

# Brief Review of Service Discovery Protocols

## Descripción de los Diferentes Protocolos Descubridores de Servicios

Mauricio Alejandro Ríos  
mrios@unisimonbolivar.edu.co  
Universidad Simón Bolívar  
Barranquilla - Atlántico

Keywords:  
Service Discovery Protocols,  
Jini, UPnP, SDP, MANETS

### Abstract

In computation, the term ubiquitous refers to environments where devices that have capacities of processing, storage and communication (mobile phones, personal digital assistant, etc.) can communicate in a smart and ordered way, having their own conscience about their surroundings, everything must be transparent to the users. Wireless networks ad hoc, also known as MANETs (Mobile ad hoc Networks), are represented as a magnificent technology of communication to this kind of environment and applications. In MANETs networks, collaboration among devices is given in an offered services form by the devices themselves (nodes), which are discovered using protocols of discoveries of services. In this article is pretended to make a review of the different protocols of discoveries of services more common in the market for the scheme of ubiquitous computation.

Palabras clave:  
Protocolos de descubrimiento  
de servicios, Jini, UPnP, SDP,  
MANETS

### Resumen

El término de computación ubicua hace referencia a ambientes donde los dispositivos que poseen capacidades de procesamiento, almacenamiento y comunicación (teléfonos móviles, personal digital assistant, etc.) puedan comunicarse de forma inteligente y ordenada teniendo una conciencia propia del entorno que los rodea, todo esto debe ser transparente para los usuarios. Las redes inalámbricas ad hoc también conocidas como MANETs (Mobile ad hoc Networks) se presentan como una magnífica tecnología de comunicación para este tipo de entorno y aplicaciones. En las redes MANETs, la colaboración entre los dispositivos se da en forma de servicios ofrecidos por los mismos dispositivos (nodos), los cuales son descubiertos utilizando protocolos de descubrimientos de servicios, En este artículo se pretende hacer una visión de los diferentes protocolos descubridores de servicios más conocidos en el mercado para el esquema de computación ubicua

## I. INTRODUCCIÓN

Hace poco somos testigos de la nueva era computacional, una era donde todos nuestros dispositivos de cómputo no se vean arraigados en un solo sitio en el espacio, este tipo de computación es conocida como computación móvil. Hoy día muchos de los dispositivos que usamos son fáciles de transportar como los PDA (personal digital assistant), teléfonos móviles, notebook o similares. Sin embargo todos estos dispositivos serian poca cosa si no

existiese una manera de comunicarlos entre si para intercambiar información dentro un grupo de usuarios que comparten una misma afición. Cabe resaltar que los usuarios no deberían tener los conocimientos de un administrador de red para poder compartir sus recursos a otros usuarios, sino que existan mecanismos para su cómoda instalación, configuración y uso, así se facilitaría el trabajo al administrador de la red y al usuario final abstraerlo de la complejidad de los protocolos de comunicación.

## II. PARADIGMA DEL MODELO CLIENTE-SERVIDOR A LA COMPUTACIÓN UBICUA.

### a. MODELO CLIENTE-SERVIDOR

A partir de los años 90 la proliferación de los sistemas distribuidos empiezan a tomar fuerzas frente a las aplicaciones desarrolladas con el modelo standalone, un sistema distribuido es una aplicación compuesta por varios elementos que comparten información mientras se ejecutan paralelamente en diferentes computadores, desde su comienzo se han hecho diferentes mecanismos y patrones de comunicación entre ellos.

El modelo que mas prevalece en un sistema distribuido es el modelo cliente-servidor, allí entra a jugar dos participantes importantes en este sistema, cliente y servidor, el cliente es aquel que realiza peticiones que serán escuchadas por un programa servidor. Por lo general el servidor se debe de encontrar siempre en ejecución y espera de solicitudes por parte de los clientes que este debe atender.

El modelo cliente-servidor posee ciertas variantes que está definida por llamada 'lógica de negocio', o dicho de otra manera, la operativa necesaria para satisfacer la funcionalidad de la aplicación.

- Si la "lógica del negocio" se encuentra del lado del servidor se dice que es un modelo cliente ligero-servidor pesado.
- Si la "lógica del negocio" se encuentra del lado del cliente se dice que es un modelo cliente pesado-servidor ligero.

No obstante existe una tercera alternativa donde un servidor puede hacer a su vez peticiones a otros servidores, dando lugar así a una arquitectura en varias capas, en la que cada capa intermedia puede comportarse como un servidor y cliente.

Un ejemplo clásico del modelo cliente-servidor es el sistema WWW. Donde un cliente en nuestro caso es el programa navegador o browser que realiza peticiones a un aplicativo de servicio web muchas veces llamado servidor web.

### b. EL MODELO P2P

A diferencia del paradigma cliente-servidor, donde se nota gran asimetría en la participación de los que suministran contenidos y servicios (comúnmente llamados servidores) y de aquellos que solo se limitan a consumir (cliente) de dichos servicios asumiendo estos últimos un papel pasivo sin aportar sus recursos a otros nodos.

Los sistemas distribuidos basados en el modelo P2P (peer-to-peer), asigna a todos los nodos participantes las mismas responsabilidades y tipos actividades, distribuyendo la información entre los nodos en vez de centralizarla. " Además, el modelo P2P asume la conectividad variable de las entidades participantes (como norma general, no estarán disponibles de manera constante) y la independencia del nodo de red (las entidades pueden participar desde un nodo distinto cada vez, sin disponer de una dirección fija). A estos sistemas distribuidos se les atribuye con frecuencia el nombre de 'redes P2P', siendo las entidades que participan en la comunicación los 'nodos miembro'. Según este modelo, se establece una disociación clara entre el nodo miembro y el nodo de red en el que se ejecuta. " [1]

La forma en que el modelo P2P distribuye la información se hace a través de un índice, lo cual facilita la participación y colaboración entre los nodos que hacen parte de esta red, un nodo individual puede hacer uso de otros nodos activos y a sus archivos compartidos, al mismo tiempo el nodo que se conecta pasa hacer parte de dicho índice, con lo que resulta visible a los demás nodos, Las redes P2P son empleadas mayormente por programas como Ares, Emule, entre otros, estos son utilizados para compartir música, video o juegos, lo cual no es un

punto positivo al comercio electrónico ya que esto conlleva asuntos legales.

“ Aunque la tecnología P2P es menos eficiente que las bases de datos distribuidas o los sistemas de apoyo a la decisión en tiempo real, es interesante dada la facilidad para incorporar nuevos participantes a la red, resolviendo problemas como aquellos relacionados con la interoperabilidad. Además, algunas de las desventajas de este paradigma pueden aliviarse al incorporar un cierto grado de control organizacional sobre la red. ”

### C. COMPUTACIÓN MÓVIL

La gran demanda de nuevos dispositivos con capacidades de almacenamiento y procesamiento con capacidad de intercambiar información sin utilizar un medio físico cableado, promueve al diseño de interconexiones inalámbricas, esto a su vez conlleva a que el propio dispositivo tenga la libertad a incorporarse o abandonar cualquier red. Los protocolos de red tales como el Bluetooth, WIFI se utilizan en redes MANETs como medio de transporte para el intercambio de transmisión. En estos entornos donde cualquier dispositivo tiene cierta promiscuidad para ingresar de una red a otra sin tener que entrar en una ardua configuración es indispensable que se tenga un mecanismo para descubrir otros dispositivos y servicios de manera dinámica.

“ Este tipo de redes se introduce un conjunto adicional de problemas a los ya definidos en las redes P2P: acceso a información móvil (no siempre está presente), problemas de gestión de la energía (baterías limitadas), sensibilidad a la posición y detección de presencia. ” [1]

### d. COMPUTACIÓN UBICUA

“ La visión de la computación ubicua denota una tendencia en la evolución de las tecnologías de la información y la comunicación según la cual los elementos de proceso se incrustan en los objetos

cotidianos dotándolos de “sensibilidad”, por ejemplo, adaptando su comportamiento dependiendo del contexto en que se encuentre el objeto. Al final, el procesamiento de la información y las capacidades de comunicación quedarán integrados en los objetos y, al menos a primera vista, desaparecerán del foco de atención del usuario, permitiéndole concentrarse completamente en la tarea que está realizando. ” [1]

La expresión computación ubicua nace gracias a Mark Weiser en 1991, en aquel entonces lograr el objetivo de este paradigma era incansable ya que las tecnologías existentes no permitían dicho desarrollo. Hoy día existen algunos productos como lo son las handheld (computadores de bolsillo), wearables, Lan inalámbricas.

### III. SERVICIOS

Como se menciona en el apartado anterior los diferentes modelos de sistemas siempre han apuntado a prestar un servicio al usuario ya sea que se tenga movilidad o no con sus dispositivos, sin embargo no hemos analizado lo que es un servicio y cuáles son los inconvenientes que este posee a la hora de encontrarlo.

Un servicio es una reunión de elementos orientados a convertir algo, con el objetivo de darle un valor agregado. En definitiva, un servicio es una unidad funcional caracterizada por una interfaz formalmente definida y unos requerimientos de comportamiento establecidos.

Un modelo de servicios debe dar respuesta a las siguientes cuestiones: [1]

- Qué lenguaje se emplea para especificar el servicio. Este debe ser conocido por todos los participantes en el negocio. La descripción del servicio debe ser independiente de las entidades que realizan o reciben el servicio, de manera que no resulte costosa la incorporación de nuevos participantes al negocio. El lenguaje debe ser capaz de captar

el valor que produce el servicio en los elementos sobre los que actúa.

- Cómo localizar entidades que realicen un determinado servicio. (Protocolo de descubrimiento). Hay que definir un repositorio centralizado o distribuido que permita el listado de operaciones soportadas por la organización y sus características (perfil de la compañía).
- Cómo puede una entidad anunciar la disponibilidad de sus servicios (registro).
- Qué mecanismos de negociación previos a la ejecución de las transacciones hay definidos.
- Cómo se transfieren los elementos a transformar. Es necesario indicar un protocolo de transporte para el intercambio de mensajes, que incluya garantías de seguridad y fiabilidad. Además, hay que definir un formato común de intercambio de mensajes
- Tipos de operaciones comunes (por ejemplo, envío de un pedido de compra).

Un modelo de servicio le permite a las organizaciones exponer sus competencias principales a los usuarios, pero estos servicios deben poseer protocolos, lenguajes e interfaces estándares abiertos de manera que los servicios puedan accederse sin intervención humana, sin embargo más adelante notaremos que no todas las organizaciones se han pues de acuerdo en unificarse.

#### IV. DESCUBRIMIENTO DE SERVICIOS

Un servicio a nivel de informática se puede definir como la capacidad que tiene un dispositivo de realizar tareas y al mismo tiempo, disponerlas a otros para su uso.

Un grupo de desarrolladores que pretenda realizar un protocolo que preste un servicio en la red, deberán

de organizar y estandarizar las entidades que van a formar parte de la comunicación, dicha organización deberá manejar un protocolo común y formato para el intercambio de mensajes.

Como muchas de las compañías comerciales desarrollan sus propios estándares de comunicación no existe un modelo universal para la realización de dicha tarea, hasta la fecha los protocolos existentes carecen de homogeneidad. Por lo general es necesario configurar los dispositivos antes de hacer uso de un servicio. Por ejemplo, configurar correctamente la dirección de red, comprobar que el servicio está operativo.

“ Ya que no podemos esperar que cada usuario tenga el conocimiento de un experto en redes, estas tareas de configuración deben ser asumidas por las propias aplicaciones. En este punto es donde entra en juego el descubrimiento de servicios. ” [2]

Con un protocolo de descubrimiento de servicios se espera que los usuarios puedan acceder a un amplio rango de servicios con solo conectar sus dispositivos a una red y pasar por una configuración sin intervención humana, el usuario gozará de los servicios de una manera transparente donde solo este se enfocará a utilizarlo y satisfacer su necesidad.

#### a. BREVE RESEÑA HISTÓRICA

En los años 80 se da iniciativa de ciertos protocolos para detectar algunos aplicativos que poseían los dispositivos en aquel entonces, Apple Inc. Desarrolla el conjunto de protocolos AppleTalk. Uno de sus protocolos era el NPB (name binding protocol) un protocolo de descubrimiento de servicio muy sencillo pero eficaz y totalmente distribuido.

Tanto el NPB como el AARP (Apple Address Resolutions Protocol) permitían a los equipos de appletalk generar automáticamente su dirección de red hicieron la diferencia entre sus rivales.

“ Appletalk identifica de forma unívoca los servicios ofrecidos en lugar de identificar el dispositivo que los ofrece. De esta forma, se permite mover un servicio de un dispositivo a otro sin necesidad de reconfigurar toda la red. ” [2] Sin embargo Appletalk poseía una limitante, para poder utilizar los servicios este debía instalar los controladores en los equipos.

En ese tiempo existían otros protocolos como Banyan VINES, Novell Netware o el sistema NetBIOS de IBM sin embargo no poseían la facilidad que brindaba Appletalk.

En 1982, se realiza un proyecto para conocer la dirección Ethernet de un dispositivo a partir de su dirección IP, address resolution protocol (ARP). Seguidamente se crea el RARP (reverse address resolution protocol) lo cual permitía realizar la operación opuesta del ARP usando un servidor central. Usualmente estos eran usados por los equipos conectados en red para obtener una dirección IP automática después de su iniciar la marcha.

“ El protocolo bootstrap protocol (BOOTP) elimino parcialmente las limitaciones de RARP, que necesitaba un servidor por cada red Ethernet existente. Este también fue creado con el fin de permitir la configuración automática de equipos que se arrancan conectadas a una red TCP/IP, asignando siempre la misma dirección a los equipos sin reutilizarlas. Al igual que su predecesor, BOOTP necesita un servidor central, pero en este caso el servidor puede controlar varias subredes.

Hasta el año 1993 no se mejoro este sistema de asignación automática de direcciones. El protocolo dynamic host configuration protocol introdujo el concepto de lease time por el cual los equipos deben renovar su dirección ip cada cierto tiempo, lo que permite reutilizarlas. ” [2]

Los anteriores protocolos no se deberían denominar como descubridores de servicios, ya que no permite identificar un servicio por su nombre y usarlo con una

mínima configuración. Ya que se debe conocer la dirección-puerto que identifica el servicio en si.

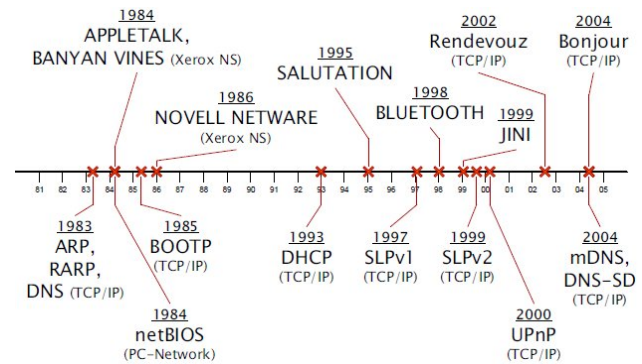


Figura 1. Evolución de los descubridores de servicios. [2]

Las dos primeras soluciones de descubrimiento de servicio para redes IP fueron el SLP (service location protocol) y Salutation, aunque estos dos fueron un gran avance todavía se poseía la limitante de instalar los controladores necesarios para los utilizar los servicios.

Con la llegada de la tecnología Bluetooth, nace la primera solución de descubrimiento de servicios específicamente para redes tipo PAN. El Bluetooth contiene SDP un mecanismo muy sencillo de publicación, descubrimiento y descripción de los servicios que se encuentran dentro de la red. Sin embargo el usuario debe instalar un software específico para cada de aplicación que se quiera hacer uso.

A mediados de septiembre de 1999 el grupo de trabajo Zeroconf de la IETF, intenta organizar las distintas tecnologías y protocolos diseñados hasta el momento. “ El objetivo fundamental era crear una arquitectura de descubrimiento de servicios que pudiera introducir esa facilidad de uso del viejo Appletalk a las redes IP ” . Todo este esfuerzo se vio reflejado en la creación de un documento request for comments RFC 3927.

El RFC 3927 describe un protocolo para asignar direcciones Ip dinámicas sin necesidad de usar un servidor central. Sin embargo esta funcionalidad ya la

estaba incorporada en los primeros ordenadores Macintosh desarrollados por Apple Inc.

El grupo Zeroconf no ha sido capaz de llegar a una solución definitiva con referente a los aspectos de descubridor de servicios, existen varias alternativas pero no se ha publicado ningún documento de carácter oficial, lo único que se llegó a un acuerdo fue la asignación automática de direcciones Ip (principalmente, las empresas Sun Microsystems, Microsoft Corp. y Apple Inc.)

De hecho, como veremos a continuación, cada una de estas compañías ha adoptado una solución distinta para resolver el problema planteado.

- Sun Microsystems presento la primera versión de su tecnología Jini en 1999.
- Microsoft Corp. y el UPnP Forum desarrollaron UPnP
- Apple Inc. lanzo Rendevous , una solución basada en las recomendaciones del grupo Zeroconf, Rendevous ha sido recientemente rebautizado como Bonjour.

Actualmente, los fabricantes de productos como impresoras, puertas de enlace residenciales o sistemas de entretenimiento comienzan a añadir funcionalidades de descubrimiento de servicio a estos. Diferentes productos serán compatibles con distintas soluciones de descubrimiento de servicios. Por ello, en este caso cobrarán una especial importancia los aspectos de interoperabilidad. Es por esto que han surgido ideas como la plataforma OSGi que intenta aglutinar varias soluciones de descubrimiento de servicios en una sola estructura modular. De esta forma se pueden desarrollar aplicaciones sin tener que preocuparnos por la arquitectura de descubrimiento de servicios subyacente. Con el comienzo del siglo ha cambiado la forma de usar Internet. Hoy en día son comunes las aplicaciones peer-to-peer de intercambio de archivos. Otros aspectos, como la multiplicación de los

servicios web o el aumento del número de accesos a Internet de banda ancha, han contribuido a esta transformación. El W3C tomo la decisión de crear una solución común para el descubrimiento de servicios web. El resultado fue el nacimiento de una alternativa a los sistemas RPC tradicionales llamada SOAP. ” [2]

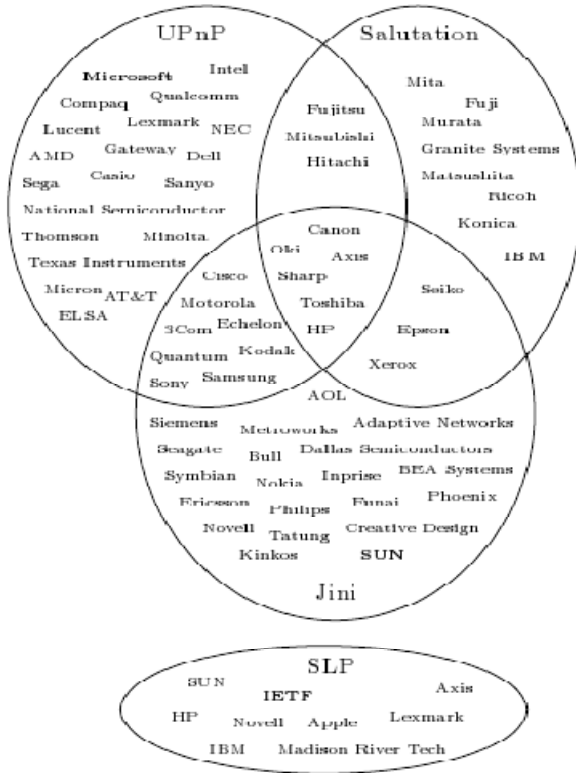


Figura 2. Empresas involucradas en el desarrollo de JINI, SALUTATION, UPnP y SLP [3]

## V. CARACTERISTICAS PRINCIPALES DE UN DESCUBRIDOR DE SERVICIO

### a. ACTORES PARTICIPANTES EN LOS PROTOCOLOS DE DESCUBRIMIENTO DE SERVICIOS

Un descubrimiento de servicio hace uso de un protocolo específico, este protocolo debe determinar cómo es la estructurada de los mensajes que se deben intercambiar a los diferentes dispositivos, existen unos mensajes destinados a identificar que servicios existen y otros tienen el objetivo de ofrecerlo a otros dispositivos que se encuentren en la red.

Para todo esto suceda se necesitan de los siguientes participantes:[4]

- Cliente: Es la entidad que requiere uno o varios servicios específicos ofrecidos por la red para ser utilizados en alguna aplicación del dispositivo.
- Servidor: Es la entidad que posee una o varias características o servicios que pueden servir a los demás dispositivos de la red. Por lo tanto es el agente que ofrece dichos servicios a través de la red.
- Directorio: Es un servidor que almacena de manera parcial o completa la información sobre los servicios que ofrece cada uno de los nodos que componen la red.

b. ARQUITECTURAS

Con respecto a la forma como se tiene distribuida la información sobre los servicios que ofrecen los diferentes dispositivos que se encuentran en la red, los protocolos de descubrimiento de servicios se pueden dividir en tres arquitecturas:

- Basada en directorios  
En esta arquitectura uno o varios dispositivos de la red toman la función de un directorio de servicios, estos registran la información obtenida por los servicios ofrecidos que les mandan los demás dispositivos.

Así, si un dispositivo necesita de un servicio puede solicitarlo directamente al directorio de servicio y este lo dirigirá al nodo que lo presta. De esta forma no se tendría que estar enviando mensajes por toda la red hasta encontrar el servicio necesitado.

- Distribuida  
Los nodos que hacen el papel de cliente y servidores se comunican directamente entre sí para conocer los servicios que se ofrecen en la red sin necesidad de

utilizar nodos directorios. Siendo así los nodos no necesitan estar sincronizándose constantemente para mantener la consistencia de su base de datos, sin

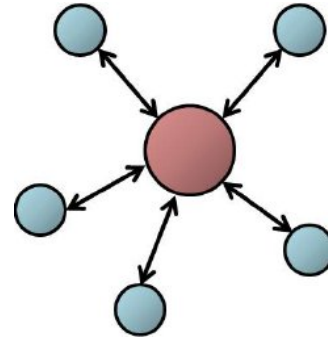


Figura 3. Arquitectura basada en directorio. [4]

embargo al no tener intermediarios las peticiones y los avisos deben ser enviados a todos los nodos de la red incrementando el tráfico de mensajes y el ancho de banda.

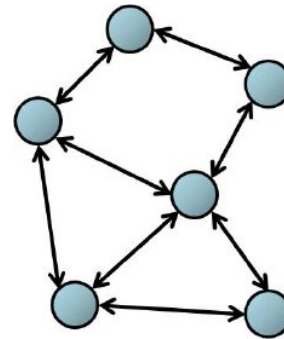


Figura 4. Arquitectura distribuida [4]

- Híbrida  
Nace por la combinación de la arquitectura basada en directorio y distribuida, la información de los servicios se encuentra tanto en los nodos directorios como en los nodos servidores.

“Si un cliente tiene a su alcance a un directorio, elegirá a éste para obtener la información que requiere como en la arquitectura basada en directorios. Si esto no es posible el cliente se comunicará directamente con el servidor como en la arquitectura distribuida. De esta manera se evita que existan puntos críticos de fallo en la red, ya que si un

directorio falla el cliente siempre tiene la posibilidad de conectarse con el servidor directamente.”[4]

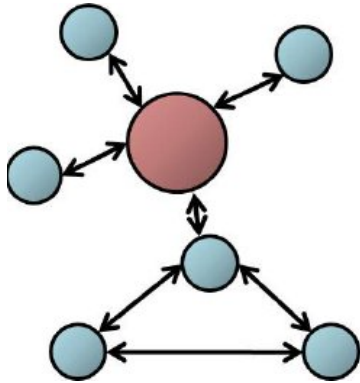


Figura 5. Arquitectura híbrida [4]

### C. MODO DE FUNCIONAMIENTO DE UN DESCUBRIDOR DE SERVICIO

Según la manera como se comunican los nodos clientes y servidores a la hora de solicitar y anunciar los servicios que ofrecen independiente de la arquitectura en que se estén, se considera que existen dos modos de funcionamiento.

1. Modo push (proactivo): Los servidores envían mensajes a los clientes de la red, anunciando los servicios que estos dispones. Los mensajes son enviados a todos los clientes sin que estos últimos lo hayan solicitado previamente.
2. Modo pull (reactivo): El cliente envía una solicitud buscando un servicio que necesita por toda la red (broadcast sino existe un directorio o unicast si existe un directorio).

## VI. DESCRIPCIÓN DE LOS PROTOCOLOS DE DESCUBRIMIENTO DE SERVICIOS

### a. JINI

Jini fue desarrollado por Sun Microsystems, este protocolo está basado bajo la tecnología java, en esencia la estructura de Jini contiene 3 protocolos:

1. Descubrimiento.

2. Registro.
3. Búsqueda.

Los dos primeros empiezan su funcionamiento cuando se conecta un dispositivo a la red, el descubrimiento sucede cuando un servicio necesita la ubicación de otro en el que pueda registrarse, mientras el registro tiene lugar cuando un servicio localiza otro que quiere registrarse.

El tercer protocolo (búsqueda) ocurre cuando un cliente o usuario localiza e invoca un servicio descrito desde su interfaz.

Jini contiene un modelo de programación que gestiona la manera en que los dispositivos se conectan para formar una comunidad desplazando el su código a través de toda la red. Jini usa java Remote Method Invocation (RMI) para mover el código.

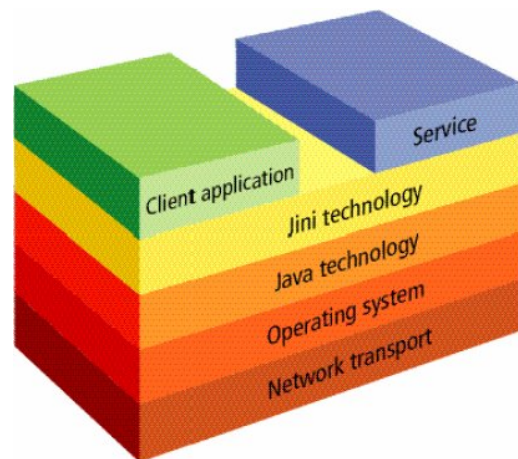


Figura 6. Arquitectura Jini [5]

### b. SALUTATION

La arquitectura SALUTATION proporciona un método estándar para aplicaciones, servicios y dispositivos para describir y anunciar sus capacidades a otras aplicaciones, servicios y dispositivos. La arquitectura posibilita búsqueda y descubrimiento basado en capacidades particulares.

Su estructura está compuesta por dos componentes:



- Salutation manager: facilita una interfaz de transporte independiente a las aplicaciones de cliente y servidor, esta interfaz (SLM-API) incluye registros de servicios, descubrimiento de servicios y funciones de acceso a servicios.
- Transport manager: es una entidad que depende de la red de transporte.

La interfaz entre el Salutation manager y Transport manager (llamado SLM-TI) obtiene independencia de protocolos de comunicación en la arquitectura SALUTATION.

Este protocolo no tiene una importancia notable debido a que se dejó de trabajar en él y surgieron otros protocolos que cumplían su función y gozaron de mayor presencia en el mercado.

### C. ZEROCONF Y BONJOUR

Bonjour, es una de las soluciones más destacadas para descubrimiento de servicio en redes de área local. Zeroconf usa DNS multicast para alcanzar el objetivo de eliminar las configuraciones en el descubrimiento de servicio. Zeroconf o Zero Configuration Networking es un conjunto de técnicas del IETF que permiten crear de forma automática una red IP sin configuración o servidores especiales.

La tecnología Zeroconf es una de las soluciones más extendidas para descubrimiento de servicio en redes de área local. Bonjour es la implementación de Zeroconf de Apple, y es una parte integral del sistema operativo MAC OS.

Bonjour también se instala en un gran número de sistemas operativos Windows, gracias a la popularidad de iTunes (aplicación de reproducción de música de Apple), el cual instala Bonjour como parte del proceso de instalación.

### d. SERVICE LOCATION PROTOCOL (SLP)

El Service Location Protocol es un estándar del Internet Engineering Task Force (IETF),

descentralizado, ligero y extensible de descubrimiento de servicio. Usa URLs de los servicios, que definen el tipo de servicio y direccionan un servicio particular. Por ejemplo "service:printer:lpr://hostname" es una URL para un servicio de "impresión" disponible en "hostname". Basado en la URL del servicio, los usuarios (o aplicaciones) pueden ver servicios disponibles en su dominio y seleccionar y usar el que se desee.[3]

En SLP existen 3 tipos de entidades:

- User Agent: Entidad que emplea SLP para hacer la petición y obtener un servicio, normalmente para realizar la petición se utiliza la URL.
- Service Agent: proceso que trabaja para uno o más servicios para publicar la información de dichos servicios, a su vez tiene este está encargado de registrar la información de los servicios en el directory Agent.
- Directory Agent: este se encarga de recoger la información de los agentes de servicios para crear un repositorio centralizado de información de servicios.

El objetivo del protocolo SLP es la localización dinámica de servicios, dentro de una red local, con un dominio administrativo común. Mediante este protocolo se intenta evitar que los servicios tengan que estar pre-configurados en cada máquina cliente, o que los usuarios tengan que recordar en qué máquina se encuentra cada servicio.[3]

### e. BLUETOOTH (SDP, SERVICE DISCOVERY PROTOCOL)

Bluetooth no surge como una arquitectura orientada a servicios, sino que se trata de un protocolo inalámbrico iniciativa de Ericsson en 1994 con el propósito de sustituir los cables entre los dispositivos electrónicos. En 1998, se unen a esta iniciativa IBM, Toshiba, Nokia e Intel.

Además del protocolo de comunicación propiamente dicho, se han definido una serie de protocolos a nivel aplicación que facilitan el desarrollo de aplicaciones Bluetooth. Entre ellos se encuentra el protocolo de descubrimiento de servicios (Service Discovery Protocol SDP).

Este protocolo permite a un dispositivo Bluetooth conocer cuáles son los servicios que proporcionan los dispositivos con lo que tiene conectividad. El protocolo sigue un esquema clásico cliente/servidor en el que la descripción de los servicios se almacena como un registro en el dispositivo que actúa como servidor SDP.

SDP soporta búsquedas por clases de servicios, y por atributos de servicios.

### f. UNIVERSAL PLUG AND PLAY (UPNP)

UpnP es una simple extensión del modelo Plug and play de los sistemas operativos Windows, plug and play facilita la instalación y configuración de un periférico (impresora, teclado, mouse, etc.) en un sistema Windows.

UPnP está diseñado para ampliar esta simplicidad a toda la red permitiendo el descubrimiento y control de los dispositivos de red como por ejemplos impresoras de red, dispositivos wireless, pcs de todo tipo. UPnP está diseñado para soportar la configuración-cero, establecimiento de una red invisible y descubrimiento automático para una gran variedad de categorías de dispositivos de una amplia gama de proveedores.

UpnP usa el estándar TCP/IP y otros protocolos de internet, dando una funcionalidad total en las redes existentes.

“Además, como UPnP es una arquitectura distribuida y abierta de red, definida por protocolos, es independiente del sistema operativo, lenguaje de programación o el medio físico. UPnP tampoco especifica las APIs de las aplicaciones que usará,

permitiendo que los proveedores del sistema operativo creen los APIs que satisfagan las necesidades de sus clientes.”[3]

### g. JXTA

JXTA es un conjunto de estándares abiertos que permiten a cualquier dispositivo conectado a la red, desde teléfonos celulares y PDA inalámbricos hasta computadores PC y servidores comunicarse y colaborar según el modelo P2P.

Los objetivos de JXTA, por tanto, son similares a JINI. Sin embargo, JXTA no está asociado a ningún lenguaje de programación concreto y se basa en XML en vez de emplear mecanismos de serialización de objetos tipo RMI.

La interoperabilidad es el objetivo central, y JXTA proporciona mecanismos que permiten que los peers se localicen unos a otros y se proporcionen servicios a través de diferentes plataformas y redes.[1]

## VII. CONCLUSIONES

Hemos visto como la computación ha evolucionado a lo largo del tiempo, en sus comienzos solo se concebía con dispositivos estáticos que solo se comunicaban con un grupo limitado de nodos conocidos que hacían parte de una red previamente ya configurados por una persona especializada, estos dispositivos no eran movidos físicamente de un lugar a otro para evitar la constante configuración manual que se requería.

Más tarde este paradigma empieza a tomar un giro donde no solo se necesita un dispositivo que nos facilite procesar datos, sino que podamos llevarlos con nosotros mismo a diferentes lugares y a su vez compartir la información con otros dispositivos para obtener una retroalimentación de información.

Sin embargo el transportar nuestros dispositivos a otros ambientes requieren configuración manual

para poder hacer parte de un grupo de trabajo y que entre ellos exista una armonía en la comunicación.

Cabe resaltar que se ha avanzado desde los primeros inicios hasta nuestros días en el aspecto de la comunicación y configuración manual mínima que se debe hacer a los dispositivos por intervención humana, sin embargo no se ha llegado a una conclusión y unificación en los mecanismos que las empresas nos brindan para tener una comunicación fiable y fácil de manipular, por lo que cada compañía una quiere regir el mercado con sus respectivas tecnologías que se brindan en el mercado.

En un ambiente de computación ubicua es indispensable tener un mecanismo de descubrimiento y anuncio de servicio debido al constante cambio de entornos, al creciente número de dispositivos y su variedad en ellos.

En los ítems anteriores se han mencionado algunas de las soluciones planteadas para el descubrimiento de servicios (unos hacen parte de un protocolo de descubrimiento como tal mientras que otras forman parte de la arquitectura de un framework de desarrollo de servicios).

Muchas de las soluciones han tenido en cuenta la problemática principal “dinamismo del entorno por la movilidad de los nodos”, la gran mayoría de las soluciones no han obtenido una solución con las limitaciones de los dispositivos que ingresan en estos entornos. Al tener un ambiente de trabajo tan dinámico los servicios estarán disponibles un tiempo limitado, por lo cual se necesita identificar de ante mano la disponibilidad e indisponibilidad de dichos servicios.

Algunas de las características que se necesitan en una tecnología de descubridor de servicio son:

1. Capacidad para funcionar con dispositivos limitados. [6]

2. Minimizar el número y la cantidad de información transmitida para que se adapten los dispositivos.
3. Adaptación a entornos dinámicos como estáticos.
4. Integración de variedad dispositivos a la red.
5. Multiplataforma.

### REFERENCIAS

- [1] J. F. Colom López y F. Maciá Pérez, Modelos de Servicios para Negocio Electrónico, 2006
- [2] José M. Sáhez Santana, universidad de Málaga, “Evaluación del protocolo de descubrimiento de servicios UPnP en redes inalámbricas”, 2005.
- [3] Iria Quiroga García, Universidad Carlos III de Madrid, “Diseño e implementación java del extensible service registration protocol (XSRP)”, 2009.
- [4] Fernando Hoyos Leyva, Universidad Carlos III de Madrid, “Descubrimiento de servicios y routing en redes Manet con dispositivos Maemo”, 2009.
- [5] Sánchez Arias, D. Universidad de las Américas Puebla, “Domótica: diseño de una casa inteligente basado en la tecnología Jini”. 2004
- [6] Celeste Campo Vázquez, Universidad Carlos III de Madrid, “Tecnologías middleware para el desarrollo de servicios en entornos de computación ubicua”, 2003.