

Wireless Corba para Robots móviles

Sergio Rodríguez Casáñez

16 de diciembre de 2002

Índice general

	IV
Agradecimientos	v
Prólogo	vi
Introducción	vii
I Conceptos	1
1. Comunicación inalámbrica	2
1.1. Historia	3
1.2. Introducción	4
1.3. Definición de WLAN	5
1.4. Composición de una red wireless	6
1.5. ¿Por qué inalámbrica?	7
1.6. Aplicaciones	9
1.7. Configuraciones WLAN	10
1.8. Tipos de tecnologías wireless	14
1.8.1. Tecnología de espectro ensanchado	15
1.8.2. Tecnología de infrarrojos	18
1.9. Alternativas a la WLAN	19
1.9.1. Bluetooth	19
1.9.2. HomeRF	20
1.9.3. IrDA	21
1.9.4. Ultra Wide Band (UWB)	21
1.10. El estándar 802.11	23
1.11. Alta velocidad WLAN: 802.11a y 802.11g	25
1.11.1. El camino a la alta velocidad	25
1.11.2. 802.11a: alta velocidad, alta capacidad	26

1.11.3. 802.11g: alta velocidad en la banda de 2.4 GHz	28
1.11.4. Cambiando el estándar a alta velocidad	29
1.12. Consideraciones a tener en cuenta	30
1.13. Mejorando la productividad y la calidad de vida	33
1.13.1. Beneficios de la productividad	35
1.13.2. Beneficios de calidad de vida	39
2. Introducción a CORBA	42
2.1. Historia	43
2.2. El grupo OMG	45
2.2.1. La misión del OMG	45
2.2.2. Organización	47
2.2.3. Políticas de enlace	50
2.3. La norma CORBA	50
2.4. Estructura	53
2.5. Lenguaje de definición de interfaces	57
2.6. El modelo de comunicaciones de CORBA	61
2.7. El modelo de objetos de CORBA	62
2.8. Adaptadores de objetos	63
2.8.1. Adaptador básico de objetos (BOA)	64
2.8.2. Adaptador portable de objetos (POA)	67
2.9. Clientes y servidores CORBA	70
2.10. Stubs y skeletons	72
2.11. Desarrollo de aplicaciones CORBA	72
2.12. Invocación dinámica	74
2.12.1. Pasos para la realización de una invocación dinámica	74
2.12.2. Los interfaces de la invocación dinámica	75
2.12.3. Tipos de invocaciones	76
2.13. Servidores multihilo	78
2.13.1. Consecuencias en los sirvientes	78
2.13.2. Estrategias multihilo	80
2.14. Servicios corba	81
2.15. Facilidades comunes	82
2.16. Dominios corba	84
2.17. El protocolo GIOP	85
2.17.1. Representación común de datos	85
2.17.2. Codificación de los datos en CDR	87
2.18. Tipos de ORB's	90

<i>ÍNDICE GENERAL</i>	III
II Desarrollo	92
3. Desarrollo Wireless	93
3.1. Requisitos	94
3.2. Tecnología aplicada	94
3.3. Elección del producto	97
3.4. Instalación y configuración	102
3.4.1. Access point	102
3.4.2. Tarjeta pcmcia	102
4. Desarrollo Corba	104
4.1. Requisitos	105
4.2. Tecnología aplicada	105
4.3. Configuración del ORB	108
4.3.1. Configuración sobre plataforma Linux	108
4.3.2. Compilación sobre plataforma windows	109
4.4. Generación del IDL	111
4.5. Generación del cliente	112
4.6. Generación de la implementación	112
4.7. Generación del servidor	112
4.8. Generación del fichero Makefile	117
4.9. Breve manual de usuario	119
III Apéndices	121
A. Glosario WIRELESS	122
B. Glosario CORBA	130
C. Abreviaciones WIRELESS	136
D. Abreviaciones CORBA	141
E. Wireless Ethernet	154

Lo importante es aprender

Carlos A. García

Ricardo Sanz Bravo

Agradecimientos

Se lo dedico a mis padres, pues ellos han hecho posible la culminación de mis estudios.

También se lo dedico a todas aquellas personas que me han apoyado en estos años.

Prólogo

Desde la aparición del hombre en la tierra, éste ha ido evolucionando a lo largo de los años, mediante una serie de inventos que van desde el fuego o la rueda hasta los inventos más sofisticados de hoy en día.

Como dijo el filósofo Epicuro: *el bienestar es el fin del hombre*, de ahí la búsqueda de inventos que hagan la vida del hombre más fácil y placentera.

En la actualidad no solo se busca la simplicidad del trabajo mediante máquinas, sino que además son necesarias en muchos casos. Por ejemplo en el campo de la medicina en la que se utilizan robots que hacen incisiones demasiado precisas para la mano del hombre, por no comentar el trabajo de éstos en condiciones donde el ser humano corre peligro como puede ser: en las profundidades del mar, en las de una mina o en las misiones a otros planetas. Además hay que tener en cuenta la rapidez y precisión, gracias a los robots en una fábrica la producción es mucho mayor.

Por esto y mucho más cada vez se construyen más máquinas, más robots, en 1995 había más de 600000 robots que se distribuían en los campos de la medicina, del automóvil, del espacio, etc.

Cada vez los robots mejoran en algún aspecto pareciéndose más al hombre y en muchos casos mejorándolo, pero todavía está por ver si en un futuro las máquinas pensarán, actuarán como el hombre o mejor ante imprevistos que no se hayan programado. Hay una gran controversia sobre esto en la que científicos de todo el mundo tienen distintos puntos de vista. Se pueda hacer esto o no, es evidente que todos ellos siguen trabajando para mejorar estos “artilugios” que cada día nos hacen la vida más agradable.

En este proyecto intentamos aportar nuestro granito de arena uniéndonos a los científicos de todo el mundo.

Introducción

Este proyecto entra a formar parte de otro que le engloba. El proyecto "padre" se realiza para la investigación de sistemas autónomos en los que se utilizan técnicas de control de robots.

Como plataforma partimos de un coche teledirigido "*robot2*" al que se le han anulado algunas capacidades que se le instalaron de fábrica, es decir: el control remoto se le ha suprimido; se le ha cambiado el servo de dirección por otro mayor, debido al aumento de peso que dicho coche va a experimentar cuando se le introduzcan baterías, sensores, varios procesadores, etc; se han cambiado los engranajes metálicos para mayor durabilidad y se han eliminado los amortiguadores originales.

Aunque el robot es un mero juguete hay que tener en cuenta que todo las investigaciones y avances logrados con él siempre se pueden extrapolar a otros vehículos mayores.

En el coche, entre otras cosas, va incorporado como procesador principal un Wafer 5820, con factor de forma de 3.5 pulgadas, que es lo que nos interesa en nuestro proyecto "hijo" en cuestión.

Lo que intentamos realizar en este proyecto es la comunicación inalámbrica *comunicación sin cables* entre el wafer del robot y un PC, el cual está situado en una sala del departamento de automática de la Escuela Técnica superior de Ingenieros Industriales. Esta comunicación debe ser tal que desde el PC controlaremos al coche para poder dirigirlo a nuestro antojo por dentro y fuera del departamento, para ello el robot llevará una cámara incorporada para poder ver por donde circula. Como el departamento está constituido por muros de mucho grosor tendremos que asegurarnos que la comunicación cuando el coche esté fuera del departamento sea segura.

El proyecto constará de dos partes: establecer la instalación de los componentes, que hacen posible la comunicación entre ambos procesadores, sobre el



Figura 1: plataforma

sistema operativo Linux y la utilización de Corba y C++ para el desarrollo del software.

Parte I

Conceptos

Capítulo 1

Comunicación inalámbrica

1.1. Historia

Las redes de área local inalámbricas "*wireless LAN (WLAN)*" funcionan desde hace más de quince años en entornos industriales y de investigación. Este tipo de redes se implementó por primera vez en el año 1979.

La casa IBM Suiza, utilizó enlaces infrarrojos creando una red de área local en una fábrica. Posteriormente se utilizaron implementaciones basadas en tecnologías de microondas según los esquemas de transmisión de espectro ensanchado.

En marzo de 1985 la Comisión Federal de Comunicaciones, FCC, organismo encargado de la regulación de las telecomunicaciones en Estados Unidos, asignó a los sistemas WLAN las bandas frecuenciales 902-928 MHz., 2.400-2.4835 GHz. y 5.725-5.850 GHz también conocidas como ISM (Industrial, Científica y Médica) y que pueden utilizarse bajo licencia administrativa. Esta asignación de una localización frecuencial fija propició una mayor actividad industrial. En este punto las redes de área local inalámbrica dejaron de ser meramente experimentales para empezar a introducirse en el mercado.

Entre los años 1985 y 1990 se trabajó en el desarrollo de productos WLAN y finalmente, en mayo de 1991, se publicaron algunos trabajos que hablaban sobre redes inalámbricas que superaban la velocidad de transferencia de 1 Mbps velocidad mínima a partir de la cual el comité IEEE considera que una red es de área local. Hasta ese momento las WLAN habían tenido una aceptación marginal en el mercado por dos razones fundamentales: falta de un estándar y precios elevados de la solución inalámbrica.

En estos últimos años se ha producido un crecimiento en el mercado de hasta un 100 % anual. Este hecho es atribuible a dos razones principales: el desarrollo del mercado de los equipos portátiles y de las comunicaciones móviles que han producido que los usuarios puedan estar en continuo movimiento manteniendo comunicación constante con otros terminales y elementos de la red. En este sentido, las comunicaciones inalámbricas ofrecen una prestación no disponible en las redes cableadas: movilidad y acceso simultáneo a los recursos de la red.

La conclusión de la definición de la norma IEEE 802.11 para redes de área local inalámbricas en junio de 1997 ha establecido un punto de referencia y ha mejorado muchos de los aspectos de estas redes.

1.2. Introducción

En los últimos años se ha producido un crecimiento espectacular en lo referente al desarrollo y aceptación de las comunicaciones móviles y en concreto de las redes de área local (Wireless LANs). La función principal de este tipo de redes es la proporcionar conectividad y acceso a las tradicionales redes cableadas (Ethernet, Token Ring...), como si de una extensión de éstas últimas se tratara, pero con la flexibilidad y movilidad que ofrecen las comunicaciones inalámbricas. La tecnología LAN inalámbrica permite al cliente comunicación computacional sin cables para acceder a los servicios de área local.

La tecnología Inalámbrica libera a los usuarios de las limitaciones físicas de una conexión por cable para comunicaciones. Los dispositivos inalámbricos se comunican sin cables a una red de comunicaciones, por ejemplo, un teléfono sin cordón en tu casa se comunica inalámbricamente a su estación de recarga/poder la cual está físicamente conectada a la red de cableado telefónico.

La idea que queremos resaltar es que los sistemas WLAN no pretenden sustituir a las tradicionales redes cableadas, sino más bien complementarlas. En este sentido el objetivo fundamental de las redes WLAN es el de proporcionar las facilidades no disponibles en los sistemas cableados y formar una red total donde coexistan los dos tipos de sistemas.

Durante los últimos siete años, las LANs inalámbricas han ganado una fuerte popularidad en un gran número de mercados, incluyendo hospitales, minoristas, manufactura, almacenamiento, y universidades. Estas industrias se han beneficiado de la productividad ganada por el uso de terminales portátiles y notebooks para transmitir información en tiempo real a servidores centrales para procesarla. Business Research Group, una firma de investigación de mercado, predice una expansión de seis veces del mercado de WLANs a nivel mundial para el año 2001, alcanzando más de \$2 billones de ingresos.

Por último señalar la importancia de no confundir la tecnología LAN inalámbrica con celular o modems de radio usados para conectividad de WAN inalámbrica a la Internet. La tecnología LAN inalámbrica provee conectividad "en-edificio."^a una red cableada de conexión a Internet.



1.3. Definición de WLAN

Una red de área local inalámbrica puede definirse como a una red de alcance local que tiene como medio de transmisión el aire.

Por red de área local entendemos una red que cubre un entorno geográfico limitado, con una velocidad de transferencia de datos relativamente alta (mayor o igual a 1 Mbps tal y como especifica el IEEE), con baja tasa de errores y administrada de forma privada.

Por red inalámbrica entendemos una red que utiliza ondas electromagnéticas como medio de transmisión de la información que viaja a través del canal inalámbrico enlazando los diferentes equipos o terminales móviles asociados a la red. Estos enlaces se implementan básicamente a través de tecnologías de microondas y de infrarrojos. En las redes tradicionales cableadas esta información viaja a través de cables coaxiales, pares trenzados o fibra óptica.

Una red de área local inalámbrica, también llamada wireless LAN (WLAN), es un sistema flexible de comunicaciones que puede implementarse como una extensión o directamente como una alternativa a una red cableada.

Este tipo de redes utiliza tecnología de radiofrecuencia minimizando así la necesidad de conexiones cableadas. Este hecho proporciona al usuario una gran movilidad sin perder conectividad. El atractivo fundamental de este tipo de redes es la facilidad de instalación y el ahorro que supone la supresión del medio de transmisión cableado. Aún así, debido a que sus prestaciones son menores en lo referente a la velocidad de transmisión que se sitúa entre los 2 y los 10 Mbps frente a los 10 y hasta los 100 Mbps ofrecidos por una red convencional, las redes inalámbricas son la alternativa ideal para hacer llegar una red tradicional a lugares donde el cableado no lo permite, y en general

las WLAN se utilizarán como un complemento de las redes fijas.

1.4. Composición de una red wireless

Una red wireless es una red inalámbrica de área local conectada a un tipo de red metropolitana, normalmente localizada en un área geográfica como puede ser una ciudad o un pequeño pueblo.

Varias interfases wireless standard existen actualmente, algunas operando con licencia y otras sin ella, algunas punto a punto, otras punta a multi-punto, y otras todavía más flexibles.

La más común en este momento es la familia 802.11, en la que se comprenden: 802.11, 802.11a, 802.11b y 802.11g.

Cada red wireless está compuesta por varios nodos conectados juntos. Un nodo es una colección de varios PC's u otros equipos conectados juntos directamente usando la red IP y sobre un rango directo de radio.

Un nodo consiste mínimo de un router y de uno o más clientes. Los clientes normalmente requieren una pequeña configuración y solo hablan con el router, mientras el router enviará sus propios datos y los datos de los clientes al resto de la red.

El nodo participará en el intercambio de información con otros nodos garantizando siempre el alcance al resto de la red.

Los nodos pueden ser conectados por radio o por otros medios. El término puede ser asociado con el router/host el cual maneja cada nodo de la propia red local.

Al haber un rango limitado de señales de radio necesitamos un gran número de nodos para proveer de cobertura a todo un pueblo, de modo que se requiere un compleja malla de conexiones entre nodos para tener una red robusta. Donde los clientes de la red completan dicha red. Los nodos sin clientes forman cada grupo de infraestructura de red.

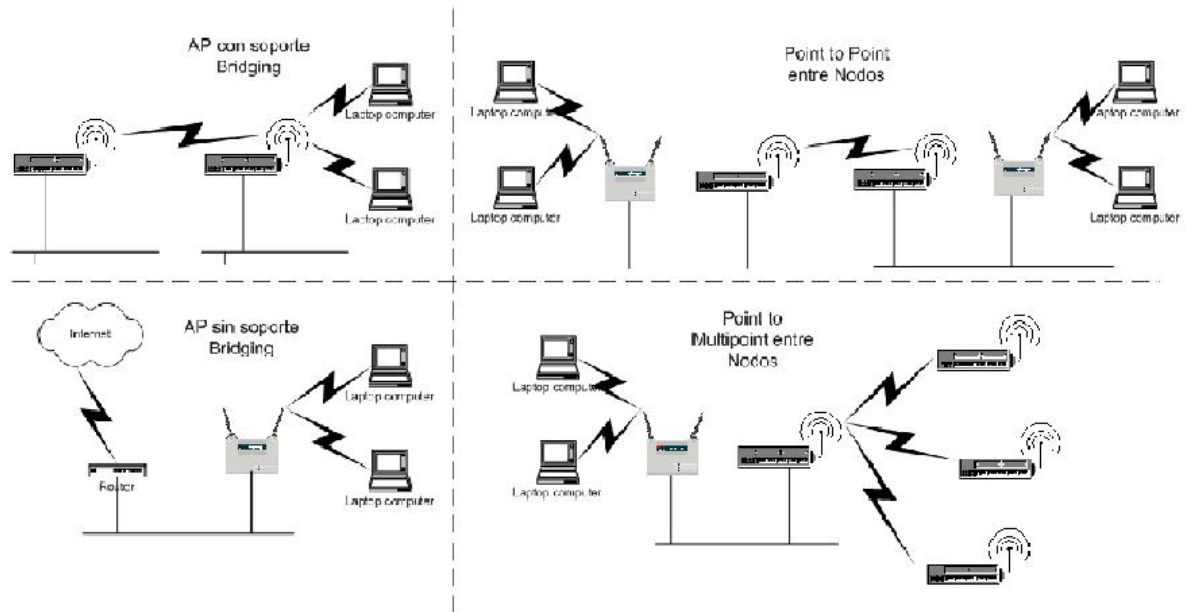


Figura 1.1: Tipos de nodos en una red wireless

1.5. ¿Por qué inalámbrica?

La generalizada confianza sobre redes en los negocios y el crecimiento meteórico de Internet y de servicios en línea son fuertes testimonios de los beneficios de compartir datos y compartir recursos. Con las WLAN, los usuarios pueden acceder a información compartida sin buscar un lugar donde enchufarse, y los administradores de red pueden establecer o aumentar las redes sin instalar o mover cables. Las WLAN ofrecen las siguientes ventajas de productividad, servicio, conveniencia, y costos sobre las tradicionales redes cableadas:

Movilidad:

sistemas de WLAN pueden proveer a usuarios LAN acceso a información en tiempo real en cualquier parte dentro de su organización. Esta movilidad soporta productividad y oportunidades de servicio imposibles con redes cableadas.

Simplicidad y rapidez de instalación:

Instalar un sistema de WLAN puede ser rápido y fácil y puede eliminar la necesidad de poner cables a través de murallas y techos.

Flexibilidad de Instalación:

La tecnología inalámbrica permite a la red ir donde los cables no pueden.

Reduce el costo del propietario:

Mientras la inversión inicial requerida para el hardware de una WLAN puede ser mas alto que el costo de hardware de una LAN cableada, en conjunto los gastos de instalación y costos de ciclo de vida pueden ser significativamente más bajos. Los beneficios del costo a largo plazo son más altos en entornos dinámicos requiriendo movimientos frecuentes, adiciones, y cambios.

Escalabilidad:

Los sistemas de WLAN pueden ser configurados en una variedad de topologías para conocer las necesidades de aplicaciones específicas e instalaciones. Las configuraciones son fácilmente cambiadas y el rango desde redes par-a-par adecuado para un pequeño número de usuarios a redes de infraestructuras enteras de miles de usuarios que permiten roaming sobre una amplia área.

1. Red de Área Local Inalámbrica (Wireless Local Area Network).
2. Red de Área Extensa Inalámbrica (Wireless Wide Area Network).
3. Red de Área Metropolitana Inalámbrica (Wireless Metropolitan Area Network).
4. Red de Área Personal Inalámbrica (Wireless Personal Area Network).

	<i>WLAN (1)</i>	<i>Puente LAN-LAN</i>	<i>WWAN (2)</i>	<i>WMAN (3)</i>	<i>WPAN (4)</i>
Área de Cobertura	En edificio o campus	Edificio a edificio	Nacional	Área Metropolitana	Unos pocos metros
Función	Extensión o alternativa a una LAN cableada	Alternativa a una conexión cableada	Extensión de una LAN	Extensión de una LAN cableada	Alternativa al cable
Cuota de Uso	No	No	Si	Si	No
Throughput Típico	1-11 Mbps	2-100 Mbps	1-32 Kbps	10-100 Kbps	0, 1-4 Mbps

Figura 1.2: Diferencias entre WLAN y otras tecnologías inalámbricas

1.6. Aplicaciones

Las aplicaciones de LAN inalámbricas ofrecen a los usuarios de computadores la confiabilidad y conectividad de alto rendimiento de LANs cableadas con la adicional flexibilidad, movilidad y accesibilidad financiera de la inalámbrica. La conectividad de LAN inalámbrica es definida como acceso de clientes a recursos de red como e-mail, la Internet, servidores de archivos, impresoras y aplicaciones de base de datos sin necesitar la conexión física a una red de área local.

Redes LAN inalámbricas son ideales para ambientes de colegio y negocios:

- (1) Donde tirar cableado de red es costoso y difícil, no es imposible, como en:
 - grandes espacios abiertos como auditorios y salón de conferencias.
 - ambientes de campus con múltiples edificios.
 - espacios multi-uso que requieren configuración flexible.
 - edificios donde materiales como ladrillos, concreto o asbestos son prevalecientes.

Con las redes inalámbricas no hay necesidad de tirar cables extra, expandir el cableado a través de un espacio grande, atravesar paredes o cavar trincheras al siguiente edificio.

- (2) Donde temporalmente las redes LAN necesitan ser rápidamente establecidas, configuradas y potencialmente movidas, como para:
 - laboratorio de computadores ad hoc en espacios multi-uso sin cablear.
 - oficinas usadas por visitantes o personal sin espacio asignado.

Con tecnología inalámbrica, una conexión LAN o red temporal puede ser establecida en cualquier parte dentro de una oficina, edificio o ambiente de campus.

- (3) Donde usuarios de computadores móviles requieren acceso a recursos de red local establecida, servicios Internet, bibliotecas y centros de datos desde varias ubicaciones. Por ejemplo:

- trabajadores cuyas tareas les requieren gastar tiempo en varias ubicaciones, incluyendo salas no cableadas.
- estudiantes y personal que van desde clase a clase, edificio a edificio y al aire libre.

Con inalámbrica, recursos de red de área local pueden ser disponible a usuarios de computadores móviles para acceso a e-mail, conexión a Internet, bibliotecas de bases de datos y más.

- (4) Redes locales para situaciones de emergencia o congestión de la red cableada.
- (5) Estas redes permiten el acceso a la información mientras el usuario se encuentra en movimiento. Habitualmente esta solución es requerida en hospitales, fábricas, almacenes...
- (6) Generación de grupos de trabajo eventuales y reuniones ad-hoc. En estos casos no valdría la pena instalar una red cableada. Con la solución inalámbrica es viable implementar una red de área local aunque sea para un plazo corto de tiempo.
- (7) En ambientes industriales con severas condiciones ambientales este tipo de redes sirve para interconectar diferentes dispositivos y máquinas.
- (8) Interconexión de redes de área local que se encuentran en lugares físicos distintos. Por ejemplo, se puede utilizar una red de área local inalámbrica para interconectar dos o más redes de área local cableadas situadas en dos edificios distintos.

1.7. Configuraciones WLAN

El grado de complejidad de una red de área local inalámbrica es variable, dependiendo de las necesidades a cubrir y en función de los requerimientos del sistema que queramos implementar podemos utilizar diversas configuraciones de red.

Peer to peer o redes ad-hoc:

La configuración más básica es la llamada *de igual a igual o ad-hoc* y consiste en una red de dos terminales móviles equipados con la correspondiente tarjeta adaptadora para comunicaciones inalámbricas.

Para que la comunicación entre estas dos estaciones sea posible hace falta que se vean mutuamente de manera directa, es decir, que cada una de ellas esté en el rango de cobertura radioeléctrica de la otra. Las redes de tipo *ad-hoc* son muy sencillas de implementar y no requieren ningún tipo de gestión administrativa.

En este caso cada cliente solo tiene acceso a los recursos de otro cliente y no a un servidor central.

Clientes y puntos de acceso:

Para aumentar el alcance de una red del tipo anterior hace falta la instalación de un punto de acceso. Con este nuevo elemento doblamos el alcance de la red inalámbrica (ahora la distancia máxima permitida no es entre estaciones, sino entre cada estación y el punto de acceso). Además, los puntos de acceso se pueden conectar a otras redes, y en particular a una red fija, con lo cual un usuario puede tener acceso desde su terminal móvil a otros recursos.

Para dar cobertura en una zona determinada habrá que instalar varios puntos de acceso de tal manera que podamos cubrir la superficie necesaria con las celdas de cobertura que proporciona cada punto de acceso y ligeramente solapadas para permitir el paso de una celda a otra sin perder la comunicación.

Cada punto de acceso puede contener muchos clientes, el número específico depende del número y naturaleza de las transmisiones involucradas. En muchas aplicaciones del mundo real un simple punto de acceso da servicio entre 15-50 clientes.

Los puntos de acceso tienen un alcance finito, del orden de 160 metros en interiores y de 210 metros en exteriores. En muchas de sus utilidades



Figura 1.3: red ad-hoc

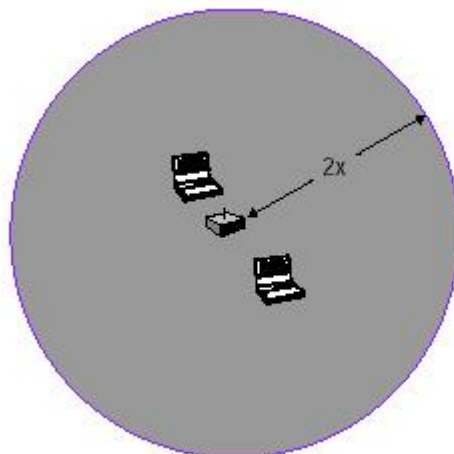
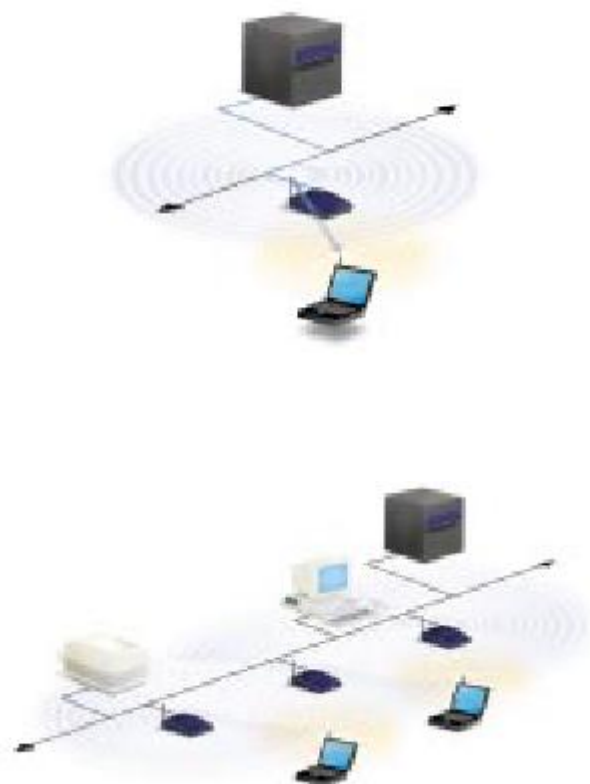


Figura 1.4: punto de acceso



como puede ser en almacenes o en universidades probablemente sean necesarios más de un punto de acceso.

Para posicionar un punto de acceso es necesario hacer un estudio cuya finalidad es cubrir un área de tal manera que el cliente pueda alcanzar totalmente este área sin perder contacto con la red.

La habilidad del cliente de moverse a la perfección entre un conjunto de puntos de acceso sin perder la conexión con la red se llama *roaming*.

Los puntos de acceso dan servicio a los clientes entre ellos cuya comunicación es invisible para el cliente, garantizando una perfecta conexión.

Uso de un punto de extensión:

Para resolver problemas particulares de topología, el diseñador de la red puede elegir usar un Punto de Extensión para aumentar la red de los Puntos de Acceso. Los Puntos de Extensión se ven y funcionan como



Figura 1.5: punto de extensión



Figura 1.6: antenas direccionales

los Puntos de Acceso, pero no están atados a una red cableada como los Puntos de Acceso. Los Puntos de Extensión funcionan justo como su nombre lo dice: ellos extienden el rango de la red transmitiendo señales desde un cliente a un Punto de Acceso o a otro Punto de Extensión.

Los Puntos de Extensión pueden ser atados juntos en orden para pasar el mensaje desde un Punto de Acceso a los clientes más lejanos, así como los humanos en una brigada pasan cubos de agua mano a mano desde una fuente de agua a un incendio.

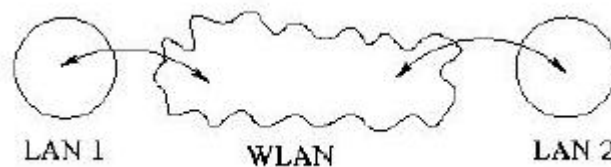
Uso de antenas direccionales:

Otra de las configuraciones de red posibles es la que incluye el uso de antenas direccionales. El objetivo de estas antenas direccionales es el de enlazar redes que se encuentran situadas geográficamente en sitios distintos tal y como se muestra en la siguiente figura:

Un ejemplo de esta configuración lo tenemos en el caso en que tengamos una red local en un edificio y la queramos extender a otro edificio. Una posible solución a este problema consiste en instalar una antena direccional en cada edificio apuntándose mutuamente. A la vez, cada una de estas está conectada a la red local de su edificio mediante un punto de acceso. De esta manera podemos interconectar las dos redes locales.

1.8. Tipos de tecnologías wireless

Los productos de wireless LAN tienen un rango de tecnología que varía dependiendo de las aplicaciones que necesitemos. Cada tecnología contiene su propio conjunto de ventajas y limitaciones.



1.8.1. Tecnología de espectro ensanchado

La tecnología de espectro ensanchado consiste en difundir la señal de información a lo largo del ancho de banda disponible, es decir, en vez de concentrar la energía de las señales alrededor de una portadora concreta lo que se hace es repartirla por toda la banda disponible. Este ancho de banda total se comparte con el resto de usuarios que trabajan en la misma banda frecuencial. Existen dos tipos de tecnologías de espectro ensanchado:

- Espectro Ensanchado por Secuencia Directa (DSSS).
- Espectro Ensanchado por Salto en Frecuencia (FHSS).

Tecnología de espectro ensanchado por secuencia directa (DSSS):

Esta técnica consiste en la generación de un patrón de bits redundante llamado *señal de chip* para cada uno de los bits que componen la señal de información y la posterior modulación de la señal resultante mediante una portadora de RF. En recepción es necesario realizar el proceso inverso para obtener la señal de información original.

La secuencia de bits utilizada para modular cada uno de los bits de información es la llamada secuencia de Barker y tiene la siguiente forma:

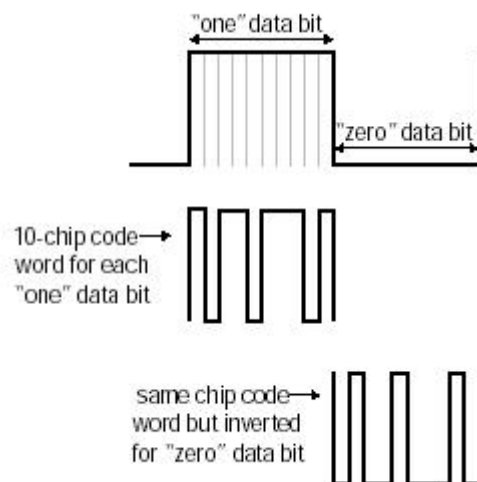


Figura 1.7: aspecto de una señal de dos bits a la cual le hemos aplicado la secuencia de Baker

+1, -1, +1, +1,-1, +1, +1, +1, -1, -1, -1

DSSS tiene definidos dos tipos de modulaciones a aplicar a la señal de información una vez se sobrepone la señal de chip tal y como especifica el estándar IEEE 802.11:

- La modulación DBPSK: Differential Binary Phase Shift Keying
- La modulación DQPSK: Differential Quadrature Phase Shift Keying

proporcionando unas velocidades de transferencia de 1 y 2 Mbps respectivamente.

En el caso de Estados Unidos y de Europa la tecnología de espectro ensanchado por secuencia directa, DSSS, opera en el rango que va desde los 2.4 GHz hasta los 2.4835 GHz, es decir, con un ancho de banda total disponible de 83.5 MHz. Este ancho de banda total se divide en un total de 14 canales con un ancho de banda por canal de 5 MHz de los cuales cada país utiliza un subconjunto de los mismos según las normas reguladoras para cada caso particular. En el caso de España se utilizan los canales 10 y 11 ubicados en una frecuencia central de 2.457 GHz y 2.462 GHz respectivamente.

DSSS PHY FREQUENCY CHANNEL PLAN

Channel	Frequency	FCC	IC	ETSI	Spain	France	MKK
		USA	Canada	Europe	Spain	France	Japan
1	2412 Mhz	x	x	x			x
2	2417 Mhz	x	x	x			x
3	2422 Mhz	x	x	x			x
4	2427 Mhz	x	x	x			x
5	2432 Mhz	x	x	x			x
6	2437 Mhz	x	x	x			x
7	2442 Mhz	x	x	x			x
8	2447 Mhz	x	x	x			x
9	2452 Mhz	x	x	x			x
10	2457 Mhz	x	x	x	x	x	x
11	2462 Mhz	x	x	x	x	x	x
12	2467 Mhz			x		x	x
13	2472 Mhz			x		x	x
14	2484 Mhz						x

En topologías de red que contengan varias celdas, ya sean solapadas o adyacentes, los canales pueden operar simultáneamente sin apreciarse interferencias en el sistema si la separación entre las frecuencias centrales es como mínimo de 30 MHz. Esto significa que de los 83.5 MHz de ancho de banda total disponible podemos obtener un total de 3 canales independientes que pueden operar simultáneamente en una determinada zona geográfica sin que aparezcan interferencias en un canal procedentes de los otros dos canales. Esta independencia entre canales nos permite aumentar la capacidad del sistema de forma lineal con el número de puntos de acceso operando en un canal que no se esté utilizando y hasta un máximo de tres canales. En el caso de España esta extensión de capacidad no es posible debido a que no existe el ancho de banda mínimo requerido (la información sobre la distribución de las frecuencias en distintas regiones del mundo se encuentra disponible en el estándar IEEE 802.11).

Tecnología de espectro ensanchado por salto en frecuencia (FHSS):

La tecnología de espectro ensanchado por salto en frecuencia consiste en transmitir una parte de la información en una determinada frecuencia durante un intervalo de tiempo llamada *dwell time* e inferior a 400 ms. Pasado este tiempo se cambia la frecuencia de emisión y se sigue transmitiendo a otra frecuencia. De esta manera cada tramo de información se va transmitiendo en una frecuencia distinta durante un intervalo muy corto de tiempo. Cada una de las transmisiones a una frecuencia concreta se realiza utilizando una portadora de banda estrecha que va

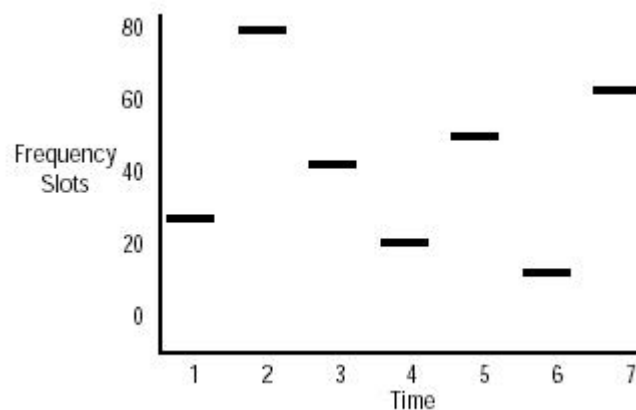


Figura 1.8: Modo de trabajo de la técnica FHSS

cambiando (saltando) a lo largo del tiempo.

Este procedimiento equivale a realizar una partición de la información en el dominio temporal. El orden en los saltos en frecuencia que el emisor debe realizar viene determinado según una secuencia pseudo aleatoria que se encuentra definida en unas tablas que tanto el emisor como el receptor deben conocer. La ventaja de estos sistemas frente a los sistemas DSSS es que con esta tecnología podemos tener más de un punto de acceso en la misma zona geográfica sin que existan interferencias si se cumple que dos comunicaciones distintas no utilizan la misma frecuencia portadora en un mismo instante de tiempo.

Si se mantiene una correcta sincronización de estos saltos entre los dos extremos de la comunicación el efecto global es que aunque vamos cambiando de canal físico con el tiempo se mantiene un único canal lógico a través del cual se desarrolla la comunicación.

Para un usuario externo a la comunicación la recepción de una señal FHSSS equivale a la recepción de ruido impulsivo de corta duración.

El estándar IEEE 802.11 describe esta tecnología mediante la modulación en frecuencia FSK, Frequency Shift Keying, y con una velocidad de transferencia de 1 Mbps ampliable a 2Mbps bajo condiciones de operación óptimas también especificadas en la rma.

1.8.2. Tecnología de infrarrojos

Una tercera tecnología, de momento no demasiado utilizada a nivel comercial para implementar WLANs, es la de infrarrojos. Los sistemas de infrarrojos se sitúan en altas frecuencias, justo por debajo del rango de frecuencias de la luz visible. Las propiedades de los infrarrojos son, por tanto, las mismas que tiene la luz visible. De esta forma los infrarrojos no pueden pasar a través de objetos opacos pero se pueden reflejar en determinadas superficies.

Las longitudes de onda de operación se sitúan alrededor de los 850-950 nm, es decir, a unas frecuencias de emisión que se sitúan entre los $3,15 \cdot 10^{14}$ Hz y los $3,52 \cdot 10^{14}$ Hz.

Los sistemas que funcionan mediante infrarrojos se clasifican según el ángulo de apertura con el que se emite la información en el emisor en:

- Sistemas de corta apertura, de haz dirigido o de visibilidad directa que

funcionan de manera similar a los mandos a distancia de los aparatos de televisión.

Esto supone que el emisor y el receptor tienen que estar orientados adecuadamente antes de empezar a transmitirse información.

- Sistemas de gran apertura, reflejados o de difusión que radian tal y como lo haría una bombilla, permitiendo el intercambio de información en un rango más amplio.

La norma IEEE 802.11 especifica dos modulaciones para esta tecnología: la modulación 16 ppm y la modulación 4 ppm proporcionando unas velocidades de transmisión de 1 y 2 Mbps respectivamente.

Esta tecnología se aplica típicamente en entornos de interior para implementar enlaces punto a punto de corto alcance o redes locales en entornos muy localizados como puede ser una aula concreta o un laboratorio.

1.9. Alternativas a la WLAN

1.9.1. Bluetooth

Bluetooth es un protocolo punto a punto realizado para conectar inalámbricamente teléfonos celulares, portátiles, ordenadores de mano, cámaras digitales e impresoras. Opera en cortas distancias de unos 10 metros, eliminando la necesidad de cables e infrarrojos. Existe una discusión sobre como alcanzar los 100 metros pero para ello debe incrementarse el poder de transmisión a 100 mW.

El radio Bluetooth transmite pequeños paquetes usando FHSS a 1,600 saltos por segundo, la velocidad de los saltos son mucho mayor que en el caso de los radios de HomeRF y otros. Saltar más rápido tiene la ventaja de utilizar menos tiempo sobre el canal de frecuencia que puede sufrir interferencias, las cuales generalmente son vistas como buenas. En otras bandas, es una producción ineficiente en términos de malgastar el tiempo durante los saltos y una carencia de tiempo al abrirse de golpe grandes paquetes.

En consideración a las interferencias en otros sistemas, la velocidad de salto puede ser vista desde dos perspectivas diferentes:

- una en la cual el objetivo es llevar a cabo un sensible retraso del tráfico, por lo tanto concierne sobre la latencia, los rápidos saltos de interferencias son privilegiados porque obtiene un canal muy pronto.
- Otra que principalmente concierne al rendimiento de los paquetes mapeados de una gran ethernet, velocidades de saltos más lentas son mejores porque la probabilidad que el paquete objetivo colisione con las fuentes de interferencias son reducidas.

Irónicamente, el común denominador entre Bluetooth y un horno microondas es la ausencia de un protocolo de escucha antes de hablar. Bluetooth ha sido diseñado para transmitir a ciegas, sin embargo la sincronización dicta como si no hubiera posibilidad de que un sistema combinado pudiera ser usado en la misma frecuencia. Ésto ha dado a la banda de 2.4 GHz el nombre algo turbio como “*mal vecino*”.

1.9.2. HomeRF

En contraste a Bluetooth, HomeRF es un protocolo WLAN con un rango de unos 50 metros y capaz de conectar muchos aparatos simultáneamente. Aunque los dos protocolos son optimizados de manera completamente diferente y diseñados para tipos diferentes de aplicaciones, hay un significado en el hardware RF. Estas permiten ambas tecnologías para cosechar

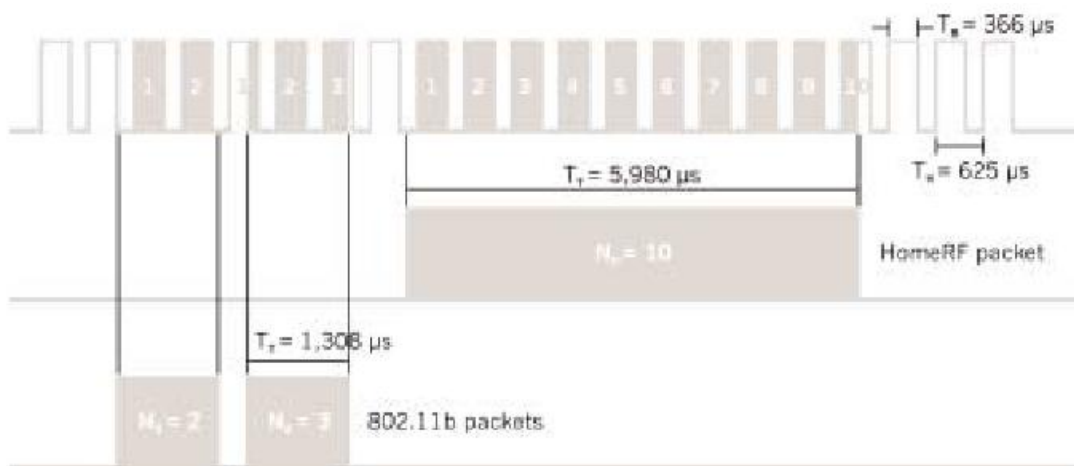


Figura 1.9: Sincronización de paquetes 802.11b, Bluetooth y HomeRF

beneficios en términos de economías de escala, como el compartir los resultados de los componentes de hardware en volúmenes combinados más altos y costo finalmente más bajo.

Mientras el estándar 802.11b era desarrollado originalmente para redes de empresas, HomeRF era ideado desde el principio para reunir las necesidades del consumidor en las aplicaciones de la red casa: asequible, robustez y facilidad de uso. Una importante diferencia de HomeRF frente a la 802.11b es la habilidad de compartir el protocolo de acceso inalámbrico (SWAP) para transportar datos asíncronos, permitiendo la integración real ordenador-telefonía. SWAP es esencialmente una combinación híbrida de de TDMA para servicios asíncronos y acceso CSMA para transmisión asíncrona de datos. Los periodos TDMA soportan hasta 8 conexiones de voz simultanea basadas en el estándar de voz (DECT), es decir, telefonía cableada de mejora digital.

1.9.3. IrDA

Esta técnica de transmisión basada en infrarrojos es la usada en controles remotos.

Desventajas:

- es punto a punto.
- es muy direccional.
- no atraviesa materiales opacos.

1.9.4. Ultra Wide Band (UWB)

Los teléfonos móviles de tercera generación han sido probados en Japón. Pero los operarios móviles que llevan gastados billones de libras adquiriendo licencias 3G ahora encaran la perspectiva de una mejor, tecnología más barata que apenas cuida sobre que banda de espectro es usada.

La banda ultra ancha es un tipo nuevo de comunicaciones de ondas de radio que puede transmitir datos sobre una corta distancias con más flexibilidad que otras frecuencias de radio

La banda ultra ancha (UWB) es una tecnología para comunicación de radio, localización de precisión y radar portátil. Ha existido desde hace 12 años, principalmente usado por los militares. Pero en los últimos tres años la Comisión Federal de Comunicaciones en los US han estado deliberando si permitir la utilización de la UWB para ser usadas en aparatos comerciales. Al darse la luz verde se ha creado un interés entre empresas de móviles y compañías inalámbricas. ¿Por qué? es muy simple, es un método barato para distribuir inalámbricamente gran cantidad de datos en una banda ancha de hasta un kilómetro de rango. Un aparato UWB trabaja emitiendo cortas series de pulsos eléctricos de baja potencia (billonésimas de segundo) los cuales no son dirigidos en una frecuencia particular en el espectro de radio pero a través del espectro entero, a través de todas las frecuencias a la vez. Es como girar sobre una bombilla brillante sin la pantalla de dicha bombilla.

Como una herramienta de comunicaciones, la banda UWB puede ser usada para construir redes inalámbricas de una gran banda ancha que soporte muchos más servidores que otros protocolos inalámbricos, tal como Bluetooth y 802.11b. UWB también puede ser usada en áreas cargadas con demasiados obstáculos por otros protocolos inalámbricos. Y porque las señales UWB son tan espaciadamente amplias que son muy resistente a obstrucciones y escuchas exteriores, conduciendo a comunicaciones inalámbricas más seguras.

Normalmente éstos obstruirían otros aparatos de radio, teléfonos móviles y radios. Sin embargo, UWB emite sus pulsos a una determinada velocidad que sólo es cogida por un destinatario que está preparado para obtener una sucesión exacta de pulsos. Por lo tanto, el destinatario tiene que saber exactamente cuando escuchar, a la orden 'oír', las transmisiones de datos. Ésto hace a la UWB muy inmune y segura a obstrucciones exteriores que el invisible código morse.

¿Por qué es enviado en todas las frecuencias?, desde las más altas hasta las más bajas, UWB puede pasar en línea recta objetos como el mar o capas de roca, y ser usado en aplicaciones de radar. Los pulsos pueden ser colocados en algún intervalo fortuito, significando el número de aparatos que pueden ser empotrados virtualmente sin límites.

¿Por qué esta técnica usa una potencia muy pequeña?, las comunicaciones UWB pueden estar por debajo del ruido de espectros usados por otros, como aparatos que usan potencias mayores, tal como teléfonos celulares y sistemas globales. Las señales de la UWB no interferirán con las señales de los otros aparatos.

UWB puede transmitir 60 megabits por segundo que es seis veces más rápido que las redes de negocio Wi-Fi. También necesita menos potencia que otros aparatos de radio convencionales.

Los pulsos de las señales UWB también puede ser usados para hacer medidas extremadamente precisas de objetos remotos, pavimentando el camino para un radar más exacto y menos costoso, imágenes y otros aparatos de detección de objeto

Así IBM, Motorola y Texas se han asociado para hacer astillas a la UWB o han investigado a la UWB desde sus principios. JVC, Panasonic, Sony, Intel y Afilado están implicados en unos estándares que les podría tomar hasta el 2004.

Sin embargo, los productos UWB estarán en el mercado entre los próximos 18 meses a dos años. Inicialmente ellos estarán probablemente en aparatos de radio para quitar los cables entre TV y video. Pero no hay nada que diga que no pueden ser usados en teléfonos móviles.

Un teléfono móvil empleando UWB podría trabajar durante semanas con una carga simple y podría tener pequeños problemas asociados a la radiación. Él podría competir con Bluetooth, aunque la tecnología está en el principio de su desarrollo.

Con la aprobación de la FCC, solo es cuestión de tiempo que los gobiernos Europeos consideren dar la prioridad a la UWB. Así pues no podemos decir que lo hemos visto todo en el mundo de la radio.

1.10. El estándar 802.11

Este estándar define el protocolo para dos tipos de redes: ad-hoc y cliente-servidor.

Como ya se comentó anteriormente una red ad-hoc es una red sencilla donde la comunicación se establece entre múltiples estaciones en un área definida, sin la necesidad de usar un punto de acceso o servidor. Especifica la "etiqueta" que cada máquina debe observar para que todas tengan un acceso equivalente a la red. Proporciona métodos para arbitrar las solicitudes de acceso al medio, para asegurar que la cantidad de datos vertida al medio sea máxima para todos los usuarios.

El cliente-servidor usa un punto de acceso que controla el tiempo de transmisión para todas las estaciones, y permite que estaciones móviles puedan acceder a la red en distintos puntos. El punto de acceso controla el tráfico desde las máquinas móviles a la red cableada de la red cliente-servidor. Esto permite la coordinación de todas las estaciones y el manejo óptimo del tráfico de datos.

En 802.11b, los dispositivos se comunican a 11Mbps siempre que sea posible. Si la intensidad de la señal o las interferencias están estropeando los datos, los dispositivos cambiarán su velocidad a 5.5Mbps, a 2Mbps y finalmente a 1Mbps. Aunque esto puede significar una reducción de velocidad permite que la red siga funcionando sin errores.

Ventajas de 802.11b:

- Es rápido
- Sin errores
- Tiene un largo alcance (de 76 a 122m en áreas cerradas y unos 300m en áreas abiertas)

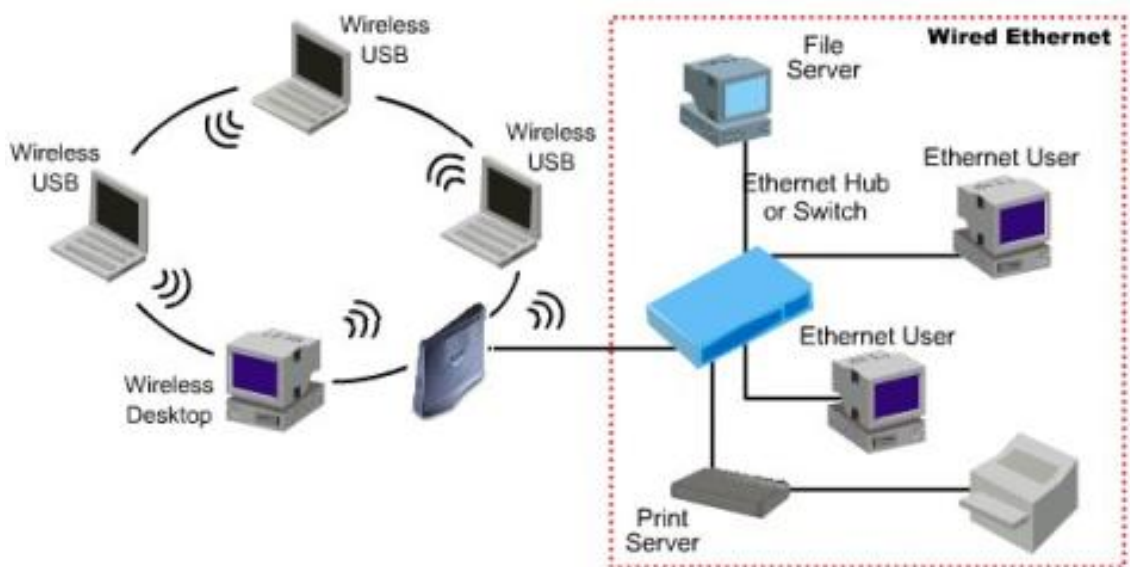


Figura 1.10: Red inalámbrica

- Es sencillo de integrar en redes ya existentes.
- Es compatible con los dispositivos 802.11 DSSS originales.

Desventajas:

- Es caro
- Requiere un punto de acceso
- Puede ser complicado de configurar
- La velocidad puede fluctuar significativamente, dependiendo de interferencias electromagnéticas en la zona y de obstáculos, como por ejemplo personas que circulen entre emisor y receptor.

1.11. Alta velocidad WLAN: 802.11a y 802.11g

La industria de WLAN ha emergido como una de los sectores con mayor crecimiento en la industria de las comunicaciones. Los envíos de los equipos de WLAN crecieron \$ 12 millones de unidades en el 2001. se espera que las tarjetas de red WLAN y las estaciones base WLAN crezcan de % 1.9 billones en el 2001 a % 5.2 billones en el 2005. Este crecimiento ha ocurrido en gran parte por la introducción de los productos base estándar de la WLAN. Estos productos, basados en el estándar 802.11b, son más rápidos, menos costosos y más simples de instalar que la generación de productos anteriores. La mayoría de los productos WLAN de hoy se comunican a 11 Mbps.

Dos nuevos estándares están ahora emergiendo y desarrollan una alta velocidad, 54Mbps. Estos nuevos estándares son conocidos como: 802.11a y 802.11g.

1.11.1. El camino a la alta velocidad

El primer estándar WLAN, 802.11, fue aprobado por el Instituto de ingeniería eléctrica y electrónica (IEEE) en 1997 y soportaba 2 Mbps.

En 1999, la IEEE aprobó los estándares 802.11a y 802.11b. El 802.11a especifica un radio de transmisión de 5 GHz y una velocidad de 54 Mbps

usando una tecnología de modulación de división multiplexada de frecuencia ortogonal (OFDM). El estándar 802.11b, conocido popularmente como WI-FI, especifica una banda de operación de 2.4 GHz, conocida como banda ISM, y una velocidad de 11 Mbps usando una tecnología (DSSS).

Debido a que la tecnología DSSS es más fácil de implementar que la OFDM, los productos 802.11b aparecieron primero en el mercado, comenzando en 1999. Desde entonces, los productos 802.11b han sido rápidamente desarrollados por las corporaciones, pequeñas oficinas, en casa residenciales y en lugares públicos (WI-FI "hotspots"). Los productos con el logo de WI-fi han pasado un test de certificación de interoperabilidad por la alianza de compatibilidad de wireless ethernet (WECA).

Ahora en el 2002, el primer producto para usuarios finales basados en el estándar 802.11a será vendido. Actualmente, estos productos están todos basados en el mismo chipset de un solo vendedor. WECA ha anunciado que es un producto que define el test de certificado de interoperabilidad, el cual será usado tan pronto como estén disponibles los productos basados en el chipset de un segundo vendedor. Los productos que pasen este test serán conocidos como productos WI-FI5 compatibles.

En el 2001, la FCC anunció nuevas reglas permitiendo modulaciones adicionales en el rango 2.4 GHz. Ésto permitió a la IEEE extender la 802.11b para soportar velocidades mayores de datos, resultando el estándar 802.11g, el cual es el anteproyecto y se espera ser completado y aprobado ahora a finales del 2002.

El 802.11g define nuevas velocidades de datos a 54 Mbps, a 2.4 GHz usando OFDM, mientras que al mismo provee compatibilidad con el 802.11b a velocidades de 11 Mbps usando DSSS.

1.11.2. 802.11a: alta velocidad, alta capacidad

Para moverse por la banda de frecuencia de 5 Ghz y usar la modulación OFDM, el estándar 802.11a provee dos principales beneficios sobre 802.11b. Incrementa la velocidad máxima por canal (de 11Mbps a 54 Mbps) e incrementa el número de canales no solapables. La banda de 5 GHz (también conocido como la banda UNII) es actualmente fabricado en tres subbandas, UNII1 (5.15-5.25 GHz), UNII2 (5.25-5.35 GHz) y UNII3 (5.725-5.825 GHz). El aumento sobre los 8 canales no solapables están disponibles cuando la UNII1 y UNII2 se están usando a la vez, frente a los 3 necesarios en la banda

	802.11	802.11a	802.11b	802.11g
Standard Approved	July 1997	September 1999	September 1999	Draft stage. Completion expected in 2002.
Available Bandwidth	83.5 MHz	300 MHz	83.5 MHz	83.5 MHz
Unlicensed Frequencies of Operation	2.4-2.4835 GHz DSSS, FHSS	5.15-5.35 GHz OFDM 5.725-5.825GHz OFDM	2.4-2.4835GHz DSSS	2.4-2.4835GHz DSSS, OFDM
Number of Non-Overlapping Channels	3 (Indoor/Outdoor)	4 Indoor (UNII1) 4 Indoor/Outdoor (UNII2) 4 Outdoor (UNII3)	3 (Indoor/Outdoor)	3 (Indoor/Outdoor)
Data Rate per Channel	2, 1 Mbps	54, 48, 36, 24, 18, 12, 9, 6 Mbps	11, 5.5, 2, 1 Mbps	54, 36, 33, 24, 22, 12, 11, 9, 6, 5.5, 2, 1 Mbps
Modulation Type	DQPSK (2 Mbps DSSS) DBPSK (1 Mbps DSSS) 4GFSK (2Mbps FHSS) 2GFSK (1Mbps FHSS)	BPSK (6, 9 Mbps) QPSK (12, 18 Mbps) 16-QAM (24, 36 Mbps) 64-QAM (48, 54 Mbps)	DQPSK/CCK (11, 5.5 Mbps) DQPSK (2 Mbps) DBPSK (1 Mbps)	OFDM/CCK (6,9, 12,18,24,36,48,54) OFDM (6,9,12,18, 24,36,48,54) DQPSK/CCK (22, 33, 11, 5.5 Mbps) DQPSK (2 Mbps) DBPSK (1 Mbps)
Compatibility	802.11	Wi-Fi5	Wi-Fi	Wi-Fi at 11Mbps and below

Figura 1.11: estándares IEEE WLAN

de 2.4 GHz. El total de ancho de banda en la banda de 5 GHz también mayor que en la banda de 2.4 GHz, 83.5 MHz frente a 300 MHz. De este modo, una WLAN basada en 802.11a soporta un gran número de usuarios simultáneos a alta velocidad sin conflictos potenciales.

Estos beneficios, sin embargo, vienen con algunas compensaciones en términos de compatibilidad y rango. Porque ellos operan en diferentes bandas de frecuencia, los productos 802.11a y 802.11b no son compatibles. Un punto de acceso 802.11b a 2.4 GHz, por ejemplo, no trabaja con una tarjeta de red de 5GHz de la 802.11a. Sin embargo, ambos estándares pueden ciertamente coexistir. Por ejemplo, un usuario de 802.11a y un usuario de 802.11b, usando separados puntos de acceso y clientes, conectados a la misma LAN, pueden operar en el mismo espacio físico y compartir los recursos de la red incluyendo el ancho de banda y el acceso a internet.

La alta frecuencia de operación de la 802.11a equivale relativamente a un corto rango. Tú necesitarás un gran número de puntos de acceso de 802.11a para cubrir el mismo área. El FCC requiere un máximo de 6dB en la antena cuando la UNII1 y UNII2 son usadas para obtener 8 canales interiores, reduciendo el rango. Sin embargo, los test iniciales muestran que los productos 802.11a todavía mantienen más o menos 3 a 1 frente a los típicos rangos de la 802.11 interiores.

1.11.3. 802.11g: alta velocidad en la banda de 2.4 GHz

El estándar 802.11g trae los beneficios de la alta velocidad, mientras mantiene la compatibilidad con el equipamiento de la 802.11b.

La 802.11g especifica operaciones en la misma banda de frecuencia y con el mismo tipo de modulación DSSS como 802.11b a velocidades superiores a 11Mbps, mientras añade más eficiencia a los tipos de modulación OFDM a alta velocidad.

Una tarjeta de red de 802.11g, por ejemplo, trabajará con unos puntos de acceso 802.11b y 802.11g y unas tarjetas de red 802.11b a la velocidad superior de 11 Mbps. Para beneficiarse de la velocidad de 54 Mbps, ambos puntos de acceso y la tarjeta de red deben ser amoldados a la 802.11g. El estándar redactado también especifica tipos opcionales de modulación (OFDM/CCK) que están intencionados para proveer eficiencia en todas las instalaciones 802.11g.

La 802.11g es una capacidad menor, en contra de la 802.11a, servir a un gran número de usuarios WLAN a alta velocidad. La modulación OFDM permite la alta velocidad, pero el ancho de banda total disponible en la banda de frecuencia 2.4 GHz continua siendo el mismo porque el 802.11g está todavía restringido a 3 canales en la banda de 2.4 GHz, distinto de los 8 canales que son habilitados en la banda de 5 GHz.

1.11.4. Cambiando el estándar a alta velocidad

Cada utilización de la WLAN es única. Es imposible proveer una simple respuesta al cambio entre la 802.11a o la 802.11g. En algunos casos, puede incluso hacerse mezclas, especialmente en la industria que introduce soluciones en modo dual (dos radios) que soportan los tres estándares. En los siguientes años, la industria realizará aparatos que soporten 11a, 11b y 11g.

Lo más importante es evaluar tus necesidades anticipadamente y tú instalación actual. Algunos criterios principales para cambiar al estándar de alta velocidad son:

Requerimientos de total capacidad:

para instalaciones que popularmente se llaman de alta densidad para usuarios de alta velocidad, el 802.11a puede ser mejor elección. Si un pequeño número de usuarios están siendo añadidos a una instalación existente de 802.11b, el 802.11g puede ser mejor elección.

Necesidad de sincronismo a alta velocidad:

si es necesario inmediatamente la alta velocidad el estándar 802.11a es el sitio a donde ir, los productos 802.11g no son esperados hasta la segunda mitad del 2002.

Planes de migración para instalaciones existentes:

si tienes una gran instalación de 802.11b y simplemente quieres añadir unos pocos usuarios a alta velocidad para el próximo año, puede tener sentido intentar ambas opciones ahora y desplegar productos en modo dual para el próximo año.

Concerniente a interferencias:

si tienes actualmente experiencia con las interferencias en la banda de frecuencia de 2.4 GHz de productos como bluetooth y teléfonos fijos, puede tener sentido cambiar a la banda de 5 GHz con el 802.11a.

1.12. Consideraciones a tener en cuenta

Comparando la LAN cableada, la LAN inalámbrica proporciona flexibilidad en la instalación y en la configuración y además una proporciona una libertad inherente en la red móvil.

Los clientes potenciales de la wireless LAN deben considerar algo o todo de los siguientes temas:

Alcance-Cobertura:

La distancia sobre la cual las ondas RF e IR pueden comunicar, son función del diseño del producto (incluyendo la potencia de transmisión y el diseño del receptor) y del camino de propagación, especialmente en entornos cerrados. Interacciones con típicos objetos de edificaciones incluyendo muros, metales e incluso personas, pueden afectar como se propaga la energía, y de esa manera conseguir un particular sistema de alcance y cobertura.

Objetos sólidos bloquean señales de infrarrojos, lo cual impone limitaciones adicionales. La mayoría de los sistemas usan RF porque las ondas de radio pueden penetrar mejor en los muros y obstáculos. El alcance (o radio de cobertura) para sistemas wireless típicos varía entre un poco menos de 20 metros hasta un poco más de 60 metros.

La cobertura puede ser extendida, y existen unas microceldas que dotan de una verdadera libertad de movilidad por medio de roaming.

Throughput:

Como en la LAN cableada, el rendimiento en las wireless LAN dependen de la relación *producto-organización*. Entre los factores que afectan al rendimiento se incluye el número de usuarios, factores de propagación, el alcance, los múltiples caminos, el tipo de sistema de LAN usado, así como la latencia y los cuellos de botella sobre la parte cableada de la LAN.

La velocidad de los datos en las wireless LAN comerciales están en un rango de 1.6 Mbps. Los usuarios de la tradicional Ethernet o Token Ring LAN generalmente experimentan una pequeña diferencia en el rendimiento cuando usan una wireless LAN. La wireless LAN proporciona un rendimiento suficiente para las aplicaciones de oficina más

comunes basadas en LAN, incluyendo intercambios de correos electrónicos, acceso a parte de los periféricos, acceso a internet y acceso a aplicaciones y base de datos para multiusuarios.

En términos de rendimiento una wireless LAN opera a 1.6 Mpps, para hacernos una idea los módems V.90 transmiten y reciben datos a una velocidad de 56.6 Kbps. Luego la wireless va a unas 30 veces más rápido.

Integridad y confiabilidad:

La tecnología de datos inalámbricos ha sido probada a través de mas de 50 años de aplicaciones inalámbricas en ambos sistemas, comercial y militar. Mientras las interferencias de radio pueden causar degradación en el throughput, tales interferencias son raras en el lugar de trabajo. Diseños Robustos de tecnología LAN inalámbrica probada y la distancia limitada sobre las cuales estas señales viajan resultan en conexiones que son lejos más robustas que las conexiones de teléfonos celulares y proveen desempeño de integridad de datos igual o mejor que las redes cableadas.

Compatibilidad con redes existentes:

La mayoría de las wireless LAN proporcionan una interconexión estándar para la industria con las redes cableadas tal como Ethernet o Token Ring. Los nodos de las wireless LAN están soportados por la capacidad de sistemas de red de la misma manera que otros nodos LAN's: usando apropiados drivers. Una vez instalado la red trata al nodo como a cualquier otro componente de la red.

Interoperatividad y escalabilidad:

DSSS 802.11 es el estándar de la industria soportado por Farallon junto con las principales compañías como Apple, 3Com, Lucent, Nokia, Nortel y otras. El estándar inalámbrico 802.11 asegura compatibilidad para atrás. Esto significa que las soluciones DSSS 802.11 de diferentes tasas de transmisión trabajarán conjuntamente. En consecuencia, cuando la emergente solución de 11Mb de DSSS 802.11b se haga disponible, ella actuará conjuntamente con las soluciones de envío actuales de 2Mb de DSSS 802.11.

Para una fácil integración con una red Ethernet cableada, DSSS 802.11 soporta todos los estándares de red Ethernet incluyendo protocolo TCP/IP, AppleTalk, NetBEUI y IPX. Adicionalmente, los modos de traducción y encapsulación permiten la conversión de paquetes 802.3

desde una red Ethernet cableada a paquetes 802.11 para transmisión inalámbrica y viceversa.

Los clientes deben estar enterados que los sistemas de WLANs de diferentes proveedores pueden no ser interoperables. Por tres razones. Primero, diferentes tecnologías no interoperarán. Un sistema basado en tecnología FHSS no se comunicará con otro basado en tecnología DSSS. Segundo, sistemas usando diferentes bandas de frecuencia no interoperarán hasta si ambos emplean la misma tecnología. Tercero, sistemas de diferentes proveedores pueden no interoperar hasta si ambos emplean la misma tecnología y la misma banda de frecuencia, debido a diferencias en la implementación por cada proveedor.

Interferencias y coexistencia:

La naturaleza sin licencia de las WLANs basadas en radio significa que otros productos que transmiten energía en el mismo espectro de frecuencia puede potencialmente proveer alguna medida de interferencia a los sistemas de WLANs. Los Hornos Microondas son una preocupación potencial, pero la mayoría de los fabricantes de WLANs diseñan sus productos considerando la interferencia del microondas. Otra preocupación es la ubicación de múltiples WLANs. Mientras WLANs de algunos fabricantes interfieran con WLANs, otros coexisten sin interferencias. Este tema es mejor dirigirlo directamente con el proveedor apropiado.

Temas de licencia:

En los Estados Unidos, la Comisión Federal de Comunicaciones (FCC) gobierna las transmisiones de radio, incluyendo aquellas empleadas en las WLANs. Otras naciones tienen su correspondiente agencia regulatoria. En el caso de Chile, la entidad regulatoria está dada por la Subsecretaría de Telecomunicaciones. Las WLANs son típicamente diseñadas para operar en porciones del espectro de radio donde la FCC no requiere que el usuario final compre licencia para usar las ondas. En Estados Unidos la mayoría de las WLANs transmiten sobre una de las bandas de ISM (Instrumentation, Scientific, and Medical). Estas incluyen 902-928 MHz, 2.4-2.483 GHz, 5.15-5.35 GHz, y 5.725-5.875 GHz. Para que las WLANs sean vendidas en un país en particular, el fabricante de la WLAN debe garantizar su certificación por la agencia apropiada en aquel país.

Simplicidad y facilidad de uso:

Los usuarios necesitan poca nueva información para conocer las ventajas de la wireless LAN. Porque la naturaleza inalámbrica de una wireless LAN es transparente a un usuario NOS, las aplicaciones trabajan del mismo que en una red cableada. Los productos wireless LAN incorporan una variedad de herramientas de diagnóstico para direccionar los temas relacionados con los elementos inalámbricos del sistema; sin embargo, los productos son diseñados tal que muchos usuarios raramente necesitan dichas herramientas. Wireless LAN simplifica muchas de sus aplicaciones y configuraciones. Sólo los puntos de acceso de la WLAN requieren cables, los administradores de la red son libres de poner cables a los usuarios finales. Al carecer de cables puedes mover, añadir y cambiar operaciones triviales sobre la WLAN.

Finalmente, la naturaleza portable de la WLAN permite a los administradores preconfigurar y resolver los problemas de toda la red antes de instalarla. Una vez configurada, la WLAN puede ser movida de un lugar a otro con una pequeña o ninguna modificación.

1.13. Mejorando la productividad y la calidad de vida

Los profesionales de tecnologías de información en las grandes organizaciones tienen ahora que hacer más elecciones de tecnología como nunca lo habían hecho anteriormente. ¿Qué sistema operativo tengo que usar?. ¿Es voz sobre IP una opción que yo pueda querer explorar?. ¿El outsourcing a un ASP será beneficioso para mi organización?

Cada vez más, los profesionales de tecnologías de información están también debatiendo si desplegar o no una WLANs. ¿Hay ciertas funciones de trabajo o aplicaciones que justifican la inversión?. ¿Qué ganará mi organización desplegando una WLAN?. ¿Cuáles son los riesgos?. ¿Cuáles son los beneficios?. ¿Tendrán más peso los beneficios que los costos?. ¿Adoptarán los usuarios finales la nueva tecnología?

Mientras que los ahorros de costos son difíciles de cuantificar, los beneficios "suaves" de usar WLANs como la productividad, flexibilidad y exactitud son generalizados y convincentes. Estos beneficios suaves los cuáles llevan in-

directamente a los actuales ahorros de costos son los chóferes principales detrás del creciente número de organizaciones que están desplegando WLANs.

Para proporcionar casos específicos de uso de WLANs, Cisco Systems comisionaron a Sage Research para conducir discusiones cualitativas con empleados de grandes organizaciones que usan WLANs. Este esfuerzo de la investigación destapó actitudes relevantes de clientes que revelan una variedad de beneficios del mundo real así como algunos desafíos para desplegar WLANs. Los datos en este reporte son basados en discusiones cualitativas con 20 participantes. Para calificar, los participantes requerían ser empleados de organizaciones con más de 1000 empleados que han estado usando WLANs por lo menos hace un mes. Los participantes debían estar relacionados con la especificación de presupuestos o la aprobación de compra de equipo de WLAN y/o tener acceso a la WLAN de la organización por lo menos una vez por semana. La muestra final incluye a participantes de varias industrias.

Cuando uno piensa en usuarios de WLAN, los primeros que se nos vienen a la cabeza son la industria de la salud y la educación. Si bien, es verdad que estas dos industrias son más liberales para adoptar la tecnología, el uso de WLAN se ha expandido también a otras industrias. De echo minoristas, gobierno, manufactura, y hasta organizaciones financieras están desplegando WLANs a una tasa creciente.

Dentro de organizaciones que usan WLANs, un promedio de 15 % de los empleados tienen acceso a esta tecnología - aunque los resultados varían ampliamente desde un mínimo de 1 % hasta un máximo de 80 % y la mayoría de las organizaciones tienen planes de expandir su despliegue de WLAN. Mientras la mayoría de los usuarios de WLAN se clasifican bajo la áreas funcionales de las tecnologías de información, dirección ejecutiva, y ventas, también vale la pena entender las funciones específicas que merecen el uso de WLANs. Algunas de estas funciones de trabajo incluyen viaje frecuente, trabajar desde el hogar sobre una base regular (trabajo a distancia), asistiendo a reuniones frecuentes, asistiendo a conferencias lejos de la oficina, trabajando al aire libre donde se han hecho disponibles Puntos de Acceso de WLAN a lo largo de la periferia del edificio, trabajando en un almacén grande o fábrica, tomando inventario de suministros, y acceso instantáneo a información basada en servidor.

1.13.1. Beneficios de la productividad

Ahorros de tiempo:

En el mundo de alta-presión de hoy, nadie puede discutir que el tiempo es dinero. La mayoría de los beneficios cuantificables son en términos de ahorros de tiempo. Basado en la primera investigación, en promedio, un usuario de WLAN puede ahorrar hasta ocho horas por semana en relación a un usuario de LAN cableada. En promedio, un usuario de WLAN ahorra a su organización \$260.50 por semana debido a ahorros de tiempo. Los datos fueron desde \$30 por semana por usuario a \$750 por semana por usuario. Al examinar estos datos considere el bajo número de entrevistas realizadas. La siguiente tabla muestra algunos ejemplos de organizaciones que están ahorrando tiempo por el uso de WLANs.

WLANs les permite a los usuarios que ahorren la mayoría de tiempo cuando responden los e-mail. Esto es especialmente verdadero para profesionales de TI. De echo, un empleado dentro del departamento de TI puede recibir más de 100 e-mails diarios. Accediendo a la WLAN mientras está en reuniones, en la cafetería, o en otras ubicaciones, los empleados están frecuentemente poniéndose al día en sus e-mails mientras hacen otras cosas.

La WLANs no solamente hacen ahorrar tiempo a su usuario, si no que a otros empleados conectados por vía fija también ahorran tiempo indirectamente. Por ejemplo, un empleado con un portátil no necesita llamar a otro colega para obtener información. En cambio, el usuario móvil puede acceder a la LAN por vía inalámbrica y obtener la información por sí mismo.

Otro ejemplo es una Enfermera trabajando en un piso de un hospital puede necesitar acceso a información sobre un paciente. En lugar de llamar a la asistente administrativo o recepcionista, la enfermera

Tipo de Organización	Tipos de usuarios de WLANs	Promedio Tiempo Ahorrado	Promedio dinero ahorrado
Gobierno	Ejecutivos y TI	5 horas/semana/usuario	\$150 /semana/usuario
Salud	Enfermeras	15 horas/semana/usuario	\$472 /semana/usuario
Finanzas	Ejecutivos y TI	15 horas/semana/usuario	\$750 /semana/usuario

Figura 1.12: Tiempo ahorrado por el uso de WLAN's

puede acceder al archivo de pacientes en la LAN utilizando un dispositivo inalámbrico. Por el uso de una red LAN inalámbrica, en este escenario se optimiza tanto la calidad del trabajo como la flexibilidad. A menudo, este beneficio tal como la calidad del trabajo como la flexibilidad obtenida que no se puede cuantificar lo que lleva a adoptar la WLAN.

Dicho por los usuarios:

Ahora puedo usar mi portátil para trabajar durante las reuniones. Me puedo comunicar y no tengo que esperar seis horas para utilizar mis anotaciones. En la sala de reuniones no tengo a donde ir, así que mi productividad aumenta. VP MIS , Finanzas.

El beneficio es que cuando necesitas revisar algo, lo puedes hacer ahí mismo en el piso de ventas. No importa lo que estás mirando, lo puedes revisar ahí mismo. En relación con la eficiencia, es más eficiente. Puedes revisar una mayor cantidad de cosas de una sola vez con eso. La mercadería puede estar ahí mismo. Administrador de Bodega, Minorista.

Yo podía decir que un usuario promedio ahorra por lo menos tres horas al día, en una semana promedio ellos ahorran 15 horas. Esta es una ganancia significativa porque estamos hablando de liberarlos de llamar al laboratorio por los resultados, mirando los resultados de Radiología. Ellos tenían que llamar al Departamento lo que está liberando gente en ambos lados porque utilizábamos a las secretarias para responder las llamadas y manejar los archivos. Gerente de TI, Hospital.

Flexibilidad:

Uno de los beneficios de instalar WLANs es la flexibilidad que ofrece a los usuarios. Ellos son capaces de acceder a la LAN desde una variedad de áreas tradicionales dentro de sus edificios, tanto como desde las áreas no tradicionales y a veces hasta desde el exterior de los edificios.

La tabla siguiente muestra las distintas localizaciones, ordenadas desde las más comunes a las menos comunes desde donde los usuarios acceden a LAN de su organización en forma inalámbrica.

Esta amplia cobertura da al usuario la flexibilidad de trabajar desde su escritorio o moverse a través de su organización, todo el tiempo conectado totalmente a la red. Debe tenerse en cuenta que la implementación de WLANs varía de una organización a otra. No todas las organizaciones suministran tal amplia variedad de accesos inalámbricos. Sin embargo, acceso desde ubicaciones menos convencionales no

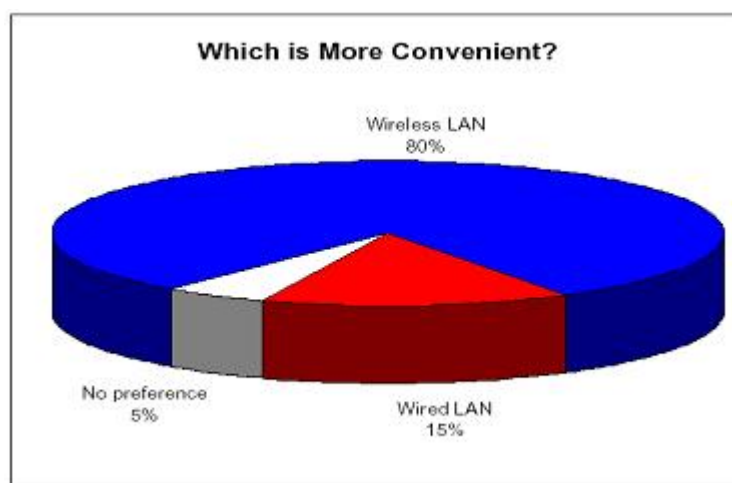
son poco comunes y, de acuerdo a los usuarios, la productividad aumenta con la habilidad de acceso desde múltiples ubicaciones. Como lo demuestra la siguiente cita, los usuarios encuentran que WLANs les ofrece la libertad que las redes fijas no son capaces de suministrar. Una vez más, aunque esta libertad no es algo que se puede cuantificar en términos del dinero ahorrado, el aumento de flexibilidad se traduce en aumento de productividad para los usuarios de la WLAN.

Dicho por los usuarios:

Es bueno no tener que preocuparse por tropezar encima de cables. Gasto una cantidad de tiempo enorme de mi tiempo en mi escritorio, y acoplado a un cable, eso está bien, pero encuentro mucho goce en la libertad que me da ser capaz de desacoplar mi notebook y salir y hablar

Acceso a WLAN desde el interior de Edificios	Acceso a WLAN desde el exterior de Edificios
Sala de Conferencias	Casa
Oficinas	Otro lugar de trabajo
Sala de Reuniones	Bodega
Embarque y área de recepción	Tren
Área de Inventario	Bus
Centro de Distribución	Biblioteca
Cafetería	Aeropuerto
Sala de Ejercicios	Campo Atlético
Sala de Clase	Restaurant/ Café

Figura 1.13: Acceso inalámbrico desde distintas ubicaciones



con un vendedor en el piso u mostrarle notas que puedo haber tomado cuando estaba con un cliente o mostrarle una presentación que pueda haber obtenido de un proveedor. Gerente Regional de Ventas, Manufactura.

Tengo la habilidad para tomar mi sistema al escritorio de otro ingeniero y mostrarle un error que estoy teniendo en un aplicación que estamos corriendo para así poder hacer pruebas. No tengo que traerlo hacia mí. Puedo tomar el mismo sistema donde yo quiera, a cualquier instalación. Si tiene capacidad inalámbrica, lo tengo en todo momento conmigo. No estoy limitado para ir a cualquier escritorio. No estoy limitado para ir a cualquier ubicación. Puedo ir donde quiera, con tal de que tengamos cobertura inalámbrica. Me puedo sentar donde quiera y trabajar donde quiera. Ingeniero de Integración, Finanzas.

Calidad de trabajo:

En adición a la Flexibilidad y al ahorro de tiempo, muchos usuarios reportan que la calidad de sus trabajos ha mejorado por el uso de WLANs. El mayor beneficio en este frente es la Exactitud. Por ejemplo, un empleado del gobierno es responsable por mantener el rastro de desechos nucleares. Anteriormente, un científico necesitaría contar los tambores a mano, y registrar el número en papel. Luego, los datos serían entrados en una aplicación basada en un servidor por un asistente administrativo en una fecha posterior. Esto producía un increíblemente gran margen de error. Ahora, con acceso a una WLAN, el científico puede usar un scanner portátil para leer código de barras que son puestos directamente en los tambores de desecho. Haciendo esto, muchos datos se alimentan en tiempo real directamente al servidor. Esto aumenta la exactitud tremendamente de saber en un 70 % dónde estaban ubicados los desechos, ahora son capaces de saber un 99.9 % la ubicación de los desechos.

El ejemplo anterior es sólo una manera en que la exactitud puede ser mejorada usando WLAN. El punto es que, mediante el uso de WLANs, los datos pueden ahora ser alimentados directamente desde varias ubicaciones en vez entrarlos a mano en una fecha posterior. Esto aumenta la exactitud de la información y puntualidad - otro extendido beneficio "suave" que los usuarios de WLANs están comprendiendo.

Dicho por los usuarios:

El beneficio número uno de usar WLANs es que los usuarios tienen acceso a la información que necesitan desde cualquier parte dentro de

las instalaciones sin tener que correr de un lado para otro. Nosotros no estamos perdiendo información debido a información mal escrita. Ellos lo escribirían y correrían a la máquina a hacer los ajustes y dirían “oh, se me olvidó...”. Ahora, ellos lo tienen con ellos. Gerente SI, Manufactura.

Uno de los chóferes principales en nuestro negocio es la exactitud. La cantidad de datos del cliente que puede ser recogida en el extremo delantero de nuestra venta, procesos administrativos o de servicio a clientes ha sido mejorado grandemente usando una WLAN. Director de Telecomunicaciones, Finanzas

Si la WLAN no estuviera presente obtendría mucho menos logrado. El inventario y los pedidos serían menos precisos. Obtendría más cosas que no necesito y menos de lo que hago. Gerente de Tienda, Minorista

Pienso que una WLAN reduce un montón de errores y malas interpretaciones. Cuando pones algo en papel, las personas pueden siempre leerlo mal. En un sistema computacional, no se aceptan errores. Tienes que entrar los datos correctamente. Gerente de TI, Transporte

1.13.2. Beneficios de calidad de vida

Tensión reducida:

Algunos usuarios de WLANs las ven como reductores de tensión. Este es el caso por una razón principal, información prontamente disponible. Los usuarios de WLANs no tienen que preocuparse tratando de encontrar una conexión cableada. Por ejemplo, si un usuario de WLAN trae la presentación equivocada a una presentación, él o ella no tiene que preocuparse porque la presentación correcta puede accederse inalámbricamente desde el servidor. De esta manera, la tensión es aliviada desde que el usuario no tiene que preocuparse por “enchufarse” para acceder a la LAN.

En suma a la habilidad para bajar archivos, los usuarios pueden responder importantes preguntas vía e-mail rápidamente sin tener que preocuparse por encontrar una conexión. Por ejemplo, un usuario de TI con una conexión inalámbrica puede ir a comer a una cafetería sin tener que preocuparse por perder e-mails urgentes. De nuevo, el punto es que la tensión es reducida en algunas maneras por tener continuo acceso a e-mail, archivos, y aplicaciones.

Dicho por los usuarios:

Las WLANs son un reductor de la tensión, porque la información es mucho más prontamente disponible a ellos (obreros). Con tal de que estén lo bastante motivados a recuperarla, ellos pueden hacer mucho más con ella. Gerente de TI, Hospital.

Diría que una WLAN disminuye la tensión porque ellos no están tan ansiosos... Pueden obtener acceso a enviar e-mail y a responder y no tienen que preocuparse de que no pueden atender tal tema o responder a él por un par de horas... Gerente de Infraestructura de Servicios, Transporte.

Si veo muchas cosas en el estante si el sistema lo contó bien, puedo revisarlo ahí mismo en vez de tener que caminar a la oficina. Diría que mi tensión es menor porque me puedo encargar de las cosas que me tengo que encargar. Gerente de Tienda, Minorista.

Vida familiar mejorada:

Algunos usuarios incluso pueden gastar más tiempo en casa debido a WLANs. Algunos usuarios de WLANs declararon que ellos pueden reducir las horas que ellos gastan en el trabajo debido a lograr más siendo móviles en la oficina. Algunos usuarios pueden ahorrar más de dos horas cada día en la oficina, e irse a casa a una hora más razonable, a las 5 en punto en vez de las 7 en punto.

De hecho, algunos usuarios encuentran que las WLANs les permiten estar más involucrados en las vidas de sus niños. Por ejemplo, algunos dijeron que pueden asistir a los eventos deportivos de los niños o eventos del coro debido a la libertad que tienen con esta tecnología.

Dicho por los usuarios:

Ya no paso el tiempo que pasaba en la oficina porque tengo las herramientas para hacer lo mismo fuera de la oficina. Ahora puedo estar en casa a una hora razonable. Gerente de Infraestructura de Servicios, Transporte.

Aumento de estatus:

Los empleados que usan las WLANs sienten que tienen una ventaja personal sobre usuarios que no tienen acceso a WLANs. Esto muestra que el sentimiento de tener acceso completo a datos en cualquier momento es muy poderoso para los usuarios de WLANs. Este sentimiento de poder, en la mente de los usuarios de WLANs, guía a sentimientos de tener una ventaja.

Un participante en particular sentía un sentimiento de aprecio hacia su organización debido a que proveían tecnología que le permitía estar un paso delante de la competencia. Esto daba al usuario de WLAN un sentimiento de seguridad dentro de su organización.

Dicho por los usuarios:

Tiene la tendencia a impresionar a las personas de algún grado que estamos al filo de la tecnología y estamos tomando al riesgo. Director de TI, Utilidad.

Capítulo 2

Introducción a CORBA

2.1. Historia

La aparición de CORBA está muy ligada a la existencia de sistemas distribuidos. En un principio, con el mainframe, el sistema se controlaba de forma centralizada. Con la introducción de los PC's se pudo descargar gran parte de la carga de trabajo que tenían que soportar los mainframes, apareciendo por lo tanto la arquitectura cliente/servidor. Así se sustituyeron estos "monstruos" por servidores UNIX mucho más económicos. Esto trajo consigo una mayor independencia de las secciones dentro una empresa para crear aplicaciones que se adaptasen mejor a sus necesidades y no depender tanto del control central.

En este tipo de sistema, los datos se hallan almacenados en el servidor, el interfaz de usuario en el cliente y la lógica de control en cualquiera de ellos. Tenía la desventaja de que al cambiar alguna parte del cliente, el nuevo programa debía ser distribuido a todos los usuarios.

Esta arquitectura evolucionó hacia la multicapa (multitier) cliente/servidor. Divide al servidor en varias capas, para poder aislar al cliente de los cambios producidos en la aplicación. De esta forma se podrán hacer cambios en la aplicación con poca probabilidad de que afecten al cliente. Además permite más flexibilidad en la aplicación al poder colocar los módulos del servidor en máquinas diferentes.

A continuación se pueden encuadrar los sistemas distribuidos. En estos sistemas todas las funciones de la aplicación se exponen como objetos, cada uno de los cuales puede usar los servicios proporcionados por otros objetos del mismo o de diferente sistema. Por lo tanto se oscurece la diferencia entre cliente y servidor: los clientes pueden crear objetos que actúen como servidores. Proporcionan una gran flexibilidad. Ésta se basa en que la interfaz (protocolo de comunicación entre dos módulos) de un módulo especifica a los demás los servicios que ofrece y cómo se usan. Mientras la interfaz no varíe, se podrán hacer todos los cambios que se deseen dentro del módulo. Se deben incluir servicios que permitan localizar a los diferentes módulos.

Llegados a este punto es donde tenemos que empezar a hablar de CORBA. En los sistemas distribuidos es muy importante la definición de la interfaz y ciertos servicios como la búsqueda de módulos, ... Lo que hace CORBA es proporcionar un estándar para poder definir estas interfaces entre módulos, así como algunas herramientas para facilitar la implementación de dichas interfaces en el lenguaje de programación escogido. Adicionalmente se

han incluido algunos servicios estándar accesibles a todas las aplicaciones en CORBA. Además CORBA proporciona todo el mecanismo que permite a los distintos módulos de una aplicación comunicarse entre sí.

Las dos principales características de CORBA es que es independiente tanto de la plataforma como del lenguaje de la aplicación. La independencia de plataforma significa que los objetos de CORBA se pueden utilizar en cualquier plataforma que tenga una aplicación CORBA ORB. La independencia de lenguaje se refiere a que los objetos CORBA y los clientes se pueden implementar en cualquier lenguaje de programación. Así, a un objeto CORBA no le hará falta saber el lenguaje en que ha sido escrito otro objeto con el que se esté comunicando.

Para diseñar aplicaciones distribuidas hay más opciones aparte de CORBA, tales como:

Programación con sockets:

De esta forma las aplicaciones se comunican entre sí a través de un canal". Es la forma más directa de comunicar componentes de una aplicación. Se escriben o leen los datos del socket. Al ser una forma de programación de bajo nivel, no se introduce mucha sobrecarga a la aplicación, pero no es muy adecuada para manejar tipos de datos complejos, especialmente cuando los módulos de la aplicación se encuentran en diferentes tipos de máquinas o están implementados en diferentes lenguajes.

Llamada a procedimiento remoto (RPC):

El programador define una función que usa sockets para comunicarse con el servidor remoto que ejecuta la función y devuelve el resultado de nuevo mediante sockets. El RPC es un mecanismo bastante potente como para ser la base de muchas aplicaciones cliente/servidor.

Entorno de computación distribuido de la OSF (DCE):

Se trata de un conjunto de estándares iniciados por el Open Software Foundation (OSF). Nunca ha tenido demasiada aceptación.

Modelo de objetos para componentes distribuidas de Microsoft (DCOM):

Ofrece capacidades similares a las de CORBA. Es un modelo de objetos relativamente robusto. Su defecto es que está disponible casi exclusiva-

mente en el entorno Windows aunque para aplicaciones específicas de Windows es una muy buena solución.

Invocación de métodos remotos de Java (RMI):

Su principal ventaja es que soporta pasar objetos por valor y su principal desventaja es que es una solución solamente para Java.

2.2. El grupo OMG

2.2.1. La misión del OMG

El Grupo de Gestión de Objetos (OMG por las siglas en inglés de *Object Management Group*) se formó con la misión de crear un mercado de programación basada en componentes, impulsando la introducción de objetos de programación estandarizados.

Los estatutos de la organización incluyen el establecimiento de guías para la industria y especificaciones detalladas para la gestión de objetos a fin de suministrar un marco de trabajo común para el desarrollo de un entorno de computación distribuida que abarque las principales arquitecturas de máquina y sistemas operativos.

Su propósito principal es desarrollar una arquitectura única, utilizando la tecnología de objetos, para la integración de aplicaciones distribuidas garantizando la reusabilidad de los componentes, la interoperabilidad y la portabilidad, y basada en componentes de programación disponibles comercialmente.

Mientras que las primeras propiedades de la arquitectura propuesta por el OMG están basadas en los beneficios del uso de la tecnología de objetos por la cual ha apostado, la última corresponde a una decisión estratégica del consorcio orientada a evitar las dificultades que han tenido los productos de otros organismos de normalización como la ISO para encontrar una adecuada respuesta en el mercado. Su objetivo, más allá de generar normas, es el de asegurarse de que estas normas sean utilizadas ampliamente, por lo que ninguna propuesta es adoptada como una norma a menos que describa una tecnología que ya se encuentre en el mercado o asegure una rápida disponibilidad.

La arquitectura del OMG está constituida entonces por componentes (objetos) interoperables con interfaces normalizadas, ofrecidos en el mercado por

diversas empresas, que los usuarios adquieren para construir sus propias aplicaciones.

A pesar de que en su nombre se menciona a los objetos en general, el OMG no pretende cubrir todos los ámbitos de aplicación de la tecnología de objetos, los cuales incluyen los lenguajes de programación, los métodos de análisis y diseño, las interfaces gráficas de usuario, las bases de datos, y los componentes (servicios o aplicaciones encapsulados como “circuitos integrado”). Su interés está centrado en la generación de normas con la especificación (de las interfaces) de los componentes requeridos para el desarrollo de aplicaciones distribuidas, o lo que es lo mismo, para la integración de aplicaciones.

La expresión “gestión de objetos” es definida por el OMG como el desarrollo de programas que modelan el mundo real representándolo con “objetos”. Estos objetos encapsulan atributos, relaciones y métodos de componentes de programa identificables. Un beneficio fundamental de un sistema orientado a objetos es su habilidad para ampliar su funcionalidad mediante la extensión de los componentes existentes y la adición de nuevos objetos al sistema. La gestión de objetos conduce a un desarrollo más rápido de las aplicaciones, un mantenimiento más fácil, una gran escalabilidad, y la reutilización de programas.

Para lograr su propósito de construir una Arquitectura para la Gestión de Objetos (OMA, Object Management Architecture), el OMG ha establecido una base común de partida para todo el trabajo alrededor de la misma, la cual está constituida por los siguientes elementos:

- Un conjunto único de términos y definiciones para los conceptos a manejar en relación con la orientación a objetos: objeto, clases, mensaje, etc.
- Un marco de abstracción común, o modelo de objetos, que define un modelo matemático riguroso para las aplicaciones en términos de objetos y para la compartición de información entre aplicaciones en términos de mensajes.
- Un modelo de referencia común o arquitectura.
- Un conjunto común de interfaces, protocolos y lenguajes.

2.2.2. Organización

El OMG fue fundado en 1989 como una organización sin ánimo de lucro, por once compañías entre las que se encontraban: Hewlett-Packard, 3Com, Canon, Philips, American Airlines, Data General y Unisys. En la actualidad cuenta con más de 800 miembros en las categorías de Contribuyentes, Contribuyentes de Dominio, Influyentes, Gubernamentales, Auditores y Universidades. Su sede principal esta situada en Framingham (Estado Unidos), y posee oficinas en Alemania, Australia, Bahrain, Brasil, India, Italia, Japón y Reino Unido. Es patrocinador de la serie de ferias comerciales y conferencias COMDEX Enterprise. Su personal de planta consiste en un pequeño equipo de trabajo, con alrededor de 35 personas de tiempo completo, encargado fundamentalmente de labores administrativas pues ellos no hacen ningún tipo de desarrollo; la actividad de estandarización es efectuada por delegados de las organizaciones miembros, que participan en distintos grupos de trabajo.

El OMG está estructurado mediante tres tipos de órganos:

- El Consejo de Directores (BOD, *Board Of Directors*), que toma las decisiones finales en la organización.
- Los Comités Técnicos, que lanzan los procedimientos para la adopción de normas y coordinan las actividades de los grupos de trabajo. Son tres:
 - El comité de Tecnología de Plataforma (PTC, *Platform Technology Committee*).
 - El Comité de Tecnología de Dominio (DTC, *Domain Technology Committee*).
 - El Consejo de Arquitectura (AB, *Architecture Board*).
- Los grupos de trabajo, asociados a cada uno de los Comités Técnicos, encargados de temas específicos para los cuales realizan estudios, evalúan propuestas, etc. Los hay de tres tipos:
 - Comisiones (PTF, *Platform Task Force*).
 - Grupos Especiales de Interés (SIG, *Domain Special Interes Group*).
 - Sub-Comités (SC, *Sub-Committee*).

El Comité de Tecnología de Plataforma es el responsable de los niveles inferiores de la arquitectura, que incluyen el núcleo con las funciones de interacción entre objetos y los servicios básicos. En los inicios del OMG era el único Comité Técnico y tuvo a su cargo el proceso de adopción del estándar original de CORBA.

El Comité de Tecnología de Dominio es el responsable de la especificación de los servicios relacionados con los dominios específicos de la aplicación, que influyen la manufactura, las finanzas, el comercio electrónico y la salud. Fue creado a raíz de la intensa actividad que surgían en los niveles superiores de la arquitectura, una vez adquirieron madurez las especificaciones de la "plomería" de los servicios básicos.

El Consejo de Arquitectura es el responsable de vigilar la consistencia y la integridad técnica con la OMA de las especificaciones adoptadas por el PTC y el DTC.

Los grupos de trabajo del PTC son:

- Object Analysis & Design PTF
- Japan PSIG
- ORB and Object Services PTF
 - Benchmarking PSIG
 - Document Management PSIG
 - Internet PSIG
 - Realtime PSIG
 - Security PSIG
 - Test & Validation PSIG
 - Wrappers Working Group

Los grupos de trabajo del DTC son:

- Analytical Data Management DSIG
- Autonomous Decentralized Service System DSIG
- Business Object DTF

- Command, Control, Computers, Communications and Inteligence (C4I) DSIG
- CORBAgis DSIG
- Enterprise Application Integration DSIG
- Enterprise Customer Interaction System DSIG
- Finance DTF
- Healthcare (CORBAmed) DTF
- Human Resources Research DTF
- Knowledge Management Working Group
- Korea Working Group
- Life Sciences Research DTF
- Manufacturing
- Retail Working Group
- Telecom DTF
- Transportation DTF
- Utilities DTF

Los grupos de trabajo del AB son:

- Object & Referencie Models SC
- Liaison SC
- Test SIG
- CORBA Users SIG
- Metrics SIG

2.2.3. Políticas de enlace

El OMG mantiene relaciones activas con muchos otros organismos y consorcios de estandarización, en una labor de gran importancia por cuanto reduce la duplicación de normas y la cosecuente confusión en el mercado. Para tal efecto se creó bajo la dependencia del Consejo de Arquitectura (AB), el Sub-Comité de Enlace (*Liaison Sub-Committee*). Entre las organizaciones con las cuales se tienen acuerdos de cooperación, figuran:

- El Consorcio World Wide Web.
- El Open Group (X/Open + SF).
- Organismos internacionales acreditados como: ISO e ITU.
- Organismos nacionales como: ANSI, IEEE e JSA.
- Organismos sectoriales como: TINA-C, Open GIS Consortium, ECMA (*European Computer Manufacturers Association*), ANSA, OSI/NMF, SWIFT (*The Society for Worldwide International Financial Telecommunication*), etc.

2.3. La norma CORBA

CORBA es una especificación normativa que resulta de un consenso entre los miembros del OMG, un consorcio que agrupa hoy por hoy a más de 700 empresas tanto de la industria informática como consumidores de la misma. Esta norma cubre cinco grandes ámbitos que constituyen los sistemas de objetos distribuidos:

- Un lenguaje de descripción de interfaces, llamado Lenguaje de Definición de Interfaces **IDL** (Interface Definition Language), traducciones de este lenguaje de especificación IDL a lenguajes de implementación (como pueden ser C++, Java, ADA, etc.) y una infraestructura de distribución de objetos llamada Object Request Broker (ORB) que ha dado su nombre a la propia norma: Common Object Request Broker Architecture (CORBA).

- Una descripción de servicios, conocidos con el nombre de **Servicios CORBA** (CorbaServices), que complementan el funcionamiento básico de los objetos de que dan lugar a una aplicación. Estas especificaciones cubren los servicios de nombrado, de persistencia, de eventos, de transacciones, etc. El número de servicios se amplía continuamente para añadir nuevas capacidades a los sistemas desarrollados con CORBA.
- Una descripción de servicios orientados al desarrollo de aplicaciones finales, estructurados sobre los objetos y servicios CORBA. Con el nombre de **Facilidades Comunes** (CorbaFacilities), estas especificaciones cubren servicios de alto nivel, como los interfaces de usuario, los documentos compuestos, la administración de sistemas y redes, etc. La ambición es aquí bastante amplia ya que CorbaFacilities pretende definir colecciones de objetos prefabricados para aplicaciones habituales en la empresa: creación de documentos, administración de sistemas informáticos, etc.
- Una descripción de servicios verticales denominados **Interfaces de Dominio** (CorbaDomains), que proveen funcionalidad de interés para usuarios finales en campos de aplicación particulares. Por ejemplo, existen proyectos en curso en sectores como: telecomunicaciones, finanzas,

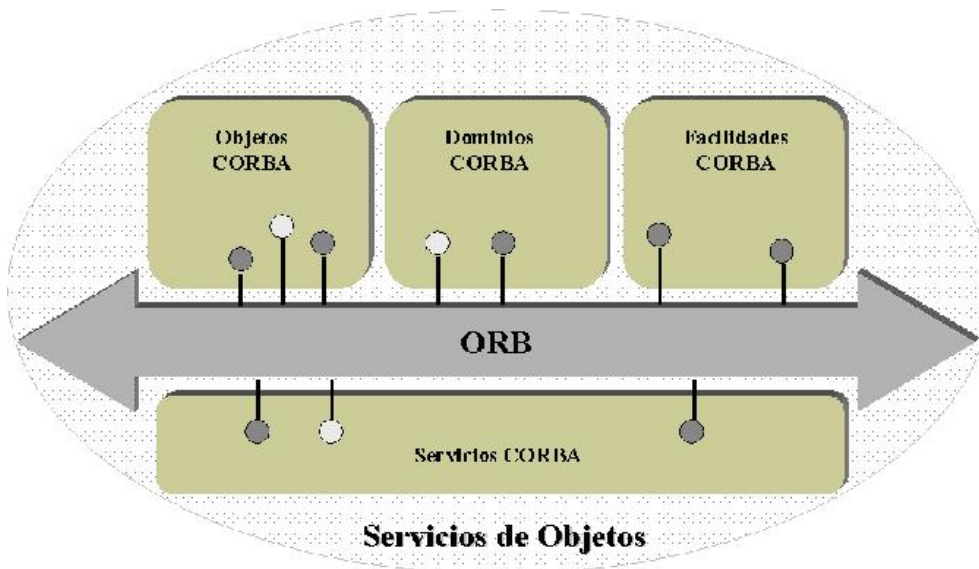


Figura 2.1: servicio de objetos

medicina, etc.

- Un protocolo genérico de intercomunicación, el Protocolo General Inter-ORB **GIOP** (General Inter-ORB Protocol), que define los mensajes y el empaquetado de los datos que se transmiten entre los objetos. Además define implementaciones, de ese protocolo genérico, sobre diferentes protocolos de transporte, lo que permite la comunicación entre los diferentes ORBs consiguiendo la interoperabilidad de elementos de diferentes vendedores. Por ejemplo el IIOP para redes con la capa de transporte TCP.

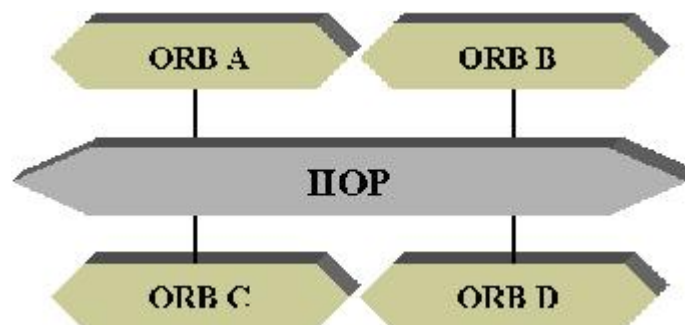


Figura 2.2: Un protocolo genérico de intercomunicación GIOP

2.4. Estructura

Object Request Broker (ORB) es el sistema intermedio (*middleware*) que establece relaciones cliente/servidor entre objetos. Mediante la utilización un de ORB, un cliente puede invocar transparentemente un método de un objeto servidor que se encuentre en la misma máquina o en otra distinta. El ORB intercepta la llamada realizada por el objeto que implementa la petición, pasa los parámetros, invoca el método y retorna los resultados. No es necesario que el cliente sepa dónde se localiza el objeto que ejecutará el método, el lenguaje en el que está programado, el sistema operativo, ni cualquier otro aspecto que no sea su interfaz. Hay que resaltar que los papeles de cliente y servidor que se dan a los objetos son simplemente para coordinar las interacciones entre ambos; estos papeles pueden cambiar ya que un objeto puede ser cliente o servidor dependiendo de la ocasión.

Su funcionamiento es el siguiente: cuando un módulo de una aplicación quiere usar un servicio proporcionado por otro módulo obtiene una referencia del objeto que provee ese servicio. Después de obtenerla, el módulo puede invocar métodos en ese objeto. La primera responsabilidad del ORB es resolver peticiones de referencias de objetos, permitiendo a los módulos de la aplicación establecer conexión entre ellos.

Otra de las responsabilidades del ORB es el marshaling y el unmarshaling. Después de que un módulo de la aplicación haya obtenido una referencia del objeto cuyos servicios quiere usar, ese módulo ya puede invocar métodos en ese objeto. Generalmente estos métodos necesitan parámetros como entrada y devuelven otros parámetros como salida. El ORB debe recibir los parámetros de entrada del módulo que llama al método y marshal estos parámetros. Esto quiere decir que el ORB traduce los parámetros a un formato (on-the-wire format) que puede ser transmitido por la red hasta el objeto remoto. El ORB también unmarshals los parámetros devueltos, convirtiéndolos a un formato que el módulo llamante entienda.

Todo esto se hace de forma transparente a la intervención del programador. Una aplicación cliente invocará un método remoto y recibirá los resultados como si el método fuese local.

Gracias a este proceso se consigue la independencia de plataforma, debido a que los parámetros se traducen en la transmisión a un formato independiente de la plataforma (el on-the-wire format forma parte de las especificaciones CORBA) y en recepción se convierten al formato específico de la

plataforma. Un cliente ejecutándose en un sistema Macintosh podrá invocar métodos de un servidor que se ejecute en un sistema UNIX. Además de la independencia del SO usado, las diferencias de hardware (como el ordenamiento de los bytes, endianness, la longitud de las palabras, etc.) son irrelevantes puesto que el ORB hace las conversiones necesarias automáticamente.

En resumen, las responsabilidades del ORB son:

- Dada una referencia a un objeto por un cliente, el ORB localiza la correspondiente implementación del objeto (el servidor).
- Cuando el servidor está localizado, el ORB asegura que el servidor está preparado para recibir la petición.
- El ORB del lado del cliente acepta los parámetros del método que se está invocando y marshals los parámetros a la red.
- El ORB del lado del servidor unmarshals los parámetros de la red y se los entrega al servidor.
- Los parámetros de retorno, si existen, se marshal/unmarshal del mismo modo.
- La mayor ventaja que ofrece el ORB es el tratamiento de los datos independientemente de la plataforma.

En la figura siguiente se muestra la estructura de un sistema distribuido basado en CORBA. En él se puede observar dos partes: una cliente y una servidora. En la parte cliente existirá un programa cliente propiamente dicho al que CORBA le añade cierta infraestructura para permitir la comunicación con el servidor a través de la red. Del otro lado, la parte servidora estará formada por el objeto que exporta su funcionalidad, integrado en el servidor (proceso en el que se ejecuta) y diversos elementos que permiten que las invocaciones realizadas por el cliente a los métodos del objeto lleguen a éste, sean procesadas y sus resultados devueltos.

- La parte cliente estará formada por:

Los Stubs del Cliente :

Proporcionan una capa intermedia entre el cliente y el núcleo del ORB. Definen cómo los clientes invocan los servicios que proporcionan los objetos servidores. Desde la perspectiva del cliente, el

stub actúa como una especie de proxy, dando la impresión de que la invocación se está realizando sobre un objeto local como en cualquier aplicación orientada a objetos. El stub se encarga de codificar la operación y sus parámetros, y de enviarla de forma remota.

El interfaz de invocación dinámica:

Permite descubrir en tiempo de ejecución métodos para ser invocados. CORBA define una biblioteca, que permite localizar el método, generar los parámetros, realizar la llamada remota y recoger los resultados.

El almacén de interfaces :

Permite obtener y modificar la descripción de todos los componentes que en él están registrados, los métodos que soporta y los parámetros que requiere. CORBA llama a esas descripciones firmas. El almacén de interfaces (*Interface Repository*) es una base de datos distribuida modificable en tiempo de ejecución.

El interfaz del ORB :

Consiste en unas pocas librerías de servicios locales para realizar labores auxiliares en la aplicación. Por ejemplo, CORBA proporciona APIs (*Application Programming Interface*, interfaces de programación de aplicaciones) para convertir referencias a objetos a cadenas. Estas llamadas pueden ser muy interesantes si se necesita comunicar o almacenar referencias a objetos.

La capacidad de soportar tanto invocaciones dinámicas como estáticas, además del almacén de interfaces, da a CORBA una gran venta-

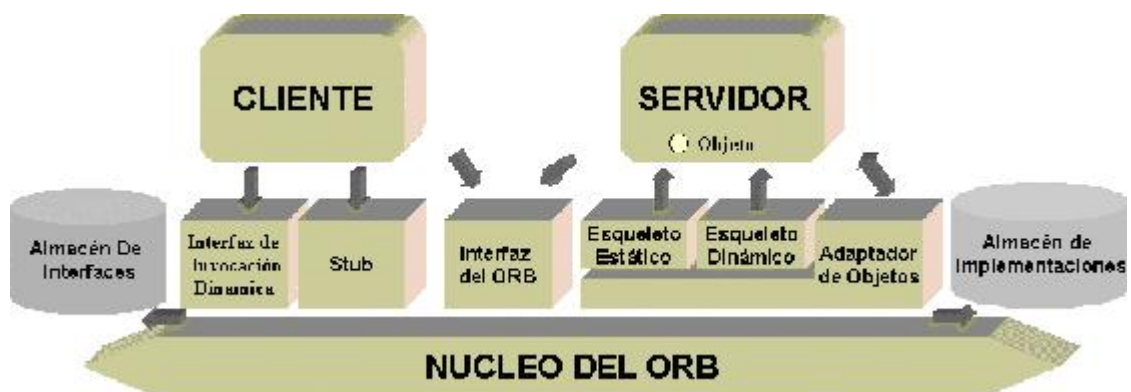


Figura 2.3: Sistema distribuido

ja sobre las plataformas intermedias competidoras. Las invocaciones estáticas son rápidas y fáciles de programar. Las invocaciones dinámicas proporcionan la máxima flexibilidad, pero son difíciles de programar. Estas últimas son muy usadas por herramientas que descubren servicios en tiempo de ejecución.

Por lo que respecta al lado del servidor no hay diferencia entre la invocación dinámica y estática, ya que ambas tienen el mismo mensaje semánticamente hablando. En los dos casos el ORB localiza al Adaptador de Objetos del objeto y le transmite un mensaje que contiene el nombre del servicio a invocar y sus parámetros. La implementación recibe a través del esqueleto del objeto los datos necesarios, ejecuta el servicio y retorna el resultado para que sea enviado al cliente en forma de un nuevo mensaje.

- Los elementos que componen la parte servidora son los siguientes:

El esqueleto del servidor:

Proporciona los elementos necesarios para que los clientes invoquen los servicios exportados por el objeto. Estos esqueletos realizan una función similar a los stubs del cliente tratando de hacer transparente todo el proceso de comunicación.

El esqueleto de interfaces dinámicos (DSI):

Proporciona un mecanismo de enlazado en tiempo de ejecución para servidores que necesitan manejar llamadas a métodos que no tienen esqueletos estáticos definidos.

El adaptador de objetos:

Proporciona en tiempo de ejecución un entorno para la instanciación de objetos en el servidor (proceso que los acoge), la asignación de referencias y la gestión las peticiones que les llegan.

El almacén de implementaciones:

Proporciona en tiempo de ejecución un almacén de información acerca de las clases que el servidor soporta, los objetos instanciados y sus identificadores.

El interfaz del ORB:

Consiste en unas pocas APIs de servicios locales iguales a las de la parte cliente.

2.5. Lenguaje de definición de interfaces

La otra pieza fundamental de la arquitectura CORBA es el uso del lenguaje de definición de interfaz (**IDL**) que especifica interfaces entre objetos CORBA. Es un lenguaje estándar.

Los interfaces descritos en IDL se pueden ajustar a cualquier lenguaje de programación lo que hace que las aplicaciones y los componentes CORBA sean independientes del lenguaje utilizado para implementarlos. El pliego de condiciones IDL es el responsable de asegurar que los datos son intercambiados correctamente entre lenguajes diferentes. Por ejemplo, el tipo `long` en IDL es un entero de 32 bits con signo, que corresponde a un `long` de C++ o a un `int` de Java.

Esta independencia del lenguaje se consigue gracias al language mapping, que es un pliego de condiciones que empareja las construcciones en IDL con las de un lenguaje de programación particular. Por ejemplo en el C++ mapping el “interface” de IDL corresponde a una “class” de C++. La OMG ha definido un número de language mapping estándar para muchos lenguajes, tales como C, C++, COBOL, Java, Smalltalk, etc. Mappings para otros lenguajes existen, pero o no son estándar o están en proceso de serlo.

Otra característica importante de IDL es que no se trata de un lenguaje de implementación (no se pueden escribir aplicaciones en IDL), su único propósito es definir interfaces. Comparando con C++, se podría decir que las definiciones IDL son análogas a los ficheros de cabecera para las clases

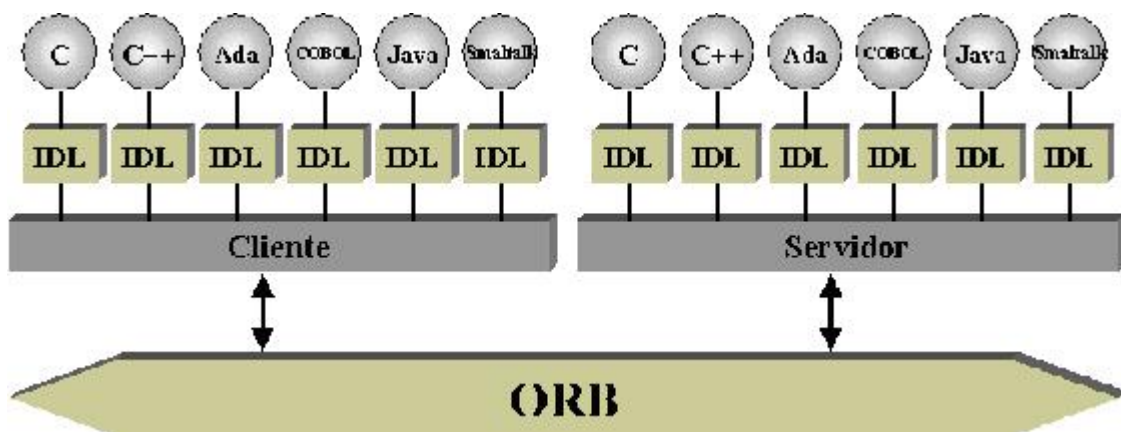


Figura 2.4: Lenguaje de definición de interfaces

(que no contienen la implementación de la clase, sino que más bien definen su interfaz). Con respecto a Java sería parecido a las definiciones de los interfaces Java.

CORBA no dicta el uso de un lenguaje en particular, por lo que deja a los diseñadores de la aplicación la elección del lenguaje que más se ajuste a sus necesidades. Incluso se pueden utilizar varios lenguajes para implementar cada parte de la aplicación. Por ejemplo, el cliente puede estar escrito en Java que asegura que pueda funcionar en cualquier máquina, mientras que el servidor se puede implementar en C++ para obtener unas buenas prestaciones.

En la descripción IDL, el programador deberá plasmar todos los elementos del objeto que serán accesibles para los clientes. Es habitual establecer una similitud entre lo que es un fichero de especificación CORBA-IDL y lo que son las partes públicas de un fichero de cabecera (".h" o ".hh") en C++.

A continuación se muestra:

una tabla con los tipos básicos CORBA-IDL y su equivalencia en C++:

CORBA-IDL	C++
short	CORBA::Short
long	CORBA::Long
long long	CORBA::LongLong
unsigned short	CORBA::UShort
unsigned long	CORBA::ULong
unsigned long long	CORBA::ULongLong
float	CORBA::Float
double	CORBA::Double
long double	CORBA::LongDouble
char	CORBA::Char
wchar	CORBA::WChar
wstring	CORBA::WString
boolean	CORBA::Boolean
octet	CORBA::Octet

Figura 2.5: Mapeado en C++

una tabla con los tipos básicos CORBA-IDL y su equivalencia en Java:

CORBA-IDL	Java
short	short
long	int
long long	long
unsigned short	short
unsigned long	int
unsigned long long	long
float	float
double	double
long double	double
char	char
wchar	char
string	java.lang.String
wstring	java.lang.String
boolean	boolean
octet	byte

Figura 2.6: Mapeado en Java

2.6. El modelo de comunicaciones de CORBA

Vamos a intentar entender el papel de CORBA en una red de ordenadores. Típicamente ésta consiste en sistemas que están físicamente conectados. Esta capa física proporciona el medio a través del cual tiene lugar la comunicación. Más allá de la capa física se encuentra la capa de transporte, que incluye los protocolos responsables de mover los paquetes de datos desde un punto hacia otro. CORBA es neutral respecto estos protocolos de red, es independiente del protocolo utilizado y podría funcionar con cualquiera.

El estándar CORBA especifica el Protocolo General Inter-ORB (GIOP) que, en un nivel alto, establece un estándar para la comunicación entre varios ORB's de CORBA. GIOP es tan sólo un protocolo general, por lo que el estándar CORBA también especifica otros protocolos que detallan GIOP para usar un protocolo de transporte particular (como TCP/IP o DCE). El utilizado para redes TCP/IP se denomina Internet Inter-ORB Protocol (IIOP). Los fabricantes deben implementar este protocolo para ser considerados conformes a CORBA. Este requerimiento ayuda a asegurar la interoperabilidad entre productos CORBA de diferentes fabricantes, aunque cada uno puede tener además sus propios protocolos.

Un ORB puede soportar cualquier protocolo, pero siempre debe que incluir el IIOP (se negocia el protocolo a usar al establecerse la conexión entre los ORB's). De todas formas, los ORB's de CORBA normalmente se comunican usando el IIOP debido, en parte, a que este protocolo es el correspondiente al protocolo TCP/IP, que es el usado en Internet. En el argot CORBA/IIOP las referencias a objeto se pasan a denominar Referencias a Objetos Interoperables o IOR's.

Resumiendo, las aplicaciones CORBA se construyen encima de los protocolos derivados de GIOP (como el IIOP). Estos protocolos están, a su vez, encima de los protocolos de transporte (TCP/IP, DCE). Las aplicaciones CORBA no están limitadas a usar sólo uno de estos protocolos, sino que se puede usar un puente para interconectar aplicaciones situadas en redes que trabajen con diferentes protocolos. Se puede ver que, más que suplantar los protocolos de transporte, la arquitectura CORBA crea otra capa (la capa del protocolo Inter-ORB) que utiliza estos protocolos como soporte. Esta es otra de las claves de la interoperabilidad entre las aplicaciones CORBA ya que no se dicta el uso de un protocolo de transporte particular.

2.7. El modelo de objetos de CORBA

En CORBA, todas las comunicaciones entre objetos se hacen a través de referencias de objeto (explicadas posteriormente). Además, la visibilidad de los objetos se obtiene únicamente pasando referencias a esos objetos. Los objetos no pueden ser pasados por valor (al menos en la especificación actual de CORBA). En otras palabras, los objetos remotos en CORBA permanecen remotos, no hay actualmente ninguna forma de mover o copiar un objeto de un sitio a otro.

Todas las arquitecturas orientadas a objetos ofrecen un modelo de objetos, que describe cómo se representan los objetos en el sistema. Tres de las principales diferencias entre el modelo de objetos de CORBA y los modelos tradicionales radican en la forma semi-transparente de distribuir los objetos en CORBA, el tratamiento de las referencias a objetos y el uso de los llamados adaptadores de objetos (como el BOA -Basic Object Adapter).

Para un cliente CORBA, una llamada a un método remoto es exactamente igual a una llamada a un método local. La naturaleza distribuida de los objetos en CORBA es transparente a los usuarios de dichos objetos. Debido a que los objetos se encuentran distribuidos hay más posibilidades de que se produzca un fallo, por lo que CORBA debe ofrecer la forma de manejar estas situaciones. Lo resuelve ofreciendo un conjunto de excepciones de sistema, que pueden ser lanzadas por cualquier método remoto.

En una aplicación distribuida, hay dos métodos posibles de obtener acceso a un objeto de otro proceso. Uno se conoce como "paso por referencia". Se explica muy bien mediante un ejemplo. Supongamos dos procesos, A y B. El primer proceso (A) pasa una referencia a un objeto suyo al segundo proceso (B). Cuando el proceso B invoca un método en ese objeto, el método lo ejecuta el proceso A puesto que él posee el objeto. El proceso B sólo tiene visibilidad del objeto (a través de la referencia al objeto) y por lo tanto sólo puede pedir al proceso A que ejecute los métodos en su lugar. Es decir, pasar un objeto por referencia significa que un proceso concede visibilidad de uno de sus objetos a otro proceso mientras retiene la propiedad del objeto. Las operaciones en el objeto a través de la referencia al objeto las procesa el propio objeto.

El segundo método de pasar un objeto se denomina "paso por valor". El estado actual del objeto se pasa al módulo que lo pide donde se creará una nueva copia del objeto. Siguiendo con el ejemplo de antes, cuando el proceso

B invoque métodos en el objeto se ejecutarán en dicho proceso (en la copia) en vez de en el proceso A, donde el objeto original reside. Por lo tanto el estado del objeto original no cambia.

Como se dijo antes, en el modelo de objetos de CORBA todos los objetos se pasan por referencia. Al pasar un objeto por valor es necesario asegurarse de que el módulo que recibe el objeto tiene implementaciones para los métodos soportados por dicho objeto. En cambio esto no es necesario cuando los objetos son pasados por referencia.

Sin embargo, hay algunos problemas asociados a pasar los objetos por referencia. Todas las llamadas a los métodos serán remotas, por lo que si un componente invoca repetidas veces métodos en un mismo objeto remoto se producirá una gran cantidad de sobrecarga en la comunicación entre los dos componentes (sería más eficiente pasar el objeto por valor, de forma que el componente pueda manipular el objeto localmente).

El estándar CORBA define una serie de adaptadores de objetos (object adapters) cuyo primer propósito es conectar la implementación de un objeto con su ORB.

2.8. Adaptadores de objetos

El Adaptador de Objetos juega un papel muy importante en la estructura de un sistema CORBA. Se sitúa como un pegamento entre las implementaciones de los objetos CORBA y el propio ORB, consiguiendo que las peticiones que llegan a través del ORB sean procesadas por la implementación del Objeto. Entre sus responsabilidades se encuentran:

Registrado de los objetos:

Los Adaptadores de Objetos deberán proporcionar funciones para registrar implementaciones para los objetos CORBA.

Generación de referencias a objetos:

Los Adaptadores de Objetos deberán generar referencias para los objetos que tengan registrados.

Activación de procesos servidores:

Los Activadores de Objetos deberán activar los procesos (servidores) donde los objetos puedan ser activados.

Activación de objetos:

Los Activadores de Objetos deberán activar los objetos registrados.

Multiplexación de peticiones a los objetos registrados:

Los Adaptadores de Objetos deberán asegurar que todas las peticiones sean recibidas por los objetos, aunque tengan múltiples conexiones, sin que ninguna se bloquee indefinidamente.

Gestión de las invocaciones:

Los Adaptadores de Objetos deben despachar las peticiones de objetos registrados.

En la norma CORBA han aparecido dos adaptadores de objetos: el actual adaptador POA (*Portable Object Adaptor*), y el original y ya desaparecido (en la versión CORBA2.2) BOA (*Basic Object Adaptor*).

2.8.1. Adaptador básico de objetos (BOA)

El adaptador de objetos BOA fue el primero en ser incluido en la norma CORBA y cubría de forma mínima las funciones del Adaptador de Objetos, como el registro de objetos, la generación de referencias, la activación de implementaciones de objetos y la gestión de las peticiones.

Hay que resaltar que CORBA distingue entre los objetos, unidades que son susceptibles de exportar la funcionalidad, y los servidores, que son los procesos que alojan a los objetos. De este modo el BOA tiene cuatro políticas de activación que define diferentes relaciones entre los objetos y los servidores que los alojan. A continuación se mostrarán estas políticas y la forma de registrar los objetos para las mismas.

- **Políticas de activación del BOA**

El BOA permite activar objetos de cuatro modos diferentes:

Servidor compartido:

En la política de servidor compartido múltiples objetos pueden residir en el mismo servidor (o proceso). El BOA activa el servidor la primera

vez que recibe una petición para uno de los objetos que tiene registrados y están alojados en ese servidor. Después de que el servidor se haya inicializado, notifica al BOA que está preparado para manejar peticiones a través de la llamada *impl_is_ready*. Todos los objetos de este servidor quedarán listos para recibir peticiones a partir de ese momento.

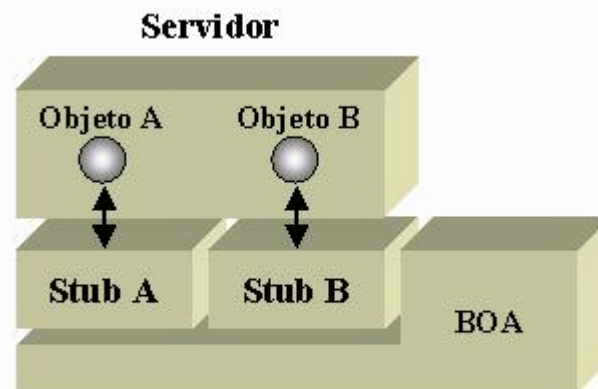
Cuando un proceso está listo para terminar, lo notifica al BOA emitiendo una llamada *deactivate_impl*. En ese punto todos los objetos instalados en ese servidor serán automáticamente desactivados.

Servidor no compartido:

En un servidor no compartido la política de activación consiste en que cada objeto reside en un proceso diferente. Se activa un servidor nuevo la primera vez que se realiza una petición sobre el objeto. Cuando el objeto se inicializa notifica al BOA que está preparado para manejar peticiones a través de la llamada *obj_is_ready*. El servidor dejará de recibir peticiones cuando se realice una llamada a *deactivate_obj*.

Servidor por método:

En la política de activación por método, se arranca un servidor siempre que se realiza una petición. El servidor permanecerá activo solamente durante la ejecución de un método. Varios procesos pueden estar activos concurrentemente dando servicio al mismo objeto. Debido a que cada nueva petición arranca un nuevo servidor no será necesario que se notifique al BOA cuando el objeto está listo o quiere ser desactivado.



Un Servidor, varios objetos

Figura 2.7: Servidor compartido

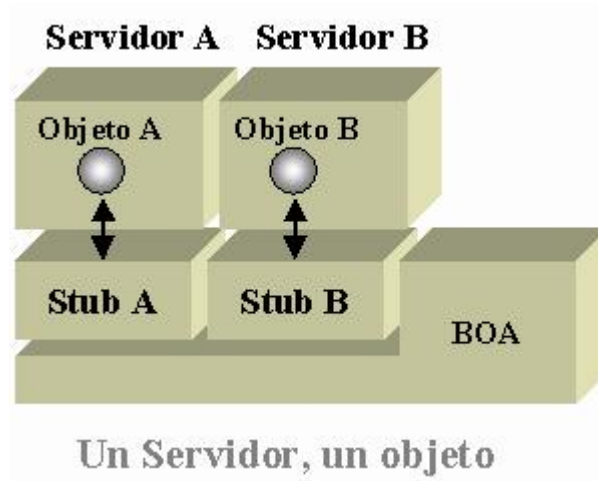


Figura 2.8: Servidor no compartido

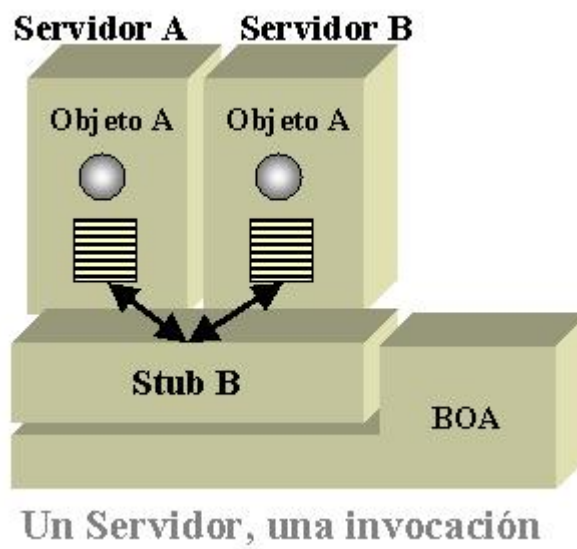


Figura 2.9: Servidor por método

Servidor persistente:

En la política de activación persistente, los servidores son activados por métodos ajenos al BOA. Cuando el proceso servidor está listo realiza la llamada *impl_is_ready*. La única diferencia con la activación compartida es que no es el BOA quien activa el servidor.

Esta variedad de políticas de activación permite elegir el tipo de comportamiento que más se adapte a un determinado servidor. Por ejemplo, un servidor que tarde mucho en arrancar trabajaría mejor como servidor persistente; por el contrario un servidor que arranque rápidamente bajo demanda podría trabajar bien como servidor por método.

2.8.2. Adaptador portable de objetos (POA)

- **Políticas del POA**

La especificación CORBA define un conjunto de políticas que especificarán como es la gestión que el POA va a realizar sobre sus objetos asociados. El programador podrá definir los valores para estas políticas cuando cree un nuevo POA. A continuación se comentarán las políticas y sus posibles valores:

Política de hilos:

Se usa para controlar los hilos que el servidor puede crear para procesar las invocaciones sobre los objetos que contiene. Existen dos posibilidades: la de un único hilo (SINGLE_CTRL_MODEL) o la de hilos controlados por el ORB (ORB_CTRL_MODEL). Si se utiliza la política de único-hilo todas las peticiones se procesarán secuencialmente. En el caso de utilizar multihilo, será el ORB el que los gestione. En este caso todos los objetos, incluidos el Mánager de Sirvientes, deben tener un código reentrante.

Política de vida:

Esta política se utiliza para especificar al POA si los sirvientes que se vayan a activar en él serán persistentes (PERSISTENT) o transitorios (TRANSIENT). Cuando un objeto es persistente puede sobrevivir al

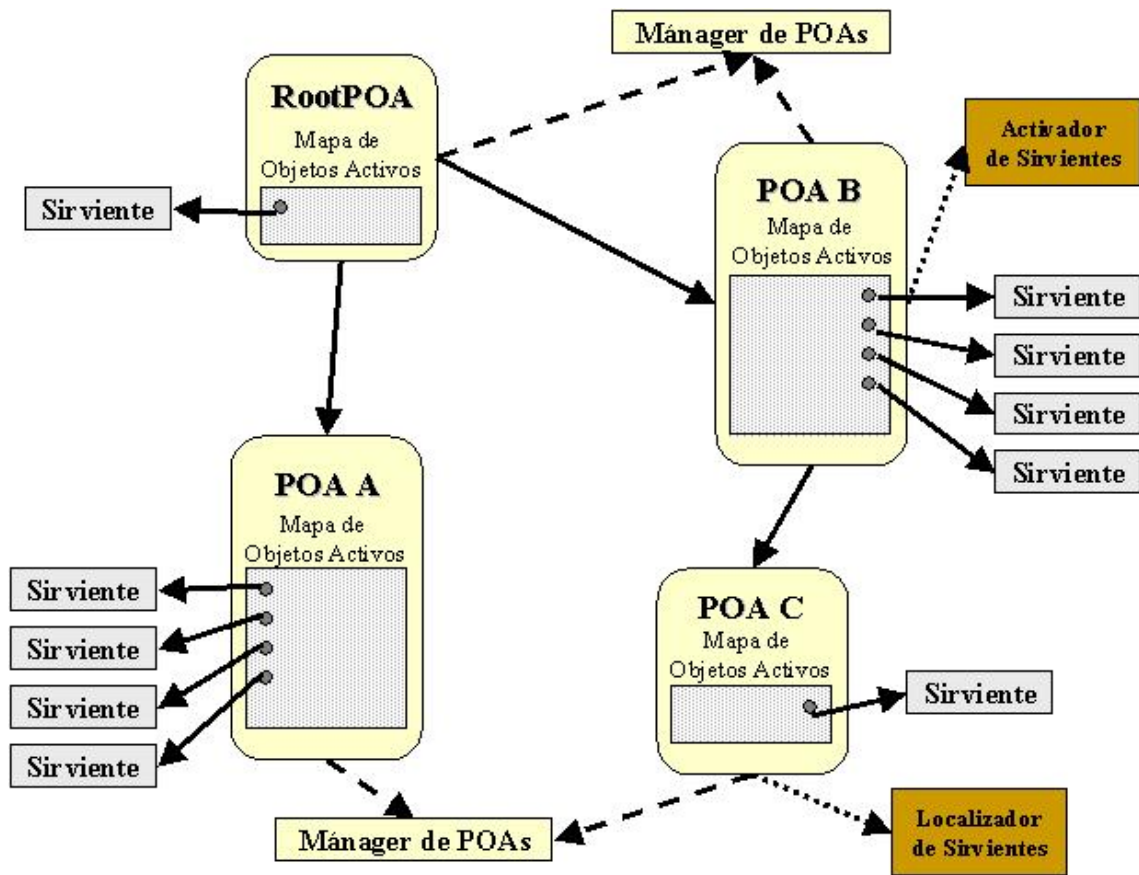


Figura 2.10: Adaptador portable de objetos

proceso que lo ha creado. Esto quiere decir que, si se desactiva un servidor con objetos persistentes en su interior, la próxima vez que vuelva a ser activado, el POA le generará idénticas referencias, con lo que para todos sus clientes el objeto será el mismo que el que estuvieron utilizando anteriormente. Por supuesto el POA no se hará cargo de restaurar el estado de los objetos, si no que deberá ser el programador el que se encargue de almacenarlo antes de la destrucción y recuperarlo al volver a activar los objetos. Por el contrario, si los objetos son transitorios, el POA al reactivarlos les generará nuevas referencias, con lo que sus clientes no podrán reconocerlos.

Política de identificador único:

La política de Identificador Único se utiliza para indicar al POA si los sirvientes deben tener un único identificador (UNIQUE_ID) o por el contrario podrán tener varios (MULTIPLE_ID). Cuando se utiliza un único identificador el POA fuerza a que cada referencia tenga un sirviente que atienda sus peticiones.

Política de asignación de identificador:

Esta política se utiliza para indicar al POA si los identificadores de los objetos van a ser creados por la aplicación (USER_ID) o por el ORB (SYSTEM_ID).

Política de activación:

Esta política se utiliza para indicar al POA si la activación implícita está permitida (IMPLICIT_ACTIVATION) o no (NO_IMPLICIT_ACTIVATION). Si lo está, un servidor podrá crear un sirviente y activarlo en el POA mediante una única operación (`_this`).

Política de retención de sirvientes:

La política de Retención de Sirvientes se utiliza para indicar al POA si debe almacenar la relación sirviente-referencia en el mapa de sirvientes activos. Sus valores pueden ser retener (RETAIN) y no retener (NON_RETAIN). Si el valor elegido es no retener deberá combinarse con los valores de la política de procesamiento de peticiones USE_DEFAULT_SERVANT o USE_SERVANT_MANAGER. La utilización de NON_RETAIN como Política de Retención de Sirvientes está recomendada para servidores que tienen una gran cantidad de sirvientes, ya que el almacenado de las asociaciones requiere una gran cantidad de memoria.

Política de procesamiento de peticiones:

Esta política se utiliza para indicar al POA cómo debe procesar las peticiones que le lleguen. Existen tres posibilidades:

- Consultar solamente el Mapa de Sirvientes Activos (`USE_ACTIVE_MAP_ONLY`): el POA comprueba si la referencia tiene entrada en el Mapa de Sirvientes Activos; si no la tiene, retornará al cliente una excepción de objeto inexistente. Esta posibilidad no podrá ser utilizada si la política de vida es `NON_RETAIN`.
- Utilizar Sirviente por Defecto (`USE_DEFAULT_SERVANT`): en este caso si el identificador de objeto no está en el Mapa de Objetos Activos, el POA intentará despachar la petición con el Sirviente por Defecto.
- Invocar al Mánager de Sirvientes (`USE_SERVANT_MANAGER`): si el POA no encuentra el identificador del objeto en el Mapa de Objetos Activos utilizará el Mánager de Sirvientes para intentar encontrar un sirviente que responda a la petición. Existen dos tipos de Mánagers de Sirvientes: el Activador de Sirvientes y el Localizador de Sirvientes. Según la política de Retención de Sirvientes el POA admitirá un tipo o el otro: Activador de Sirvientes para `RETAIN` y Localizador de Sirvientes para `NON_RETAIN`.

La utilización de Sirviente por Defecto y de Localizador de Sirvientes esta desaconsejada en objetos que tienen estado. En el caso del Sirviente por Defecto, debido a que varios clientes pueden utilizar un mismo sirviente y en el caso del Localizador de Sirvientes porque existe la posibilidad que las peticiones de un mismo cliente sean tratadas por sirvientes diferentes

La utilización de Mánagers de Sirvientes permite al programador establecer criterios en la asignación de sirvientes a referencias, control del número de sirvientes activos, establecer límites en el número de sirvientes activos simultáneos, etc. Con ello puede conseguir una gestión del servidor que aproveche de forma más eficiente los recursos del sistema.

2.9. Clientes y servidores CORBA

Tradicionalmente, en una aplicación cliente/servidor, el servidor es el componente que provee un servicio a otros componentes de la aplicación,

mientras que un cliente es un componente que utiliza servicios proporcionados por un servidor. En CORBA, igualmente, un componente es considerado un servidor si contiene objetos CORBA cuyos servicios están accesibles para otros objetos. De la misma forma, un componente es considerado un cliente si accede a servicios de otros objetos CORBA. Pero cuando consideramos una invocación a un método remoto los papeles de cliente y servidor se pueden invertir temporalmente, ya que un objeto CORBA puede participar en múltiples interacciones simultáneamente.

Si un componente crea un objeto y proporciona a otros componentes visibilidad a ese objeto (permite a otros componentes obtener referencias a ese objeto), ese componente actúa como un servidor para ese objeto; cualquier petición hecha a ese objeto por otro componente será procesada por el componente que creó el objeto. Ser un servidor CORBA significa que el componente (el servidor) ejecuta métodos en un objeto particular en nombre de otros componentes (los clientes).

Pero frecuentemente un módulo de una aplicación puede proporcionar servicios a otros módulos mientras a su vez accede a servicios de otros módulos. En este caso el módulo está actuando como cliente y servidor a la vez. De hecho, dos módulos pueden actuar simultáneamente como clientes y servidores uno del otro.

Consideremos el siguiente escenario: un primer módulo (A) recibe una referencia a un objeto creado por un segundo módulo (B) e invoca un método en ese objeto. Aquí el módulo A actúa como cliente y el módulo B como servidor. Ahora supongamos que como un parámetro del método invocado se debe pasar una referencia a un objeto. Si el módulo B ahora invoca algún método en ese objeto, para esta invocación en particular el módulo A actúa como servidor mientras que el módulo B actúa como cliente. Los dos módulos no han cambiado sus papeles globales en la aplicación, pero han invertido temporalmente sus funciones como cliente y servidor. Por eso los términos cliente y servidor dependen del contexto del método invocado y de en qué módulo resida el objeto del método.

Aunque un módulo de una aplicación puede funcionar tanto como cliente como servidor, es usual etiquetar al módulo como uno de los dos (no ambos a la vez). Si la función global de un módulo es usar los servicios proporcionados por otro módulo, y sólo proporciona objetos como argumentos a los métodos de módulo invocado, nos podemos referir al primer módulo como cliente y al segundo como servidor. Los métodos invocados de esta forma se denominan *client callback methods* (un *callback* es un método implementado por un

cliente e invocado por un servidor). Son realmente importantes en CORBA debido a su carencia de pasar objetos por valor.

2.10. Stubs y skeletons

Después de crear las definiciones de la interfaz de un módulo usando IDL se ejecutan los ficheros IDL resultantes en un compilador IDL. Éste genera lo que se conoce como client stubs y server skeletons (no es necesario escribirlos). Sirven como una especie de "pegamento" que conecta las especificaciones de la interfaz IDL independientes del lenguaje con la implementación en un código de un lenguaje específico.

Un client stub es una pequeña parte de código que hace que la interfaz de un servidor CORBA esté disponible para un cliente, y se compila conjuntamente con la parte cliente de la aplicación, es decir, los client stubs de cada interfaz se proporcionan para incluirlos en los clientes que usan ese interfaz. El client stub de un interfaz determinado proporciona una maqueta de la implementación de cada uno de los métodos de esa interfaz. Los métodos del client stub simplemente se comunican con el ORB para marshal y unmarshal parámetros.

Un server skeleton es una pequeña parte de código que proporciona la "armazón" sobre la cual se construye el código de implementación del servidor para un determinado interfaz. Para cada método de una interfaz el compilador IDL genera un método vacío en el server skeleton. El programador debe entonces proporcionar una implementación para cada uno de estos métodos.

2.11. Desarrollo de aplicaciones CORBA

La creación de una aplicación CORBA es un proceso claramente definido tanto en las etapas que lo forman como en el orden de las mismas. Esto permite al desarrollador, una vez conocido el proceso, generar aplicaciones de forma rápida y sistemática.

Los pasos para la creación de una aplicación serán los siguientes:

1. **Definir los interfaces de los objetos que van a participar en la aplicación mediante el Lenguaje de Definición de Interfaces (IDL):**

El interfaz de los objetos será su ventana al mundo y el que permita que los demás objetos conozcan las operaciones que tienen disponibles y cómo deben ser invocadas. Mediante el lenguaje de definición IDL se definen: los métodos que el objeto exporta, los parámetros de los mismos, los atributos, las excepciones que genera, etc.

2. **Compilar los interfaces de los objetos con el compilador CORBA-IDL:**

En este paso, mediante la compilación del fichero de interfaces con el compilador CORBA-IDL, se obtienen los esqueletos del cliente y del servidor (en el lenguaje de implementación).

3. **Crear la implementación de los objetos, el servidor y el cliente:**

Se crearán las clases que dan la funcionalidad al objeto y las implementaciones del servidor (programa desde donde se crearán los objetos y se registrarán en el Activador de Objetos) y el cliente.

4. **Compilar el código:**

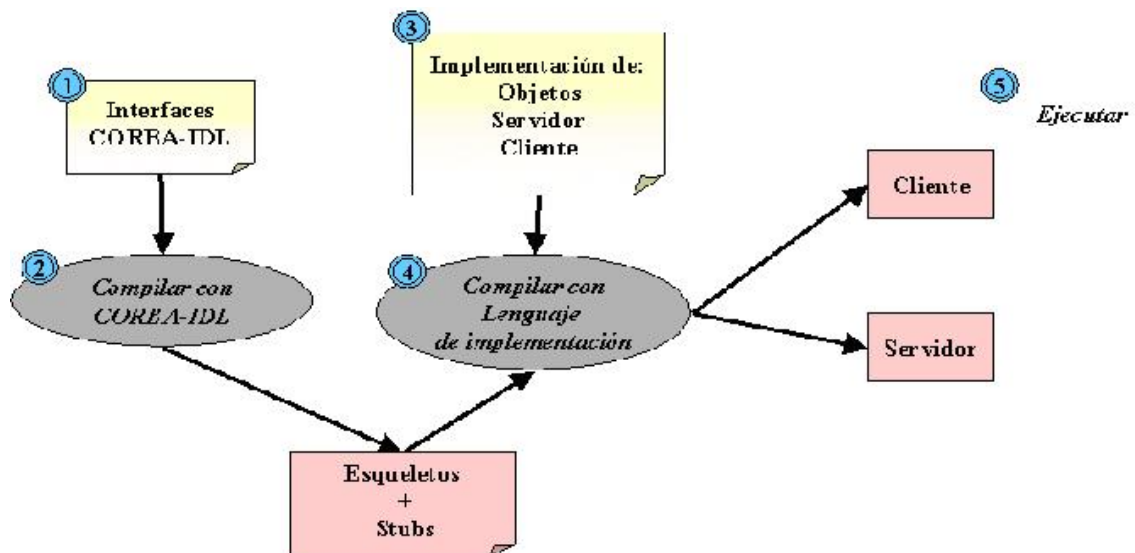


Figura 2.11: Aplicación CORBA

En este paso se procederá a la creación de los ejecutables de la futura aplicación. Por un lado se compilará la implementación del servidor y la de los objetos con los esqueletos para dar lugar a la parte servidora de la aplicación. Por otro lado, se compilará el cliente conjuntamente con los stubs para dar lugar a la parte cliente.

5. Ejecutar la aplicación:

La forma de ejecutar la aplicación distribuida dependerá del ORB que se esté utilizando. En algunos casos, previamente a la ejecución habrá que registrar el servidor en el almacén de implementaciones; en otros, el registro será automático; y en otros casos, al registrar el servidor será el Almacén de implementaciones el que arranque el servidor. Como norma general deberá arrancarse primero el servidor y posteriormente se podrá comenzar la ejecución del cliente.

Existen dos formas de desarrollar aplicaciones CORBA:

- Por herencia.
- Por delegación.

2.12. Invocación dinámica

Existen ocasiones en las que el cliente de un sistema distribuido basado en CORBA no conoce el interfaz de los objetos con los que desea trabajar en tiempo de compilación. En estos casos la invocación dinámica permite en tiempo de ejecución, la localización del interfaz y la generación dinámica de una invocación.

2.12.1. Pasos para la realización de una invocación dinámica

Una vez se ha obtenido la referencia al objeto del que se desea invocar un método dinámicamente los pasos a llevar a cabo serán los siguientes:

1. Obtención del nombre del interfaz.

2. Obtención de la descripción del método en el almacén de interfaces.
3. Creación de la lista de argumentos.
4. Creación de la petición.
5. Realización de la petición.

2.12.2. Los interfaces de la invocación dinámica

Los servicios que el usuario necesita para realizar invocaciones dinámicas, son parte del núcleo CORBA. Estos servicios están distribuidos en cuatro interfaces:

CORBA::Object

El interfaz CORBA::Object define 3 operaciones importantes a la hora de realizar invocaciones dinámicas. La primera de ellas es la operación *get_interface*, a través de ella es posible obtener la descripción del interfaz residente en el Almacén de Interfaces de un objeto. Las otras dos operaciones son *_create_request* y *_request*, ambas se utilizan para la creación de peticiones de forma dinámica, su diferencia radica en los parámetros que han de pasárseles que en el caso de segunda son mucho más sencillos.

CORBA::Request

El objeto CORBA::Request es el objeto petición. Contiene operaciones para introducir parámetros a la operación como *add_arg* y operaciones para enviar la petición al servidor y recibir el resultado. Existen diferentes formas de transmitir una petición, a través de PRCs, con data-gramas, o con el método de respuesta diferida (enviar/recibir).

CORBA::NVList

El objeto CORBA::NVList se utilizar para construir listas de parámetros para las peticiones (request). Contiene métodos para introducir parámetros como *add_arg*, para introducirles valores *add_value*, para conocer el número de parámetros introducidos *get_count*, etc.

CORBA::ORB

El objeto CORBA::ORB es un objeto de propósito general. Contiene algunas operaciones interesantes desde el punto de vista de la invocación dinámica como son: *create_operation_list* que facilita la creación de listas de parámetros para las peticiones, y las operaciones para envío y recepción de múltiples respuestas, *send_multiple_request_oneway*, *send_multiple_request_deferred*, *poll_next_response* y *get_next_response*.

2.12.3. Tipos de invocaciones**Invocaciones RPC:**

Las invocaciones tipo RPC se realizan mediante el método *invoke* de la petición que se acaba de generar.

En este caso mediante una única invocación se realiza la petición y se obtienen los resultados.

Invocaciones datagrama:

En este caso la petición se envía como un datagrama mediante la operación *send_oneway*. Este tipo de invocación solamente se utiliza para caso en los que no se necesita respuesta del objeto.

```
CORBA::Request_var peticion= obj->_request("resultado");
//...
peticion->invoke();
```

Figura 2.12: Invocación rpc

```
CORBA::Request_var peticion= obj->_request("resultado");
//...
peticion->send_oneway();
```

Figura 2.13: Invocación datagrama

Invocaciones con respuesta diferida (enviar/recibir):

En este tipo de invocación el proceso se realiza en dos partes, en una primera se envía la petición mediante la operación *send_deferred* devolviendo el control al programa y en la segunda parte se recibe la respuesta del servidor a través de las funciones *poll_response* (detecta y ha llegado la respuesta) y *get_response* (retorna el valor recibido).

Para cualquiera de los tipos de invocación los resultados obtenidos con las mismas quedarán almacenados en las variables asociadas a los parámetros out e inout. El valor de retorno podrá ser leído a través de la operación de la petición *return_value*.

```
CORBA::Request_var peticion= obj->_request("resultado");  
  
    //...  
  
peticion->send_deferred();  
while(!peticion->poll_response())  
{  
    //espera por la respuesta  
}  
  
peticion->get_response();
```

Figura 2.14: Invocación con respuesta diferida

2.13. Servidores multihilo

CORBA especifica desde su versión 2.2 la creación de servidores multihilo. Para crear servidores multihilo debe especificarse en el POA al que está asociado el o los objetos la política *ThreadPolicy* como **ORB_CONTROL_MODEL**. En este caso es el ORB decide en cada momento el número de hilos activos. Existen tres criterios para la gestión y asignación de estos hilos:

Hilo por petición(*thread-per-request*)

En este caso el servidor genera un hilo para cada una de las peticiones que llegan al sirviente. Esta opción se recomienda para sirvientes que tengan pocos accesos simultáneos. Ya que de no ser así la continua creación y destrucción de los hilos relentizaría el procesamiento de la peticiones.

Hilo por conexión(*thread-per-connection*)

El servidor genera un hilo para cliente conectado al sirviene. Se recomienda la utilización de este criterio en sirvientes que tienen conectados clientes durante un largo periodo de tiempo.

Conjunto de hilos(*thread pool*)

Con esta estrategia el servidor genera un conjunto inicial de hilos y los va asignado a las peticiones según van llegando. Si al llegar la petición hay hilos desocupados se le asigna uno que la procese y sino se encola hasta que quede alguno libre.

Mediante la utilización de la thread pool el diseñador del sistema puede controlar la utilización de los recursos del servidor, puesto que inicialmente establece un límite sobre el número de hilos que simultáneamente podrán estar trabajando. De esta forma impedirá que picos en la cantidad de peticiones recibidas puedan bloquear el servidor o hacer el procesamiento muy ineficiente.

2.13.1. Consecuencias en los sirvientes

Una consecuencia clara en los sirvientes será la necesidad de proteger el acceso a datos y variable compartidas. Puesto que el procesamiento de la peticiones ya no se procesan de forma secuencial los sirvientes deberán proteger tanto los datos que comparten con otros objetos como sus propias

variables. En el caso de los datos compartidos dos invocaciones sobre dos objetos podrían acceder simultáneamente al mismo dato.

En el segundo caso si un objeto recibe dos invocaciones y esta se procesan simultáneamente podrían interactuar sobre la misma variable, obteniendo resultados impredecibles.

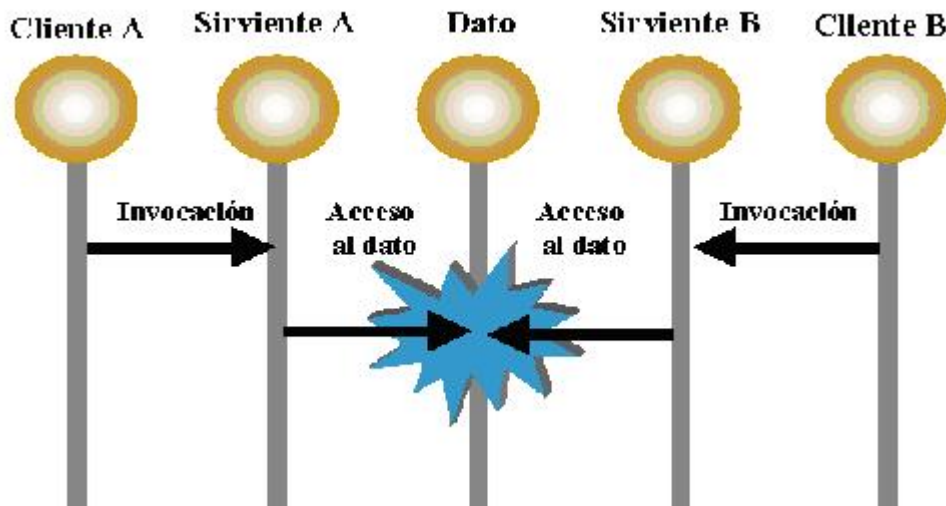


Figura 2.15: Primer caso de la consecuencia de los sirvientes

2.13.2. Estrategias multihilo

Existen dos formas de paralelizar servidores:

- Paralelización de grano grueso.
- Paralelización de grano fino.

En la paralelización de grano grueso solamente se permite un hilo por sirviente en cada momento. Por tanto, se cerrará un mutex al comienzo de cada método y no se abrirá hasta el final del mismo.

En la paralelización de grano fino se permite que múltiples hilos trabajen sobre un sirviente simultáneamente. Los mutex en este caso se situaran guardando las secciones críticas. En este tipo de paralelización el programador deberá ser extremadamente cuidadoso para evitar los interbloques (*deadlocks*).

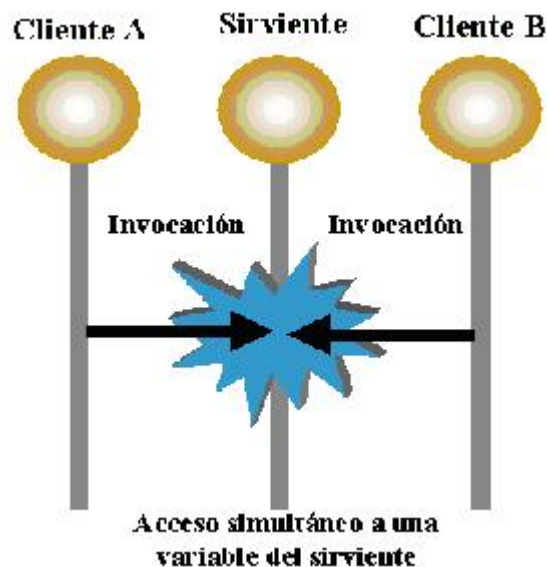


Figura 2.16: Segundo caso de la consecuencia de los sirvientes

2.14. Servicios corba

Los servicios CORBA son conjuntos de objetos pre-implementados que complementación la funcionalidad de otros objetos para dar lugar a una aplicación final. Su propósito es general, es decir, son independientes del dominio de aplicación.

La Norma CORBA especifica los siguientes servicios:

- Arranque
- Ciclo de Vida
- Colección
- Concurrencia
- Consulta
- Externalización
- Eventos
- Licencia
- Negociación
- Nombrado

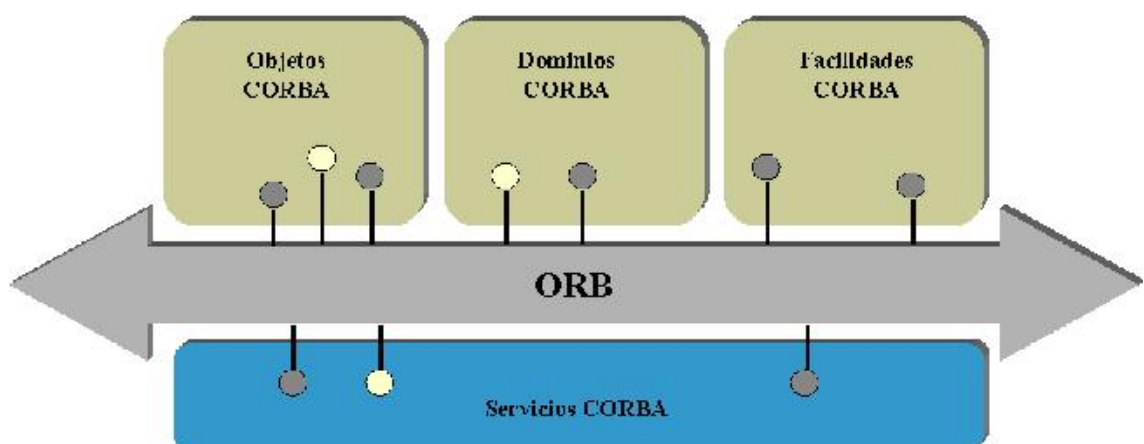


Figura 2.17: Servicios Corba

- Persistencia
- Propiedades
- Relación
- Seguridad
- Tiempo
- Transacciones

2.15. Facilidades comunes

Servicios de alto nivel, a manera de funciones de aplicación genéricas, que pueden ser configurados para una necesidad específica en cualquier campo de aplicación. Se conocieron inicialmente como Facilidades Comunes Horizontales, para diferenciarlas de las Facilidades Comunes Verticales (llamadas actualmente Interfaces de Dominio) que son específicas para determinados ámbitos de aplicación; hoy en día se denominan **CORBAfacilities**.

Mientras que los Servicios de Objetos están orientados a la manipulación de los objetos distribuidos, las Facilidades Comunes están orientadas al usuario final, brindándole servicios tales como impresión de documentos,

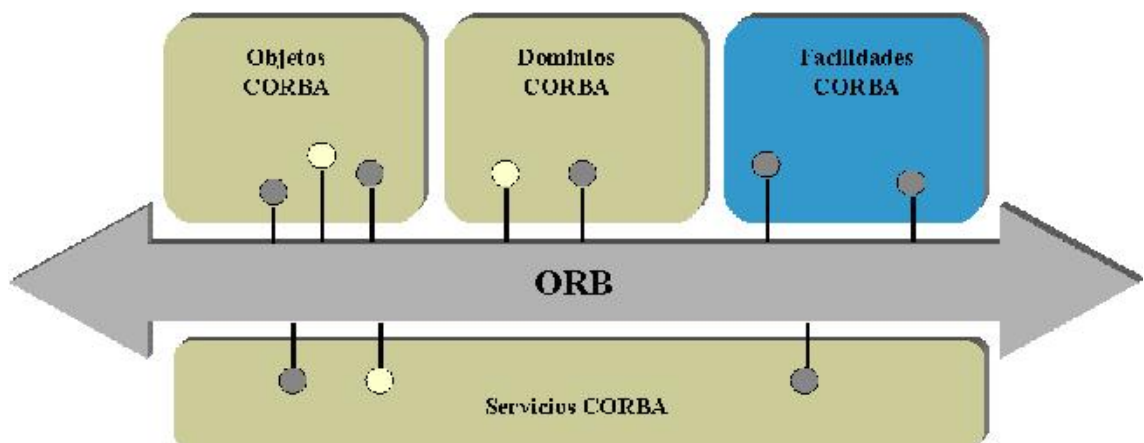


Figura 2.18: Facilidades comunes

correo electrónico, etc. El OMG define originalmente cuatro categorías principales de Facilidades Comunes:

Interfaz de usuario:

Facilidades para acceso del usuario a las aplicaciones y también para la representación de la información. Una de ellas es la Facilidad de Presentación Compuesta (*Compound Presentation Facility*), que provee un entorno para la compartición y subdivisión de las ventanas de despliegue de una arquitectura de documentos compuestos.

Gestión de información:

Facilidades para el modelado, almacenamiento, recuperación e intercambio de información. Un ejemplo es la Facilidad para el Almacenamiento y Recuperación de Información (*Information Storage and Retrieval Facility*), que es aplicable a diferentes servicios de información como acceso a bases de datos, directorios, búsquedas en Internet, etc.

Gestión de sistemas:

Facilidades para la gestión de sistemas de información, como: control, monitorización (monitoreo) seguridad, configuración y políticas de operación (adición de usuarios, permisos, instalación de programas, etc).

Gestión de tareas:

Facilidades para la automatización del trabajo, brindando soporte tanto a los procesos de los usuarios como a los procesos del sistema. Por ejemplo, la Facilidad de Flujo de Trabajo (*Workflow Facility*), ofrece servicios para la gestión y coordinación de objetos que son parte de procesos de trabajo, como en el caso de un proceso compras.

En un comienzo, el OMG había creado la Comisión de Facilidades Comunes (*Common Facilities Task Force*) a cargo del trabajo alrededor de este tema. Sin embargo, la actividad en este componente de la arquitectura se ha reducido sensiblemente, principalmente a causa del mayor interés que han despertado las Interfaces de Dominio, al punto que esta comisión ha sido absorbida por el ORBOS PTF (*ORB and Object Services Platform Task Force*), al dentro del cual continúa con los temas de gestión de documentos e Internet. Por otro lado, algunos asuntos que correspondían a esta comisión están siendo manejados por otras comisiones, como es el caso de las Facilidades de Flujo de Trabajo, que han quedado a cargo de la Comisión de Objetos de Negocio (*Business Object Task Force*).

2.16. Dominios corba

Así como las Facilidades Comunes, las Interfaces de Dominio definen funciones de aplicación genéricas, pero que en este caso corresponden a dominios de aplicación específicos como comercio electrónico, finanzas, medicina, manufactura, telecomunicaciones, transporte, etc.

Estos servicios se conocen como **CORBA domains**, y en su implementación pueden combinar algunas Facilidades Comunes y Servicios de Objetos. En la actualidad existe un gran número de Comisiones, Grupos Especiales de Interés y Grupos de Trabajo dedicados a su estudio, los cuales están desarrollando una actividad muy intensa que se refleja en las especificaciones en proceso y aprobadas.

- CORBA Manufacturing
- CORBA Finance
- CORBAmed
- CORBA Telecoms

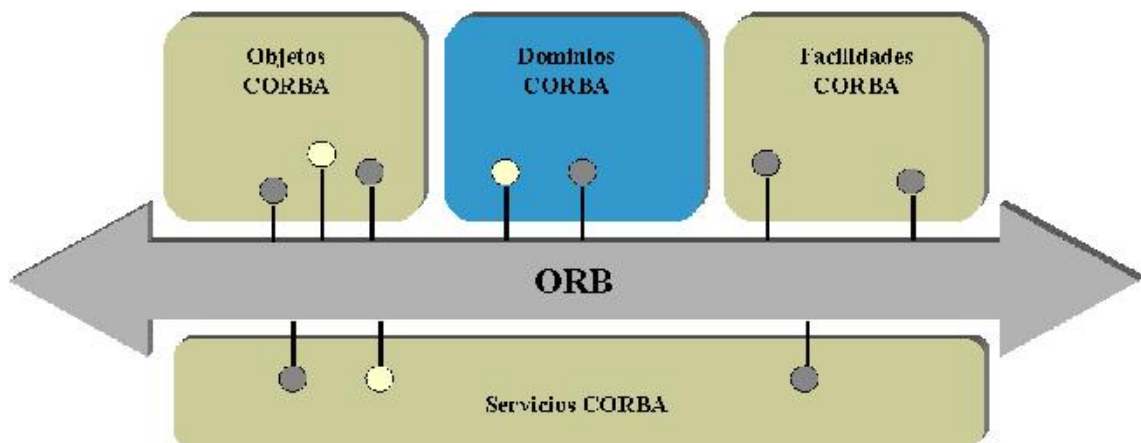


Figura 2.19: Dominios corba

2.17. El protocolo GIOP

El Protocolo General Inter-ORB GIOP (*General Inter-ORB Protocol*) es el artífice de la interoperabilidad entre ORBs de diferentes vendedores. El GIOP se implementa sobre protocolos de transporte orientados a conexión.

La especificación del protocolo GIOP consiste en los siguientes elementos:

- El CDR (representación común de datos). Define cómo deben codificarse los datos para su transferencia entre clientes y servidores.
- El formato de los mensajes GIOP. Define los mensajes que se intercambian el ORB del cliente y el del servidor cuando se realizan invocaciones sobre objetos, se localizan implementaciones y se controlan los canales de comunicación.
- Condiciones de la capa de transporte. La especificación GIOP describe condiciones que la capa de transporte debe cumplir para poder transmitir mensajes GIOP. La especificación también describe cómo se han de gestionar las conexiones y las restricciones en el orden de los mensajes GIOP.

La especificación IIOP añade los siguientes elementos a la especificación GIOP:

- El transporte de mensajes IOP en Internet. La especificación IIOP (*Internet IOP*) describe cómo los agentes abren conexiones TCP/IP y las usan para transferir mensajes GIOP.

El IIOP no es una especificación separada, sino que es una especialización o mapeado del GIOP al protocolo específico TCP/IP. La especificación GIOP se considera una base para futuras implementaciones a otros protocolos de transporte.

2.17.1. Representación común de datos

El protocolo GIOP define la representación común de datos (Common Data Representation, CDR) que determina la capa binaria de los tipos CORBA-IDL para la transmisión. El CDR tiene las siguientes características:

- El CDR soporta tanto representación big-endian como little-endian.

Los mensajes GIOP codifican los datos según un bit que indica la ordenación de los bytes que constituyen esos datos. Con ello se permite que tanto las máquinas “big-endian” como las “little-endian” puedan enviar sus datos según su representación nativa. Si el emisor y el receptor utilizan una ordenación de los datos diferente será el receptor el responsable de reordenarlos. Este modelo recibe el nombre de “el receptor debe ponerlo bien” (receiver makes it right). Su ventaja estriba en que si el modelo de ordenación de los datos en el emisor y en el receptor es el mismo, no han de hacer ninguna reordenación, y si el modelo es diferente, únicamente se hará una en cada transmisión. Si se toma como base la política utilizada por algunas implementaciones de las llamadas a procedimientos remotos, como por ejemplo las ONC RPCs de SUN, podrá observarse su mejora. Los RPCs transforman todos los datos que van a ser transmitidos a “big-endian”. Si ambas máquinas utilizan ese esquema de representación para sus datos nativos, no será necesario realizar ninguna reordenación. Pero si alguna de ellas utiliza el esquema “little-endian” deberá hacerse un cambio, y si las dos máquinas son “little-endian” los cambios deberán ser dos. (ver figura 2.20).

- Los tipos de datos primitivos se alinean sobre los límites naturales.

El CDR alinea los tipos primitivos de datos sobre límites de bytes, que es lo natural en la mayoría de las arquitecturas. Por ejemplo, los valores short se alinean sobre límites de 2-bytes, los valores long se alinean sobre límites de 4-bytes y los double se alinean sobre límites de 8-bytes.

ORDENACIÓN	CORBA	RPC
big-endian big-endian	0 cambios	0 cambios
big-endian little-endian	1 cambio	1 cambio
little-endian little-endian	0 cambios	2 cambios

Figura 2.20:

Realizar este tipo de alineación provoca que el flujo de datos que se transmite tenga algunos bytes de relleno. A pesar de esto, el CDR es más eficiente que la transmisión compacta que haría excesivamente costoso el empaquetado y desempaquetado de los datos.

- Los datos codificados con CDR no son auto-identificados.

CDR es una codificación binaria que no es auto-identificada. Por ejemplo, si un método requiere dos parámetros `int`, un `long` seguido de un `double`, el empaquetado de los datos consistirá en 16 bytes. Los 4 primeros contendrán el valor de `long`, los 4 siguientes serán relleno (con contenido indefinido) para mantener la alineación, y los 8 bytes finales contendrán el valor del `double`. El receptor simplemente observará 16 bytes de datos, por lo que deberá conocer por adelantado qué tipos contiene para poder desempaquetarlos correctamente.

Esto significa que la codificación CDR requiere que tanto emisor como receptor deban estar de acuerdo acerca de los tipos de los datos que se van a intercambiar. Este entendimiento será fijado por las definiciones IDL que ambos conocerán en tiempo de compilación. En este tipo de codificación el receptor no tiene capacidad para impedir la interpretación de los datos si el emisor ha variado los tipos. Por ejemplo, si el emisor enviara dos valores `double` en vez de un `long` seguido de un `double`, el receptor interpretaría los 4 primeros bytes del primer `double` como un valor `long`.

La codificación CDR es un compromiso a favor de la eficiencia. De ahí que se soporten representaciones tanto “little-endian” como “big-endian” y alineación de los datos en sus límites naturales. Por otro lado esto provoca que los errores en los tipos no puedan ser detectados en tiempo de ejecución.

2.17.2. Codificación de los datos en CDR

En esta apartado se presentarán las ideas básicas sobre las reglas de alineación de datos en CDR.

Codificación de tipos de datos simples:

Cada tipo simple debe comenzar en un byte determinado relativo a su tamaño según su aparición en el flujo de transmisión. La alineación de los tipos simples será la vista en la figura 2.21:

Codificación de strings:

Los strings y los wide strings se codifican como un unsigned long que indica la longitud de la cadena (incluido el byte del carácter nulo de terminación) seguido de los bytes de contenido y de un byte con un carácter nulo de terminación. Por ejemplo, la cadena “Perro” ocuparía 10 bytes. Los primeros 4 bytes son un unsigned long que contendrá el valor 6, los 5 bytes siguientes contendrán los caracteres “Perro”, y el último byte contendrá el carácter ASCII nulo.

Por supuesto como el primer elemento de la codificación de los string y los wstrings es un long, éstos se alinearán a los límites naturales del tipo long.

Codificación de estructuras:

Las estructuras en CDR se codifican como una secuencia de los miembros de la estructura según el orden en el que hallan sido definidos en IDL. Cada miembro de la estructura es alineado de acuerdo con el orden definido para su tipo. Para mantener la alineación se introducirán bytes de relleno cuando sea necesario. Considérese la siguiente estructura:

```
struct CD
{
char c;
double d;
};
```

Esta estructura contiene un carácter, el cual puede ocupar cualquier posición dentro del flujo de transmisión, seguido por un valor de tipo

ALINEACIÓN	TIPOS IDL
1	char, octet, boolean
2	short, unsigned short
4	long, unsigned long, float, tipos enumerados
8	long long, unsigned long long, double, long double

Figura 2.21:

double, que deberá ser alineado con límites de 8 bytes. Si la estructura CD tuviera que ser transmitida como el único parámetro de un método, ocuparía 16 bytes. (ver figura 2.22).

Hay que indicar que el tamaño de la estructura podrá variar según la posición que ocupe con respecto al resto de los datos con los que comparta el flujo de transmisión.

```
interface Objeto
```

```
{
void metodo(in string cadena, in CD estructura);
};
```

Cuando un cliente empaquete una invocación al método mostrado en el código de ejemplo anterior, enviará todos los parámetros in de acuerdo con las reglas de codificación CDR. Si se supone que el string contiene la cadena “Perro” y que los parámetros comenzarán con un offset alineado a 8 bytes se obtendrá la siguiente codificación (ver figura 2.23):

La codificación del valor “Perro” consumirá 10 bytes: 4 bytes para la longitud y 6 para almacenar la cadena. El segundo parámetro del método será una estructura de tipo CD. Puesto que el dato miembro c es un carácter y puede ser alineado en cualquier byte se situará a continuación del final del string. No sucederá lo mismo para el d, ya que al ser un dato de tipo double se alinea a 8 bytes, con lo que no podrá empezar hasta el byte 16, teniendo entonces que introducir 5 bytes de relleno.



Figura 2.22:

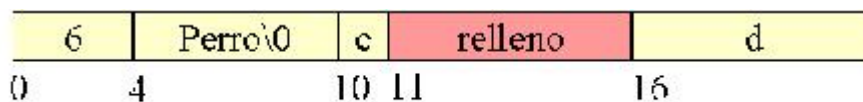


Figura 2.23:

En este caso la codificación de la estructura CD ocupa 13 bytes, mientras que en el ejemplo anterior ocupaba 16. Con esto queda claro que el tamaño de la codificación de un conjunto de datos dependerá de la posición que los mismos ocupen dentro del flujo de transmisión.

Codificación de uniones:

Las uniones en CDR se codifican como el valor del discriminante seguido de la codificación del campo activo en el momento de transmisión.

Codificación de arrays:

Como los arrays son de longitud fija, la codificación CDR se limitará a la codificación de cada uno de los elementos del vector.

Codificación de secuencias:

La secuencia se codifica como un **long** que representa el tamaño, seguido de los elementos que la forman.

Codificación de enumeraciones:

Los valores de una enumeración son codificados como **long**. Los valores numéricos asociados a los identificadores de la enumeración se determinan por el orden en el cual aparecen en la declaración de la enumeración. Al primer identificador de la enumeración le corresponderá el valor cero, y los siguientes se numerarán sucesivamente de izquierda a derecha.

2.18. Tipos de ORB's

Aunque hay unos 15 ORB's reconocidos por la OMG vamos a mostrar algunos de los más usados:

ORB's

- ORB MICO:

Mico es un ORB educacional que no necesita equipos de grandes prestaciones. Ha sido desarrollado por la Universidad de Frankfurt y aunque su rendimiento no es muy bueno es muy interesante para empezar.

www.mico.org

- ORB BACUS:

Es un ORB de libre distribución para fines no comerciales. Su rendimiento es bastante bueno. Ya está actualizado a la norma CORBA 2.3. Ha superado sus pequeñas incompatibilidades en el mapeado a C++. También permite el mapeado a Java. Realmente es un ORB muy a tener en cuenta.

www.occ.com/ob

ORB's para tiempo real

- ORB TAO:

TAO es un ORB desarrollado en la Universidad de Washington por Douglas Schmidt y su equipo. Está considerado hoy por hoy como el ORB con mejores prestaciones. Permite la creación de servidores multihilo y tiene servicios para tiempo real.

www.cs.wustl.edu/schmidt/TAO.html

- ORBexpress:

El ORBexpress es un ORB desarrollado por Object Interface Systems. Es ideal para aplicaciones empotradas. Permite la creación de servidores multihilo, incluido el thread pool.

www.ois.com

Parte II

Desarrollo

Capítulo 3

Desarrollo Wireless

3.1. Requisitos

- El host y el robot móvil deben estar conectados inalámbricamente.
- La distancia operativa de transmisión debe ser mayor de 20 metros.
- La comunicación inalámbrica será tal que permita la transmisión de imágenes de vídeo desde el robot, así como la de los diversos sensores.

3.2. Tecnología aplicada

Las tecnologías barajadas para el desarrollo de la red inalámbrica han sido: 802.11b, Bluetooth, HomeRF , IrDA y Ultra Wide Band.

La tecnología escogida para el desarrollo de la red inalámbrica del cliente es la 802.11b del IEEE. Esta tecnología cumple los requisitos de flexibilidad, dimensionamiento, prestaciones, seguridad y costes. Las otras tecnologías estudiadas han sido rechazadas por los siguientes motivos:

Bluetooth:

Los motivos principales para la desestimación de esta tecnología fueron la escasa velocidad (1Mbps) y su corto alcance (10m). A favor de esta tecnología cabe citar la amplia acogida en el mercado (Bluetooth agrupa a 1200 miembros en su SIG) y el bajo coste en relación a las demás alternativas. Además se están desarrollando puntos de acceso que permitirán ofrecer coberturas de hasta 100m y se está trabajando para que la versión 2 de este estándar soporte transmisiones a 4 Mbps.

HomeRF:

Los motivos principales para la desestimación de esta tecnología fueron su escasa velocidad (2 Mbps) y el enfoque de dirigir los productos al hogar.

IrDA:

Tiene las desventajas de ser point-to-point, muy direccional y no atravesar materiales opacos.

Ultra Wide Band:

Esta tecnología es la más avanzada actualmente, disponiendo de las velocidades de transmisión más altas. A pesar de ello, todavía es una tecnología muy reciente.

Si esta nueva tecnología es aprobada por la FCC (existe la preocupación de que interfiera las actuales transmisiones de GPS, tráfico aéreo y televisión) podría conducir al desarrollo de una tecnología muy superior a las LAN inalámbricas del estándar 802.11b. UWB ofrece menor costo, un ancho de banda mayor y un mayor soporte al usuario que la 802.11b y Bluetooth. La señal de UWB no emplea portadora de señal, su consumo es bastante pequeño, en torno a los 50 a 70 mW.

Las características ofrecidas 802.11b permiten, no solo cumplir con las expectativas del cliente, sino disponer de prestaciones adicionales. A continuación vinculamos las necesidades del cliente con las características disponibles en 802.11b.

- Velocidad de transmisión de 11 Mbps, equiparable a redes convencionales de cable, que permitirá en la zona de almacenaje actualizar de manera rápida los datos de inventario, y en las oficinas y sala de juntas disponer de los datos actualizados en cualquier momento.
- Rango de cobertura de 100m, que proporcionará conectividad a la red en cualquier punto del edificio.
- Topología de estrella, que permitirá la administración de la red desde un único punto (al igual que los hubs en las redes cableadas), además de minimizar el impacto derivado del fallo de un equipo de un usuario final.
- Roaming o itinerancia, que permitirá estar continuamente enganchado a la red en cualquier desplazamiento por el interior del edificio.
- Seguridad en las comunicaciones, maximizando la confidencialidad de los datos de la compañía que viajan por la red.
- Interoperabilidad con redes ethernet, que permitirá el acceso a los recursos de la red Fast Ethernet ya existentes, sin suponer un costo adicional en equipamiento de interconexión.
- Bajo consumo, que concede mayor autonomía a los equipos portátiles.

- Soporte de voz y vídeo, pese a no ser una necesidad del cliente, éste puede verse beneficiado por una instalación de este tipo.

	Bluetooth	HomeRF	802.11b
Características técnicas			
Espectro	2.4 GHz	2.4 GHz	2.4 GHz
Velocidad máx.	1 Mbps	2 Mbps	11 Mbps
Alcance (entornos de oficina)	~10 m	~100 m	~100 m
Redes	PAN	WLAN	WLAN
Voz y datos	Sí	Sí	Sí
Seguridad	Sí	Sí	Sí
Topologías	Punto a punto	Punto a punto y estrella	Punto a punto y estrella
Selección de frecuencias.	FHSS	FHSS	DSSS
Otras características			
Tipo de red	PAN	WLAN	WLAN
Órgano estandarizador	Bluetooth SIG	HomeRF Working Group	IEEE
Disponibilidad de productos	Media	Media	Alta
Coste	Medio	Medio	Alto

Figura 3.1: Características de las distintas tecnologías

3.3. Elección del producto

Cuatro han sido las soluciones estudiadas para este proyecto. Los productos analizados fueron: Airconnect de 3COM, WL de Compaq, 2011 de Intel y Aeronet de CISCO. La siguiente tabla resume sus características:

Finalmente, la solución escogida fue la de Compaq, basándonos en cinco características fundamentales:

Coste medio:

Inferior a la de 3Com, 2011 de Intel y Aeronet de cisco.

Garantía de los productos:

Igual a la de Intel y 3Com y superior a la de Cisco.

Disponibilidad de productos:

Puntos de acceso, tarjetas PC Card para portátiles, tarjetas PCI y dispositivos USB para equipos de sobremesa (Intel no disponía de tarjetas PCI). Además Compaq tiene presencia en España a través Hp con las consiguientes ventajas de proximidad e idioma.

Prestaciones:

Similares a las de las demás soluciones.

Experiencia de la compañía:

Compaq como una de las empresas líderes en equipamiento de red.

Se ha dedicado mucho tiempo al estudio de las diversas alternativas para disponer de una buena comunicación inalámbrica. Se ha decidido usar una wireless PC-Card WL110 de Compaq . Esta tarjeta PCMCIA necesita de un adaptador para conectarla al bus PC104 del Wafer 5820.

Conjuntamente con las tarjetas PCMCIA se adquirió un Access Point de Compaq. Este permite comunicar la red inalámbrica formada por las tarjetas PCMCIA con la red LAN del laboratorio.

Las pruebas realizadas demuestran que si bien la distancia operativa es aceptable, la velocidad de transmisión no alcanza los niveles esperados.

	3COM AirConnect	Compaq Wireless	Intel Wireless 2011	CISCO Aeronet
Distribuidor	3COM	Compaq	Intel	CISCO
Especificación	IEEE 202.11b	IEEE 202.11b	IEEE 202.11b	IEEE 202.11b
Certificación Wi-Fi	Sí	Sí	Sí	Sí
Soluciones disponibles				
Punto de Acceso	3CRWE74796B	WL410	WEAP2011EU	AIR-AP552E2C
Adaptador PC Card	3CRWE73796B	WL110	WPC2011EU	AIR-PCMB52
Adaptador PCI	3CRWE777A	WL210	N.D.	AIR-PCI352
Características de Radio				
Banda de Frecuencias	ICM	ICM	ICM	ICM
Tipo de Señal	DSSS	DSSS	DSSS	DSSS
Tasa de transferencia				
Ajuste dinámico	Sí	Sí	Sí	Sí
11 Mbps	Sí	Sí	Sí	Sí
5.5 Mbps	Sí	Sí	Sí	Sí
2 Mbps	Sí	Sí	Sí	Sí
1 Mbps	Sí	Sí	Sí	Sí
Modo de transmisión	Half duplex	Half duplex	Half duplex	Half duplex
Acceso al medio	CSMA/CA	CSMA/CA	CSMA/CA	CSMA/CA
Modulación				
CCK a 5.5 y 11 Mbps	Sí	Sí	Sí	Sí
DQPSK a 2 Mbps	Sí	Sí	Sí	Sí
DBPSK a 1 Mbps	Sí	Sí	Sí	Sí
Topologías				
Punto a punto	Sí	Sí	Sí	Sí
Punto de acceso	Sí	Sí	Sí	Sí
Encriptación				
WEP (40 bits)	Sí	Sí	Sí	No
WEP (128 bits)	Sí	Sí	Sí	Sí
RC4	Sí	Sí	N.D.	N.D.
Software				
Punto de acceso	Site Survey Dynamic Access	Site Survey APManager	Site Survey	Site Survey
Cliente	Sí	Sí	Sí	Sí
Reglamentaciones				
Certif. FCC Clase B	Sí	Sí	Sí	Sí
CSA	Sí	Sí	Sí	Sí
Marca CE	Sí	Sí	Sí	Sí
Garantía	3 años	1 año (AP) y 3 años (PC Card y PCI)	3 años (AP) e ilimitada (PC Card)	1 año (AP) y 3 años (PC Card y PCI).
Sistemas Operativos				
Windows 9x	Sí	Sí	Sí	Sí
Windows NT 4.0	Sí	Sí	Sí	Sí
Windows 2000	Sí	Sí	Sí	Sí
Características del punto de acceso				
Interfaz de red	10BASE-T(RJ45)	10BASE-T(RJ45)	10BASE-T(RJ45)	10/100BASE-T(RJ45)
Puerto serie RS-232	Sí	No	Sí	Sí
Nº máx. de clientes	64	N.D.	127	N.D.
Diversidad de antena	Sí	Sí	Sí	Sí
Transformador	Externo	Externo	Externo	No
Alimentación a través de línea Ethernet	Sí	Sí	Opcional	Sí
PVP (ptas)				
Punto de acceso	153.750	130.268	158.223	146.741
Adaptador PC Card	30.750	29.631	33.586	48.537
Adaptador PCI	38.727	37.086	N.D.	51.171

Figura 3.2: Soluciones estudiadas

La distancia máxima a la que se ha podido mantener una comunicación entre la tarjeta y el access point es de aproximadamente 50 metros, a través de los gruesos muros exteriores del laboratorio.

Respecto a la velocidad de transmisión, si bien la máxima especificada es de 11 Mbps, en laboratorio y bajo condiciones idóneas, solo hemos obtenido una velocidad de 5.5 Mbps, con picos de 8 Mbps.

Una de las principales razones de la elección de esta tarjeta a parte de las enumeradas anteriormente ha sido la garantía de que existen drivers para Linux, sistema operativo que será empleado en el Wafer 5820.

Se ha utilizado una antena que se adapta a la compaq wl110 para ampli-



Figura 3.3: Compaq WL110



Figura 3.4: Compaq WL410

ficar la señal obtenida, pero los resultados obtenidos no fueron los deseados. La señal no mejoró prácticamente nada.

Existen antenas caseras, fabricadas por grupos que se dedican o tienen como afición las comunicaciones inalámbricas, las cuales han alcanzado hasta 9Km. de distancia. Son las antenas de Pringles.



Figura 3.5: Antena amplificadora de señal



Figura 3.6: Antena Pringles

3.4. Instalación y configuración

3.4.1. Access point

Instalación:

Un Punto de Acceso, o estación base, es un receptor de radio y transmisor que se conecta a tu red cableada Ethernet. A través de este dispositivo nodos inalámbricos, como un portátil equipado con una tarjeta wl110 inalámbrica, tienen acceso a servicios de red LAN cableada como e-mail, Internet, impresoras y más. El Rango Operativo, administración de capacidades, seguridad de red inalámbrica y número de usuarios soportados son determinados por las capacidades del Punto de Acceso.

Consta de una conexión RJ45 donde le conectamos un cable de red y el otro extremo es conectado a un hub, dicho hub a su vez está conectado o forma parte de una red alámbrica (LAN) de tal forma que todos los nodos inalámbricos entren a formar parte de la propia LAN, como si no existiera una comunicación inalámbrica.

Configuración:

El access point wl410 usado para nuestro proyecto no depende del sistema operativo utilizado, no necesita configuración alguna.

3.4.2. Tarjeta pcmcia

Instalación:

El objetivo del proyecto es que podamos comunicar un PC con el robot móvil, pero también que podamos dirigir el robot desde un portátil, de modo que tanto el robot como el portátil deben estar equipados con una tarjeta inalámbrica wl110.

El portátil ya viene equipado con una ranura pcmcia, en la cual se inserta la tarjeta wl110 y ya está instalada.

El robot consta de un wafer 8520, el cual por si mismo no tiene ranura pcmcia para instalar la tarjeta inalámbrica. Así pues hay que dotarlo de un adaptador para pcmcia. Una vez que el adaptador ha sido instalado se inserta la tarjeta wl110 igual que si fuera un portátil.

Configuración:

La configuración de la tarjeta depende del sistema operativo utilizado. En el proyecto se ha usado el sistema operativo Linux en concreto la distribución del Red Hat 7.3.

Al introducir la tarjeta inalámbrica en la ranura pcmcia el Red Hat 7.3 la detecta directamente sin hacer nada más, pues ya viene con el driver para la compaq wl110 por defecto. En otras distribuciones habría que instalar dicho driver.

Una vez insertada la tarjeta y detectada por el sistema operativo tenemos que configurarla, para ello vamos al archivo:

```
/etc/pcmcia/network.opts
```

y editamos en dicho archivo:

- La dirección IP (IPADDR).
- La máscara de red (NETMASK).
- La puerta de enlace (GATEWAY).
- Las direcciones DNS (DNS_1 y DNS_2)

Por ejemplo:

```
*,*,*,*)
```

```
IF_PORT=""
```

```
BOOTP="n"
```

```
IPADDR="dirección IP"
```

```
NETMASK="255.255.255.0"
```

```
NETWORK="por defecto la que viene"
```

```
BROADCAST="por defecto la que viene"
```

```
GATEWAY="puerta de enlace"
```

```
DOMAIN="por defecto la que viene"
```

```
DNS_1="DNS 1"
```

```
DNS_2="DNS 2"
```

```
::
```

Con esto la tarjeta ya funciona.

Esta tarjeta también la hemos utilizado con el sistema operativo windows xp, el cual la detecta también como el red Hat 7.3 y solo hay que configurarla como cualquier otra tarjeta de red normal.

Capítulo 4

Desarrollo Corba

4.1. Requisitos

Tres son los requisitos fundamentales para el desarrollo de nuestro proyecto:

- La arquitectura elegida debe poder soportar el desarrollo de aplicaciones distribuidas.
Las aplicaciones distribuidas son más fáciles de comprender, implementar, depurar y mantener.
- El sistema operativo puede ser cualquiera.
- Debe haber una independencia en el lenguaje de programación utilizado.

4.2. Tecnología aplicada

Las tecnologías barajadas han sido: DCOM de Microsoft, RMI de Sun y CORBA de la OMG.

Las tres arquitecturas cumplen el requisito de poder soportar el desarrollo de aplicaciones distribuidas.

CORBA será la tecnología elegida por disponer de una interfaz de programación independiente de las plataformas usadas.

CORBA ya ha sido mencionado en el capítulo 2, a continuación mostraremos desventajas de las otras dos tecnologías que muestran por sí solas los motivos por las cuales han sido rechazadas.

RMI:

- No provee un mecanismo de meta-información y auto-descripción. Los clientes no pueden “despertar” en un ambiente que se describa a sí mismo.
- Sólo soporta un lenguaje, Java, lo que hace difícil la integración de software ya construido.

- Los objetos son pasados por valor, lo cual hace que el sistema no sea escalable a medida que el tamaño de los objetos aumenta (el sistema se hace más ineficiente en el tiempo).
- No existe protocolo de paso por *firewall*.
- No favorece la programación totalmente distribuida, sino que los roles cliente y servidor deben ser bien establecidos al principio.

DCOM:

- Su configuración es una auténtica pesadilla.
- No es escalable.
- No se basa en un modelo de objetos clásico.
- Sólo soporta herramientas específicas de Microsoft. Está ligado a entornos Windows.

A continuación se muestra una tabla con las características que van desde los sockets a los objetos distribuidos.

Característica	CORBA	DCOM	RMI	HTTP/CGI	Sockets ^a
Nivel de abstracción	★★★★	★★★★	★★★★	★★	★
Integración con Java	★★★★	★★★	★★★★	★★	★★
Soporte multiplataforma	★★★★	★★	★★★★	★★★★	★★★★
Implementación total en Java	★★★★	★	★★★★	★★★★	★★★★
Soporte de tipos	★★★★	★★★★	★★★★	★	★
Facilidad de configuración	★★★	☆	★★★	★★★	★★★
Invoc. de métodos distribuida	★★★★	★★★	★★★	☆	☆
Estado entre invocaciones	★★★★	★★★	★★★	☆	★★
Meta-información	★★★★	★★★	☆	☆	☆
Invocaciones dinámicas	★★★★	★★★★	★	☆	☆
Rendimiento (<i>ping</i>)	★★★★	★★★	★★★	☆	★★★★
	3.3 ms	3.9 ms	5.5 ms	603.8 ms	2.0 ms
Seguridad	★★★★	★★★★	★★★	★★★	★★★
Transacciones	★★★★	★★★	☆	☆	☆
Objetos persistentes	★★★★	★	☆	☆	☆
Soporte multilenguaje	★★★★	★★★	☆	★★★	★★★★
Protocolo común multilenguaje	★★★★	★★★★	☆	★★★★	☆
Escalabilidad	★★★★	★	★	★★	★★★★
Estándar abierto	★★★★	★★	★★	★★★★	★★★★

^a★★★★ = mejor; ★ = peor; ☆ = N/A, muy mala o inexistente.

Figura 4.1: Características CORBA/RMI/DCOM

4.3. Configuración del ORB

Como ORB elegimos **TAO**, porque permite la creación de servidores multihilo y sobre todo porque tiene servicios para tiempo real que sería el próximo paso a dar después de lo descrito en este proyecto.

4.3.1. Configuración sobre plataforma Linux

A continuación escribiremos los pasos necesarios para compilar el ORB sobre linux que es el sistema operativo que vamos a utilizar.

1. Bajamos las fuentes ACE+TAO, la versión ACE-5.2+TAO-1.2.tar, por ejemplo de la página:

```
http://deuce.doc.wustl.edu/Download.html
```

2. Descomprimos dichas fuentes, dentro del directorio /usr/ace.

```
# cd /usr
```

```
# mk dir ace
```

```
# cd ace
```

```
# tar xvzf ACE-5.2+TAO-1.2.tar
```

3. Dentro del directorio /etc/profile incluimos lo siguiente:

- Añadimos las variables de entorno ACE_ROOT y TAO_ROOT que apunten al directorio donde se han descomprimido las fuentes. Ej:
export ACE_ROOT=/usr/ace/ACE_wrappers
export TAO_ROOT=/usr/ace/ACE_wrappers/TAO
- Añadimos la variable de entorno LD_LIBRARY_PATH para poder usar las librerías dinámicas durante la compilación:
export LD_LIBRARY_PATH=\$ACE_ROOT/ace:\$LD_LIBRARY_PATH
- Añadimos lo siguiente para que podamos utilizar el IDL desde cualquier directorio.
export PATH=\$TAO_ROOT/TAO_IDL:\$PATH

Se pone dentro del directorio /etc/profile para no tener que escribir lo mismo cada vez que iniciamos sesión.

4. Reiniciamos el sistema.
5. Tocar el fichero `$ACE_ROOT/ace/OS.h` si queremos cambiar el host por defecto, puertos, etc. que utilizarán las aplicaciones. (Esto es opcional).
6. Crear enlaces que apunten desde `$ACE_ROOT/ace/config.h` y desde `$ACE_ROOT/include/makeinclude/platform_macros.GNU` a los ficheros que contengan la configuración adecuada para la plataforma sobre la que estamos compilando:
 - Dentro del directorio `$ACE_ROOT/ace/config.h` ponemos:
`ln -s config-linux.h config.h`
 - Dentro del directorio `$ACE_ROOT/include/makeinclude/platform_macros.GNU` ponemos:
`ln -s platform_linux.GNU platform_macros.GNU`
7. Hacer make en los siguientes directorios:
`$ACE_ROOT/ace`
`$ACE_ROOT/apps/gperf`
`$TAO_ROOT/tao`
`$TAO_ROOT/TAO_IDL`
`$TAO_ROOT/orbsvcs/orbsvcs`

Con lo anterior ya hemos compilado el ORB y podemos empezar a compilar nuestros programas.

4.3.2. Compilación sobre plataforma windows

Hay que tener en cuenta que utilizamos como compilador el Visual C++.

Windows 95, 98 y NT:

1. Fijar las variables de entorno:
Entramos en `autoexec.bat` en modo edición (no queremos abrirlo, queremos escribir en él).

- SET ACE_ROOT=C:/ace/ACE_wrappers
C:/ace/ es el directorio donde hemos metido el ACE_wrappers.
 - SET TAO_ROOT= %ACE_ROOT%/TAO
 - SET TAO_IDL_PREPROCESSOR= C:/Archivos de programa/Microsoft Visual Studio/VC98/Bin/cl.exe.
2. Modificar config.sys
shell=C:/WINDOWS/command.com/e:4096/p
C:/WINDOWS/command.com es la ruta completa al command.com
 3. Añadir al Visual C++.
Una vez dentro del Visual C++, abrimos la ventana “tools”, después vamos a “options”, dentro de “options” vamos a “directories”, una vez aquí dentro de “Show directories for:” elegimos las opciones que describimos a continuación añadiendo lo siguiente:
 - Include files:
C:/ace/ACE_wrappers
C:/ace/ACE_wrappers/TAO
C:/ace/ACE_wrappers/TAO/TAO
 - Executables files:
C:/ace/ACE_wrappers/Bin
 - Library files:
C:/ace/ACE_wrappers/ACE
C:/ace/ACE_wrappers/TAO/TAO
C:/ace/ACE_wrappers/TAO/ORBSVCS/ORBSVCS
C:/ace/ACE_wrappers/TAO/TAO/PORTABLESERVER
 4. Librerías para el client y el server:
Entramos en “Project”, luego en “setting” y finalmente en “link” e incluimos lo siguiente:
aced.lib
taod.lib
TAO_PortableServerd.lib

Windows 2000 y XP:

1. Variable de entorno:
Vamos a “MI PC”, con el botón derecho pulsamos, vamos a “propiedades”, luego a “avanzado”, por último entramos en “Variables de entorno”, e introducimos las nuevas variable.
2. No hay que modificar el config.sys.

3. Añadir al Visual C++.

Añadimos lo mismo que en el caso del windows 95, 98 y NT.

4.4. Generación del IDL

Nuestro IDL se compone de una sola función, la cual tiene como objetivo que el robot se mueva para adelante, para atrás, a la derecha y a la izquierda.

Es una función muy simple, pero sirve de un primer paso para la construcción de las funciones que debe realizar el robot.

```
CL      /I %ACE_TAO_ROOT%\tao /I %ACE_TAO_ROOT% /I %ACE_TAO_ROOT%\orbsvcs /I
      %ACE_ROOT% /link /libpath:%ACE_TAO_ROOT%\tao;%ACE_ROOT%\ace

ACE_ROOT      C:\ace\ACE_Wrappers

ACE_TAO_ROOT      C:\ace\ACE_Wrappers\Tao

LIB      %ACE_ROOT%\ace; %LIB%; %ACE_ROOT%\TAO\tao\PortableServer

PATH      C:\Archivos de
      programa\MiKTeX\miktex\bin;%ACE_ROOT%\bin;%ACE_TAO_ROOT%
      \orbsvcs\orbsvcs;%ACE_ROOT%\bin\Release;%PATH%

TAO_IDL_PREPROCESSOR      C:\Archivos de programa\Microsoft Visual
      Studio\VC98\Bin\Cl.exe

TAO_ROOT      %ACE_TAO_ROOT%
```

Figura 4.2: Variables de entorno del windows 2000 y XP

```
interface robot
{
    void mover (in char letra);
};
```

Figura 4.3: Código IDL

4.5. Generación del cliente

El código del cliente tiene como objetivo construir un bucle abierto de manera que en cualquier momento podamos dar instrucciones para que el coche vaya para delante(pulsar tecla Q), para atrás(pulsar tecla A), para la derecha(pulsar tecla P) y para la izquierda(pulsar tecla O).

Teniendo en cuenta por supuesto la inicialización del orb y el NameService.

4.6. Generación de la implementación

El código implementación es un código del cual el servidor va a hacer uso, es decir, va a formar parte del servidor.

Este código se va a dividir en dos: una cabecera (llamada robot_impl.h) y un código fuente (llamado robot_impl.cpp).

El código de la implementación básicamente lo que hace es sacar por el puerto serie del ordenador que hace de servidor (en este caso el ordenador del robot) unas señales que el microprocesador, que también está en el robot, tiene que recibir.

Hay que tener en cuenta que nuestro proyecto es parte de otro más grande y en el proyecto “padre” es el micro el que hace que se mueva el servo, el que controla los sensores, etc.

4.7. Generación del servidor

En el servidor inicializamos el orb, el POA, definimos las políticas del POA y usamos el NameService.

```

#include "robotC.h"
#include <orbsvcs/orbsvcs/CosNamingC.h>

int main (int argc, char* argv[])
{
    try
    {
        // comprobar argumentos

        if (argc=3)
        {
            cerr<<"Use: client IOR_string"<<endl;
            throw 0;
        }

        CORBA::ORB_var orb = CORBA::ORB_init(argc, argv);

        CORBA::Object_var objNam =
        orb->resolve_initial_references("NameService");

        CosNaming::NamingContext_var r_context =
        CosNaming::NamingContext::_narrow(objNam);

        CosNaming::Name comp;
        comp.length(1);
        comp[0].id = CORBA::string_dup ("Robot");
        comp[0].kind = CORBA::string_dup ("");
        CORBA::Object_var nObj = r_context->resolve(comp);
        robot_var Server = robot::_narrow(nObj);
        try {
            for(;;)
            {
                cout<<"\n Pulsa las teclas q,a,p,o para moverte\n\n"<<endl;
                char capturar;
                cin >> capturar;
                Server->mover (capturar);
            }
        }
        catch (CORBA::Exception &e){
            cout << "Excepcion: " << e << endl;
        }
        catch (CORBA::Exception &e){
            cout << "Excepcion: " << e << endl;
        }
    }
}

```

Figura 4.4: Código del cliente

```

#ifndef ROBOT_IMPL_H
#define ROBOT_IMPL_H
#include "robotS.h"

class robot_impl: public POA_robot {
public:
    virtual void mover (
        CORBA::Char letra
    )
        ACE_THROW_SPEC ((
            CORBA::SystemException
        ));

private:
};

#endif

```

Figura 4.5: Código cabecera de la implementación

```

#include "robot_impl.h"

#include <sys/types.h> //
#include <sys/stat.h> //
#include <fcntl.h> // para el puerto serie
#include <unistd.h> //

// función mover

void robot_impl::mover (CORBA::Char letra) ACE_THROW_SPEC ((CORBA::SystemException))
{
    cout << "Recibida una " << letra << endl;
    int fd;
    cout << "Abriendo ttyS0 ..." << endl;
    fd=open("/dev/ttyS0", O_WRONLY|O_NONBLOCK);
    cout << "Abierto!" << endl;
    //if(1==read(0, (char*)&buffer, 1)) {
    if(letra=='a' || letra=='q' || letra=='o' || letra=='p') {
        write(fd, (char*)&letra, 1);
    }
    //}
    cout << "movimiento completado" << endl;
}

```

Figura 4.6: Código fuente de la implementación

```

#include <orbsvcs/orbsvcs/CosNamingC.h>
#include "robot_impl.h"

int main(int argc, char* argv[])
{
    robot_impl Servant;

    // Inicializacion del ORB

CORBA::ORB_var orb = CORBA::ORB_init(argc, argv);

    // Se toma una referencia al RootPOA

CORBA::Object_var obj = orb->resolve_initial_references("RootPOA");
PortableServer::POA_var rootPOA = PortableServer::POA::_narrow(obj.in());
PortableServer::POAManager_var poa_manager = rootPOA->the_POAManager();

    // Se toma una referencia al root_context del NameService

CORBA::Object_var objNaming = orb->resolve_initial_references("NameService");
CosNaming::NamingContext_var root_context;

    try
    {
        root_context = CosNaming::NamingContext::_narrow(objNaming.in());
    }

    catch (const CORBA::Exception&)
    {
        cout << endl << "|¡ATENCIÓN! No se encuentra el servidor de nombres" <<
endl << endl;
        exit(1);
    }

    // Creacion de politicas para nuestro POA persistente

CORBA::PolicyList policies;
policies.length(1);
policies[(CORBA::ULong)0] = rootPOA->
>create_lifespan_policy(PortableServer::PERSISTENT);

    // Creacion de myPOA con las politicas correctas

PortableServer::POA_var myPOA = rootPOA->
>create_POA("CorbaServer_agent_poa",poa_manager,policies);

    // Se fija el Id del Servant

```

Figura 4.7: 1ª parte del código del servidor

```
PortableServer::ObjectId_var managerId = PortableServer::string_to_ObjectId("CORBA

    // Activacion del Servant con el Id en myPOA

myPOA->activate_object_with_id(managerId, &Servant);

    // Activacion del POA manager

poa_manager->activate();

    // Asociacion de la implementacion con el nombre en el root context

CosNaming::Name comp;
comp.length(1);
comp[0].id = CORBA::string_dup("Robot");
comp[0].kind = CORBA::string_dup("");
root_context->rebind(comp, Servant._this());

    // Inicializamos el servidor CORBA

cout << "Servidor preparado y a la espera de peticiones ..." << endl;
orb->run();
}
```

Figura 4.8: 2ª parte del código del servidor

4.8. Generación del fichero Makefile

Como todo fichero Makefile contiene una serie de instrucciones de tal manera que no se tengan que escribir cada vez que arranquemos nuestra aplicación.

```

#-----
#      Makefile
#-----

ifndef TAO_ROOT
  TAO_ROOT = $(ACE_ROOT)/TAO
endif # !TAO_ROOT

LDLIBS = -ITAO_CosNaming -ITAO
CPPFLAGS += -I$(TAO_ROOT)/orbsvcs

IDL_FILES = robot
IDL_SRC = robotC.cpp robotS.cpp
BIN2 = Server Client

#### If the TAO orbsvcs library wasn't built with sufficient components,
#### don't try to build here.
TAO_ORBSVCS := $(shell sh $(ACE_ROOT)/bin/ace_components --orbsvcs)
ifeq (Naming,$(findstring Naming,$(TAO_ORBSVCS)))
  BIN = $(BIN2)
endif # Time

SRC = $(addsuffix .cpp, robotServ robot_impl Client) $(IDL_SRC)

SERVER_OBJS = $(addsuffix .o, robotServ robot_impl robotC robotS)

CLIENT_OBJS = $(addsuffix .o, Client robotC)

TAO_IDLFLAGS=-in

#-----
# Include macros and targets
#-----
include $(ACE_ROOT)/include/makeinclude/wrapper_macros.GNU
include $(ACE_ROOT)/include/makeinclude/macros.GNU
include $(TAO_ROOT)/rules.tao.GNU
include $(ACE_ROOT)/include/makeinclude/rules.common.GNU
include $(ACE_ROOT)/include/makeinclude/rules.nonested.GNU
include $(ACE_ROOT)/include/makeinclude/rules.local.GNU
include $(TAO_ROOT)/taoconfig.mk

#-----
#      Local targets
#-----

.PRECIOUS: $(foreach ext, $(IDL_EXT), robot$(ext))

Server: $(addprefix $(VDIR),$(SERVER_OBJS))
        $(LINK.cc) $(LDFLAGS) -o $@ $^ $(VLDLIBS) $(POSTLINK) $(SNTXLIBS) $(LSOCKET)

Client: $(addprefix $(VDIR),$(CLIENT_OBJS))
        $(LINK.cc) $(LDFLAGS) -o $@ $^ $(VLDLIBS) $(POSTLINK) $(SNTXLIBS) $(LSOCKET)

realclean: clean
        -$(RM) $(foreach ext, $(IDL_EXT), robot$(ext))

# DO NOT DELETE THIS LINE -- g++dep uses it.
# DO NOT PUT ANYTHING AFTER THIS LINE, IT WILL GO AWAY.

# IF YOU PUT ANYTHING HERE IT WILL GO AWAY

```

Figura 4.9: Código del Makefile

4.9. Breve manual de usuario

Una vez que ya hemos compilado nuestro ORB, y hemos escrito el código realizamos una serie de tareas para llevar a buen puerto nuestra aplicación:

1. Metemos todo el código en la misma carpeta, llamando a dicha carpeta con el nombre que queramos para nuestra aplicación.

2. Compilamos el fichero idl:

```
tao_idl robot.idl
```

3. Compilamos el cliente y el servidor con las siguientes instrucciones:

```
make Client
```

```
make Server
```

Al cliente le llamamos “Cliente” y al servidor “Server”, porque así lo hemos definido en el fichero Makefile.

4. Compilamos el servidor de nombres, para ello debemos ir al directorio:

```
/usr/ace/ACE_wrappers/TAO/orbsvcs/Naming_Service/
```

y ejecutar el comando:

```
make
```

5. Habilitamos el multi-cast. Este paso solo hay que hacerlo una vez en cada máquina que utilices, cada vez que arranques la aplicación no es necesario hacerlo otra vez.

En cualquier directorio podemos ejecutar lo siguiente:

```
route add -net 224.0.0.0 netmask 240.0.0.0 dev eth0
```

Si escribes “route” en la consola verás lo que has añadido.

6. Arrancamos el NameService. Nos vamos otra vez al directorio:

```
/usr/ace/ACE_wrappers/TAO/orbsvcs/Naming_Service/
```

y escribimos lo siguiente:

```
Naming_Service -ORBDebug Level 10 -ORBEndPoint iiop://localhost:90001
```

NOTA: el NamingServer solo es necesario arrancarlo en una máquina, no en todas las que utilizas. El “localhost” normalmente es la IP de la

¹si este comando o cualquier otro no funciona tal y como viene en el texto anteponer lo siguiente:“./”.

máquina y el puerto, en este caso “9000” es el número del puerto que tú quieras ponerle.

7. Arrancamos el servidor:

```
./Server -ORBInitRef NameService = iiop://localhost:9000/NameService
```

NOTA: el “localhost” es la IP de la máquina donde instalas el NameService y el puerto es el que le has puesto al arrancar el NameService, en este caso: “9000”.

8. Arrancamos el cliente:

```
./Client -ORBInitRef NameService = iiop://localhost:9000/NameService
```

NOTA: el “localhost” es la IP de la máquina donde instalas el NameService y el puerto es el que le has puesto al arrancar el NameService, en este caso: “9000”.

9. Una vez que ya corre nuestra aplicación, para abortarla se utiliza:

Control + C ó con el comando “kill”.

Por último señalar que hasta este punto el breve manual se ha escrito para realizar la aplicación en un solo ordenador, basta con abrir tres terminales. En el primero arrancas el NameService, en el segundo arrancas el servidor y en el tercero el cliente. En ese mismo orden.

Cuando utilizas varios ordenadores, las instrucciones son las mismas lo único que en vez de tres terminales utilizas un ordenador por cada cliente ó servidor que tenga la aplicación. Pero en cada ordenador hay que escribir en el directorio “etc/hosts” lo siguiente:

1. 127.0.0.1 localhost (viene por defecto).
2. IP dominio nombre del ordenador.
3. Además en ordenador en el que arranques el NameService² debes incluir la IP, dominio y nombre de todos los ordenadores que utilicen dicho NameService.

²el NameService se arranca solo una vez en cualquiera de los ordenadores que estés utilizando.

Parte III

Apéndices

Apéndice A

Glosario WIRELESS

- 10Base-T:** An IEEE standard for Ethernet networking over unshielded twisted pair (UTP) cable.
- 100Base-TX:** IEEE 802.3u standard for 100Mbps Fast Ethernet over unshielded twisted pair cable. 100Base-TX runs over 2-pair Category 5 cable.
- 10BASE-T-compliant:** Conforms to the IEEE 802.3 10BASE-T standard.
- Active star:** A network topology where the center of the star is a hub with network branch cables radiating outward.
- Adapter:** See Ethernet adapter.
- Address:** A unique numeric identifier of a node on a local area network.
- AppleTalk:** Protocols that allow Macintosh computers to communicate with printers, file servers, and other devices.
- Auto-negotiation:** Feature that allows Fast Ethernet cards, hubs and switches to automatically determine the maximum speed of a connection between devices. Auto-negotiation will select the fastest speed (10Mb half-duplex, 10Mb full-duplex, 100Mb half-duplex, 100Mb full-duplex) common to both devices on each end of a connection.
- Backbone:** A network topology consisting of a length of cable with multiple network connection points.
- Bandwidth:** Based on the transmission speed of the network. It defines the raw data throughput theoretically possible over the network to be shared with other users within the same collision domain. For example, 10Base-T provides 10Mbps of bandwidth.
- Bridge:** A device that links two or more similar networks forming a single, larger network. Bridges isolate network traffic by monitoring the destination network address within an individual data packet to if that packet should be passed onto the other side of the network. Bridges are not protocol dependent, therefore they do not care what type of data is actually inside the packet.
- Coaxial:** One of the types of cabling used in Ethernet networks. Inside its outer insulator, a coaxial cable has two concentric conductors separated by a layer of insulation.

Collapsed backbone: When a single hub or switch is used to connect multiple hubs and switches together via unshielded twisted pair cabling.

Collision: A situation that occurs when two devices on a network try to transmit at the same time and their transmissions bump into each other. When a collision occurs, both sets of data must be retransmitted. Retransmission is handled automatically by the Ethernet devices.

Collision domain: Collision domain A single collision domain consists of one or more Ethernet hubs and nodes connected together. Each device within the collision domain shares the available network bandwidth with the other devices in the same domain. Switches and bridges are used to separate collision domains that are too large in order to improve performance and network reliability.

CSMA/CD: An acronym that stands for Carrier Sense Multiple Access/Collision Domain. It defines how an Ethernet network works by detecting collisions after they have occurred and automatically correcting for them.

Daisy-chain: A network topology where all devices are connected serially (one, to the next, to the next) to each other.

Device: Electronic equipment used on a network, such as a hub, router, workstation, or printer.

Driver: A program that controls the network hardware or implements the protocol stacks through which higher level applications communicate with the network hardware.

Ethernet: IEEE 802.3 standard defining a type of local area network characterized by a 10-megabit per second (Mbps) data rate. Ethernet networks can run over different types of cabling. The most popular is 10Base-T Ethernet which uses unshielded twisted-pair cabling.

Ethernet adapter: A component installed in a device that enables the device to connect to, and communicate on, an Ethernet network; usually an Ethernet interface card with Ethernet driver software.

Fast Ethernet: IEEE 802.3u standard defining a type of local area network characterized by a 100-megabit per second (Mbps) data rate. Fast Ethernet networks can run over different types of cabling. The most popular is 100Base-TX Fast Ethernet which uses category 5 unshielded twisted-pair cabling.

Fiber Optic cable: Special cable that uses light, instead of electrical signals, to transmit network data at high speeds and over longer distances.

Filtering: The process that a switch or bridge uses to examine an Ethernet packet and either pass it to the other side of the network or isolate it from the rest of the network. (See the definition for a Bridge.)

Full Duplex: When two network devices, such as a card and a switch, can simultaneously send and receive data effectively doubling the bandwidth of a connection.

Gigabit Ethernet: A new standard for transmitting Ethernet packets at 1000Mbps.

Hop count: From the IEEE 802.3 Ethernet standard, the hop count for an Ethernet network is defined by the number of hubs an Ethernet signal must pass through to get to its final destination. 10Base-T rules limit the maximum number of hops to four hubs, while Fast Ethernet limits the number of hops to two.

Hub: A multi-port repeater; the center of an active star network.

IEEE (Institute of Electrical and Electronic Engineers): An international professional society that issues its own standards related to electrical and electronic devices.

IEEE 802: A committee responsible for setting standards for LANs.

IEEE 802.X: A set of specifications for Local Area Networks (LAN) from The Institute of Electrical and Electronic Engineers (IEEE). Most wired networks conform to 802.3, the specification for CSMA/CD Ethernet networks. The 802.11 committee completed a standard for 1 and 2 Mbps wireless LANs in 1997 that has a single MAC layer for the following physical-layer technologies: Frequency Hopping Spread Spectrum, Direct Sequence Spread Spectrum, and Infrared. IEEE 802.11 HR, an 11 Mbps version of the standard is expected to be completed by the end of 1999.

Independent network: A network that provides (usually temporarily) peer-to-peer connectivity without relying on a complete network infrastructure.

- Infrastructure network:** A wireless network centered around an access point. In this environment, the access point not only provides communication with the wired network but also mediates wireless network traffic in the immediate neighborhood.
- LAN (Local Area Network):** Workstations and shared devices connected in a limited area, usually a single building.
- LocalTalk:** The cabling specification for AppleTalk running at a speed of 230.4 kilobits per second.
- Mbps (Megabits per second):** A measure of the rate of data transmission equal to a million bits per second.
- Microcell:** A bounded physical space in which a number of wireless devices can communicate. Because it is possible to have overlapping cells as well as isolated cells, the boundaries of the cell are established by some rule or convention.
- Multipath:** The signal variation caused when radio signals take multiple paths from transmitter to receiver.
- Network administrator:** A person who coordinates the design, installation and management of a network. A network administrator is also responsible for adding new users to the network and troubleshooting.
- Network topology:** A physical arrangement of devices and cable in a network.
- Node:** A single, addressable device on a network. Computers, networked printers, and routers are nodes.
- Packet:** A group of digital information (known as bits) that has been formatted together to be transmitted across a network .
- Port:** An access point for data to enter, exit, or be repeated across a network; a hardware receptacle on a network device, such as a socket or jack, into which you can plug a connector.
- Protocol:** A language or set of rules used for network communication. Popular protocols used in most networks include TCP/IP, IPX and AppleTalk.

Puntos de Acceso: Un Punto de Acceso, o estación base, es un receptor de radio y transmisor que conecta a tu red cableada Ethernet. A través de este dispositivo nodos inalámbricos, como el PowerBooks de Macintosh o notebooks equipados con una tarjeta Farallon SkyLINE Inalámbrica, tienen acceso a servicios de red LAN cableada como e-mail, Internet, impresoras y más. El Rango Operativo, administración de capacidades, seguridad de red inalámbrica y número de usuarios soportados son determinados por las capacidades del Punto de Acceso.

Puntos de Acceso “Flexible”: Como una alternativa a destacar un Punto de Acceso para conectividad inalámbrica a una red cableada Ethernet, un computador que está físicamente conectado a una red Ethernet, equipado con una tarjeta inalámbrica como la SkyLINE, y corriendo un software de solución de ruta como el Vicomsoft Internet Gateway (www.vicomsoft.com), puede actuar como el gateway entre la red cableada y la red inalámbrica.

Radio Frequency (RF) Terms: GHz, MHz, Hz: The international unit for measuring frequency is Hertz (Hz), which is equivalent to the older unit of cycles per second. One Megahertz (MHz) is one million Hertz. One Gigahertz (GHz) is one billion Hertz. For reference: the standard US electrical power frequency is 60 Hz, the AM broadcast radio frequency band is 0.55 -1.6 MHz, the FM broadcast radio frequency band is 88-108 MHz, and microwave ovens typically operate at 2.45 GHz.

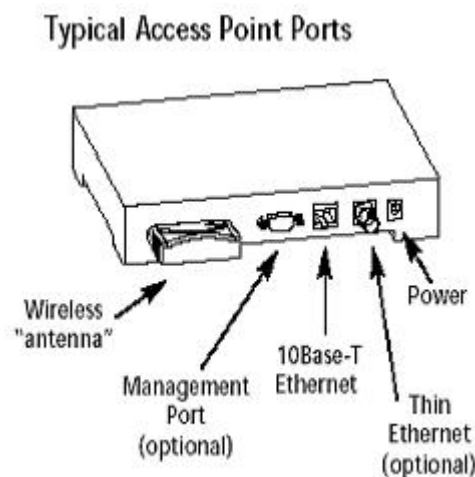


Figura A.1: Puntos de acceso

Rango Operativo: Factores que afectan el rango operativo de cualquier dispositivo inalámbrico incluyen la potencia del Punto de Acceso, el número de murallas dentro del edificio, los materiales de construcción usados dentro del edificio (concreto vs. acero vs. madera), la velocidad de transmisión de datos, y hasta las personas. La mayoría de los fabricantes de Puntos de Acceso ofrecen antenas mejoradas para un rango aumentado. Los fabricantes recomiendan que el Punto de Acceso sea desplegado 50m aparte para asegurar una cobertura completa y máximas tasas de transmisión de datos para usuarios de computadores que transitan. Adicionalmente, un grado menor en velocidad de transmisión de datos aumentará el rango operativo. Entonces, un cliente inalámbrico comunicándose a 2Mbps tendrá mejor rango que un cliente a 11Mbps. Puntos de Acceso pueden ser usados para extender el rango de una red inalámbrica hasta en situaciones donde no hay necesidad de conectar atrás a una red cableada Ethernet. Desplegando un Punto de Acceso como un “hub inalámbrico” la distancia entre dos computadores remotos puede ser extendida a 100m (50 metros desde cada computador al Punto de Acceso) dependiendo de la configuración del edificio y de la ubicación de los Puntos de Acceso.

Repeater: A device that rebroadcasts a network signal, allowing it to travel for longer distances. Repeaters have no effect on the speed of a network signal. Hubs are multi-port repeaters.

RJ-45: A telephone-industry connector type usually containing eight pins.

Router: A device that connects two or more networks; a router receives data from other network devices and retransmits the data to its proper destination over the most efficient path from one network to another.

Star topology: A network design where several network branches are connected at one central location.

Switch: An advanced type of Ethernet, Fast Ethernet or Gigabit Ethernet hub that uses the destination address of a data packet to intelligently direct network traffic to specific devices on the network, thus optimizing overall network performance.

Throughput: The actual speed in which data is transmitted over the network. In some cases, this may be significantly lower than the bandwidth available.

Topology: See network topology.

Transceiver: Hardware that transmits and receives network signals. Every device on an Ethernet network needs a transceiver. Some Ethernet transceivers are integrated, such as the transceiver circuitry built into the Ethernet cards; others are external, such as the Farallon's EtherMac Transceiver.

Twisted-pair: Cable type as specified by 10BASE-T in IEEE 802.3; 8-wire telephone cable using RJ-45 connectors.

Uplink: The auxiliary port of an Ethernet hub or switch which can be used for connecting to a larger Ethernet network.

WAN (Wide Area Network): Two or more LANs in separate geographic locations that are connected using long distance communications methods, such as telephone lines and satellites.

Wiring closet: A central location where a building's telephone and network wiring is connected. Telephone lines in a wiring closet branch out to individual extensions. Multi-story buildings with ten floors have a main wiring closet in the basement and satellite wiring closets on each floor.

Apéndice B

Glosario CORBA

Accountability: Users are accountable for their security-relevant actions. A particular case of this is non-repudiation, where responsibility for an action cannot be denied.

Administrador de Consumidores o ConsumerAdmin: es un gestor que se encarga de añadir Consumidores a un canal de eventos o de destruir dicho canal.

Administrador de Suministradores o SupplierAdmin: es un gestor que se encarga de añadir Suministradores a un canal de eventos o de destruir dicho canal.

Administration: of security information (for example, security policy) is also needed. This visible security functionality uses other security functionality such as *cryptography*, which is used in support of many of the other functions but is not visible outside the Security services. No direct use of cryptography by application objects is proposed in this specification, nor are any cryptographic interfaces defined.

Authorization and access control: deciding whether a principal can access an object, normally using the identity and/or other privilege attributes of the principal (such as role, groups, and clearance) and the control attributes of the target object (stating which principals, or principals with which attributes) can access it.

Availability: Use of the system cannot be maliciously denied to authorized users. Availability is often the responsibility of other OMA components such as archive/restore services, or of underlying network or operating systems services. Therefore, this specification does not address all availability requirements.

Cliente: una aplicación que puede pedir acceso a los métodos de un objeto remoto, o también el rol que desempeñan los objetos que realizan peticiones.

Concurrency Service: métodos para soportar el acceso concurrente de varias aplicaciones o procesos al mismo recurso. Se emplea junto al anterior servicio.

Confidentiality: Information is disclosed only to users authorized to access it.

Consumer o Consumidor: aquella entidad que recibe el evento.

Datastore: el medio de almacenamiento.

Emisor: aquel objeto que envía un evento.

Event Channel o Canal de Eventos: sirve como medio de desacople entre Suministradores y Consumidores, actuando como Proxy (quiere esto decir que será Suministrador visto desde el lado de los Consumidores y como Consumidor si nos ponemos en el lado de los Productores).

Event Service: métodos para un modelo de eventos basado en la delegación.

Evento: es un mensaje.

Evolutionary Deployment: An important property of object technology is the ability to “wrapper” existing programs (coarse grain objects) to allow these functions to serve as building blocks for new business applications. This technique has been successfully used to marry object-oriented end-user interfaces with commercial business logic implemented using classical procedural techniques.

Flexible transaction propagation control: Both client and object implementations can control transaction propagation.

Interfaz: declaración de un conjunto de métodos y atributos.

Identificador del PO o PID: el que identifica al PO dentro del medio de almacenamiento, se puede convertir en cadena, almacenarse y recorrerse en sesiones posteriores.

Identification and authentication of principals (human users and objects which need to operate under their own rights): to verify they are who they claim to be.

Integrity: Information is modified only by users who have the right to do so, and only in authorized ways. It is transferred only between intended users and in intended ways.

Interoperabilidad: CORBA define un protocolo de comunicaciones para que todos los ORBs se comuniquen entre sí, aunque sean distintos.

Interfaz: Es una declaración de las operaciones que un objeto puede realizar y que pueden ser invocados por otros objetos externos.

Invocación asíncrona: cuando el emisor que envía un evento o mensaje continúa con sus tareas y no se preocupa por recibir una respuesta o confirmación de la recepción. Ejemplo: operaciones oneway.

Invocación síncrona: cuando el emisor envía un evento, se pone a la espera de la confirmación de su recepción que le enviará el receptor, por lo tanto se bloquea hasta que recibe una respuesta. Ejemplo de esto son las SII (invocaciones estáticas) salvo las declaradas como oneway.

Invocación síncrona-diferida (deferred-synchronous): el emisor no se bloquea tras enviar un mensaje, pero puede consultar en instantes posteriores cuál ha sido en resultado de su envío.

LifeCycle Service: métodos para crear, destruir, copiar y mover objetos.

“Mapear”: generar código fuente (en C++ por ejemplo) a partir de la compilación de un fichero .idl.

Middleware: Es software cuya función es integrar software.

Model Interoperability: Customers desire the capability to add object implementations to existing procedural applications and to augment object implementations with code that uses the procedural paradigm. To do so in a transaction environment requires that a single transaction be shared by both the object and procedural code.

Naming Service: métodos para obtener la referencia de un objeto, conocido su nombre, o para registrar objetos en dicho NS.

Network Interoperability: Customers require the ability to interoperate between systems offered by multiple vendors.

Non-repudiation: provides irrefutable evidence of actions such as proof of origin of data to the recipient, or proof of receipt of data to the sender to protect against subsequent attempts to falsely deny the receiving or sending of the data.

Objeto Persistente o PO: es el que controla su propio almacenamiento o permite al cliente que lo gestione, corresponde al interfaz PO.

Object Properties Service: métodos para asignar propiedades a objetos, ya registrados, sin modificar sus interfaces IDL (es decir, de forma dinámica).

- Object Query Service:** métodos para la búsqueda de objetos en una base de datos orientada a objetos (OODB).
- Object Startup Service:** métodos para el arranque de objetos ante una petición del cliente.
- Object Transaction Service:** métodos para establecer una secuencia de operaciones conocidas como transacción y ordenar su realización, bloqueando el recurso utilizado de posibles acciones ordenadas por otros procesos y permitiendo una vuelta (rollback) al estado inicial previo a ejecutar la transacción , si el resultado no fuese el deseado.
- OMG:** Grupo de personas que se dedican a crear estándares como por ejemplo el CORBA.
- Persistent Data Service o PDS:** implementa el interfaz Protocol, coordinando las operaciones de almacenamiento.
- Persistent Object Manager o POM:** el que toma las decisiones realmente sobre el almacenamiento de un PO.
- Persistent Object Service:** métodos para almacenar el estado de un objeto y hacerlos persistentes.
- Receptor:** es el objeto que recibe un evento.
- Relationship Service:** métodos para agrupar objetos, asignando roles.
- Security auditing:** to make users accountable for their security-related actions. It is normally the human user who should be accountable. Auditing mechanisms should be able to identify the user correctly, even after a chain of calls through many objects.
- Security of communication between objects:** which is often over insecure lower-layer communications. This requires trust to be established between the client and target, which may require authentication of clients to targets and authentication of targets to clients. It also requires integrity protection and (optionally) confidentiality protection of messages in transit between objects.
- Security Service:** métodos para establecer la política de seguridad de la aplicación (en lo referente a comunicaciones).

Servidor: una aplicación que atiende a lo que le pide el cliente y le proporciona ciertos servicios, o también el rol de los objetos que reciben y gestionan peticiones.

Sirviente: el objeto que implementa un interface, para proporcionarle la funcionalidad deseada.

Skeleton: una clase de C++ que define los mismos métodos y atributos que el cliente y que el objeto sirviente hereda para implementarlos. Es la interfaz entre el objeto y el ORB.

Stub: una clase de C++ que define unos métodos y atributos para que sean llamados por el cliente. Es la interfaz entre el ORB y el cliente.

Supplier o Suministrador: la entidad que genera o emite un evento.

Support for TP Monitors: Customers need object technology to build mission-critical applications. These applications are deployed on commercial transaction processing systems where a TP Monitor provides both efficient scheduling and the sharing of resources by a large number of users. It must be possible to implement the Transaction Service in a TP monitor environment.

Trader Service: un servicio que permite obtener una referencia a un objeto del que no se conoce su nombre, pero se conoce la necesidad que impulsa a su búsqueda. Es algo similar a las Páginas Amarillas, para la búsqueda de objetos y se basa en las invocaciones dinámicas (DII).

Apéndice C

Abreviaciones **WIRELESS**

AP:

Access point.

BPSK:

Binary Phase Shift Keying.

BSS:

Basic Service Set.

CCK:

Complementary Code Keying.

CRC:

Cyclic redundancy check.

CSMA/CA:

Carrier Sense Multiple Access with Collision Avoidance.

CSMA/CD:

Carrier Sense Multiple Access with Collision Detection.

CTS:

Clear to Send.

DCF:

Distribution Coordination Function.

DHCP:

Dynamic Host Configuration Protocol.

DS:

Distribution system.

DSSS:

Direct sequence spread spectrum.

ESS:

Extended Service Set.

ETSI:

European Telecommunications Standards Institute.

FCC:

Federal Communications Commission (USA).

FHSS:

Frequency Hopping Spread Spectrum.

IBSS:

Independent Basic Service Set.

IEEE:

Institute of Electrical and Electronics Engineers.

IETF:

Internet Engineering Task Force.

IP:

Internet Protocol.

IPSec:

Internet Protocol Security.

ISA:

Integrated Services Architecture.

ISM:

Industry, Scientific, and Medical.

ISO:

International Organization for Standardization.

LLC:

Logical Link Control.

MAC:

Media Access Control.

MIB:

Management information base.

MKK:

Radio Equipment Inspection and Certification Institute (Japan).

NIC:

Network interface card.

NOS:

Network operating system.

OFDM:

Orthogonal frequency division multiplexing.

OFDM/CCK:

Orthogonal frequency division multiplexing/complimentary code keying.

PCF:

Point Coordination Function.

PCI:

Peripheral Component Interconnect.

PRNG:

Pseudo random number generator.

QAM:

Quadrature amplitude modulation.

QPSK:

Quadrature Phase Shift Keying.

RC4:

Ron's Code or Rivest's Cipher.

RTS:

Request to Send.

SNMP:

Simple Network Management Protocol.

TCP/IP:

Transmission Control Protocol/Internet Protocol.

UNII:

Unlicensed National Information Infrastructure, refers to 5 GHz unlicensed frequency band.

WECA:

Wireless Ethernet Compatibility Alliance.

WEP:

Wired Equivalent Privacy.

WLAN:

wireless local area network.

WLANA:

Wireless LAN Alliance.

Apéndice D

Abreviaciones CORBA

3GL:

Third generation language.

4GL:

Fourth generation language.

ACID:

Atomicity, consistency, isolation, durability.

ACL:

Access control list.

ADT:

Abstract datatype.

AFC:

Application foundation classes [Microsoft].

ANSI:

American National Standards Institute.

API:

Application program interface.

AQ:

Advanced queuing [Oracle8].

ASCII:

American standard code for information interchange.

AWT:

Abstract windowing toolkit [Java].

BDK:

Beans developer kit [Java].

BLOB:

Binary large object.

BOA:

Basic object adapter [CORBA].

BSD:

Berkeley system distribution [UNIX].

C/S:

Client/server.

CGI:

Common gateway interface.

CICS:

Customer information control system [IBM].

CLI:

Call level interface [SAG].

CLOB:

Character large object.

COM:

Common object model [Microsoft].

CORBA:

Common object request broker architecture [OMG].

DB:

Database.

DBA:

Database administrator, database administration.

DBMS:

Database management system.

DCE:

Distributed computing environment [OSF].

DCOM:

Distributed common object model [Microsoft]

DDCF:

Distributed document component facility.

DDE:

Dynamic data exchange [Microsoft].

DDL:

Data definition language [SQL].

DLL:

Dynamic link library [Microsoft].

DLM:

Distributed lock manager [Oracle8].

DML:

Data manipulation language [SQL].

DOS:

Disk operating system.

DSOM:

Distributed system object model [IBM].

DSS:

Decision support system.

DTP:

Distributed transaction processing.

EBCDIC:

Extended binary-coded decimal interchange code [IBM].

EJB:

Enterprise JavaBean.

ERP:

Enterprise resource planning.

ESIOP:

Environment-specific inter-orb protocol.

FTP:

File transfer protocol.

GB:

Gigabyte.

GIF:

Graphics interchange format.

GIOP:

General inter-orb protocol.

GUI:

Graphical user interface.

GUID:

Globally-unique identifier.

HTML:

Hypertext markup language.

HTTP:

Hypertext transfer protocol.

IDE:

Integrated development environment, interactive development environment.

IDL:

Interface definition language.

IEEE:

Institute of Electrical and Electronics Engineers.

IIOP:

Internet inter-ORB protocol.

IP:

Internet protocol.

IPC:

Interprocess communication.

IS:

Information services.

ISAM:

Indexed sequential access method.

ISO:

International standards organization (translation).

ISP:

Internet service provider.

ISQL:

Interactive SQL [Interbase].

ISV:

Independent software vendor.

IT:

Information technology.

JAR:

Java archive (on analogy with tar, q.v.).

jdb:

Java debugger [Sun].

JCK:

Java compatibility kit [Sun].

JDBC:

Java database connectivity.

JDK:

Java developer kit.

JFC:

Java foundation classes.

JIT:

Just in time.

JNDI:

Java naming and directory interface.

JNI:

Java native interface.

JOB:

Java Objects for Business [Sun].

JPEG:

Joint photographic experts group.

JSP:

Java server pages [Sun].

JTA:

Java transaction API.

JTS:

Java transaction service.

KB:

Kilobyte.

LAN:

Local area network.

LDAP:

Lightweight directory access protocol.

LDIF:

LDAP data interchange format.

LOB:

Large object.

MB:

Megabyte.

MIS:

Management information services.

MOM:

Message-oriented middleware.

MPEG:

Motion picture experts group.

NCLOB:

National character large object.

NIC:

Network information center [internet].

NNTP:

Net news transfer protocol.

NSP:

Network service provider.

NT:

New Technology [Microsoft].

OCI:

Oracle call interface.

OCX:

OLE common control [Microsoft].

ODBC:

Open database connectivity [Microsoft].

ODBMS:

Object database management system.

ODL:

Object definition language [Microsoft].

ODMG:

Object Database Management Group.

OEM:

Original equipment manufacturer.

OID:

Object identifier.

OLE:

Object linking and embedding.

OLTP:

On line transaction processing.

OMA:

Object management architecture [OMG].

OMG:

Object Management Group.

OO:

Object-oriented, object orientation.

OODBMS:

Object-oriented database management system.

OQL:

Object query language.

ORB:

Object request broker.

ORDBMS:

Object relational database management system.

OS:

Operating system.

OSF:

Open System Foundation.

OSI:

Open systems interconnect.

OSQL:

Object SQL.

OTM:

Object transaction monitor.

OTS:

Object transaction service.

OWS:

Oracle Web Server.

PB:

Petabyte.

PDF:

Portable document format [Adobe].

PGP:

Pretty good privacy.

PL/SQL:

Procedural language/SQL [Oracle].

POA:

Portable object adapter [CORBA].

RAM:

Random access memory.

RAS:

Remote access service [Microsoft NT].

RCS:

Revision control system.

RDBMS:

Relational database management system.

RFC:

Request for comments.

RFP:

Request for proposal.

RMI:

Remote method invocation [Sun].

ROM:

Read only memory.

RPC:

Remote procedure call.

RTF:

Rich text file.

SAG:

SQL Access Group.

SCSI:

Small computer system interface.

SDK:

Software developer kit.

SET:

Secure electronic transaction.

SGML:

Standard generalized markup language.

SID:

System identifier [Oracle].

SLAPD:

Standalone LDAP daemon.

SMP:

Symmetric multiprocessing.

SMTP:

Simple mail transfer protocol.

SPI:

Service provider interface.

SQL:

Structured query language.

SQLJ:

SQL for Java.

SRAM:

Static (or synchronous) random access memory.

SSL:

Secure socket layer.

tar:

Tape archive, tape archiver [UNIX].

TB:

Terabyte.

TCPS:

TCP for SSL.

TCP/IP:

Transmission control protocol/internet protocol.

TP:

Transaction processing.

TPC:

Transaction Processing Council.

TPCW:

TPC Web benchmark.

TPF:

Transaction processing facility.

TPM:

Transaction processing monitor.

tps:

Transactions per second.

UCS:

Universal character set [ISO 10646].

UDP:

User Titograd protocol.

UI:

User interface.

UML:

Unified modeling language [Rational].

URL:

Universal resource locator.

VAR:

Value-added reseller.

VRML:

Virtual reality modeling language.

WAN:

Wide area network.

WIPS:

Web interactions per second [TPCW].

WWW:

World wide web.

XA:

Extended architecture [X/Open].

XML:

Extended markup language.

Apéndice E

Wireless Ethernet



CARACTERÍSTICAS

Accesorios

Ref.: 191808-B21

Tarjeta PCMCIA inalámbrica Compaq WL110



Con la tarjeta PCMCIA WL110, podrá utilizar un PC de bolsillo o un PC portátil para conectarse a alta velocidad a redes corporativas y acceder a Internet con total seguridad, lo que le permitirá desplazarse con completa libertad.

Precio: **115,00 €** (19.134 pts)*

Formato	PCMCIA
Alimentación	Fuente de alimentación 5 V / +25 mA
Velocidad	11, 5,5, 2 y 1 Mbps; Selecciones de velocidad automática (ARS)
Dimensiones	117,8 x 53,95 x 8,7 mm (Tarjeta PC)
Otras características	Difusión: Secuencia Barker de 11 chips
Compatibilidades	Canales secundarios certificados en todo el mundo (FCC/ETS/JP/FR), 11 canales
Comunicaciones	Modulación por salto de frecuencias directo (CCK, DQPSK, DBPSK)
Conexión	Alcance en metros a 11Mbps: Oficina Abierta: 160m Oficina semiabierta: 50m Oficina cerrada: 25m Sensibilidad del receptor dBm: -82 Retardo (a REF de <1%): 65 ns
Consumo	Modo en espera: 9 mA Modo de recepción: 185 mA Modo de transmisión: 285 mA
Frecuencias	Espectro de 2.400 - 2.483,5 MHz
Cumplimiento de normativas	Estándares IEEE 802.11b y WiFi
Interfaz	Tarjeta PC, PCI
Potencia	15 dBm
Seguridad	Encriptación WEP (Wired Equivalent Privacy) de 128 bits Tasa de error mejor que 10 ⁻⁵ Protocolo de Acceso CSMA/CA (evasión de colisiones) con ACK
Temperaturas de funcionamiento	0-55 o C Humedad máxima 95% (sin condensación)
Sistemas operativos compatibles	Windows 95/98/ME/NT4 (SP4)/2000 Windows CE/Pocket PC
Cobertura internacional	Europa/APA: ETS 300-328, Sello CE EE.UU.: FCC (47 CFR) Parte 15C, Artículo 15.247 Canadá: ISC RSS 139 Japón: Normativas de radio MPT
Requisitos	1 ranura PCMCIA; Windows 95, Windows 98, Windows 2000, Windows NT, o Windows Me



CARACTERÍSTICAS

Accesorios

Ref.: 191811-021

Punto de Acceso inalámbrico Compaq WL410



Pensado para pequeñas y medianas empresas. Este dispositivo amplía el alcance de la red Ethernet existente o evita el cableado. Una solución rentable para aquellos que no precisen de una red inalámbrica dual.

Precio: **469,00 €** (78.035 pts) *

Formato	Sobremesa / Mural
Alimentación	Unidad de pared externa: Detección automática 100/240 VCA 47-63 Hz, 9W/1,1A Preparado para Active Ethernet (Alimentación mediante Ethernet)
Dimensiones	145 x 175 x 70 mm (incluye soporte para mesa) 130 x 175 x 45 mm (incluye soporte para pared)
Otras características	- Encendido - Actividad LAN Ethernet - Actividad LAN inalámbrica,
Comunicaciones	Conexión Ethernet 10 Base T (RJ45)
Conexión	Alcance en metros a 11Mbps: Oficina Abierta: 160m Oficina semiabierta: 50m Oficina cerrada: 25m Sensibilidad del receptor dBm: -82 Retardo (a REF de <1%): 65 ns
Especificaciones mecánicas	Cubierta de plástico Placa de montaje que permite montaje en pared o mesa Conector de antena externo y accesible
Cumplimiento de normativas	Cumplimiento IEEE 802.11b (WIFI)
Interfaz	Ethernet 802.3 10Base-T (Conector RJ 45)
Peso	0,50 kg
Seguridad	Encriptación de 128 bits WEP
Temperaturas de funcionamiento	0 °C a + 40 °C Humedad máxima de funcionamiento: 90% (condensación no permitida)

* precio orientativo sin IVA para Península y Baleares

Imprimir



CARACTERÍSTICAS

Accesorios

Ref.: 230268-B21

Antena Amplificadora Wireless LAN

Esta antena omnidireccional aumenta la flexibilidad de su red. Colóquela en la pared o en el techo para mejorar la calidad de la conexión inalámbrica.

Precio: **111,00 €** (18.469 pts) *

Formato	Antena Omnidireccional
Dimensiones	230 x 25 x 8 mm. longitud de 1,5 m Diámetro de 70 mm (base)
Otras características	Oficina abierta: 50% de aumento con respecto al alcance de una tarjeta PC de Compaq estándar. Oficina semiabierta: 15% de aumento con respecto al alcance de una tarjeta PC de Compaq.
Conexión	Propia (compatible con FCC).
Especificaciones mecánicas	Diseño Omnidireccional. Diversidad de la antena: sí. Montaje sobre mesa y soporte para montaje en pared.
Frecuencias	2.400 a 2.483,5 MHz
Peso	150 gr

* precio orientativo sin IVA para Península y Baleares

SPECIFICATIONS	
Frequency Band	2400-2483.5 MHz
Number of Selectable Sub-channels	Worldwide Certified (FCC/ETSI/JP/FR), 11 channels
Modulation Technique	Direct Sequence Spread Spectrum (CCK, DQPSK, DBPSK)
Spreading	11-chip Barker sequence
Bit Error Rate	Better than 10^{-5}
Media Access Protocol	CSMA/CA (Collision Avoidance) with ACK
Interface	PC Card, PCI
Size / Viewable Image Size (diagonal)	15 in / 13.8 in
OS Support	Windows 95/98/ME/NT4 (SP4)/2000/Windows CE/Pocket PC
Dimensions	117.8 mm x 53.95 mm x 8.7 mm (PC card)
Speed Options	11, 5.5, 2 and 1 Mbps; Automatic Rate Selection (ARS)
Output Power	15dBm
Power Consumption (5 volt power supply)	Doze mode - 9mA Receive mode - 185 mA Transmit mode - 285 mA
Temperature Range (operational)	0-55° C 95% max. Humidity (non condensing)
Standards	IEEE 802.11b
Regulations	US: FCC (47 CFR) Part 15C, Section 15.247 Canada: ISC RSS139 Europe/APA: ETS 300-328, CE Marked Japan: MPT Radio Regulations

Speed Options	11 Mbps	5.5 Mbps	2 Mbps	1 Mbps
Range in meters(feet) Open Office	160 m (525ft)	270 m (885 ft)	400 m (1300 ft)	550 m (1750 ft)
Semi Open Office	50 m (165 ft)	70 m (230 ft)	90 m (300 ft)	115 m (375 ft)
Closed Office	25 m (80 ft)	35 m (115 ft)	40 m (130 ft)	50 m (165 ft)
Receiver Sensitivity	-82	-87	-91	-94
Delay Spread (at FER of < 1%)	65ns	225ns	400ns	500ns

SPECIFICATIONS	
Features	Integrated 11 Mbps radio Supports -- Power Over Ethernet 128 bit key security using RC4 encryption 10 Mbps Ethernet IEEE 802.11b (Wi-Fi) compliant Spanning Tree Algorithm IEEE 802.1D Transparent Bridging Selective protocol filtering Access Control Table and Radius Authentication DHCP and BOOTP Multi-channel support Roaming support
Management	Compaq AP Manager Software SNMP MIB ii, 802.3, 802.1D, and 802.11 MIB compliant Windows based user interface TRAPS: power up, authentication, link up/down Site Survey Tools Remote Link Test
LEDs	Power Ethernet LAN Activity Wireless LAN Activity
Interface	Ethernet 802.3 -10Vase-T (RJ 45 Connector)
Dimensions	145 mm x 175 mm x 70 mm (includes table mount bracket) 130 mm x 175 mm x 45 mm (includes wall mount bracket)
Weight	0.50 kg
Power Supply	External wall plug-in unit: Auto sensing 100/240 VAC 47-63 Hz, 9w/1.1 Ready for - Power Over Ethernet
Temperature Range (operational)	0° to +40° C Operating humidity max 90% (no condensing allowed)

Speed Options	11 Mbps	5.5 Mbps	2 Mbps	1 Mbps
Range in meters(feet) Open Office	160 m (525ft)	270 m (885 ft)	400 m (1300 ft)	550 m (1750 ft)
Semi Open Office	50 m (165 ft)	70 m (230 ft)	90 m (300 ft)	115 m (375 ft)
Closed Office	25 m (80 ft)	35 m (115 ft)	40 m (130 ft)	50 m (165 ft)
Receiver Sensitivity	-82	-87	-91	-94
Delay Spread (at FER of < 1%)	65ns	225ns	400ns	500ns

QuickSpecs

Compaq WL110 Wireless PC Card

MODELS

WL110 Wireless PC Card
191808-021

OVERVIEW

The Compaq WL110 Wireless PC Card can be used worldwide for the Enterprise, ISP, Public Access or SMB markets using any of the access point infrastructure products. The card provides high-speed wireless networking and secure access to network resources and the Internet. The WL110 PC card is for use in notebooks, desktops and the iPAQ Pocket PC. Benefits include:

- The same high-performance connectivity as wired systems, with the added freedom to roam around your building or campus
- Enhanced security with 128-bit Wired Equivalent Privacy (WEP) encryption
- WiFi certified and compatible with IEEE 802.11b compliant PC Cards
- Compaq Client Manager Software

COMPATIBILITY

Compaq Notebook
iPAQ Pocket PC

SERVICE AND SUPPORT

WL110 Wireless PC Card has a three-year limited warranty or the remainder of the warranty of the Compaq product in which it is installed. Technical support is available seven days a week, 24 hours a day, by phone as well as online support forms. Certain restrictions and exclusions apply.

SPECIFICATIONS

Frequency Band	2400-2483.5 MHz
Number of Selectable Sub-channels	Worldwide Certified (FCC/ETS/URF/FR), 11 channels
Modulation Technique	Direct Sequence Spread Spectrum (DSSS, DQPSK, DBPSK)
Spreading	11-chip Barker sequence
Bit Error Rate	Better than 10 ⁻⁵
Media Access Protocol	CSMA/CA (Collision Avoidance) with ACK
Interface	PC Card, PCI
OS Support	Windows 95/98/ME/NT4 (SP4)/2000; Windows CE/Pocket PC
Dimensions	4.6 in x 2.12 in x 0.34 in / 117.8 mm x 53.95 mm x 8.7 mm (PC card)
Speed Options	11, 5.5, 2 and 1 Mbps, Automatic Rate Selection (ARS)
Output Power	15dBm
Power Consumption (5 volt power supply)	Data mode – 9mA Receive mode – 185 mA Transmit mode – 295 mA
Temperature Range (operations)	32° to 130° F / 0° to 55° C 95% maximum Humidity (non condensing)
Standards	IEEE 802.11b
Regulations	US: FCC (47 CFR) Part 15C, Section 15.247 Canada: ISC RSS139 Europe/APA: ETS 300-328, CE Marked Japan: MPT Radio Regulations

COMPAQ

0A-16799 North America — Version 2 — June 21, 2001

1

QuickSpecs

Compaq WL110 Wireless PC Card

SPECIFICATIONS (continued)

Speed Options	11 Mb/s	5.5 Mb/s	2 Mb/s	1 Mb/s
Range in meters (feet) Open Office	160 m (525 ft)	270 m (885 ft)	400 m (1300 ft)	650 m (1750 ft)
Semi Open Office	50 m (165 ft)	70 m (230 ft)	90 m (300 ft)	115 m (375 ft)
Closed Office	25 m (80 ft)	35 m (115 ft)	40 m (130 ft)	50 m (165 ft)
Receiver Sensitivity	-82	-87	-91	-94
Delay Spread (at FER of < 1%)	55ns	225ns	400ns	500ns

Compaq PCs use genuine Microsoft® Windows®

<http://www.microsoft.com/piracy/howtotell>

Microsoft is a trademark of Microsoft Corporation in the U.S. and other countries.

QuickSpecs

Compaq WL410 Wireless SMB Access Point

MODELS

WL410 Wireless SMB Access Point
191811-001

OVERVIEW

The Compaq WL410 is a full featured wireless access point that acts as a bridge between the wired Ethernet backbone and your wireless clients. The small form factor of the WL410 Access Point makes it easy to locate either on a flat surface or easily mounted on a wall. Benefits of the WL410:

- Secure for placement in public areas with a Kensington locking system
- Comes with an integrated antenna, but also has a jack for adding an external antenna for enhanced performance
- Supports both 110/240 VAC power from local AC outlet or Power Over Ethernet
- Enhanced security with 128-bit Wired Equivalent Privacy (WEP) encryption

COMPATIBILITY

Compaq Desktop
Compaq Notebook
iPAQ Pocket PC

SERVICE AND SUPPORT

WL410 Wireless SMB Access Point has a one-year limited warranty. Technical support is available seven days a week, 24 hours a day, by phone as well as online support forms. Certain restrictions and exclusions apply.

SPECIFICATIONS

Features	Integrated 11 Mb/s radio Supports Power Over Ethernet 128 bit key security using RC4 encryption 10 Mb/s Ethernet IEEE 802.11b (Wi-Fi) compliant Spanning Tree Algorithm IEEE 802.1D Transparent Bridging Selective protocol filtering Access Control Table and Radius Authentication DHCP and BOOTP Multi-channel support Roaming support
Management	Compaq AP Manager Software SNMP MIB II, 802.3, 802.1D, and 802.11 MIB compliant Windows based user interface TRAPS: power up, authentication, link up/down Site Survey Tools Remote Link Test

COMPAQ

QuickSpecs

Compaq WL410 Wireless SMB Access Point

SPECIFICATIONS *(continued)*

LEDs	Power Ethernet LAN Activity Wireless LAN Activity			
Interface	Ethernet 802.3 -10Base-T (RJ 45 Connector)			
Dimensions	5.7 in x 6.9 in x 2.8 in/145 mm x 175 mm x 70 mm (includes table mount bracket) 5.1 in x 6.9 in x 1.8 in/130 mm x 175 mm x 45 mm (includes wall mount bracket)			
Weight	1.1 lbs/0.50 kg			
Power Supply	External wall plug-in unit: Auto sensing 100/240 VAC 47-63 Hz, 9W/1.1A Ready for -Power Over Ethernet			
Temperature Range (operational)	32° to 104° F/0° to +40° C Operating humidity maximum 90% (no condensing allowed)			
Speed Options	11 Mb/s	5.5 Mb/s	2 Mb/s	1 Mb/s
Range in meters (feet) Open Office	160 m (525 ft)	270 m (885 ft)	400 m (1300 ft)	550 m (1750 ft)
Semi Open Office	50 m (165 ft)	70 m (230 ft)	90 m (300 ft)	115 m (375 ft)
Closed Office	25 m (80 ft)	35 m (115 ft)	40 m (130 ft)	50 m (165 ft)
Receiver Sensitivity	-82	-87	-91	-94
Delay Spread (at FER of < 1%)	65ns	225ns	400ns	500ns

Compaq PCs use genuine Microsoft® Windows®

<http://www.microsoft.com/piracy/howtotell>

Microsoft is a trademark of Microsoft Corporation in the U.S. and other countries.

Índice de figuras

1.	plataforma	VIII
1.1.	Tipos de nodos en una red wireless	7
1.2.	Diferencias entre WLAN y otras tecnologías inalámbricas	8
1.3.	red ad-hoc	11
1.4.	punto de acceso	12
1.5.	punto de extensión	13
1.6.	antenas direccionales	13
1.7.	aspecto de una señal de dos bits a la cual le hemos aplicado la secuencia de Baker	15
1.8.	Modo de trabajo de la técnica FHSS	17
1.9.	Sincronización de paquetes 802.11b, Bluetooth y HomeRF	20
1.10.	Red inalámbrica	24
1.11.	estándares IEEE WLAN	27
1.12.	Tiempo ahorrado por el uso de WLAN's	35
1.13.	Acceso inalámbrico desde distintas ubicaciones	37
2.1.	servicio de objetos	51
2.2.	Un protocolo genérico de intercomunicación GIOP	52

<i>ÍNDICE DE FIGURAS</i>	166
2.3. Sistema distribuido	55
2.4. Lenguaje de definición de interfaces	57
2.5. Mapeado en C++	59
2.6. Mapeado en Java	60
2.7. Servidor compartido	65
2.8. Servidor no compartido	66
2.9. Servidor por método	66
2.10. Adaptador portable de objetos	68
2.11. Aplicación CORBA	73
2.12. Invocación rpc	76
2.13. Invocación datagrama	76
2.14. Invocación con respuesta diferida	77
2.15. Primer caso de la consecuencia de los sirvientes	79
2.16. Segundo caso de la consecuencia de los sirvientes	80
2.17. Servicios Corba	81
2.18. Facilidades comunes	82
2.19. Dominios corba	84
2.20.	86
2.21.	88
2.22.	89
2.23.	89
3.1. Características de las distintas tecnologías	96
3.2. Soluciones estudiadas	98
3.3. Compaq WL110	99

<i>ÍNDICE DE FIGURAS</i>	167
3.4. Compaq WL410	99
3.5. Antena amplificadora de señal	100
3.6. Antena Pringles	101
4.1. Características CORBA/RMI/DCOM	107
4.2. Variables de entorno del windows 2000 y XP	111
4.3. Código IDL	111
4.4. Código del cliente	113
4.5. Código cabecera de la implementación	114
4.6. Código fuente de la implementación	114
4.7. 1ª parte del código del servidor	115
4.8. 2ª parte del código del servidor	116
4.9. Código del Makefile	118
A.1. Puntos de acceso	127

Bibliografía

- [1] R. Flickenger, "Building wireless community networks," January 2002.
- [2] E. of IEEE, "Ieee 802.11, wireless lan mac and physical layer specifications," juny 1997.
- [3] M. M. Martin, "Wireless local area networks," vol. <http://www.csm.ohio.edu/comt391w/mmartin/final.html>.
- [4] M. Schwartz, "Redes de telecomunicaciones," vol. Ed. Addison-Wesley Iberoamericana, México, 1994.
- [5] W. Stallings, "Comunicaciones y redes de computadoras," *Prentice Hall*, 1997.
- [6] I. Prepared by Sage Research, "Wireless lans: Improving productivity and quality of life," vol. www.sageresearch.com., May 2001.
- [7] "What is a wireless lan?," *White Paper*, vol. www.proxim.com.
- [8] "Dr. farallons guide to wireless lan connectivity," vol. www.farallon.com.
- [9] "Migration of wireless lan technologies in the enterprise deploy 802.11b today, and prepare for 802.11a tomorrow," vol. www.proxim.com.
- [10] "Introduction to wireless lans," vol. www.wlana.com.
- [11] D. C. Schmidt, "Developing distributed object computing applications with corba,"
- [12] S. V. M. Henning, "Programación avanzada en corba con c++," *Addison-Wesley*, 2002.
- [13] S. Baker, "Corba distributed objects using orbix," *Adison Wesley*, 1997.

- [14] B. Blakley, “Corba security, an introduction to safe computing with objects,” *Adison Wesley*, 2000.
- [15] “Distributed systems research group,,”
- [16] “www.3com.com,”
- [17] “www.cisco.com,”
- [18] “www.orinocowireless.com,”
- [19] “www.stechcomm.com,”
- [20] “<http://standards.ieee.org/getieee802/portfolio.html?agree=accept>,”
- [21] “Wireless lan alliance,” vol. <http://www.wlana.com>.
- [22] D. C. S. Irfan Pyarali