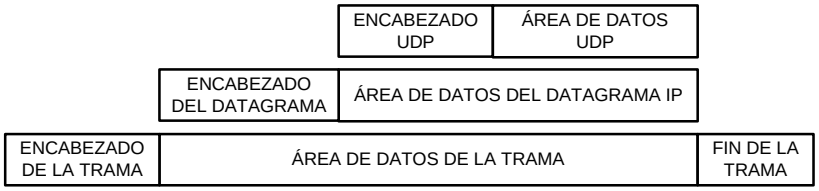


Figura 4.8 UDP dentro del área del datagrama IP.

Un segmento UDP cosiste en una cabecera de 8 bytes seguida de los datos. Como lo muestra la figura 4.9.

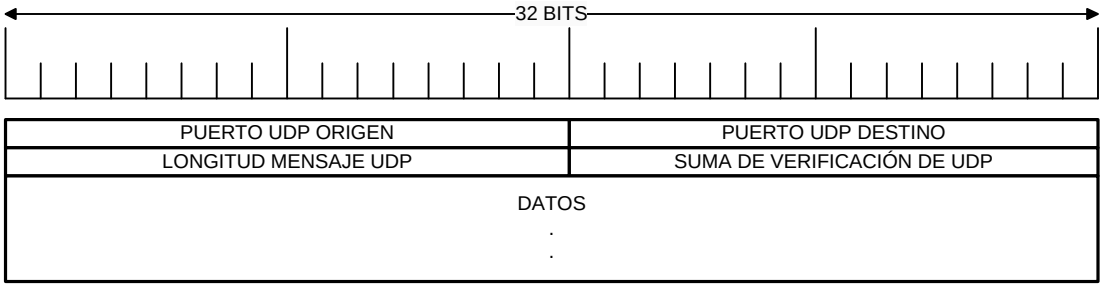


Figura 4.9 Formato del segmento UDP.

- **Puerto UDP origen** (16 bits, opcional). Número de puerto de la máquina origen.
- **Puerto UDP destino** (16 bits). Número de puerto de la máquina destino.
- **Longitud del mensaje UDP** (16 bits). Especifica la longitud medida en bytes del mensaje UDP incluyendo la cabecera. La longitud mínima es de 8 bytes.
- **Suma de verificación UDP** (16 bits, opcional). Suma de comprobación de errores del mensaje. Para su cálculo se utiliza una pseudo-cabecera que también incluye las direcciones IP origen y destino. Para conocer estos datos, el protocolo UDP debe interactuar con el protocolo IP.
- **Datos**. Aquí viajan los datos que se envían las aplicaciones. Los mismos datos que envía la aplicación origen son recibidos por la aplicación destino después de atravesar toda la red de redes.

4.1.3.3 PROTOCOLO TCP.

El TCP (Transmisión Control Protocol, Protocolo de Control de Transmisión) esta basado en el protocolo IP que no es fiable y no orientado a conexión, y sin embargo es un protocolo confiable orientado a conexión que permite que una corriente de bytes originada en una máquina se entregue sin errores en cualquier otra máquina de la Internet. De esta forma, las aplicaciones que lo utilicen no tienen que preocuparse de la integridad de la información: dan por hecho que todo lo que reciben es correcto.

La relación y ubicación dentro del modelo de referencia entre IP, TCP y UDP se muestran en la figura 4.10.

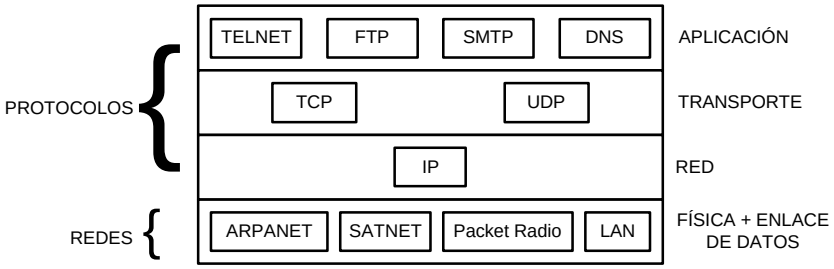


Figura 4.10 Protocolos y redes en el modelo TCP/IP.

Este protocolo fragmenta la corriente de bytes en mensajes discretos y pasa cada uno de ellos a la capa de Internet. En el destino el proceso TCP receptor reensambla los mensajes recibidos para formar la corriente de salida. El TCP también se encarga del control de flujo para asegurar que un emisor rápido no pueda abrumar a un receptor lento con más mensajes de los que pueda manejar.

El flujo de datos entre una aplicación y otra viajan por un circuito virtual. Sabemos que los datagramas IP pueden seguir rutas distintas, dependiendo del estado de los encaminadores intermedios, para llegar a un mismo sitio. Esto significa que los datagramas IP que

transportan los mensajes siguen rutas diferentes aunque el protocolo TCP logré la ilusión de que existe un único circuito por el que viajan todos los bytes uno detrás de otro (algo así como una tubería entre el origen y el destino).

Para que esta comunicación pueda ser posible es necesario abrir previamente una conexión. Esta conexión garantiza que los todos los datos lleguen correctamente de forma ordenada y sin duplicados. La unidad de datos del protocolo es el byte, de tal forma que la aplicación origen envía bytes y la aplicación destino recibe estos bytes.

Sin embargo, cada byte no se envía inmediatamente después de ser generado por la aplicación, sino que se espera a que haya una cierta cantidad de bytes, se agrupan en un segmento y se envía el segmento completo. Para ello son necesarias unas memorias intermedias o buffers. Cada uno de estos segmentos viaja en el campo de datos de un datagrama IP. Si el segmento es muy grande será necesario fragmentar el datagrama, con la consiguiente pérdida de rendimiento; y si es muy pequeño, se estarán enviando más cabeceras que datos. Por consiguiente, es importante elegir el mayor tamaño de segmento posible que no provoque fragmentación.

El protocolo TCP envía un flujo de información no estructurado. Esto significa que los datos no tienen ningún formato, son únicamente los bytes que una aplicación envía a otra. Ambas aplicaciones deberán ponerse de acuerdo para comprender la información que se están enviando.

Cada vez que se abre una conexión, se crea un canal de comunicación bidireccional en el que ambas aplicaciones pueden enviar y recibir información, es decir, una conexión es full-dúplex.

4.1.3.3.1 FORMATO DEL SEGMENTO TCP.

Como ya se ha mencionado, el flujo de bytes que produce una determinada aplicación se divide en uno o más segmentos TCP para su transmisión. Cada uno de estos segmentos viaja en el campo de datos de un datagrama IP. Para facilitar el control de flujo de la información los bytes de la aplicación se numeran. En la figura 4.11 se muestra la distribución de un segmento TCP.

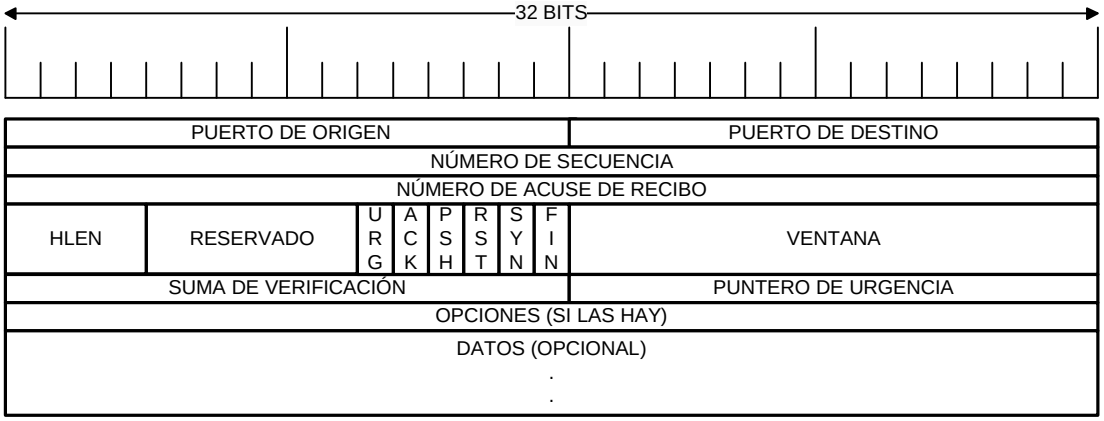


Figura 4.11 Cabecera TCP.

Cada segmento comienza con una cabecera de formato fijo de 20 bytes. La cabecera fija puede ir seguida de opciones de cabecera. Tras las opciones, si las hay, puede continuar hasta 65535 bytes totales – 20 bytes de cabecera – 20 bytes de opciones = 65515 bytes de datos, donde los primeros 20 se refieren a la cabecera IP y los segundos a la cabecera TCP. Los segmentos sin datos son legales y se usan por lo común para acuses de recibido y mensajes de control.

- **Puerto de Origen** (16 bits). Puerto de la máquina origen. Al igual que el puerto destino es necesario para identificar la conexión actual.
- **Puerto de Destino** (16 bits). Puerto de la máquina destino.
- **Número de Secuencia** (32 bits). Indica el número de secuencia del primer byte que trasporta el segmento.
- **Número de Acuse de Recibo** (32 bits). Indica el número de secuencia del siguiente byte que se espera recibir. Con este campo se indica al otro extremo de la conexión que los bytes anteriores se han recibido correctamente.

- **HLEN** (4 bits). Longitud de la cabecera medida en múltiplos de 32 bits (4 bytes). El valor mínimo de este campo es 5, que corresponde a un segmento sin datos (20 bytes).
- **Reservado** (6 bits). Bits reservados para un posible uso futuro.
- **Bits de Código** o indicadores (6 bits). Los bits de código determinan el propósito y contenido del segmento.
- **URG.** El campo Puntero de urgencia contiene información válida.
- **ACK.** El campo Número de acuse de recibo contiene información válida, es decir, el segmento actual lleva un ACK. Observemos que un mismo segmento puede transportar los datos de un sentido y las confirmaciones del otro sentido de la comunicación.
- **PSH.** La aplicación ha solicitado una operación push (enviar los datos existentes en la memoria temporal sin esperar a completar el segmento).
- **RST.** Interrupción de la conexión actual.
- **SYN.** Sincronización de los números de secuencia. Se utiliza al crear una conexión para indicar al otro extremo cual va a ser el primer número de secuencia con el que va a comenzar a transmitir (veremos que no tiene porqué ser el cero).
- **FIN.** Indica al otro extremo que la aplicación ya no tiene más datos para enviar. Se utiliza para solicitar el cierre de la conexión actual.
- **Ventana** (16 bits). Número de bytes que el emisor del segmento está dispuesto a aceptar por parte del destino.
- **Suma de verificación** (24 bits). Suma de comprobación de errores del segmento actual. Para su cálculo se utiliza una pseudo-cabecera que también incluye las direcciones IP origen y destino.
- **Puntero de urgencia** (8 bits). Se utiliza cuando se están enviando datos urgentes que tienen preferencia sobre todos los demás e indica el siguiente byte del campo Datos que sigue a los datos urgentes. Esto le permite al destino identificar donde terminan los datos urgentes.

  Nótese que un mismo segmento puede contener tanto datos urgentes (al principio) como normales (después de los urgentes).
- **Opciones** (variable). Si está presente únicamente se define una opción: el tamaño máximo de segmento que será aceptado.
- **Datos.** Información que envía la aplicación.

4.1.3.3.2 ESTABLECIMIENTO DE CONEXIÓN.

Antes de transmitir cualquier información utilizando el protocolo TCP es necesario abrir una conexión, como lo muestra la figura 4.12. Un extremo hace una apertura pasiva y el otro, una apertura activa. El mecanismo utilizado para establecer una conexión consta de tres vías.

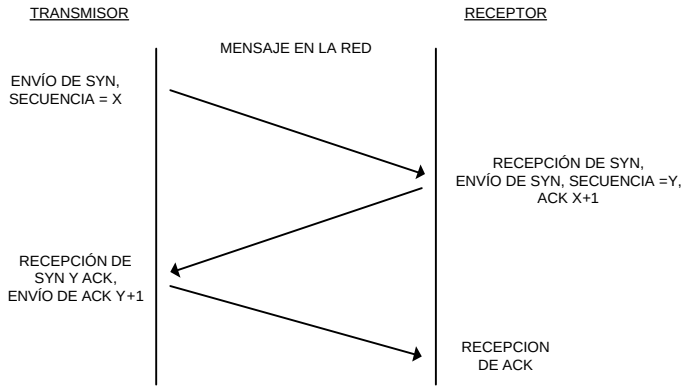


Figura 4.12. Establecimiento de una conexión con TCP.

La máquina que quiere iniciar la conexión hace una apertura activa enviando al otro extremo un mensaje que tenga el bit SYN activado. Le informa además del primer número de secuencia que utilizará para enviar sus mensajes.

La máquina receptora (un servidor generalmente) recibe el segmento con el bit SYN activado y devuelve la correspondiente confirmación. Si desea abrir la conexión, activa el bit SYN del segmento e informa de su primer número de secuencia. Deja abierta la conexión por su extremo.

La primera máquina recibe el segmento y envía su confirmación. A partir de este momento puede enviar datos al otro extremo. Abre la conexión por su extremo.

La máquina receptora recibe la confirmación y entiende que el otro extremo ha abierto ya su conexión. A partir de este momento puede enviar ella también datos. La conexión ha quedado abierta en los dos sentidos.

Observamos que son necesarios 3 segmentos para que ambas máquinas abran sus conexiones y sepan que la otra también está preparada.

4.1.3.3.3 ESTABLECIMIENTO DE COMUNICACIÓN.

La manera como es posible enviar información fiable basándose en un protocolo no fiable, es decir, que los datagramas que transportan los segmentos de TCP no se pierdan y puedan llegar de forma correcta al destino es que cada vez que llega un mensaje se devuelva una confirmación (ACK, acknowledgement) para que el emisor sepa que ha llegado correctamente. Si no le llega esta confirmación pasado un cierto tiempo, el emisor reenvía el mensaje. Esta es la manera más sencilla (aunque ineficiente) de proporcionar una comunicación fiable. El emisor envía un dato, arranca su temporizador y espera su confirmación (ACK). Si recibe su ACK antes de agotar el temporizador, envía el siguiente dato. Si se agota el temporizador antes de recibir el ACK, reenvía el mensaje. La figura 4.13 muestra este comportamiento. [27]

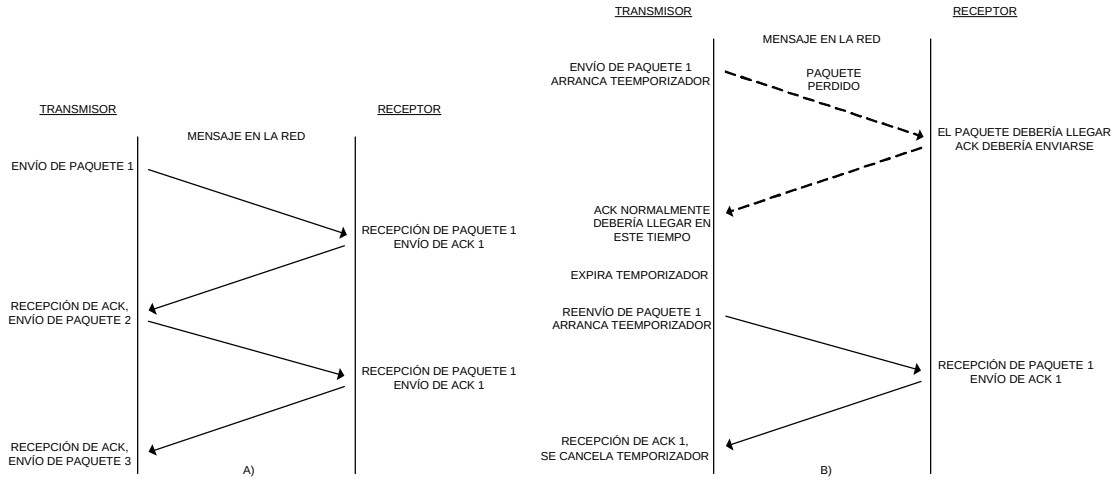


Figura 4.13 Envío de paquetes en TCP. A) En funcionamiento normal. B) Cuando se pierde algún paquete.

Este método es perfectamente válido aunque muy ineficiente debido a que sólo se utiliza un sentido de la comunicación a la vez y el canal está desaprovechado la mayor parte del tiempo. Para solucionar este problema se utiliza un protocolo de ventana deslizante, que se resume en la figura 4.14. Los mensajes y las confirmaciones van numerados y el emisor puede enviar más de un mensaje antes de haber recibido todas las confirmaciones anteriores.

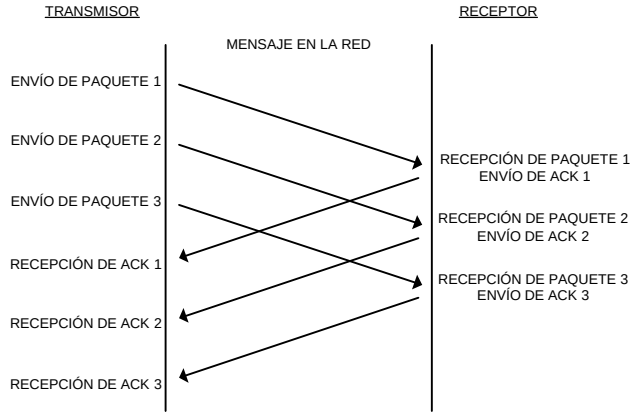


Figura 4.14 Protocolo de Ventana Deslizante.

4.1.3.3.4 CIERRE DE CONEXIÓN.

Cuando una aplicación ya no tiene más datos que transferir, el procedimiento normal es cerrar la conexión utilizando una variación del mecanismo de tres vías explicado anteriormente, como lo muestra la figura 4.15.

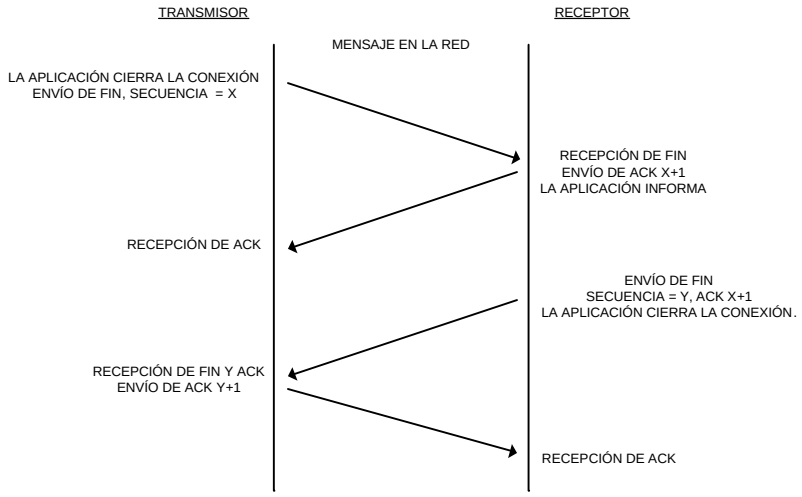


Figura 4.15 Cierre de una conexión con TCP.

El mecanismo de cierre es algo más complicado que el de establecimiento de conexión debido a que las conexiones son full-duplex y es necesario cerrar cada uno de los dos sentidos de forma independiente.

La máquina que ya no tiene más datos que transferir, envía un segmento con el bit FIN activado y cierra el sentido de envío. Sin embargo, el sentido de recepción de la conexión sigue todavía abierto.

La máquina receptora recibe el segmento con el bit FIN activado y devuelve la correspondiente confirmación. Pero no cierra inmediatamente el otro sentido de la conexión sino que informa a la aplicación de la petición de cierre. Aquí se produce un lapso de tiempo hasta que la aplicación decide cerrar el otro sentido de la conexión.
La primera máquina recibe el segmento ACK.

Cuando la máquina receptora toma la decisión de cerrar el otro sentido de la comunicación, envía un segmento con el bit FIN activado y cierra la conexión.

La primera máquina recibe el segmento FIN y envía el correspondiente ACK. Observemos que aunque haya cerrado su sentido de la conexión sigue devolviendo las confirmaciones.
La máquina receptora recibe el segmento ACK.

4.1.4 CAPA DE APLICACIÓN.

El modelo TCP/IP no tiene capas de sesión ni de presentación. No se pensó que fueran necesarias, así que no se incluyeron. La experiencia con el modelo OSI ha comprobado que esta visión fue correcta ya que se utiliza muy poco en la mayoría de aplicaciones.

La capa de 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 servidor de ficheros (FTP). Hay tantos protocolos como aplicaciones distintas y puesto que continuamente se desarrollan nuevas aplicaciones el número de protocolos crece sin parar.

Cabe aclarar que el usuario normalmente no interactúa directamente con el nivel de aplicación. Suele interactuar con programas que a su vez interactúan con el nivel de aplicación pero ocultando la complejidad subyacente. Así por ejemplo un usuario no manda una petición "HTTP/1.0 GET index.html" para conseguir una página en html, ni lee directamente el código html/xml. [28]

4.1.4.1 HTTP.

El Protocolo de Transferencia de Hipertexto (HTTP, HyperText Transfer Protocol) es el protocolo usado en cada transacción de la Web (WWW). El hipertexto es el contenido de las páginas Web, y el protocolo de transferencia es el sistema mediante el cual se envían las peticiones de acceso a una página y la respuesta con el contenido.
También sirve el protocolo para enviar información adicional en ambos sentidos, como formularios con campos de texto.

HTTP es un protocolo sin estado, es decir, que no guarda ninguna información sobre conexiones anteriores. Al finalizar la transacción todos los datos se pierden. Por esto se popularizaron las cookies, que son pequeños archivos guardados en la propia computadora que puede leer un sitio Web al establecer conexión con él, y de esta forma reconocer a un visitante que ya estuvo en ese sitio anteriormente. Gracias a esta identificación, el sitio Web

puede almacenar gran número de información sobre cada visitante, ofreciéndole así un mejor servicio.

La versión actual de HTTP es la 1.1.

El protocolo HTTP está basado en el modelo cliente-servidor. Un cliente HTTP abre una conexión y envía su solicitud al servidor, el cual responderá con el recurso solicitado —si está disponible y su acceso es permitido— y la conexión se cierra.

Por ejemplo: Para obtener un recurso con el http://www.tuhost.example/index.html

1. Se abre un *socket* con el host www.tuhost.example, puerto 80 que es el puerto por defecto para HTTP.
2. Se envía un mensaje en el siguiente estilo:

```
GET /index.html HTTP/1.0
Host: www.example.com
User-Agent: HTTPTool/1.0
[Línea en blanco]
```

La respuesta del servidor está formada por encabezados seguidos del recurso solicitado, en el caso de una página Web:

```
HTTP/1.0 200 OK
Date: Fri, 31 Dec 2003 23:59:59 GMT
Content-Type: text/html
Content-Length: 1221

<html>
<body>
<h1>Página principal de tuHost</h1>
(Contenido)
  .
  .
  .
</body>
</html>
```

Al recibirse la respuesta, el servidor cierra la comunicación.

4.1.4.2 FTP.

El Protocolo de Transferencia de Archivos (FTP, File Transfer Protocol) es un protocolo de transferencia de archivos entre sistemas conectados a una red TCP, utilizando normalmente el puerto de red 20 y el 21, basado en la arquitectura cliente-servidor, de manera que desde un equipo cliente nos podemos conectar a un servidor para descargar archivos desde él o para enviarle nuestros propios archivos independientemente del sistema operativo utilizado en cada equipo.

Un problema básico de FTP es que está pensado para ofrecer la máxima velocidad en la conexión, pero no la máxima seguridad, ya que todo el intercambio de información, desde el login y password del usuario en el servidor hasta la transferencia de cualquier archivo, se realiza en texto plano sin ningún tipo de cifrado, con lo que un posible atacante lo tiene muy fácil para capturar este tráfico, acceder al servidor, o apropiarse de los archivos transferidos.

Para solucionar este problema son de gran utilidad aplicaciones como scp y sftp, incluidas en el paquete SSH, que permiten transferir archivos pero cifrando todo el tráfico.

Este protocolo se utiliza escribiendo la dirección del servidor al que queremos conectar, indicando con **ftp://** (servidor ftp) y no con **http://** (servidor Web).

4.1.4.3 MTP.

El Protocolo Simple de Transferencia de Correo Electrónico (SMTP, Simple Mail Transfer Protocol), esta basado en texto utilizado para el intercambio de mensajes de correo electrónico entre computadoras o distintos dispositivos (PDA's, teléfonos móviles, etc.).

Con el tiempo se ha convertido en uno de los protocolos más usados en Internet. Para adaptarse a las nuevas necesidades surgidas del crecimiento y popularidad de Internet se han hecho varias ampliaciones a este protocolo, como poder enviar texto con formato o archivos adjuntos.

SMTP se basa en el modelo cliente-servidor, donde un cliente envía un mensaje a uno o varios receptores. La comunicación entre el cliente y el servidor consiste enteramente en líneas de texto compuestas por caracteres ASCII. El tamaño máximo permitido para estas líneas es de 1000 caracteres.

Las respuestas del servidor constan de un código numérico de tres dígitos, seguido de un texto explicativo. El número va dirigido a un procesado automático de la respuesta por autómata, mientras que el texto permite que un humano interprete la respuesta. En el protocolo SMTP todas las órdenes, réplicas o datos son líneas de texto, delimitadas por el carácter <CRLF>. Todas las réplicas tienen un código numérico al comienzo de la línea. En el conjunto de protocolos TCP/IP, el SMTP va por encima del TCP, usando normalmente el puerto 25 en el servidor para establecer la conexión.

4.1.4.4 POP.

El Protocolo de Oficina Postal (POP, Post Office Protocol), se utiliza en clientes locales de correo para obtener los mensajes de correo electrónico almacenados en un servidor remoto. La mayoría de los suscriptores de los proveedores de Internet acceden a sus correos a través de POP3.

Las versiones del protocolo POP (informalmente conocido como POP1 y POP2) se han hecho obsoletas debido a las últimas versiones de POP3. En general cuando uno se refiere al término POP, nos referimos a POP3 dentro del contexto de protocolos de correo electrónico.

El diseño de POP3 y sus predecesores permite que los usuarios con conexiones intermitentes (tales como las conexiones módem), descarguen su correo-e cuando se

encuentren conectados de tal manera que puedan ver y manipular sus mensajes sin necesidad de permanecer conectado. Cabe mencionar que la mayoría de los clientes de correo incluyen la opción de dejar los mensajes en el servidor, de manera tal que, un cliente que utilice POP3 se conecta, obtiene todos los mensajes, los almacena en la computadora del usuario como mensajes nuevos, los elimina del servidor y finalmente se desconecta. En contraste, el protocolo IMAP permite los modos de operación conectado y desconectado.

Los clientes de correo electrónico que utilizan IMAP dejan por lo general los mensajes en el servidor hasta que el usuario los elimina explícitamente. Esto y otros factores hacen que la operación de IMAP permita a múltiples clientes acceder al mismo buzón de correo. La mayoría de los clientes de correo electrónico soportan POP3 o IMAP; sin embargo, sólo unos cuantos proveedores de Internet ofrecen IMAP como un valor agregado a sus servicios.

Al igual que otros viejos protocolos de Internet, POP3 utilizaba un mecanismo de firmado sin cifrado. La transmisión de contraseñas de POP3 en texto plano aún se da. En la actualidad POP3 cuenta con diversos métodos de autenticación que ofrecen una diversa gama de niveles de protección contra los accesos ilegales al buzón de correo de los usuarios. Uno de estos es APOP, el cual utiliza funciones MD5 para evitar los ataques de contraseñas.

4.1.4.5 TELNET.

El protocolo Telnet (y del programa informático que implementa el cliente), sirve para acceder mediante una red a otra máquina, para manejarla como si estuviéramos sentados delante de ella. Para que la conexión funcione, como en todos los servicios de Internet, la máquina a la que se acceda debe tener un programa especial que reciba y gestione las conexiones. El puerto que se utiliza generalmente es el 23.

Sólo sirve para acceder en modo terminal, es decir, sin gráficos, pero fue una herramienta muy útil para arreglar fallos a distancia, sin necesidad de estar físicamente en el mismo sitio que la máquina que los tenía. También se usaba para consultar datos a distancia, como datos personales en máquinas accesibles por red, información bibliográfica, etc.

Su mayor problema es de seguridad, ya que todos los nombres de usuario y contraseñas necesarias para entrar en las máquinas viajan por la red como texto plano (cadenas de texto sin cifrar). Esto facilita que cualquiera que espíe el tráfico de la red pueda obtener los nombres de usuario y contraseñas, y así acceder él también a todas esas máquinas. Por esta razón dejó de usarse, casi totalmente, hace unos años, cuando apareció y se popularizó el SSH, que puede describirse como una versión cifrada de Telnet.

Para iniciar una sesión con un intérprete de comandos de otro computador, puede emplear el comando telnet seguido del nombre o la dirección IP de la máquina en la que desea trabajar, por ejemplo si desea conectarse a la máquina unam.mx deberá teclear telnet unam.mx, y para conectarse con la dirección IP 1.2.3.4 deberá utilizar telnet 1.2.3.4.

Una vez conectado, podrá ingresar el nombre de usuario y contraseña para iniciar una sesión en modo texto a modo de consola virtual. La información que transmita (incluyendo su clave) no será protegida o cifrada y podría ser vista en otros computadores por los que se transite la información (la captura de estos datos se realiza con Sniffers). Una alternativa más segura para Telnet, pero que requiere más recursos del computador es SSH que cifra la información antes de transmitirla, que autentica la máquina a la cual se conecta y que puede emplear mecanismos de autenticación de usuarios más seguros.

Hay tres razones principales por las que el Telnet no se recomienda para los sistemas modernos desde el punto de vista de la seguridad:

Telnet tienen varias vulnerabilidades descubiertas sobre los años, y varias más que podrían aún existir.

Telnet, por defecto, no cifra ninguno de los datos enviados sobre la conexión (contraseñas inclusive), así que es trivial escuchar detrás de las puertas en las comunicaciones, y utilizar la contraseña más adelante para propósitos maliciosos.

Telnet carece de un esquema de autentificación que permita asegurar que la comunicación esté siendo realizada entre los dos anfitriones deseados, y no interceptado en el centro.

En ambientes donde es importante la seguridad, por ejemplo en el Internet público, Telnet no debe ser utilizado. Las sesiones de Telnet no son cifradas. Esto significa que cualquiera que tiene acceso a cualquier router, switch, o gateway localizado en la red entre los dos anfitriones donde se está utilizando Telnet puede interceptar los paquetes de Telnet que pasan cerca y obtener fácilmente la información de la conexión y de la contraseña (y cualquier otra cosa que se mecanografía). Estos defectos han causado el abandono y depreciación del protocolo Telnet rápidamente, a favor de un protocolo más seguro y más funcional llamado SSH, lanzado en 1995. SSH provee de toda la funcionalidad presente en Telnet, la adición del cifrado fuerte para evitar que los datos sensibles tales como contraseñas sean interceptados, y de la autentificación mediante llave pública, para asegurarse de que el computador remoto es realmente quién dice ser.

4.1.4.6 SIP.

El Protocolo de Inicialización de Sesiones (SIP, Session Initiation Protocol) es un protocolo desarrollado por el IETF MMUSIC Working Group con la intención de ser el estándar para la iniciación, modificación y finalización de sesiones interactivas de usuario donde intervienen elementos multimedia como el video, voz, mensajería instantánea, juegos online y realidad virtual. SIP es uno de los protocolos de señalización para voz sobre IP, junto con H.323.

Los clientes SIP usan el puerto 5060 en TCP y UDP para conectar con los servidores SIP. SIP es usado simplemente para iniciar y terminar llamadas de voz y video. Todas las comunicaciones de voz/video van sobre RTP (Real-time Transport Protocol).

Un objetivo de SIP fue aportar un conjunto de las funciones de procesamiento de llamadas y capacidades presentes en la red pública conmutada de telefonía. Así, implementó funciones típicas que permite un teléfono común como son: llamar a un número, provocar que un teléfono suene al ser llamado, escuchar la señal de tono o de ocupado. La implementación y terminología en SIP son diferentes.
SIP es un protocolo punto a punto (también llamado p2p).

4.1.4.7 SNMP.

El Protocolo Simple de Administración de Red (SNMP, Simple Network Management Protocol) es un protocolo de la capa de aplicación que facilita el intercambio de información de administración entre dispositivos de red. SNMP permite a los administradores supervisar el desempeño de la red, buscar y resolver sus problemas, y planear su crecimiento.
Las versiones de SNMP más utilizadas son dos: SNMP versión 1 (SNMPv1) y SNMP versión 2 (SNMPv2). Ambas versiones tienen un número de características en común, pero SNMPv2 ofrece mejoras, como por ejemplo, operaciones adicionales.

SNMP en su última versión (SNMPv3) posee cambios significativos con relación a sus predecesores, sobre todo en aspectos de seguridad, sin embargo no ha sido mayoritariamente aceptado en la industria.

Una red administrada a través de SNMP consiste de tres componentes claves:

- Dispositivos administrados.
- Agentes.
- Sistemas administradores de red (NMS).

Un dispositivo administrado es un nodo de red que contiene un agente SNMP y reside en una red administrada. Estos recogen y almacenan información de administración, la cual es puesta a disposición de los NMS usando SNMP. Los dispositivos administrados, a veces llamados elementos de red, pueden ser routers, servidores de acceso, switches, bridges, hubs, computadores o impresoras.

Un agente es un módulo de software de administración de red que reside en un dispositivo administrado. Un agente posee un conocimiento local de información de administración, la cual es traducida a un formato compatible con SNMP.

Un NMS ejecuta aplicaciones que supervisan y controlan a los dispositivos administrados. Los NMS proporcionan el volumen de recursos de procesamiento y memoria requeridos para la administración de la red. Uno o más NMS deben existir en cualquier red administrada.

Los dispositivos administrados son supervisados y controlados usando cuatro comandos SNMP básicos: Lectura, escritura, notificación y operaciones transversales.

El comando de lectura es usado por un NMS para supervisar elementos de red. El NMS examina diferentes variables que son mantenidas por los dispositivos administrados.

El comando de escritura es usado por un NMS para controlar elementos de red. El NMS cambia los valores de las variables almacenadas dentro de los dispositivos administrados.

El comando de notificación es usado por los dispositivos administrados para reportar eventos en forma asíncrona a un NMS. Cuando cierto tipo de evento ocurre, un dispositivo administrado envía una notificación al NMS.

Las operaciones transversales son usadas por el NMS para determinar qué variables soporta un dispositivo administrado y para recoger secuencialmente información en tablas de variables, como por ejemplo, una tabla de rutas.

4.1.4.8 DNS.

Generalmente no se trabaja con direcciones IP sino con nombres de dominio del estilo de www.unam.mx o www.hotmail.com. Para que esto pueda ser posible es necesario un proceso previo de conversión de nombres de dominio a direcciones IP, ya que el protocolo IP requiere direcciones IP al enviar sus datagramas. Este proceso se conoce como resolución de nombres.

En los orígenes de Internet, cuando sólo había unos cientos de computadoras conectadas, la tabla con los nombres de dominio y direcciones IP se encontraba en un archivo de nombre HOST.TXT, almacenado en una única computadora. Todas las computadoras debían consultarle a éste cada vez que tenían que resolver un nombre. Este archivo contenía una estructura plana de nombres, funcionaba bien ya que la lista sólo se actualizaba una o dos veces por semana.

Sin embargo, a medida que se fueron conectando más ordenadores a la red comenzaron los problemas: el archivo HOSTS.TXT comenzó a ser demasiado extenso, el mantenimiento se hizo difícil ya que requería más de una actualización diaria y el tráfico de la red hacia esta computadora llegó a saturarla.

Es por ello que fue necesario diseñar un nuevo sistema de resolución de nombres que distribuyese el trabajo entre distintos servidores. Se ideó un sistema jerárquico de resolución conocido como DNS (Domain Name System, Sistema de Resolución de Nombres), el cual es una base de datos distribuida y jerárquica que almacena información asociada a nombres de dominio en redes como Internet. La asignación de nombres a direcciones IP es ciertamente la función más conocida de los protocolos DNS. Por ejemplo, si la dirección IP del sitio FTP de unam.mx. es 200.64.128.4, la mayoría de la gente llega a este equipo especificando www.unam.mx y no la dirección IP. Además de ser más fácil de recordar, el nombre es más fiable. El DNS nació de la necesidad de recordar fácilmente los nombres de todos los servidores conectados a Internet.

Para la operación práctica del sistema DNS se utilizan tres componentes principales:

- **Los Clientes DNS** (resolvers), un programa cliente DNS que se ejecuta en la computadora del usuario y que genera peticiones DNS de resolución de nombres a un servidor DNS (Por ejemplo: ¿Qué dirección IP corresponde a nombre.dominio?);
- **Los Servidores DNS** (name servers), que contestan las peticiones de los clientes, los servidores recursivos tienen la capacidad de reenviar la petición a otro servidor si no disponen de la dirección solicitada.
- **Las Zonas de autoridad**, porciones del espacio de nombres de dominio que almacenan los datos. Cada zona de autoridad abarca al menos un dominio y posiblemente sus subdominios, si estos últimos no son delegados a otras zonas de autoridad.

4.2 CONCLUSIONES CAPITULARES.

La tecnología aplicada en el área de las telecomunicaciones permiten tener acceso a una amplia gama de sistemas de comunicación alámbricas e inalámbricas, de las cuales Internet ha destacado en los últimos años por ser uno de los que cuentan con una gran infraestructura respaldada por fabricantes, usuarios e investigadores; además de evolucionar constantemente a los cambios y necesidades actuales, su universalidad y algo muy importante, su bajo costo.

Los modelos de referencia OSI y TCP/IP, este último con mayor dominio cada día, son una herramienta muy eficaz para la comunicación remota. Pero es la profunda comprensión y adecuada selección de la gran variedad de protocolos que los constituyen el lograr que una transmisión de datos, voz, video, audio o archivos, tenga éxito.

Algo muy importante es que del conjunto de protocolos, que se encuentran en cada capa del modelo de referencia, no se puede determinar cual de ellos es el mejor, el mejor o los mejores son aquellos que solucionen de la mejor manera un problema en específico.

5. WEB SERVER EMBEBIDO.

OBJETIVO: Documentar el proceso que conlleva el diseño de un Web Server embebido, el cual está constituido por un sistema electrónico embebido, cuyo sistema funcional es un conjunto de páginas Web alojadas en un microcontrolador.

Este sistema ofrece una solución económica a los problemas de comunicación remota ya que se puede monitorear y controlar, vía Ethernet/Internet, con tan sólo una computadora con un explorador Web.

5.1 INTRODUCCIÓN.

Un Web Server embebido es un sistema electrónico de tipo embebido que sustituye a un servidor Web, es decir, en el podemos colocar una página Web y poder hacer uso de ella desde algún punto de la red local o a través de Internet. [29]

Esto lo convierte en un adecuado medio de comunicación bidireccional, ya que además de poder desplegar información, se puede usar para poder monitorear y controlar las salidas digitales y analógicas de otro sistema e inclusive de él mismo.

El Web Server, propuesto en este trabajo, es un sistema electrónico de aplicación general y de bajo costo de implementación que con ciertos cambios en Hardware, Software y Firmware se puede adaptar a una aplicación en especial.

El Hardware del sistema estará constituido por un microcontrolador, un LCD, un teclado, una memoria serial, conectores a puertos analógicos y digitales, fuente de alimentación y cualquier otro dispositivo que requiera la aplicación.

El Firmware, que estará alojado en el microcontrolador, es programado en lenguaje C para microcontroladores y cubre todo el funcionamiento del microcontrolador pero principalmente la pila de protocolos de comunicación.

El Software abarca la programación, en lenguaje HTML, de las páginas Web que estarán alojadas en el microcontrolador.

El monitoreo y control del sistema se logra cuando un usuario, a través de una computadora con explorador Web, realiza la conexión con el sistema electrónico, el cual responde enviando las páginas Web, hacia el explorador, las cuales el usuario usará como interfaz entre él y el sistema.

El diseño de este Web Server se fundamenta en varias áreas del conocimiento, como son redes de cómputo y sus protocolos de comunicación, electrónica analógica y digital, sistemas embebidos, programación Web, a nivel ensamblador y a alto nivel, además de las áreas de la aplicación del sistema.

5.2 OBJETIVOS.

Diseñar e implementar un sistema electrónico de comunicación bidireccional que permita controlar y monitorear variables analógicas y digitales de manera remota haciendo uso de alguno de los sistemas de comunicación existentes.

El sistema además debe de contar con las características de ser reducido en tamaño, con una interfaz fácil de utilizar y con un bajo costo en su implementación, en su funcionamiento y en su actualización.

5.3 JUSTIFICACIÓN.

Hoy en día, uno de los sectores que más producen y consumen tecnología es el industrial, el cual a través de su amplia gama de aplicaciones hace indispensable la presencia de un sistema embebido.

Pero hay otros sectores de consumo como el hogar, el entretenimiento, administrativo, e inclusive el educativo que están optando por aplicar muchas de las tecnologías existentes para hacer más fácil muchas de sus actividades o inclusive impulsar la creación de muchas nuevas.

Aunque las aplicaciones en cada uno de estos sectores sean contrastantes, se pueden notar muchas áreas que son comunes en varios sectores, por ejemplo, las comunicaciones, ahorro de energía, seguridad, automatización, el control y monitoreo.

La comunicación de datos es una de las áreas que han permitido que las aplicaciones en muchos sectores, no sólo en la industria y en el hogar, estén en aumento, esto es debido a la posibilidad de que el usuario pueda realizar algunas actividades de manera remota, -desde

algunos metros de distancia hasta miles de kilómetros-, por ejemplo, envío de mensajes, encender o apagar un equipo, monitorear alguna o algunas variables, activar o desactivar alarmas, etc.

Es también un hecho de que existe una amplia gama de tecnologías que permiten el intercambio de datos, las cuales se adaptan a la también amplia gama de aplicaciones y necesidades que existen, como puede ser distancia, velocidad, protocolos, accesibilidad y costo.

5.4 DESCRIPCIÓN DEL PROYECTO.

En el presente capítulo se describirá el proceso de diseño e implementación de un sistema electrónico que por su funcionalidad es conocido como Web Server embebido, el cual fue divido en tres partes: Hardware, Software y Firmware.

El Hardware cubrió todos los aspectos relacionados con los dispositivos electrónicos que se utilizaron para la implementación física del sistema, principalmente el dispositivo embebido que realiza la comunicación, para este caso resultó elegido el microcontrolador MC9S12NE64 de la compañía Freescale; el Software abarcó todo la programación relacionada con la elaboración de la interfaz de usuario, en esta ocasión se utilizó lenguaje HTML y Java Script; y por último el Firmware, el cual cubrió todo lo relacionado con la programación en lenguaje ensamblador, C y C++ del dispositivo embebido, elemento principal de este sistema electrónico.

5.5 DISEÑO.

Una vez que se tendieron las bases del proyecto, se inicia el proceso de diseño e implementación. Para esto fue necesario elaborar un diagrama a bloques funcionales del

sistema, este permite delimitar funciones de los dispositivos y archivos que en Hardware, Software y Firmware se utilizarán. [30]

La figura 5.1 muestra el diagrama a bloques funcionales del sistema. En el se puede observar como algunos bloques tienen comunicación con otros y en forma bidireccional o unidireccional.

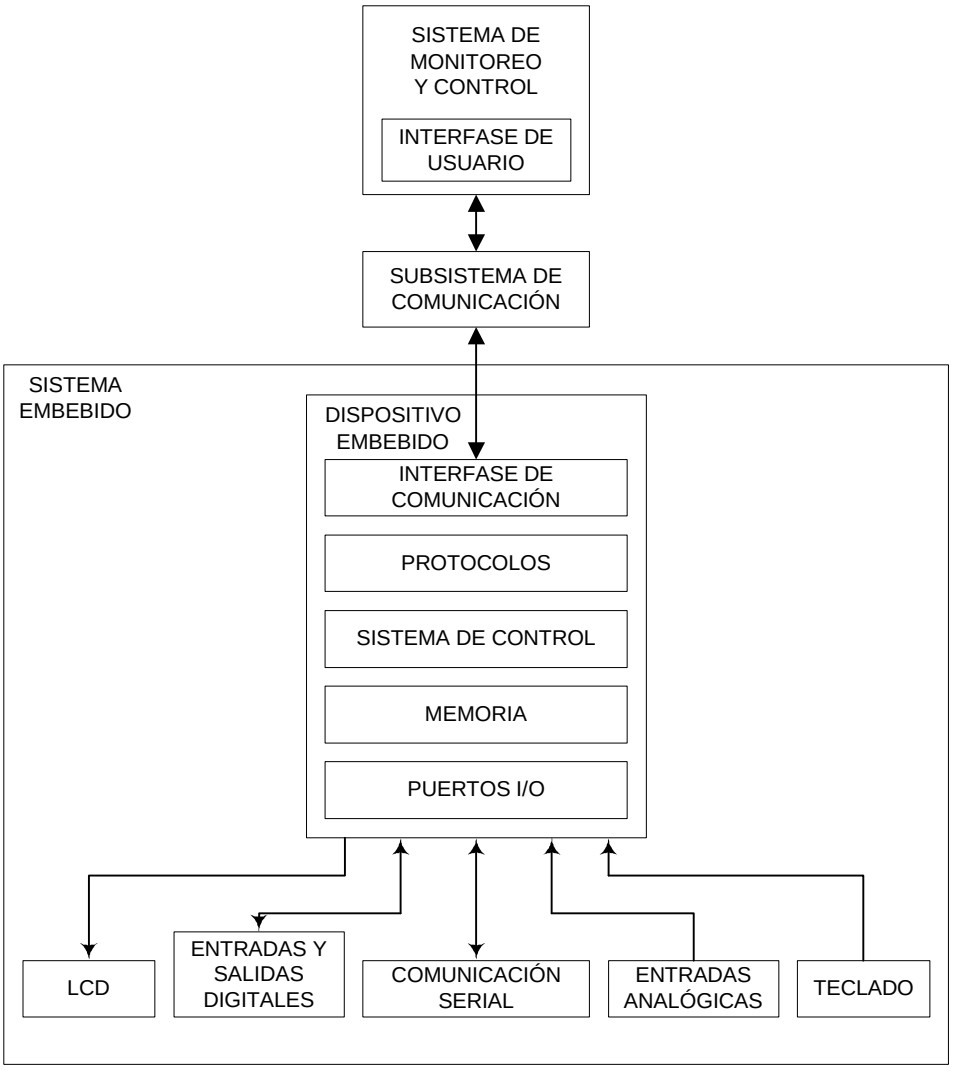


Figura 5.1 Diagrama a bloques del sistema propuesto.

5.5.1 SUBSISTEMA DE COMUNICACIÓN.

El factor más importante que se tomo en cuenta, antes de iniciar el diseño, fue el seleccionar el sistema electrónico de comunicación que permita enlazar el sistema propuesto en este proyecto con otro que lo pueda controlar y monitorear de manera remota.

Actualmente existen en el mercado de las telecomunicaciones diferentes sistemas que permiten un enlace entre dos o más sistemas electrónicos, cada uno de ellos ofrece alguna ventaja sobre otro en velocidad, costo, protocolos, alcance, ancho de banda, etc.

La tabla 5.1 resume las características, que se tomaron en cuenta para este proyecto, de los principales sistemas de comunicación que existen.

Sistema	Alcance	Costo de Servicio	Costo de Equipo
Telefonía Fija	Limitado	Medio	Bajo
Telefonía Móvil	Limitado	Alto	Madio
Internet	Total	Bajo	Bajo
Radio Frecuencia	Limitado	Alto	Alto

Tabla 5.1 Principales Sistemas de comunicación y características evaluadas para su selección.

Hay que aclarar que en todos los sistemas de comunicación su alcance puede ser total, pero esto eleva el costo, por ejemplo en la telefonía fija o móvil, si se requiere ampliar la zona de cobertura de un país a otro, el factor larga distancia se tendrá que tomar en cuenta. Lo mismo sucede con algún sistema de radio frecuencia, si se desea ampliar el área de cobertura se tendrá que hacer uso de repetidoras o hasta satélites, lo que también eleva el costo.

En el caso de Internet, para ampliar su cobertura también requiere el uso de distintas tecnologías, pero esto no influye en el precio que paga el usuario. Por lo tanto, el sistema elegido para este proyecto debido a su bajo costo y gran alcance es Internet.

5.5.2 SISTEMA DE MONITOREO Y CONTROL.

Este es el sistema o equipo que, a través del subsistema de comunicación elegido, permita al usuario monitorear y controlar el sistema electrónico embebido.

Debido a que el subsistema de comunicación elegido resultó ser Internet, el equipo más adecuado para realizar el monitoreo y control del sistema electrónico embebido es una computadora personal, aunque actualmente existen en el mercado otras opciones como un teléfono móvil que el equipo resulta más barato que una computadora pero el servicio de conexión se eleva considerablemente.

5.5.2.1 INTERFAZ DE USUARIO.

Para el diseño de una interfaz de usuario, se tomo en cuenta que el sistema se puede conectar dentro de una red o a través de Internet, debera de contar con un ambiente gráfico fácil de entender y usar, además de que su elaboración sea con Software económico.

Existen diferentes opciones para elaborar una interfaz de usuario que permita el monitoreo y control de otro sistema, además de la conexión a Internet; de ellos se puede mencionar algún lenguaje de programación como C, Visual Basic o LabView entre otros, pero la opción elegida fue el explorador Web con que cuenta todo sistema operativo como Windows o Linux y que requiere tan sólo de su instalación para usarlo. Todo explorador Web como Explorer o Mozilla Firefox, responden al protocolo HTTP, es decir, reproducen las páginas Web que se encuentran alojadas en un servidor Web y su elección dependerá totalmente del usuario.

Esto significa que varias páginas Web estarán almacenadas en el sistema electrónico embebido, las cuales serán mostradas en el explorador Web cuando se realice una conexión, esta es la razón por la cual el sistema propuesto en este trabajo es conocido como Web Server. [31]

5.5.2.2 HTML.

Para la programación en ambiente Web existe una amplia variedad de lenguajes y herramientas de programación, pero de ellos sobresale HTML (Lenguaje de Marcación de Hipertexto), el cual es un lenguaje que no requiere compilador, ya que es el propio navegador de Internet el encargado de convertir el contenido de una archivo con extensión HTM ó HTML en un ambiente gráfico.

Un archivo html o htm se crea de una manera muy fácil en cualquier procesador de texto lineal, son archivos pequeños y puede vincular un archivo de imagen, video, y animación multimedia para una mejor presentación de la interfaz.

Una propiedad de html es que permite auxiliarse de otras herramientas y lenguajes como CSS o Java Script, los cuales permiten hacer una página Web de una forma más compacta, práctica y rápida.

5.5.2.3 IMPLEMENTACIÓN DE LA INTERFAZ DE USUARIO.

Como el objetivo de una interfaz es proporcionarle al usuario un medio gráfico con el cual pueda interactuar con el sistema embebido que se pretende controlar y monitorear, la interfaz diseñada para este proyecto tiene una representación gráfica de los puertos analógicos y digitales que el usuario puede acceder en el sistema electrónico.

Las páginas Web diseñadas con el objetivo de ser alojadas en un sistema embebido deben de tener la característica de ser sencillas y compactas, pero bien estructuradas; esto es debido a que entre mayor y más elaborada sea una página Web, más grande es el archivo html que la contenga y más espacio en memoria del microcontrolador ocupará. Esta es la

razón por la que la interfaz o página para este proyecto se diseño de una forma muy básica y de la siguiente manera:

Se utilizó la técnica de frames de html con el fin de que la página Web aparente ser sólo una, más sin embargo está constituida por tres páginas, una de las cuales se actualiza constantemente y es la que corresponde a los puertos del microcontrolador.

Con CSS se logró que las páginas que constituyen la interfaz fueran más compactas debido a la manera de manejar estilos en las páginas html; y con la ayuda de Java Script se implementan funciones que ahorran código.

La figura 5.2 muestra la página Web que se utilizó como interfaz del proyecto.

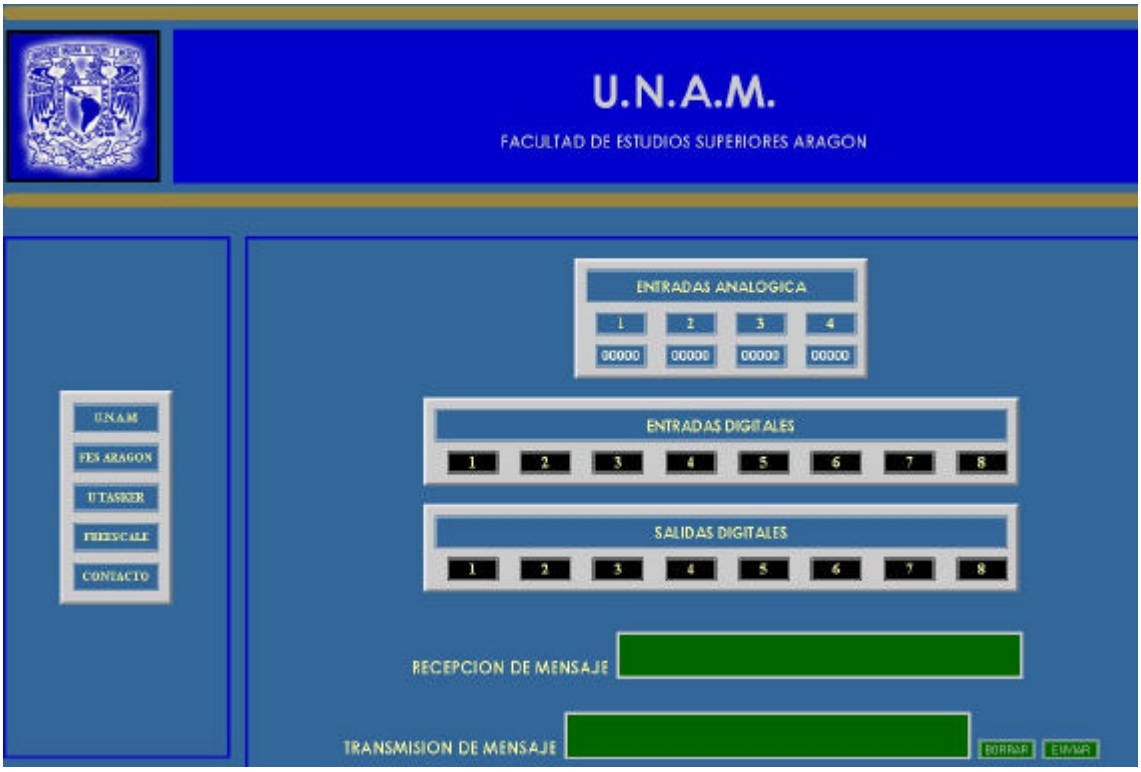


Figura 5.2 Página Web como interfaz de usuario.

En la figura 5.2 se puede observar la parte superior, que corresponde a una página, un encabezado de página con un logotipo; en la parte izquierda, que corresponde a otra página, se puede observar una parte de ligas hacia otras páginas; y por último la parte derecha es el área en donde están colocados a manera de botones los puertos digitales y analógicos que el usuario puede manipular, además de una área en donde el usuario puede enviar un mensaje de ochenta caracteres y un área, también de ochenta caracteres, en donde se puede desplegar un mensaje proveniente de el sistema electrónico.

5.5.3 DISPOSITIVO EMBEBIDO.

Una vez que se seleccionó a Internet como el subsistema de comunicación, se procedió a seleccionar el dispositivo que cumplieran con el funcionamiento del sistema propuesto en la figura 5.1, es decir, que tenga con las siguientes características:

- **Interfaz de comunicación** – es la parte del dispositivo que se encargará de realizar la comunicación, al tratarse de Internet, este módulo corresponderá a la capa física del modelo OSI o al acceso de red del modelo TCP/IP.

- **Protocolos** – son los protocolos que cumplen con el modelo OSI o TCP/IP, estos deben ser implementados y ubicados dentro del Firmware del dispositivo, es decir dentro de la memoria de programa.

- **Sistema de control** – el la unidad de control que contiene la mayoría de dispositivos embebidos, la cual debe tener la característica de velocidad y amplitud en bits de sus bus de datos, los suficientes para poder transmitir a la par de la tecnología ethernet.

- **Memoria** – es el espacio que debe contener cualquier dispositivo embebido para almacenar de manera temporal algunas variables o registros, memoria RAM; y para almacenar de manera permanente el programa que ejecutará, memoria FLASH o EEPROM de programa.

- **Puertos de entrada/salida** – son los puertos analógicos y digitales que el dispositivo debe contener para poder conectar en el todo aquello que el usuario requiera para su aplicación.

5.5.3.1 SELECCIÓN DE MC912NE64.

Actualmente existen en el mercado de los dispositivos electrónicos una amplia variedad de productos que pueden cumplir con una tarea específica. La tabla 5.2 resume las características más importantes de los principales dispositivos que pueden cumplir con el objetivo propuesto.

Dispositivo	Marca	Características	Tarjeta de Evaluación	Precio
PIC18F97J60	Microchip	Bus de 8 Bits, 128Kb de memoria, Oscilador 25Mhz	No	NA
AT91SAM7X	Atmel	Bus de 32 Bits, 512Kb de memoria, Oscilador 20Mhz	Sí	$4200.00
MC912NE64	Freescale	Bus de 16 Bits, 64Kb de memoria, Oscilador 20Mhz	Sí	$900.00
M5223X	Freescale	Bus de 32 Bits, 256Kb de memoria, Oscilador 60Mhz	Sí	$1200.00
STR91XF	ST	Bus de 32 Bits, 256Kb de memoria, Oscilador 25Mhz	Sí	$4000.00
LPC23XX	NXP	Bus de 32 Bits, 512Kb de memoria, Oscilador 24Mhz	Sí	$3600.00

Tabla 5.2 Principales dispositivos y sus características que cumplen con el objetivo del proyecto.

Todos los dispositivos citados en la tabla anterior tienen la característica de poseer un controlador de acceso al medio (MAC) de y transmisor-receptor Ethernet (EPHY), es decir que no requieren un circuito controlador de red o NIC para hacer una conexión física a una red.

El criterio de selección fue el siguiente:

- Que el fabricante contaran con una tarjeta con el dispositivo colocado (ya que todos poseen encapsulado de montaje superficial).
- Bajo costo.
- Compilador en lenguaje C sin costo adicional.
- Que el dispositivo sea conocido y por lo tanto posea un soporte en recursos.

El dispositivo que más se apego a los criterios de selección fue el microcontrolador MC9S12NE64 de Freescale, ya que el fabricante ofrece una tarjeta a un bajo costo, con compilador de C incluido, además de que dicho microcontrolador posee un núcleo de su antecesor el muy conocido HC12.

5.5.3.2 FIRMWARE.

Esta es la parte del diseño e implementación más compleja, debido a que el Firmware es quien más relaciona todas las partes del diseño, es decir, si en el Software o página Web se decidió manejar un determinado puerto como salida, el Firmware debe de determinar que nivel lógico se desea a la salida de dicho puerto y reflejarlo en su terminal correspondiente del Hardware.

Es por lo tanto el Firmware, como en la mayoría de los sistemas embebidos, quien realiza la mayor parte del funcionamiento del sistema.

En este trabajo, el Firmware, complementado por las características propias del microcontrolador, ejecuta el manejo del LCD, el manejo de los puertos analógicos y digitales, un teclado, comunicación serial y lo más importante, la pila del protocolo de comunicación vía Ethernet.

5.5.3.3 LENGUAJE EMSAMBLADOR Y C PARA MICROCONTROLADORES.

Es el lenguaje ensamblador, el lenguaje típico de programación de cualquier microcontrolador, por lo regular es proporcionado un compilador sin costo alguno al diseñador, en el se programa línea a línea las instrucciones que en su momento ejecutará el microcontrolador; pero muchos fabricantes proporcionan, a veces con un costo adicional, un compilador de lenguaje de alto nivel. De los lenguajes de alto nivel que podemos encontrar para programar un microcontrolador, C sobresale por su popularidad, pero sobre todo por sus características propias del lenguaje. Esto lo convierte en la herramienta más útil en la programación de sistemas electrónicos embebidos, ya que permite elaborar programas más compactos y fáciles de depurar y modificar.

Para este microcontrolador, utilizado en este proyecto, Freescale proporciona sin costo alguno un compilador de lenguaje C, CodeWarrior V 3.1, que aunque esta limitado en la cantidad de programa y archivos a compilar, es suficiente para este proyecto.

5.5.3.4 LIBRERIAS.

Es común hoy en día que compañías o personas dedicadas a la programación, proporcionen, de forma gratuita o por medio de un pago, librerías o programas que realizan una función específica. Compañías como CMX Systems, Viola Systems o uTasker proporcionan de manera gratuita un conjunto de librerías para diferentes microcontroladores. uTasker sobresale de los demás por poseer una gran variedad de librerías sencillas y compactas, razones por las cuales fueron utilizadas en el presente proyecto.[32]

La lista de librerías de uTasker utilizadas para el MC9S12NE es:

- ARP
- DHCP
- DNS
- ETHERNET
- FTP
- HTTP
- ICMP
- IP
- POP
- SMTP
- TCP
- TCPIP
- TELNET
- UDP
- WEB
- KEYPAD
- LCD
- I^2C

5.5.3.5 IMPLEMENTACIÓN DEL FIRMWARE.

Una vez que fue implementada la interfaz, se determinó que puertos del Hardware se manejarán y habiendo elegido aquellas librerías que podrán realizar algunas las funciones que ejecutará el sistema, se procedió a conjuntar en el Firmware la parte de comunicación y de control.

La primera parte y la más importante de todas es la de implementar en el Firmware el protocolo de comunicación, lo anterior se fundamento en el modelo OSI y TCP/IP, como lo muestra la figura 5.3.

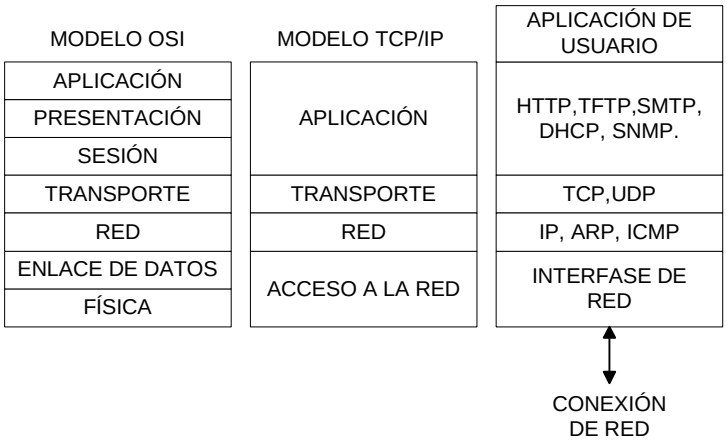


Figura 5.3 Protocolos implementados en Firmware.

La figura anterior muestra el conjunto de protocolos que fueron implementados para el funcionamiento de la sección de comunicación del sistema. Cabe aclarar que no todos los protocolos funcionan al mismo tiempo, esto depende del tipo de aplicación de usuario que se utiliza, por ejemplo, en esta aplicación donde el usuario utiliza un navegador de Internet, los protocolos son IP en la capa de Red, TCP en la capa de Transporte y HTTP en la aplicación.

El siguiente paso fue implementar un método para incorporar al microcontrolador las páginas Web que se crearon para la interfaz de usuario. Para lo anterior se utiliza FTP, que comúnmente es usado para bajar archivos de una página Web, pero en este caso sólo se utiliza para incorporar una página al sistema.

El tercer paso consistió en hacer que el sistema respondiera a alguna acción que el usuario realizará en la página Web desde el explorador de Internet. Para esto, a cada acción posible que pueda realizar el usuario, se le asignó un carácter diferente que es reenviado al sistema cada el usuario ejecuta una acción, una vez que el sistema recibe el carácter, decodifica cual de ellos es, ejecuta la acción, modifica la página Web y finaliza reenviado la página actualizada. Por ejemplo, si el usuario hace clic en alguno de los botones de las salidas digitales, se reenvía un carácter que el sistema identifica como una acción o petición del

usuario de cambiar el estado lógico de un bit del puerto de salida digital, esto es realizado y además la parte de memoria del microcontrolador correspondiente a la página Web y especial a la parte del color del botón que se oprimió se cambia de color y es reenviada la página al usuario.

Por último se implementó la parte correspondiente al envío de un mensaje del sistema al usuario, para esto se utilizó un teclado por medio del cual se ingresará caracter por caracter el mensaje a enviar, y cual también caracter por caracter es desplegado en el LCD; una vez que se finalizó la escritura del mensaje se oprime una función del teclado que hace que e sistema coloque los caracteres en el área de memoria de la página correspondiente al área de texto y reenvié la página al usuario.

5.5.4 HARDWARE.

A la par de la implementación del Firmware y de las páginas Web, se implemento también el Hardware, el cual consistió de lo siguiente:

- Un display de cristal liquido (LCD) monocromático el cual desplegará el mensaje enviado por el usuario desde una computadora y el que podrá enviar otro usuario desde el sistema embebido.
- Un puerto de 8 bits para señales digitales de entrada.
- Un puerto de 8 bits para señales digitales de salida.
- Un puerto de 4 bits entradas analógicas.
- Un teclado que permite introducir mensajes al LCD que posteriormente serán enviado al usuario, así como comandos de instrucción al sistema.

- Una memoria externa que permitirá ampliar la memoria interna del microcontrolador o alojar las páginas Web.
- Un MAX 232 (protocolo serial RS 232) hacia a una computadora, ya que es el medio por el cual se proporciona Firmware programado al microcontrolador.

- Acoplamiento, permite el acoplamiento de impedancias entre la red y el microcontrolador.

- Fuente de alimentación, proporciona los voltajes de alimentación para todos los dispositivos.

5.5.4.1 IMPLEMENTACIÓN FÍSICA DEL HARDWARE.

La implementación física de la tarjeta con todos los componentes se realizó con el Software Protel DXP de la empresa Altium. Cabe aclarar que antes de la elaboración física de la tarjeta fueron simuladas todas sus etapas en el compilador del microcontrolador, y no fue hasta que se logró una simulación adecuada que indicara un correcto funcionamiento cuando se decidió armar la tarjeta.

La fase selección del subsistema de comunicación, de la selección y diseño de la interfaz de usuario, de la selección del microcontrolador, del diseño del Firmware y del diseño y desarrollo del Hardware hicieron que el sistema propuesto en la figura 5.1 sufriera algunos cambios, dando como resultado el diagrama a bloques de la figura 5.4

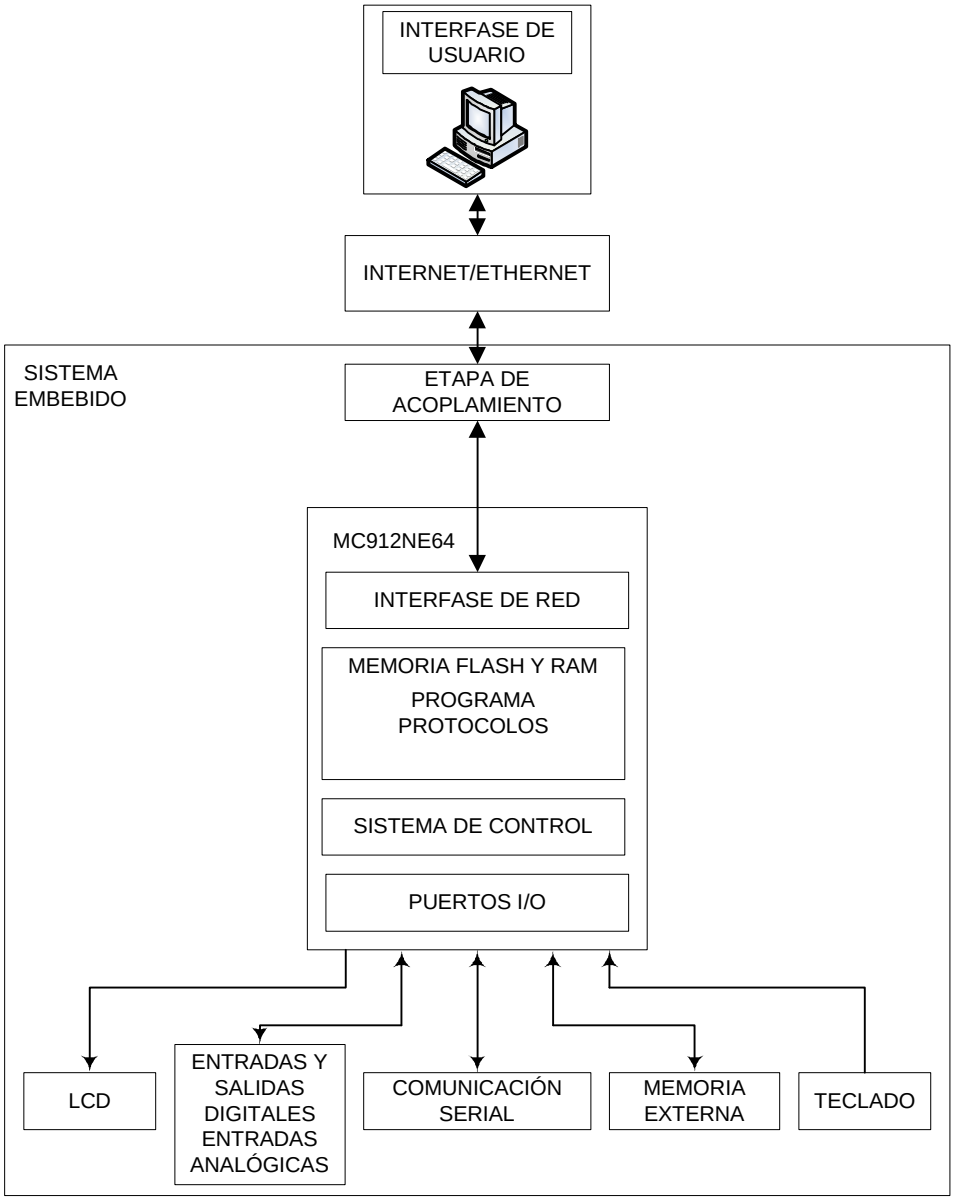


Figura 5.4 Diagrama del Sistema.

La figura 5.5 muestra el sistema implementado en este proyecto, en el cual se puede observar la tarjeta que vende el fabricante, Freescale, y la tarjeta implementada que contiene el LCD, los conectores para los puertos analógicos y digitales y la fuente de alimentación.

Figura 5.5 Sistema implementado total.

5.6 INSTALACIÓN Y USO DEL SISTEMA.

La instalación de este sistema es sumamente sencilla, sólo basta con ingresar, en el Firmware, una dirección IP, una máscara de subred y las puertas de enlace que corresponden a la red en donde estará instalado el sistema, si se quiere usar a nivel red LAN; ahora si se quiere hacer la conexión vía Internet, los datos ingresados al Firmware tienen que ser los que proporciona el proveedor de Internet.

Para hacer la conexión remota se abre el navegador de Internet y se escribe la dirección http://*direccion asignada al sistema*, en este momento se realiza la conexión respondiendo el sistema con la página Web. A partir de este momento el usuario es capaz de monitorear y controlar de manera remota el sistema.

Para asegurarse de los comandos o caracteres que el usuario envía al sistema se puede hacer uso de algún Sniffer, para este proyecto se usó Ethereal, que es un programa que lee y muestra al usuario la cadena de caracteres del datagrama que envía y recibe.

5.7 RECOMENDACIONES DE DISEÑO Y APLICACIONES.

Para sistemas de este tipo, es recomendable elaborar una página Web concisa y pequeña, ya que al ocupar espacio en memoria puede restar recursos a otra parte del sistema. También se recomienda programar en el Firmware sólo los protocolos y los recursos que se utilizará el sistema, ya que esto ahorra espacio en la memoria del microcontrolador. Para el Hardware se recomienda implementar las mayores salidas digitales y analógicas posibles, ya que tratándose de un sistema de aplicación general sólo bastará con activarlas o desactivarlas desde Firmware y en la página Web, pero tenerlas disponibles físicamente cuando la aplicación lo requiera.

Este sistema puede ser aplicado en área de monitoreo de datos, en la domótica o en áreas en donde los datos no requieran ser almacenados en grandes volúmenes, ya que esto lo tendría que realizar el microcontrolador, el cual se enfrenta con su limitante de memoria. Esto último se puede solucionar con dos opciones, una de ellas es emigrar la aplicación a un microcontrolador de mayor capacidad, y la otra opción es que la aplicación o interfaz de usuario sea un programa, como Visual Basic o LabView, los cuales permiten procesar y almacenar grandes cantidades de información provenientes del sistema electrónico.

5.8 CONCLUSIONES CAPITULARES.

En este capítulo se pudo comprobar como la correcta combinación y planeación de Software, Hardware y Firmware, puede dar como resultado la implementación de un sistema con las características de ser económico, personalizado, compacto y de amplia aplicación.

También en este capítulo se pudo comprobar que la correcta elección de herramientas se ve reflejada en los esperados y adecuados resultados.

Es posible implementar un Web Server Embebido con nueva tecnología en microcontroladores y con tecnología de comunicaciones ya existente y ampliamente conocida como es el Internet.

CONCLUSIONES GENERALES.

En este trabajo se pudo confirmar que el mercado de las telecomunicaciones ofrece grandes opciones para la transmisión de datos, de los cuales Internet sobresale por ser económico, muy accesible, universal y por permitir la conexión no sólo de sistemas de cómputo, si no también de sistemas y dispositivos electrónicos.

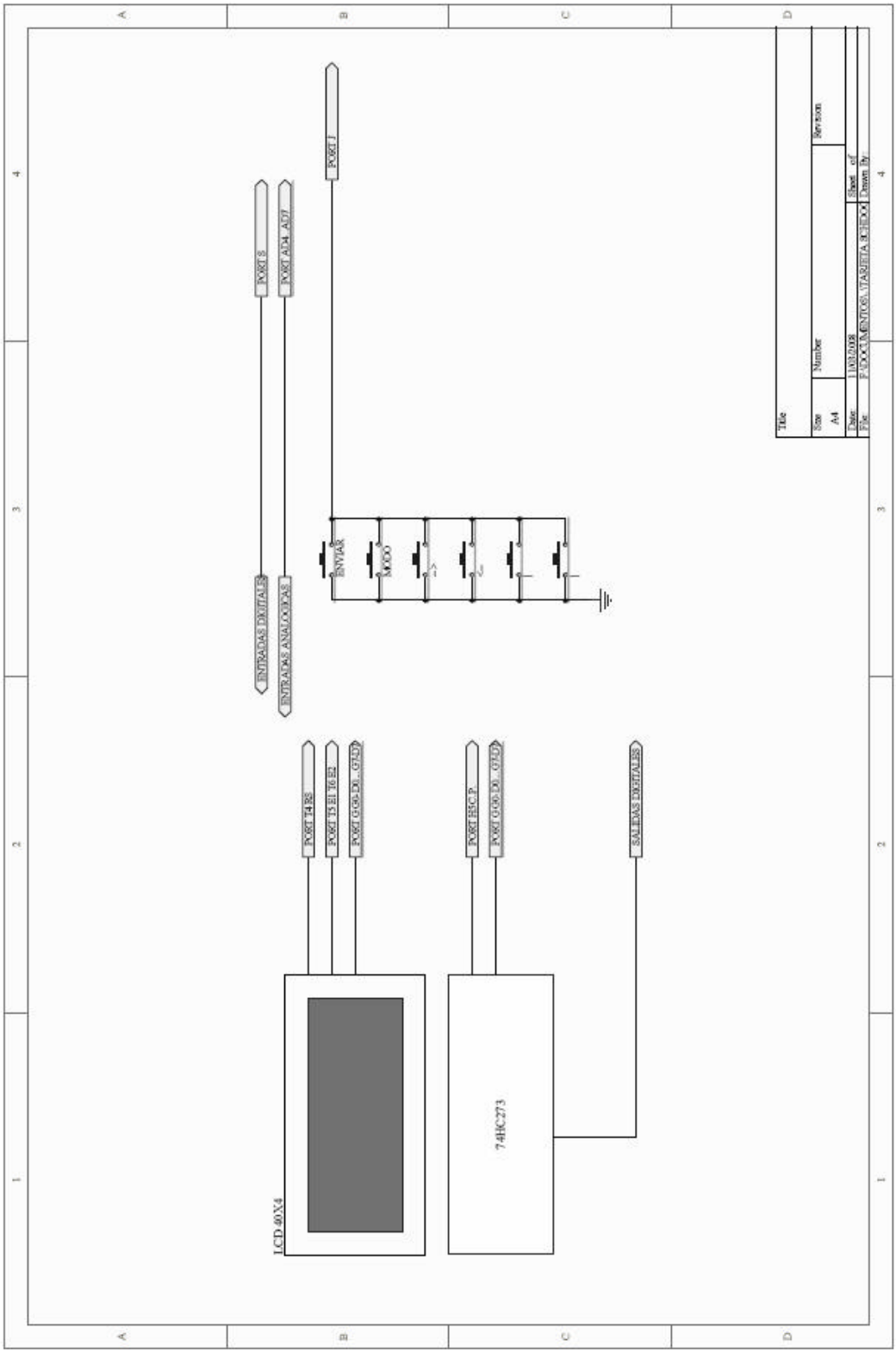
La amplia variedad de marcas y dispositivos embebidos permiten, hoy en día, la implementación de sistemas electrónicos compactos, económicos, personalizados e innovadores, capaces de satisfacer muchas de las necesidades que el ser humano enfrenta en la actualidad como la comunicación y el envió de información y datos.

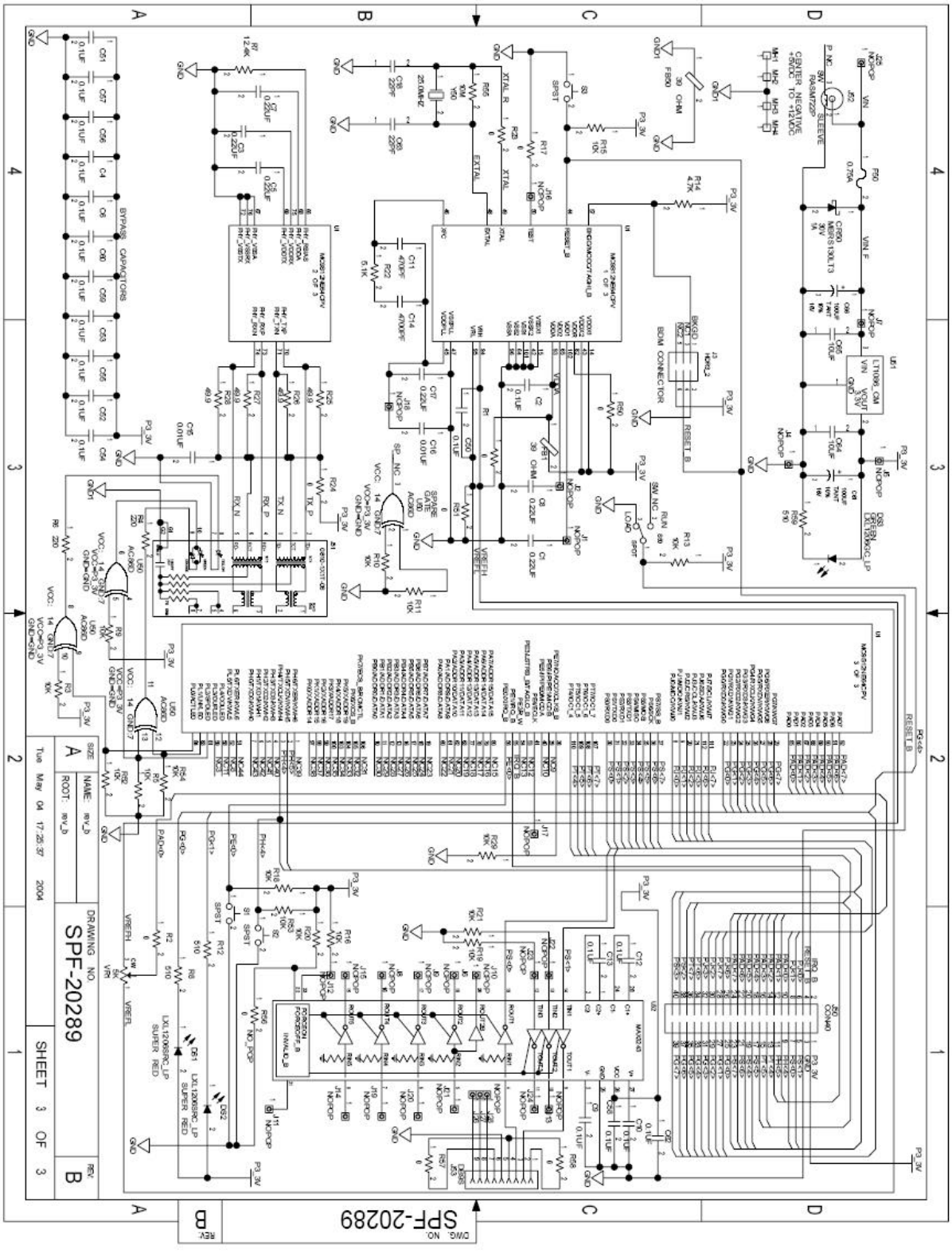
Los grandes adelantes tecnológicos en el área de la computación y en el de los semiconductores ofrecen una gran variedad en herramientas de Software, hardware y Firmware, capaces de ahorrar al diseñador tiempo, dinero y espacio, lo cual se ve reflejado en que la mayoría de áreas del conocimiento humano haya un sistema electrónico y si no, pronto lo habrá.

La correcta selección de un medio de comunicación y el adecuado conocimiento de sus protocolos, la adecuada elección de un dispositivo embebido, el correcto diseño de hardware, Software y Firmware; así como de la elección y adecuado uso de las herramientas de diseño pueden dar como resultado un eficaz medio de comunicación de aplicación general destinado para el control y monitoreo de variables, como es el Web Server embebido presentado en este trabajo.

APÉNDICE A. DIAGRAMA DEL SISTEMA.

En las siguientes dos figuras se muestra el diagrama completo del Web Server embebido.




SPF-20289
DRAWING NO.
REV. B
SHEET 3 OF 3
DRAWING NO SPF-20289
REV B

BIBLIOGRAFÍA.

[1]
VAHID, Frank & GIVARGIS, Tony: *Embedded System Design: A Unified Hardware/Software Approach*. EEUU: University of California Riverside, 1999.
[2]
BARR, Michael: *Programming Embedded Systems in C and C++*. EEUU: O'Reilly, 1999.
[3]
SUTTER, Ed: *Embedded Systems Firmware Demystified*. EEUU: CMP Books & CMP Media LLC, 2002.
[4][5]
BERGER, Arnold S.: *Embedded Systems Design: An Introduction to Processes, Tools, and Techniques*. EEUU: CMP Books & CMP Media LLC, 2002.
[6]
TORRES, Steven: AN2836: *Web Server Development with MC9S12NE64.* EEUU: Freescale Semiconductor, 2004.
[7]
TANENBAUM, Andrew S.: *Redes de Computadoras*. México: Prentice Hall, 2004.
[8]
ARTAIL, Hassan A.: *A distributed system of network-enabled microcontrollers for controlling and monitoring home devices*. EEUU: IEEE, 2002.
[9]
TORRES, Steven: AN2700: Basic *Web Server Development with MC9S12NE64 and CMX-Micronet TCP/IP Stack.* EEUU: Freescale Semiconductor, 2004.
[10]
TORRES, Steven: AN2692: Basic *MC9S12NE64 Integrated Ethernet Controller.* EEUU: Freescale Semiconductor, 2004.
[11]
MINOLI, Daniel: *First, Second and Next Generation LAN's*. EEUU: McGraw-Hill, 1994.
[12]
HSU, John Y.: *Computer Networks*. EEUU: Artech House, 1996.
[13]
BERTSEKAS, Dimitri: *Data Networks*. EEUU: Prentice-Hall, 1997.

[14]
FORTIER, Paul J.: *Handbook of LAN Technology*. EEUU: McGraw-Hill, 1992.
[15]
MOLINA, Francisco J.: *Redes de Área Local*. México: Alfaomega, 2004.
[16]
BRENTON, Chris: *Multiprotocol Network Design and Troubleshooting*. EEUU: Network Press, 1997.
[17]
SCHWARTZ, Mischa: *Redes de Telecomunicaciones*. México: Addison-Wesley, 1994.
[18]
NORRIS, Mark & PRETTY Steve: *Designing The Total Area Network*. EEUU: John Wiley & Sons, 2000.
[19]
CHORAFAS, Dimitris N.: *Local Area Network Reference*. EEUU: McGraw-Hill, 2000.
[20]
STEVENS, Richard W.: *TCP/IP Illustrated Vol. 1*. EEUU: Addison-Wesley, 1995.
[21]
FEIT, Sydney: *TCP/IP*. EEUU: McGraw-Hill, 1993.
[22]
HELD, Gilbert: *TCP/IP Professional Reference Guide*. EEUU: Averbach, 2001.
[23]
LYNCH, Daniel C. & ROSE, *Marshall T.: Internet System Handbook*. EEUU: Addison-Wesley, 1993.
[24]
QUORTEMAN John S. & CART-MITCHELL: *The Internet Conection*. EEUU: Addison-Wesley, 1994.
[25]
HALABI, Bassam: *Internet Routing Architectures*. EEUU: Cisco Press, 1997.
[26]
BLACK, Uyless: *TCP/IP and Related Protocols*. EEUU: Prentice-Hall, 1992.
[27]
STEVENS, Richard W.: *TCP/IP Illustrated Vol. 2*. EEUU: Addison-Wesley, 1995.
[28]

HELD, Gilbert: *Managing TCP/IP Networks*. EEUU: John Wiley & Sons, 2000.

[29]

BENTHAM, Jeremy: *TCP/IP Lean*, EEUU: CMP Books, 2002.

[30]

BALL, STUART R.: *Embedded Microprocessor Systems*: Real World Design: EEUU: 2002.

[31]

BURGESS, Mark: *Principles of Network and System Administration*: John Wiley & Sons, 2004.

[32]

www.uTasker:com