



**I
N
A
O
E**

Integración de Redes y Protocolos con SDR (Software Defined Radio)

Jorge Estudillo-Ramírez, René Cumplido

Reporte Técnico No. CCC-00-004
18 de julio de 2005

© **Coordinación de Ciencias Computacionales**
INAOE

Luis Enrique Erro 1
Sta. Ma. Tonantzintla,
72840, Puebla, México.



Integración de Redes y Protocolos con SDR (Software Defined Radio)

Jorge Estudillo Ramírez¹, René Cumplido²

^{1,2}Coordinación de Ciencias Computacionales,
Instituto Nacional de Astrofísica, Óptica y Electrónica,
Luis Enrique 1, Sta. Ma. Tonantzintla,
72840, Puebla, México

¹jestudillo@ccc.inaoep.mx, ²rcumplido@ccc.inaoep.mx

Abstract. Software defined radio is a concept that it is revolutionizing the technology of the communications toward the integration of multiple architectures of wireless telecommunications nets. However, the hardware reconfiguración through software also demands a similar behavior of the software oriented to regulate the exchange of information. In other words, it is important to have outlines that allow the reconfigurability of the software of protocols used in the current and future communications for their internetworking. In this work it will be analyzed the current structure of the protocols stacks and their communication. The importance of the APIs is profiled with regard to the outline based on SAPs of the current protocols stacks. Finally, three focuses are approached that they try directly on aspects of software of protocols reconfigurability, giving with this a future vision toward the integration between dynamic communication hardware - dynamic software of protocols.

Resumen. Radio definido por software es un concepto que esta revolucionando la tecnología de las comunicaciones rumbo a la integración de múltiples arquitecturas de redes de telecomunicaciones inalámbricas. Sin embargo, la reconfiguración del hardware a través de software exige también un comportamiento similar del software destinado a regular el intercambio de información. En otras palabras, es importante contar con esquemas que permitan la reconfigurabilidad del software de protocolos empleados en las comunicaciones actuales y futuras para su intercomunicación. En este trabajo se analiza la estructura actual de las pilas de protocolos y su comunicación. Se perfila la importancia de las APIs con respecto al esquema basado en SAPs de las pilas de protocolos actuales. Finalmente, se abordan tres enfoques que tratan directamente sobre aspectos de reconfigurabilidad de software de protocolos, dando con ésto una visión futura hacia la integración entre hardware de comunicación dinámico – software de protocolos dinámico.

Palabras clave. Radio Defined Software, Protocols Integration, reconfigurability, Active Network, Flexible Protocol Stacks, Adaptive Protocols, Multiprotocol Network.

1. Introducción.

El término “Software Radio” fue acuñado por Joe Mitola en 1991 para referirse a la clase de “radios” re-programables o re-configurables [1]. En otras palabras, una misma pieza de hardware puede realizar diferentes funciones en diferente tiempo. Con lo anterior podemos especular el hecho de tener un dispositivo hardware de “propósito general” en un ámbito de comunicaciones. Este concepto abre un amplio panorama de oportunidades para la industria de las comunicaciones y la investigación en ese mismo sentido.

En [1] se define SDR como “Software Radio es un radio sustancialmente definido en software y cuyo comportamiento en capa física puede ser significativamente alterado a través de cambios en su software”. Otra definición dada por [2] dice que “SDR es un radio en el cual la digitalización en el receptor es realizada en algunos escenarios de bajada desde la antena, típicamente después del filtrado del ancho de banda, reducción de amplificación de ruido, y la conversión a bajas frecuencias en subsecuentes escenarios (con un proceso inverso en la digitalización del transmisor)”. En [3] “término utilizado para describir control de software para una aplicación de radio el cual provee técnicas de modulación, operaciones de banda angosta y banda ancha, funciones de seguridad en comunicaciones y requerimientos de forma de onda”.

Desde la creación y evolución constante de estándares tales como 2.5G, 3G y 4G, existe una amplia incompatibilidad entre tecnologías de redes inalámbricas utilizadas por diferentes países. Desde una perspectiva comercial y global, este problema inhibe el uso de servicios de roaming y otras facilidades [3]. Esta es una de las grandes oportunidades de SDR, ya que el aspecto económico es de suma importancia, para la integración transparente de diversas arquitecturas de comunicación.

Sin embargo, para considerar la integración de múltiples arquitecturas de comunicaciones no sólo se debe tomar en cuenta el nivel de hardware sino también la parte de software que establece los pasos a seguir para realizar una comunicación. Los protocolos o pila de protocolos en capas superiores del modelo OSI permiten la comunicación y la coherencia de la misma en una comunicación de datos. La comunicación de la capa física –nivel hardware- hasta la capa de aplicación – software – está directamente ligada y es imposible pasar de un nivel al otro sin requerir de servicios y funciones especificados en capas intermedias.

Este trabajo trata de la orientación de la parte software con respecto al escenario dinámico de cambio que propone la tecnología SDR. Específicamente se trata de dar una respuesta a la pregunta: ¿Qué sucederá con el protocolo o pila de protocolos requeridos para complementar un proceso completo de comunicación entre dos aplicaciones, en un ambiente en donde el hardware puede adoptar cambios temporales de manera dinámica?

2. Perspectiva de Protocolos en el Ámbito SDR.

2.1 Pila de Protocolos y su comunicación.

Una pila de protocolos es una agregación de muchos protocolos simples donde cada uno implementa en una pila cierta funcionalidad y servicios para una tarea claramente especificada [4]. Los protocolos en general se clasifican mediante mecanismos de estratificación con base en su funcionalidad y servicios ofertados. Cada capa puede ser vista como una caja negra en la que no puede existir redundancia de funcionalidad y puede contener un grupo de protocolos provenientes de diferentes desarrolladores o protocolos propietarios.

El principio de estratificación para acceso a servicios o recibir servicios es utilizado en la mayoría de redes de telecomunicaciones móviles y fijas. Funciones o servicios de protocolos son ofrecidos o recibidos de capas inmediatamente vecinas. La comunicación entre las diferentes capas de una pila de protocolos es realizado vía lo que se denomina “Punto de Acceso al Servicio” (SAP). Un SAP provee acceso vía un conjunto de primitivas que habilitan el acceso a determinadas capas vecinas.

Los servicios de protocolos son un conjunto de operaciones realizada en una capa particular de un protocolo [4]. Un servicio para protocolo es, como el nombre sugiere, un servicio. Eso significa que no se pasan mensajes “a través” del servicio, dado que desde un punto de vista lógico

no es una etapa en el flujo de mensajes. Un servicio exporta una interfaz bien definida, y las etapas que necesitan su funcionalidad simplemente llaman a subprogramas en ella, como harían con cualquier otro módulo software normal (por ejemplo, uno para la entrada/salida por consola) [5]. La descripción de un servicio de protocolo define solamente el tipo de operación que una capa de protocolo particular debe ofrecer al siguiente nivel vía un SAP inherente (Figura 1).

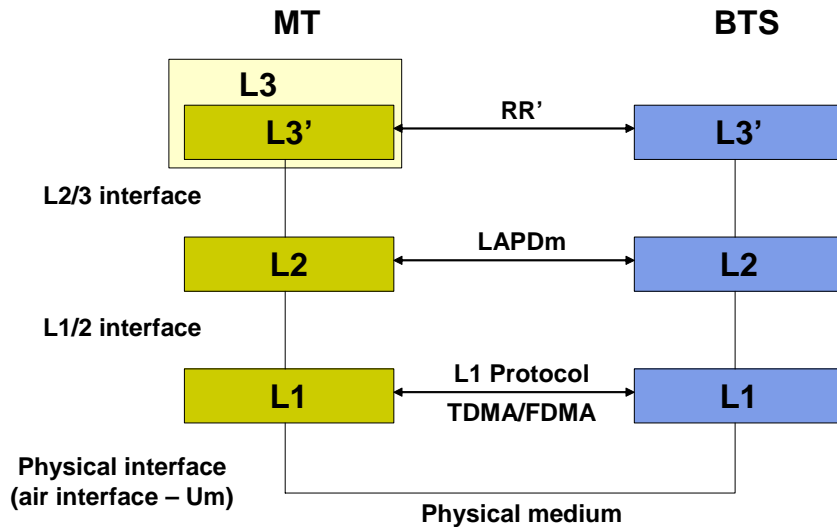


Figura 1. Esquema de comunicación por capas [4].

En la figura 1 podemos ver que la capa 2 (L2) ofrece un servicio a su capa superior vecina (L3) a través de una interfaz SAP (L2/3). Podemos observar también que existe una comunicación virtual capa – capa en donde se hace necesario utilizar un mismo protocolo.

Un punto de acceso al servicio no solamente provee servicios para su capa de protocolo correspondiente, también es útil para ocultar la complejidad de la implementación y descripción de una capa particular. Las definiciones de servicio se desarrollan de manera formal desde la capa inferior (física) hacia arriba, hasta la capa superior. La ventaja de este enfoque es que el diseño de la capa $N + 1$ puede basarse en las funciones ejecutadas en la capa N , evitando dos funciones que realizan la misma tarea en dos capas adyacentes. Un punto de acceso al servicio identifica que funciones de la actual capa de protocolos será requerida por una capa superior o vecina. En la pila de protocolos, un SAP es identificado por una única dirección (puerto).

La introducción de puntos de acceso al servicio y su estandarización fue un gran paso hacia plataformas abiertas [4]. Lo anterior debido a que la estandarización ha habilitado implementaciones propietarias de funcionamiento de protocolos manteniéndolas ocultas y con esto habilitando implementaciones de protocolos de diferentes proveedores haciendo que éstos co-existan en una pila de protocolos, ya que únicamente interesa acceder al servicio sin la necesidad de conocer detalles de cómo será realizado.

Sin embargo, una de las mayores desventajas de SAPs es su carencia de flexibilidad. Cualquier cambio, alteración o mejora en la pila de protocolo, requiere que SAPs y protocolos deban ser re-estandarizados. Un ejemplo de re-estandarización es la evolución gradual del estándar GSM hacia el Servicio Generalizado de Paquetes por Radio (GPRS), el cual requirió un nuevo diseño de pila de protocolos.

2.2 APIs vs SAPs.

A simple vista concluiríamos que servicios proporcionados vía SAPs y vía Programación de Interfaces son iguales. Sin embargo, existen un número de diferencias entre las dos técnicas que deben ser comprendidas. Se muestra primeramente los puntos importantes que consideran cada uno de éstos:

Punto de Acceso al Servicio.

- Número de puertos.
- Ocultamiento de complejidad.
- Un paso hacia plataformas abiertas.
- Poca flexibilidad.
- Un cambio menor provoca una re-estandarización.

Interfaz de Programación de Aplicaciones.

- Clases (definición e implementación).
- Organización jerárquica.
- Permite la aplicación de métodos y técnicas orientadas a objetos.
- Extensión y herencia.
- Permite la compatibilidad entre versiones diferentes de la misma interfaz.

Podemos ver que mientras SAPs son definidos como puertos para acceder las primitivas definidas en una capa, programación de interfaces son definidos en interfaces de clases relacionados a una implementación actual de la capa. El punto importante de las APIs es su capacidad de extensión y herencia que permite partir de interfaces genéricas hacia una implementación más específica (Figura 2).

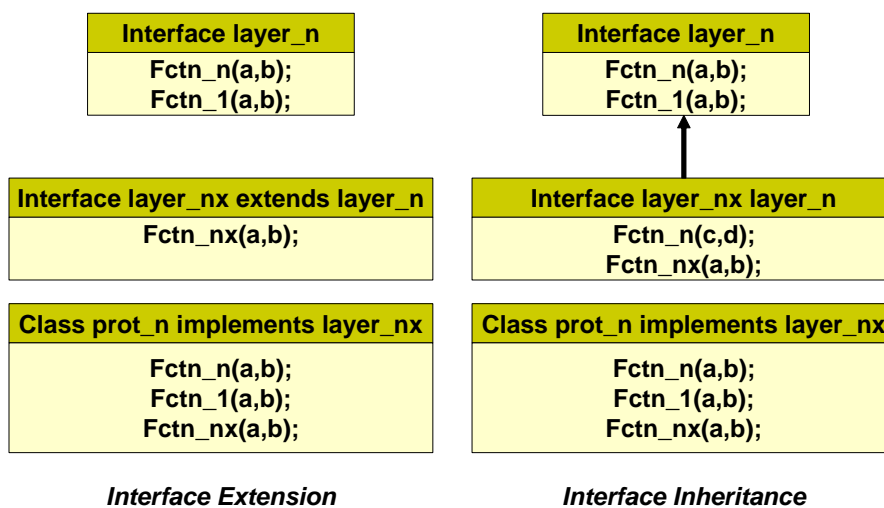


Figura 2. Extensión y herencia de una interfaz [4].

En la figura 2, la columna izquierda muestra la definición de una interfaz genérica (llamada layer_n), su extensión en “layer_nx” y finalmente su implementación en “prot_n”. La columna derecha muestra la misma interfaz genérica; sin embargo, esta vez “layer_nx” hereda la funcionalidad de “layer_n”, sustituyendo algunas definiciones por las de la interfaz original, finalmente, la interfaz es implementada en “prot_n” [4].

Lo anterior hace a la interfaz extensible en varios sentidos permitiendo, sobre todo, la compatibilidad entre versiones diferentes de la misma interfaz. Lo anterior significa que un protocolo mejorado en alguna capa de una pila de protocolo, puede ser compatible con una versión anterior del mismo protocolo en otra pila de protocolo. Lo anterior no se mantiene con SAPs debido a su carencia de flexibilidad.

3. Re-configuración y Configuración de Protocolos.

Una visión futura de la integración de diferentes sistemas de comunicación considerando como base a SDR es la re-configurabilidad de software de red [4], [6] y [7]. En la figura 3 podemos observar un proyecto real en el que se implementa esta idea. Este proyecto se llama End to End Reconfigurability (E2R).

El proyecto End-to-End Reconfigurability apunta hacia un total beneficio de la valiosa diversidad en sistemas de radio, formado de un amplio rango de sistemas tales como celular, redes locales inalámbricas y broadcast. El objetivo de E2R es crear, desarrollar y probar diseños arquitecturales de dispositivos reconfigurables y funciones para dar soporte a sistemas y ofrecer un conjunto extendido de opciones a los usuarios, proveedores de servicios y aplicaciones, operadores y reguladores en el contexto de sistemas heterogéneos de radio móviles [7].

Los protocolos de sistemas abiertos son definidos y estandarizados de una forma muy rígida resultando en poca reconfigurabilidad o capacidad para adaptarse a cambios de requerimientos en la comunicación. La construcción de pilas de protocolos más versátiles es necesaria para permitir y simplificar reconfigurabilidad de terminales, así como re-conocer ambientes de comunicación integrados. De esta forma se destacan tres enfoques orientados hacia el concepto de reconfigurabilidad software.

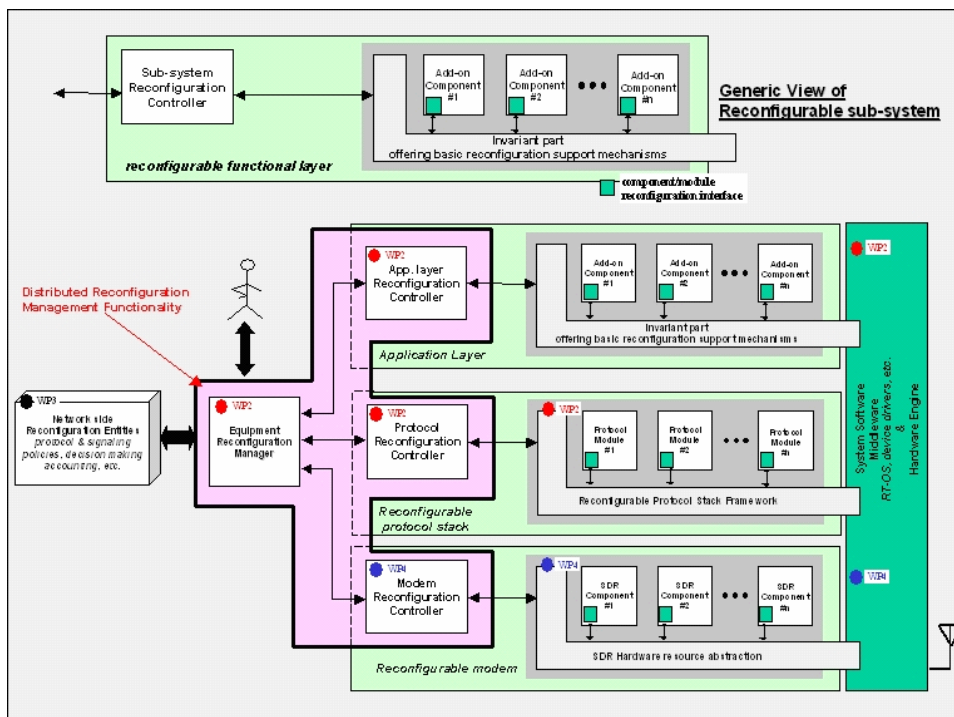


Figura 3. End-to-End Reconfigurability [7].

Los enfoques que abordan la reconfigurabilidad de software para la integración de múltiples sistemas de comunicación basados en radio son: Adaptable, “Composable” y Protocolo reconfigurable [4]. En otras palabras son enfoques que abordan la problemática de no permitir la configuración de muchos protocolos, sino permitir la configuración de cualquiera.

Los enfoques de protocolos adaptable y “composable” son técnicas que proporcionan versatilidad. La base de estas dos tecnologías es el hecho de que protocolos o capas en la pila de protocolos, son una mera aglomeración de numerosas funciones de protocolos simples los cuales pueden ser implementados independientemente de sus capas asignadas. Estos dos enfoques retoman el principio de funciones de protocolos básicas combinados en una pila de protocolos genérica o ensamblándolos durante el periodo de “boot time”, respectivamente.

Considerando el modelo OSI como guía, los enfoques adaptable y “composable”, para la clasificación de funcionalidades en un grupo de funciones de protocolos, las funcionalidades de la capa física y las capas de aplicación y presentación no deben ser consideradas como parte del grupo de funciones genéricas debido a sus altas dependencias a la plataforma hardware y sistemas de operación respectivamente. Las funciones de las otras capas de protocolos son distribuidas de acuerdo a la tarea particular de cada capa.

3.1 Protocolos Adaptables.

Este enfoque consiste, en principio, de capas de protocolo genérico que implementa un conjunto de funciones de protocolos comunes [4]; y de una segunda parte que implementa extensiones estandarizadas a dicho protocolo genérico o común generando, de esta forma un protocolo específico. La figura 4 muestra el principio de este protocolo adaptable, utilizando GSM, DECT y UMTS como ejemplos. La implementación práctica de este esquema resulta en un esqueleto-pila de protocolo conteniendo todos los elementos comunes de esos protocolos para implementar uno de ellos. Mediante la extensión de funciones específicas de algún protocolo (GSM, DECT, UMTS) a la pila de protocolos común, se tienen uno de los tres protocolos específicos mencionados en este ejemplo.

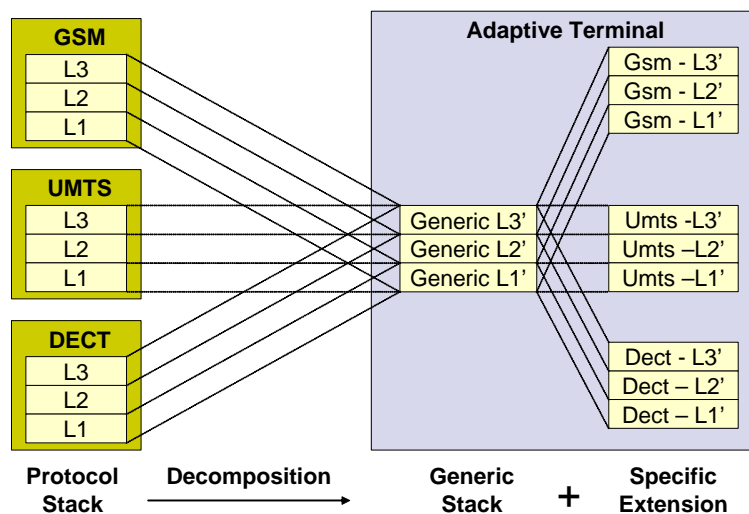


Figura 4. *Enfoque Adaptable* [4].

En la figura 4 se puede observar que de las tres capas de cada protocolo (L1, L2, L3) se tiene una pila genérica (L1', L2', L3'). En cada capa de esta pila genérica se agrupan funciones de

protocolos simples que son similares a la misma capa de cada pila de protocolo original correspondiente (GSM, UMTS, DECT). Por otro lado, aquellas características que son diferentes de cada capa, es decir que no se corresponden en similitud al mismo nivel, se conservan en pilas más pequeñas, denominadas extensiones específicas. De este modo, al tener una pila genérica y ligarla a una de las extensiones específicas se tiene como resultado una única pila de protocolo de propósito específico. Así, se tiene un esquema de reconfiguración de pila de protocolos en el que únicamente basta agregar extensiones mínimas para tener un esquema completo de comunicación. La desventaja de este esquema es el hecho de tener pocas funciones comunes entre las diferentes pilas de protocolos a integrar. Poca similitud trae consigo una pila genérica muy pobre y en cambio pilas de extensiones específicas de alto volumen. Mayor volumen de transmisión para actualizar un protocolo hace de este esquema bastante ineficiente, ya que el tiempo de transmisión es mucho mayor y en consecuencia el tiempo de reconfiguración.

3.2 Protocolos “Composable”.

La funcionalidad de protocolos y pilas de protocolos completas pueden ser divididos en funciones de protocolos simples y un grupo de estas funciones para construir pilas de protocolos personalizadas o estandarizadas durante el periodo de “boot time”. Uno de los proyectos que han sido iniciados para explorar e implementar este principio es “DaCaPo” [4].

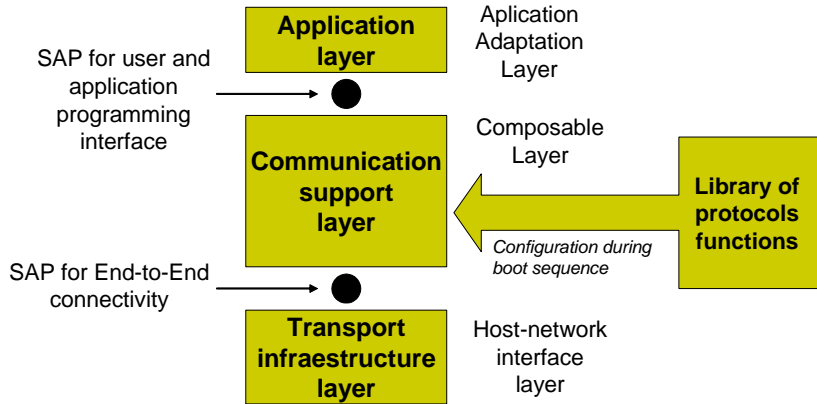
DaCaPo es un trabajo de dominio público que implementa configuración de protocolo durante el tiempo de ejecución (boot time) en vez de en tiempo de compilación. La intención de este trabajo es crear protocolos estandarizados que provean de parámetros de QoS para la conexión actual/proyectada. La base de la arquitectura es una estructura de pila con un reducido número de capas: solo tres capas son definidas en DaCaPo.

- Capa A. Capa de aplicación.
- Capa C. Capa de apoyo a la comunicación.
- Capa T. Capa de infraestructura de transporte.

Mientras la capas A y T, las capas de adaptación, son dependientes de la aplicación y de los principales mecanismos de transporte (ATM, LAN, MAC, etc), respectivamente, la capa C es la capa configurable (figura 5). Esta capa esta formada de bloques de construcción de protocolos granular aglomerados, cada uno definiendo una tarea de protocolo simple.

DaCaPo emplea cuatro entidades co-operacionales para controlar los mensajes entre los bloques de construcción y la unión de éstos con la capa C. Estas cuatro entidades son:

- CoRA (Método para configuración y distribución de recursos). Determinación de configuración de protocolos adecuados en tiempo de ejecución.
- Administrador de conexión. Establecimiento de control, administración de error y liberación de conexión.
- Ambiente de coordinación de ejecución en tiempo de corrida (runtime) del procesamiento con la capa.
- Entidad para monitorear otros componentes y control de la disponibilidad de recursos en la comunicación y sistemas.



DaCaPo Project.

Figura 5. Protocolo “Composable” (DaCaPo) [4].

3.3 Pila Reconfigurable.

El enfoque de pila de protocolo reconfigurable esta basado en la re-definición de interfaces entre capas de protocolos, clasificación de interacción entre diferentes capas en la pila del protocolo y provisión de una arquitectura para dar soporte a la pila y configuración del protocolo [4]. Este enfoque introduce e implementa interfaces de programación activa en forma de objetos que son parte de la pila de protocolo, utilizando métodos de diseño orientado a objetos para definir esta arquitectura de pila de protocolos y sustituir implementaciones de protocolos durante tiempo de ejecución.

Interfaces de Programación de Aplicación (APIs) han sido utilizados por décadas en el campo de la computación para simplificar desarrollos de aplicaciones a alto nivel. Mientras tanto, en el campo de “networking” y las telecomunicaciones la necesidad de un beneficio potencial de interfaces abiertas comunes en la capa de aplicación ha sido ampliamente reconocida. Muchos proyectos de investigación han surgido para explorar la implementación y aplicación de interfaces y plataformas de programación abierta y sus usos en terminales móviles y nodos de red.

Utilizando interfaces de programación activa estandarizada a todas las capas, introduce un grado adicional de libertad para reconfigurar pilas de protocolos en terminales y redes. Una mayor función requerida para terminales SDR es la habilidad de intercambiar software de protocolo “en el vuelo”, implicando la re-configuración dinámica de la pila de protocolo. OPtIMA es un proyecto desarrollado en esta última dirección.

OPtIMA está basado en la separación (descomposición) de pilas de protocolo en un número de entidades funcionales. Estas entidades incluyen capas de protocolo (pro-layer), interfaces y threads, descritos en clases genéricas organizadas en librerías de clases las cuales permiten ligarse dinámicamente en tiempo de ejecución.

El uso de interfaces de protocolo activo agrega más complejidad al sistema, pero también proporciona la ventaja de intercambiar protocolo durante tiempo de ejecución más que durante el tiempo de compilación o “boot time”. Las interfaces de protocolos activos son objetos que encapsulan un conjunto de interpretación de mensajes y métodos de distribución de mensajes. La estructura se muestra en la figura 6.

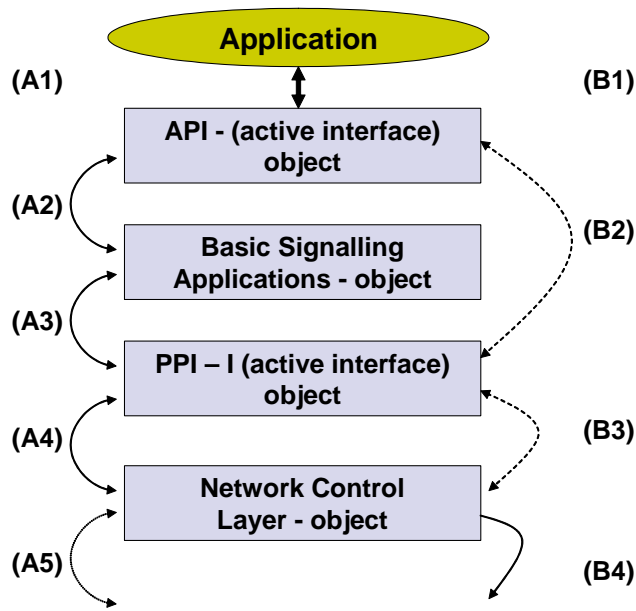


Figura 6. *Objetos de interfaz de protocolo activo* [4].

Los objetos de interfaz activa recuperan información requerida para el procesamiento de los mensajes desde la cabecera de los mensajes entrantes. El objeto de interfaz activa procesa o interpreta el mensaje y lo pasa al “pro-layer” destino. La figura 7 muestra este principio.

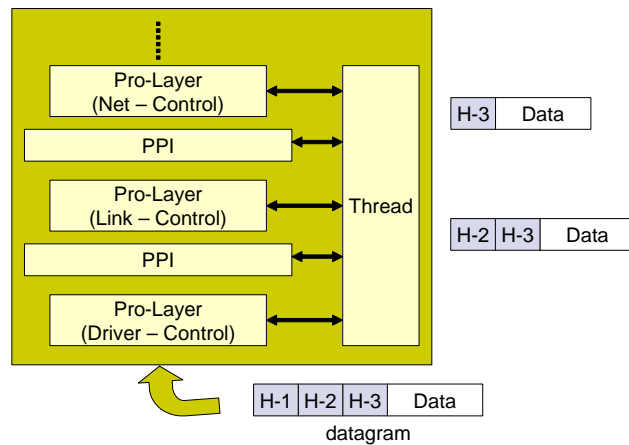


Figura 7. *Mensaje pasando en OPtIMA* [4].

En la figura 8, podemos observar un proyecto reciente, que se está realizando entre importantes compañías como Motorola, Panasonic, Siemens, y otras; Este proyecto se está desarrollando para integración de equipos SDR en redes móviles futuras [6]. Esto refuerza lo expuesto en este trabajo con respecto a la tendencia en la integración de SDR con los protocolos de capas superiores en las pilas de protocolos. Específicamente podemos ver la implementación de APIs para la comunicación entre capas, en donde no necesariamente debe ser con respecto a capas vecinas inmediatas.

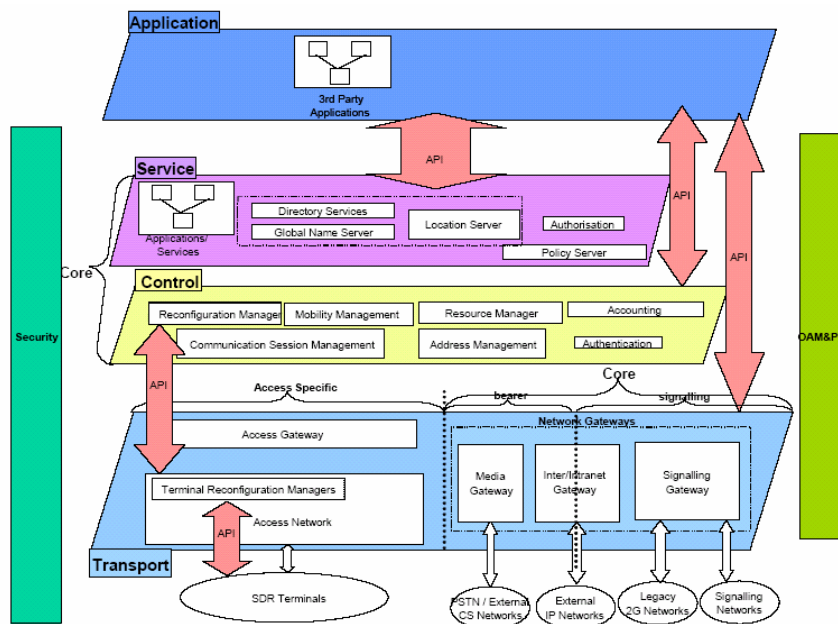


Figura 8. Proyecto Scout (Panasonic, Motorola, Seamens) [6].

3.4. Redes Activas.

Una red es considerada activa si permite que las aplicaciones inyecten en la red programas personalizados para modificar el comportamiento de los nodos de red [8]. Si se puede colocar funciones de procesamiento importantes en la red a través de sus dispositivos móviles, se puede adaptar conexiones específicas en tiempos específicos.

En una red activa, los paquetes de datos son entidades de información. Tales entidades llamadas **SmartPackets** [8], contienen las direcciones de origen y destino, datos, y algún tipo de “método” que puede ser ejecutado localmente. Así como las arquitecturas x-kernel permiten pilas de protocolos complejas partiendo de funciones simples de protocolos, se considera a los SmartPackets como portadores de funciones personalizadas simples de protocolos para mejorar módulos de protocolos en los nodos de red. Los nodos de la red son considerados activos en el sentido en que identifican SmartPackets, extraen los módulos de protocolos y los ejecutan. Estos módulos pueden ser totalmente nuevos o extensiones de funcionalidades de módulos bien conocidos.

Un ejemplo de este enfoque para acomodar diferentes protocolos y pilas de protocolos en un nodo de red es IEEE P1520 propuesto para redes activas. Este estándar apunta en la definición de un conjunto de APIs para redes. Las APIs son definidas para interpretar etiquetas de protocolos que están ligados a mensajes transmitidos en ambientes de red heterogéneas. La arquitectura P1520 se muestra en la figura 9. Esta arquitectura también considera el uso de interfaz de programación, por lo que es muy parecido al último enfoque tratado en este documento.

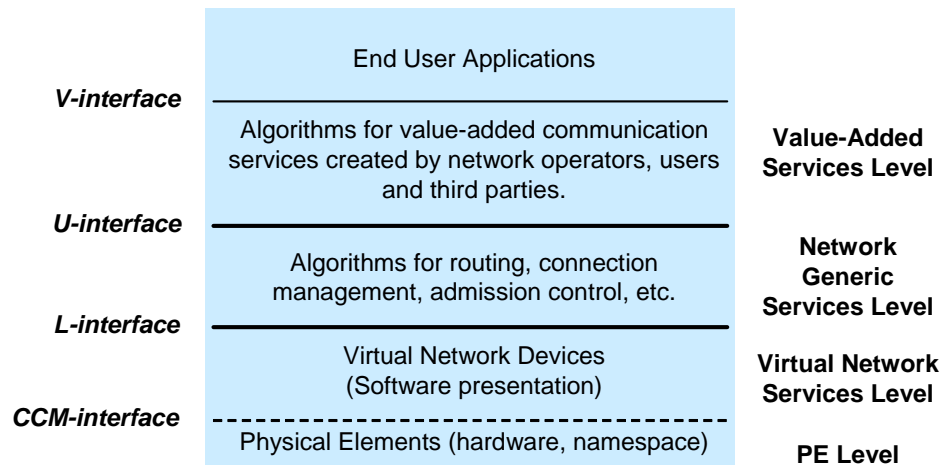


Figura 9. Modelo de referencia P1520 [4].

Los cuatro niveles representados en el modelo de la figura 9 se detallan a continuación [4]:

Nivel de servicio de valor agregado. Entidades funcionales en este nivel implementan algoritmos punto a punto para valorar servicios proporcionados por las capas inferiores. Estos valores pueden ser mejoras y capacidades orientadas a usuarios.

Nivel de servicio genérico de red. Las funciones implementadas en este nivel incluyen algoritmos que definen la funcionalidad general de redes; éstos son principalmente algoritmos de encaminamiento y ruta ya sea de camino virtual o circuito virtual.

Nivel de dispositivo de red virtual. Incluye las funcionalidades de representación lógica y abstracta de estados variables de entidades hardware en el nivel PE.

Nivel de elementos físicos. Son los elementos físicos actuales de la red, tales como switches, routers, o puntos finales de comunicación. Los elementos funcionales a este nivel son accedidos y controlados por protocolos, propietarios y protocolos de administración.

Las interfaces entre estos niveles funcionales marcan la separación entre aplicaciones de usuario final, valores de servicios agregados, servicios proporcionados de la red y los recursos principales de la red. El modelo de referencia P1520 identifica 4 interfaces, las cuales son:

V-interfaz. Es un nivel de usuario que proporciona APIs para escribir software de usuario personalizado; también provee acceso al nivel de valores de servicios agregados.

U-interfaz. Permite a los usuarios acceder a servicios de red genéricos tales como peticiones o terminación de conexiones o ancho de banda. Lo anterior dependiendo de la demanda del usuario y de la disponibilidad de recursos.

L-interfaz. Proporciona un conjunto de APIs que permiten el acceso directo a, y manipulación de, los estados de los nodos y recursos de la red local. También permite la implementación de cualquier servicio de comunicación.

CCM-interfaz. La interfaz de control de conexión y administración consiste de varios protocolos que habilitan el intercambio de estados e información de control entre los elementos físicos de hardware.

4. Conclusiones.

Como hemos visto, el uso de APIs permite una mayor flexibilidad con respecto a los SAPs existentes actualmente en la comunicación entre capas de una pila de protocolo. Por lo que se prevé su implementación, que actualmente se puede observar en proyectos como OPtIMA y Scout, en futuros protocolos tal como se muestra en el enfoque de protocolos reconfigurables. Los primeros dos enfoques, adaptable y “composable”, son altamente dependientes de las funcionalidades comunes de las pilas de protocolos a considerar en su base genérica, lo cual los hace candidatos no muy fuertes para consideraciones futuras, sin embargo, son dos opciones más hacia el concepto de reconfigurabilidad de software de protocolos.

La idea de contar con protocolos que se configuren “al vuelo” es muy atractiva y más aún la idea de que las funcionalidades entre capas estén perfectamente definidas y se considere el establecer comunicación entre cualquier par de capas. Esto último orientado a la convergencia entre la variedad de protocolos estratificados en niveles variables de capas.

Todo parece indicar que la dirección hacia donde se mueve el concepto de reconfigurabilidad de software de protocolos está orientado a la conservación de protocolos, creación de nuevos protocolos y modificación del esquema de comunicación intercapa. Las pilas de protocolos se mantendrán debido a la importancia de mantener comunicación con sistemas existentes. Aún que el esquema de estratificación se conservará, no así el de comunicación entre capas, el cual será sustituido por el uso de APIs en lugar de SAPs. En otras palabras, versiones nuevas de protocolos serán agrupadas en cada capa, según correspondan sus funciones, y será mediante APIs las que realizarán la comunicación entre cualquier capa, según sean requeridas, y dentro de la capa con la versión de protocolo o protocolo necesario para complementar la comunicación.

Con lo anterior, quizá, el uso de dispositivos de internetworking se haga innecesario. Si dos dispositivos terminales se pueden configurar para comunicarse “con un mismo tipo hardware” y además con “las mismas reglas de comunicación” (protocolos), entonces un dispositivo intermedio que hiciera las veces de gateway, empaquetando la información recibida en un tipo protocolo a otro tipo de protocolo, ya no es necesario.

Referencias.

1. Jeffrey H. Reed, *Software Radio: A Modern Approach to Radio Engineering*, Prentice Hall, Segunda Edición, 2002.
2. Stephen M. Blust, *Software Base Radio*, In: *Software Defined Radio – Enabling Technologies*, Editado por Walter Tuttlebee, Wiley, 2002.
3. Jon Bock, *Software Defined Radio*, 13 Sept 2004, URL: <http://sdr.compbuy.co.uk/content/view/9/27/>, Consulta: Julio 11, 2005.
4. Klaus Moessner, *Protocols and Network Aspects of SDR*, In: *Software Defined Radio – Enabling Technologies*, Editado por Walter Tuttlebee, Wiley, 2002.
5. Jesús M. González Barahona, *Arquitectura de Comunicaciones para Grupos de Procesos Replicados*, Tesis doctoral, Universidad Politécnica de Madrid, 1998.
6. Didier Bourse, Markus Dillinger, et al., *SDR Equipment in Future Mobile Networks*, Motorola Labs, Paris, France. URL: www.ctr.kcl.ac.uk/publications/papers/8_SDR%20Equipment%20in%20Future%20Mobile%20Networks.pdf, Consulta: Julio 5, 2005.

7. URL: <http://e2r.motlabs.com/> , Consulta: Julio 6, 2005.
8. Amit B. Kulkarni, Gary Minden, Victor Frost, Joseph Evans, *An Active Network Architecture for ATM WANs*, 1996.