

Capítulo 1

Estado del Arte

No sé lo que hace... pero debe ser muy complicado

Ana María del Rosario García Andrés

Este capítulo muestra un estudio de las plataformas móviles y su impacto en las redes de telecomunicaciones de tal forma que son el motor que empuja la evolución de éstas, acercándolas cada vez más a un enfoque cognitivo. Se describe qué son y qué ventajas aportan este tipo de redes, por último presentamos estudios actuales e implementaciones de estas redes en el campo de los teléfonos inteligentes.

1.1. Plataformas Móviles

La definición de un teléfono inteligente no es un concepto muy definido, en general un teléfono móvil que disponga de un sistema operativo completo y que cuente con una mayor conectividad que un dispositivo tradicional son características válidas para obtener la denominación de teléfono inteligente. Un hito que suele recoger ambas características es la de poseer de soporte completo al correo electrónico.

Se tiene por tanto una evolución desde el *firmware* propietario cuyo diseño está completamente adaptado al *hardware* del dispositivo desarrollando sólo las características indispensables y que vayan a ser usadas, a el desarrollo de un sistema operativo completo que abstraiga al *hardware* y dote de funcionalidad completa llegando a asemejarse a una computadora personal. Este sistema operativo puede ser de código abierto (**Ubuntu Touch**) o privativo (**BlackBerry OS**), impulsado por operadores (**Vodafone360**), fabricantes de dispositivos (**iOS**) o empresas de IT (**Android**) por citar algunos ejemplos.

Como consecuencia se tiene un dispositivo con una gran conectividad y almacenamiento de datos. Esto habilita la realización de una experiencia *web* completa e independiente. Precursores de los teléfonos inteligentes se valían de la ayuda de su ISP quién facilitaba la navegación *web* adaptando el contenido al navegador del dispositivo. De tal forma que páginas estándar son traducidas al protocolo WAP [1], a sus diferentes versiones: la versión 1.0 no soporta el protocolo HTTP, por lo que el uso de una pasarela se hace indispensable y su versión 2.0 la pasarela o *content adapter* completa la adaptación que sigue siendo necesaria pues tanto el contenido como protocolos usados para proveerlo no estaban pensados para un

mundo móvil, ejemplo de un *content adapter* es **OpenWave** usando en la operadora Vodafone. A esta tecnología le sucedió *i-mode* [2] pero no es hasta 2007, cuando un móvil incorpora el soporte nativo del modelo tradicional TCP/IP, cuando se puede hablar de una experiencia completa, dándole la denominación completa de teléfono inteligente.

El marco de conectividad se completa con las diferentes interfaces de comunicación que permiten el intercambio de información a través de más de una tecnología, entre las más comunes: *Wi-Fi*, *Bluetooth*, señalización móvil (SMS), datos sobre la interfaz móvil (GPRS, UMTS, o LTE), NFC, *Wi-Fi Direct*...

Antes de comentar las peculiaridades de las plataformas móviles analizadas, veamos cómo se reparte la cuota de mercado global

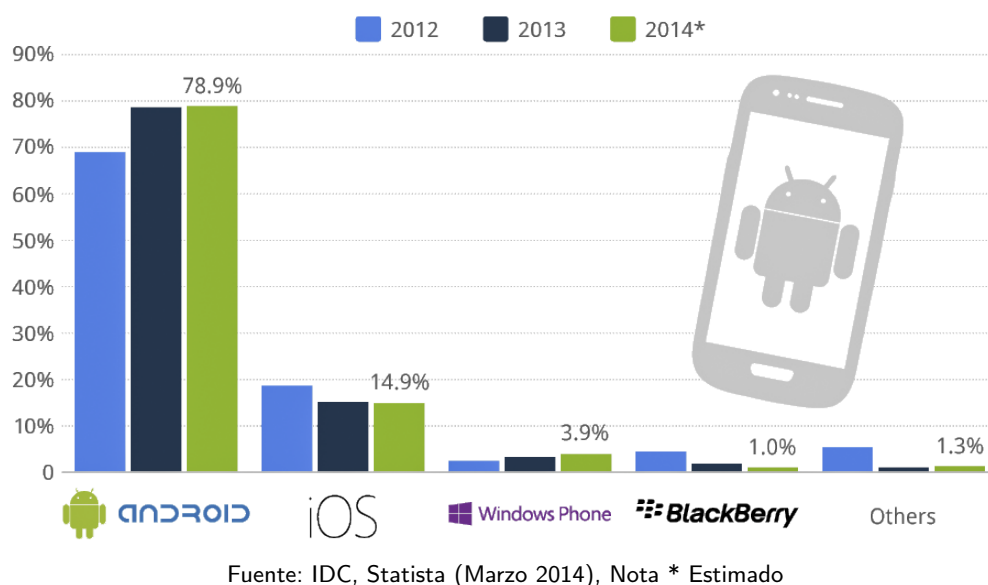


Figura 1.1: Cuota de mercado de teléfonos inteligentes, global por plataforma

Se puede observar que el mercado prácticamente se reparte entre **Android** e **iOS**, las cifras de dispositivos son:

Android Unos 900 millones de activaciones y 4800 millones de aplicaciones descargadas a fecha de mayo de 2013 [3].

iOS 800 millones de dispositivos y 7500 millones de aplicaciones descargadas a fecha de junio de 2014 [4].

Sin embargo los datos más recientes sugieren que las descargas en la plataforma de Google exceden alrededor del 60 % a las de Apple [5], sin embargo Apple defiende que sus dispositivos son usados al rededor de un 50 % más que los de **Android**.

Descontinuados

A modo de anécdota comentamos varios sistemas operativos cuyo desarrollo se ha visto interrumpido y cuyo soporte está llamado a desaparecer.

MeeGo

Este sistema operativo de código abierto basado en Linux nace tras la alianza entre Intel y Nokia. Cada uno aporta un código fuente a partir del cual evolucionar, se trata de los sistemas *Mobin* y *Maemo* respectivamente, que tenía como objetivo desarrollar un sistema único para dispositivos empotrados como *netbooks*, tabletas, ordenadores en vehículos, televisiones, teléfonos...

Tras el anuncio por parte de Nokia de la alianza con Microsoft y los cambios en la dirección de la compañía, se decide abandonar el proyecto, que continua Intel con la elaboración de **Tizen**. Sin embargo la propia comunidad de *software* libre que también participaba en el desarrollo del proyecto, así como ex-trabajadores de Nokia, continúan el desarrollo de *MeeGo* en lo que se denomina **Mer**. Desarrollos comerciales de este sistema son llevados a cabo por Jolla, una *startup* finlandesa formada por ex-trabajadores de Nokia, desembocando en el sistema operativo **Sailfish OS** que como el terminal Nokia N9 presenta una interfaz sin botones cuya interacción es mediante gestos únicamente.

Symbian

Parte como la evolución de EVOC un sistema operativo para PDA (asistente digital personal), cuando en 1998 se forma el consorcio entre Ericsson, Motorola y Nokia. Hasta 2008 otros fabricantes de móviles se unen para crear diferentes plataformas *software* creadas para **Symbian**, es en esta fecha cuando Nokia bajo una fundación sin ánimo de lucro adquiere el sistema operativo del que mantiene dos repositorios: uno privado con los últimos desarrollos y librerías y componentes privados y otro público reglado por licencia de código libre. Sin embargo en noviembre de 2010 ayudado por las condiciones adversas en el mercado decide abandonar su rama abierta. A partir de este momento las fuerzas se diluyen lo que desemboca en la externalización del desarrollo a una consultora para crear las últimas versiones de este sistema llamadas **Nokia Belle**, a principios de 2014 Nokia deja de dar soporte a su sistema, que hasta finales de 2010 era el más popular.

Vodafone360

Nace en octubre de 2009 como esfuerzos por extender la experiencia de usuario de su navegación *Live* dotándola de un sistema operativo completo bajo licencia de código abierto denominado **LiMo**. Esta distribución basa en Linux se amparaba bajo el paraguas de un consorcio formado por Motorola, NEC, NTT DoCoMo, Panasonic Mobile, Samsung y Vodafone. Tras el poco éxito cosechado con el lanzamiento de los terminales H1 y M1 fabricados por Samsung y el creciente éxito de Android lleva a la discontinuación del servicio, y la integración de la fundación **LiMo** en **Tizen**.

En activo

Sin embargo los potenciales sistemas operativos sobre los que desarrollar nuestra aplicación son los que mantienen un soporte. Entre ellos podemos destacar una pequeña lista cuyas características más importantes pasamos a resumir:

1.1.1. Android

El sistema operativo se empezó a desarrollar para cámaras digitales en 2003 bajo la empresa Android, pero rápidamente cambia el enfoque al darse cuenta que no estaban apuntando al

mercado correcto y deciden crear un sistema para competir con **Symbian** y **Windows Mobile**. Sin embargo, es el 17 de agosto de 2005 cuando Google adquiere la compañía al tomar la decisión de entrar en el mercado de los teléfonos inteligentes, cuando realmente el proyecto empieza a tomar forma. En 2007 nace la *Open Handset Alliance*: un compendio de empresas en las que se incluyen Google, algunas operadoras como Sprint o T-Mobile, fabricantes como HTC, Sony o Samsung y fabricantes de componentes electrónicos como Qualcomm o Texas Instruments, con la intención de desarrollar estándares abiertos para dispositivos móviles, constituyendo **Android** como un sistema operativo para teléfonos inteligentes construido bajo la versión 2.6 del núcleo de Linux. El alcance del sistema operativo ha ido cambiando con el tiempo y ahora no sólo alcanza a teléfonos inteligentes si no también a tabletas, relojes, televisores y sistemas para automóviles [6].

Android se distribuye bajo licencia de código abierto cuyo desarrollo está liderado por Google quien libera el código en cada versión, aunque la comunidad puede contribuir a su desarrollo gracias al proyecto AOSP [7], sin embargo es común que el *software* venga acompañado de ciertos componentes privativos de los distintos fabricantes como algunos *drivers* o elementos usados para diferenciarse del resto de fabricantes. Samsung por ejemplo, pone en manos de los desarrolladores unas herramientas adicionales a las que provee la plataforma para controlar sus singularidades [8].

Aunque se puede considerar que es una distribución más de Linux, de hecho ha habido contribuciones mutuas al desarrollo del núcleo, lo cierto es que **Android** no incluye la librería C de GNU, si no que integra la librería Bionic [9] desarrollada específicamente para este sistema. Tampoco se da soporte a la mayoría de librerías de GNU, lo que hace difícil portar aplicaciones de Linux. Para facilitar este fin se desarrolla el NDK, un conjunto de herramientas que permiten ejecutar código escrito en C/C++ en la plataforma de **Android**, su uso está recomendado para aplicaciones que requieran de mucha carga computacional o tareas muy relacionadas con una programación de bajo nivel.

El desarrollo de aplicaciones se lleva a cabo con las herramientas [10] incluidas en el SDK, entre ellas destaca ‘Android Debug Bridge’ (adb) que nos permite mediante línea de comandos comunicarte bien con el emulador o bien con un dispositivo real, permitiendo extraer las trazas de depuración, instalar o borrar aplicaciones y hasta simular conexiones o SMS. El lenguaje utilizado para desarrollar aplicaciones es Java, aunque a principios 2014 Google lanzó ‘Adobe Cordova’ [11] para portar las aplicaciones del navegador Chrome desarrolladas en HTML5 a **Android**. Esta tecnología es usada por el *framework* externo **PhoneGap** que permite escribir aplicaciones con HTML, Javascript y CSS3 empaquetándolas de tal forma como si fuesen aplicaciones nativas para **iOS**, **Android**, **Windows Phone 7**, **BlackBerry OS**, **Symbian**, **Bada** y **WebOS**, por lo que aunque no forme parte de la plataforma es posible escribir aplicaciones en casi todos los sistemas con lenguajes web.

Entre las interfaces de comunicación [12] soportadas podemos destacar: *Wi-Fi Direct*, *Bluetooth*, *NFC*, *Wi-Fi* [13] y datos móviles [14]. La característica *Ad-Hoc* está restringida por la plataforma aunque la mayoría de *chipsets* lo soportan pero se necesitan permisos de superusuario para manejarlo.

1.1.2. BlackBerry OS

Propietario de la canadiense RIM, desarrolla su primer dispositivo en 1999 al que integra el correo y un teclado QWERTY. Este sistema operativo es conocido por su enfoque al mundo de la empresa para lo que desarrollo lo que se conoce como sistema *Push*, que es un sistema

por el cual la información es recibida en el mismo momento que se envía, para ellos RIM tiene su propia infraestructura que se conecta con los distintos ISP. Este sistema centralizado permite un control total y garantizar ciertos parámetros de calidad sobre las conexiones de correo electrónico y mensajería instantánea pero tiene como principal debilidad la disponibilidad [15], aunque debido, a una mala implementación [16], también tiene fallos en la seguridad.

En 2013 sale el sistema operativo **BlackBerry 10**, cuyo primer dispositivo en incorporarlo fue el Z10, luego se han ido sumando a este sistema los modelos Q10, Z30, Z3, Passport, Q5, P'9982 y P'9983. Este sistema nace tras la crisis interna por los problemas en la compañía debido a los fallos de la disponibilidad del servicio y como intento de frenar el bipolio entre *Android* e *iOS*. Es un sistema basado en distribución de tiempo real QNX (Unix) que gracias a su tecnología de micro-módulo presenta características de estabilidad avanzadas frente a fallos en las aplicaciones. El sistema es privativo y permitía cuatro formas de desarrollo:

- **Nativo:** Aunque existen varias capas para adecuarse al tipo de desarrollador, la primera permite desarrollar aplicaciones sobre Cascades, un *framework* basado en QT que combina QML y C++. Se soportan librerías gráficas como Unity, Box2D que están basadas (y también se puede desarrollar en ellas) en OpenGL, EGLm Scoreloop. Entre toda la plataforma nos interesan los módulos QtConnectivity, QtMobility y QtNetwork. Sin embargo investigando éstos no se ha visto que se puedan realizar conexiones *Ad-Hoc*, pero sí *Wi-Fi Direct*.
- **HTML5:** Se utiliza el SDK: 'BlackBerry 10' WebWorks para su desarrollo añadiendo *plugins* para unir las funciones escritas en Javascript (HTML5) con funciones nativas de la plataforma. El módulo Connection nos permite obtener información sobre la conexión pero no con permisos de escritura.
- **Adobe AIR y Runtime Android:** Se integran maquinas que interpretan el código desarrollado para AIR y Android. Para la versión 10.3.1 se quita el soporte de AIR en favor de HTML5 y Android. Es necesario empaquetar de nuevo la aplicación nativa de *Android* para que pueda ejecutarse e instalarse en un dispositivo de **Blackberry 10**, hay dos maneras para portar una aplicación: automática que puede incurrir en fallos de compatibilidad, o manual dónde el desarrollador elige las API apropiadas que quiere usar para evitar incompatibilidades y obligada se usa el NDK para conservar su eficiencia.

1.1.3. Firefox OS

Este sistema operativo [17] está enfocado a dispositivos móviles: teléfonos inteligentes y tabletas, específicamente en el sector de gama baja. El modelo de desarrollo es código abierto, con núcleo Linux y utilizando Gecko como motor renderizador de páginas web. La forma de desarrollar aplicaciones para este sistema operativo se basa en HTML5 (HTML, CSS y JavaScript) y WebAPI para acceder a los componentes *hardware* del teléfono. Además Mozilla quiere estandarizar estas API de tal forma que sean accesibles en otros navegadores para que las aplicaciones desarrolladas para **Firefox OS** puedan ser portadas a otras plataformas con cada vez menos cambios. Asimismo las WebAPI no son exclusivas del sistema operativo si no que están implementadas en toda la plataforma de Mozilla: **Firefox OS**, navegador de escritorio y navegador Firefox para *Android*.

El desarrollador es libre de distribuir la aplicación por el canal que desee, sin embargo algunas aplicaciones deben ser revisadas y firmadas por Mozilla si se utilizan ciertas WebAPI que son demasiado sensibles respecto a la seguridad para que cualquiera pueda ejecutarlas. Esto pudiera ser un riesgo para el desarrollo de ciertas características cognitivas, delegando

a la modificación del propio sistema operativo para añadir ciertas funcionalidades si no son aprobadas por el equipo de Mozilla. Hay dos modelos de aplicaciones: aquellas que están ubicadas en un servidor remoto tal como si se tratase de una web normal y que son descargadas cuando no están disponibles desde la caché, permitiendo un mantenimiento más sencillo. Y aquellas que son distribuidas como un ‘zip’ e instaladas en el dispositivo teniendo acceso a todas las WebAPI.

Tras un breve análisis, podemos destacar que tanto *Bluetooth* [18] como *Wi-Fi* [19] son soportados pero ambas WebAPI necesitan del permiso para ser distribuidas por el *Marketplace*, sin embargo para un mayor control de las interfaces de comunicación se necesita que las aplicaciones estén preinstaladas en los dispositivos con la firma de Mozilla, ejemplos son *Bluetooth* [20] y interfaz móvil [21]. Por otro lado parece que el modo *Ad-Hoc* en la conexión *Wi-Fi* no está aun soportado [22].

1.1.4. iOS

Lanzado en 2007 como *iPhone OS* por Apple, es un sistema operativo para dispositivos móviles: teléfonos, tabletas, reproductores de música y reproductores para televisión. Tanto el *software* privativo como el *hardware* no licenciado para otras compañías, hacen que junto a *Blackberry OS* sean las plataformas más restrictivas, de hecho el modelo de desarrollo posiblemente sea el más restrictivo de todos. Esto se debe a que las herramientas de desarrollo sólo están disponibles para el sistema operativo *MAC OS X* que sigue el mismo tipo de licenciamiento. Además para poder probar el desarrollo en un dispositivo real se tiene que tener una cuenta de desarrollador con un coste asociado renovable cada año (para ciertas universidades hay programa especial sin coste).

El núcleo del sistema está basado en XNU [23] de licencia libre y que fue integrado por *Darwin BSD*: el sistema que subyace en *MAC OS X*. Este núcleo híbrido provee multitarea apropiativa y memoria protegida entre otros. Precisamente estos dos sistemas operativos comparten muchas capas en común: núcleo y servicios principales (*Fundation kit*). Pero en la capa de presentación debido a las diferencias de computación e interacción *iOS* tiene como interfaz de usuario *Cocoa Touch* en lugar de *Cocoa*.

Las diferentes versiones del sistema operativo han añadido diferentes características, la primera versión no disponía de un SDK propiamente dicho revelando el desarrollo de aplicaciones a herramientas web sin tener acceso al *hardware*, la segunda versión introduce además la tienda de aplicaciones y la denominación *iPhone OS*. Otra versión importante del sistema es la versión 4 cuando adopta la denominación de *iOS* que introduce la multitarea (con limitaciones a través de una API especial) en aplicaciones de terceros. La única forma de desarrollar aplicaciones para estos dispositivos es bajo el conjunto de herramientas suministradas utilizando el lenguaje Objective-C, además recientemente se puede usar el lenguaje Swift [24] para programar de forma más concisa y segura sobre la capa de presentación.

Las conexiones que permite la plataforma son por una parte: *Wi-Fi* y datos móviles presentados como una capa de comunicación [25] común en la que el desarrollador parece que no tiene manera de modificar los parámetros de conexión. En la versión 2 del sistema operativo hubo una API que permitía desconectar y conectar la interfaz *Wi-Fi*, con las nuevas versiones esta característica se ha visto relevada a un plano interno [26] que releva el desarrollo a fuera de los márgenes de Apple. Aunque las redes *Ad-Hoc* están soportadas no hay manera de recrear una red de este tipo con los dispositivos actuales [27]. Por otra parte la tecnología NFC sólo está disponible en el dispositivo ‘iPhone 6’ y por el momento

sólo se utiliza como pasarela de pagos. La única tecnología que nos puede dar conexiones P2P de manera independiente (*Wi-Fi* necesita de una infraestructura externa) y de forma completa es *Bluetooth* [28]. Dejándonos una sensación de comunicación incompleta en torno a las necesidades de la radio cognitiva que no cumplen las expectativas depositadas en la plataforma.

1.1.5. Tizen

Tras el renombramiento de LiMo en Tizen [29] junto con la *Linux Foundation* continua el proyecto de crear un sistema operativo para todos los dispositivos móviles. Con respecto a los teléfonos, el único dispositivo de referencia (RD-PQ) se basa en el ‘Samsung Galaxy III’. El proyecto aunque abierto a la comunidad aún se encuentra en fase de desarrollo en la versión 2.2.1, que ha traído un cambio en el licenciamiento debido al modelo de patentes de Samsung, quién espera lanzar un piloto comercial en el mercado indio a finales de 2014.

Este nuevo modelo de licencias, podría suponer un riesgo, aunque el código será libre siempre, el SDK pertenece a Samsung quien lo distribuye bajo una licencia de código no abierto y otros componentes están bajo licencia Flora que no es aprobada por la ‘Open Source Initiative’.

Con respecto al desarrollo de aplicaciones [30] existen dos enfoques:

- **Web:** incluye todas las tecnologías de desarrollo web como HTML5: JavaScript (jQuery y jQuery mobile), así como todas las API necesarias para controlar los distintos componentes del *hardware*.
- **Nativo:** si se requiere un control más preciso, se puede desarrollar programas en C++ de los cuales pueden hacer uso de las librerías Libxml2, EFL, Qt y GPL entre otras, al igual que una serie de funciones publicadas bajo varias API.

Entre las conexiones [31] soportadas caben destacar: *Wi-Fi Direct* y *Wi-Fi Ad-Hoc* [32]. El ciclo de las aplicaciones recuerda mucho a *Android*, de hecho aplicaciones desarrolladas en este sistema operativo pueden ser ejecutadas en **Tizen**.

1.1.6. Ubuntu Touch

Presentado a primeros de 2013 por Canonical, este sistema operativo para dispositivos móviles y tabletas quiere proporcionar una experiencia de usuario muy similar a la versión de escritorio de la distribución Linux Ubuntu. De hecho ambos sistemas comparten las mismas tecnologías núcleo de tal forma que aplicaciones desarrolladas para las últimas versiones de la plataforma de escritorio pueden ser ejecutadas en el dispositivo móvil y viceversa.

Ubuntu Touch [33] utiliza el *framework* Qt5 (QML y C++) para el desarrollo de aplicaciones nativas y la plataforma ‘Adobe Cordova’ para el desarrollo con tecnologías web HTML5. Otros componentes integrados son OpenGL y otros provenientes de MeeGo (oFono: pila telefónica), así como libhybris: una capa para proveer compatibilidad a la librería de C de GNU que descende de Bionic lo que hace que sea más fácil portar este sistema a dispositivos **Android**, ejemplo de esto es que los únicos dispositivos por el momento que soportan todas las características son los modelos Nexus aunque con peculiaridades [34].

Con respecto a las interfaces de comunicación, debido a que el sistema aún está en fase de desarrollo, no están expuestas todas las características en la versión para desarrolladores [35]. La característica *Ad-Hoc* se desconoce si será soportada de forma oficial.

1.1.7. Windows Phone

El sistema operativo móvil privativo de Microsoft nace a finales del 2010 como sucesor de **Windows Mobile** con la versión 7.0 pero no es hasta mayo de 2011 cuando se libera la versión 7.5 cuando podemos hablar de un sistema operativo completo tras la alianza con Nokia. La compañía finlandesa provee gran parte de los modelos que ejecutan este sistema operativo: la gama Lumia. Los desarrolladores tienen acceso a un SDK [36] propio para desarrollar características exclusivas o acceder de manera completa al *hardware* del teléfono.

A finales de 2012 sale **Windows Phone 8** una versión que integra un nuevo núcleo, hasta ahora basado en ‘Windows CE’. Se trata del núcleo ‘Windows NT’ que da soporte a procesadores de varios núcleos. Esta versión sólo está disponible para nuevos dispositivos, lo que supone un corte. Al ser un núcleo que comparte muchas características con la versión de escritorio **Windows 8** y poseer la misma interfaz de usuario: **Metro** (ahora conocida como **Modern UI**) las aplicaciones pueden ser fácilmente portadas entre ambas plataformas.

El modelo de desarrollo se basa en las herramientas de .Net a través de XAML [37] (C++/CX, C# o VisualBasic) o HTML5 (HTML, JavaScript: WinJs). Entre las API publicadas [38] se puede observar que tenemos control sobre *Bluetooth* y *Wi-Fi Direct*, pero faltan el control de *Wi-Fi* y la conexión de datos móvil, al igual que parece no haber un soporte a conexiones *Ad-Hoc* [39].

1.2. Redes de comunicaciones

Las plataformas móviles cursan sus datos por las redes de comunicaciones repartidas por todo el planeta, veremos su estado de crecimiento en los momentos actuales y una proyección de futuro enfocada tanto al crecimiento como a la evolución en cuanto a tecnología y características.

1.2.1. Crecimiento de las comunicaciones

Las comunicaciones de banda ancha están experimentando un crecimiento en los últimos años, sobre todo las móviles como podemos ver en el informe para dos mil catorce de la ITU [40]. El número de líneas fijas (ver figura 1.2) en países emergentes alcanzó en 2013 al de los países desarrollados, debido a que, aunque en ambos mundos se aprecia que la tasa de crecimiento se está reduciendo, el mayor ratio sigue estando en los países emergentes donde se pasa de un crecimiento del 18 % en 2011 al 6 % en 2014, mientras que en países desarrollados el crecimiento pasa del 4.8 % en 2011 a el 3.5 % en 2014. El 44 % de las líneas de banda ancha fija están localizadas en Asia-Pacífico (APAC) sin embargo, en Europa sólo representa el 24 % del total pero el grado de penetración es el mayor, tónica general de los países desarrollados.

Con respecto a la banda anchas en las comunicaciones móviles (ver figura 1.3) la penetración es del 32 % (tasa que se ha visto doblada a nivel de 2011 y cuadruplicada a nivel de 2009) llegando a tener 3200 millones de suscripciones de las cuales el 55 % se localizan en países emergentes donde la penetración es del 21 %. La tasa de crecimiento en estos países se espera que alcance el 26 %, cifra que dobla a la tasa de crecimiento de los países desarrollados que se augura que sea del 11.5 %. Sin embargo, al igual que pasa en las comunicaciones fijas el grado de penetración en estos países es mayor, situándose en un 84 %.

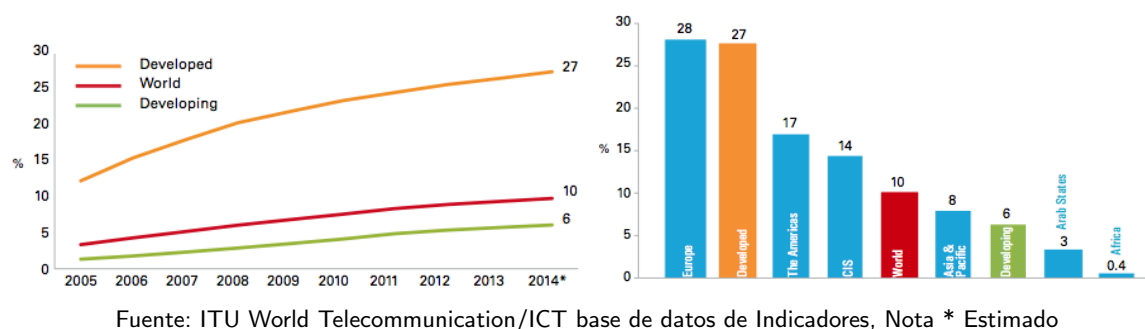


Figura 1.2: Líneas de banda ancha fijas (suscripciones por cada 100 habitantes) y estimación por región en 2014

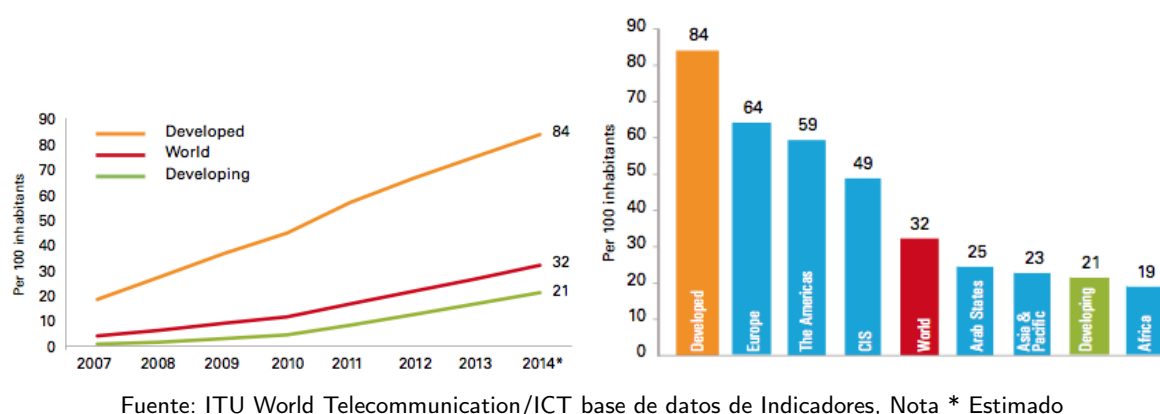
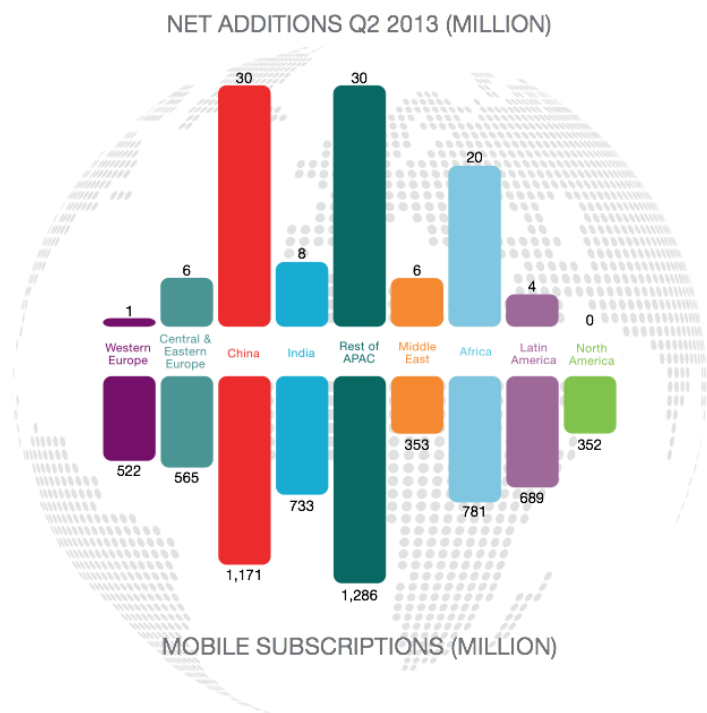


Figura 1.3: Líneas de banda ancha móvil (suscripciones por cada 100 habitantes) y estimación por región en 2014

Centrándonos exclusivamente en el mundo móvil comunicaciones de voz y datos podemos destacar según los informes de movilidad de Ericsson (EMR) del segundo trimestre del dos mil trece [41] y del dos mil catorce [42] que se ha incrementado en más de tres cientos millones el número de líneas móviles entre ambos ejercicios, alcanzando la cifra de 6800 millones en el segundo trimestre de dos mil catorce. En el segundo trimestre de 2014 se vendieron alrededor de 300 millones de teléfonos inteligentes que representa un 65 % de las ventas totales, un año anterior suponían un 40 %. Las activaciones netas de líneas móvil se mantienen pasando de un 7 % en el ejercicio de 2013 a un 6 % en el del 2014. (ver figuras 1.4 y 1.5).

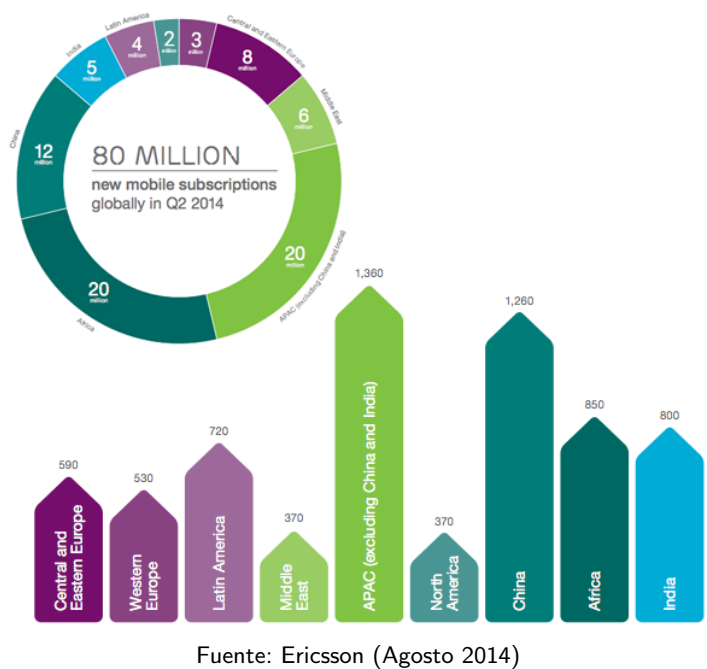
El grado de penetración evoluciona desde el 91 al 94 % en un año, pero como vemos en la figura 1.6 en algunas regiones la tasa es mayor del 100 %, esto es debido a que no se trata de usuarios únicos, se cree que sólo 4500 millones de personas contratan las 6800 millones de suscripciones móviles.

Todo esto se traduce en un aumento desmedido del tráfico cursado. En el último trimestre de dos mil nueve se produce un hito al sobrepasar el tráfico de datos al de voz. A partir de este momento se aprecia con más notoriedad una tendencia exponencial que se traduce en casi un duplicamiento del tráfico entre los segundos trimestres de 2012 y 2013 y en un incremento del más del 60 % entre los mismos periodos del año siguiente, alcanzando un tráfico de unos 2700 PetaBytes/mes (esta cifra está fuera de la gráfica 1.7).



Fuente: Ericsson (Agosto 2013), Notar que Turquía se incluye desde este momento como parte de Oriente Medio en vez de Centro y Este de Europa. Por esta razón, no se deben hacer comparaciones entre esta y ediciones previas del informe EMR

Figura 1.4: Líneas móviles (total e incremento neto) por región, segundo trimestre de 2013



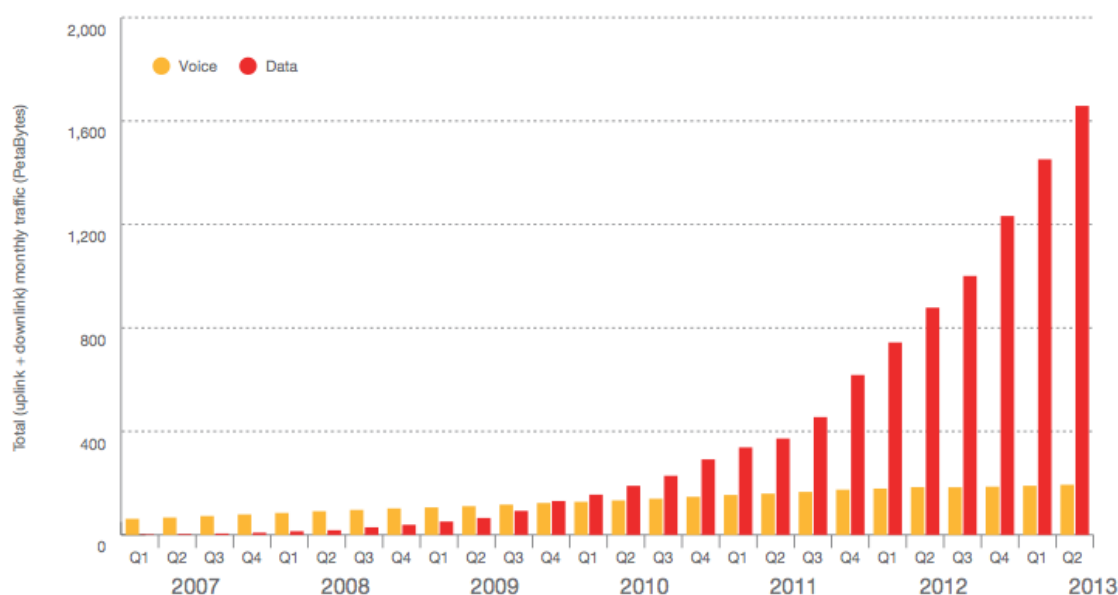
Fuente: Ericsson (Agosto 2014)

Figura 1.5: Líneas móviles: incremento neto (círculo) y total (barras) por región, segundo trimestre de 2014



Fuente: Ericsson (Agosto 2014)

Figura 1.6: Penetración líneas móviles por región, segundo trimestre 2014



Fuente: Ericsson (Agosto 2013), Notar que no se incluye el tráfico proveniente de DVB-H, Wi-Fi, o WiMax. Además el tráfico de voz no incluye voz sobre IP (VoIP)

Figura 1.7: Tráfico de datos global cursado en redes móviles

1.2.2. Futuro de las comunicaciones

La perspectiva para el futuro mantiene las tendencias actuales (exponencial y lineal) aunque a medida que avanzamos en la gráfica podemos intuir una tendencia cada vez más lineal. Con respecto a las suscripciones móviles se espera que se alcance la cifra de 9200 millones de líneas móviles en 2019, de las cuales el 80 % cuentan con banda ancha, entre las tecnologías que soporten esta conectividad 2600 millones serán LTE, aunque el grueso todavía se espera que sean las tecnología dual actual de UMTS y GSM. Al final de la década habrá 5600 millones de teléfonos inteligentes y unos 700 millones de dispositivos con conexión móvil: tabletas, ordenadores.

Con respecto a la localización de estas conexiones, el mayor crecimiento (entorno al 60 %) se espera en Oriente Medio y África, pero el mayor número de éstas corresponderán a Asia y

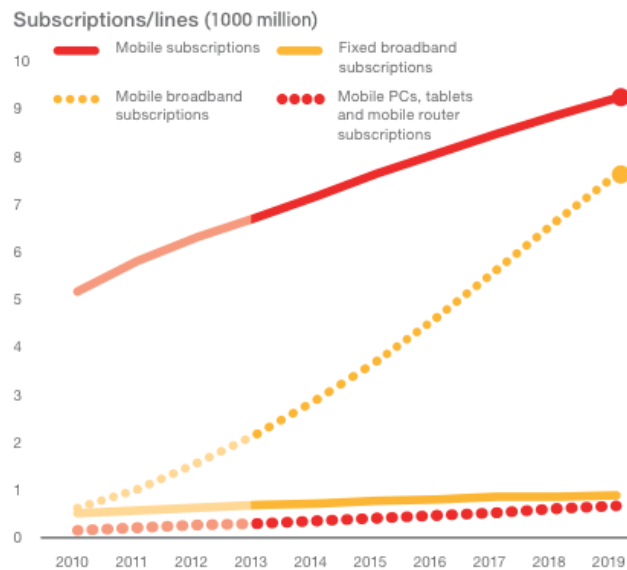


Figura 1.8: Estimación número de suscripciones hasta 2019

Pacífico. Además el 80 % de éstas contarán con la tecnología UMTS y LTE debido a que las nuevas suscripciones son el motor de su crecimiento.

Todo esto nos lleva de nuevo a un crecimiento exponencial del tráfico cursado por las redes móviles saturando el espectro cada vez más, a pesar de la mayor eficiencia de las modulaciones móviles. Experimentando un factor diez en el crecimiento del tráfico entre los años de 2013 y 2019, alcanzando los 12 ExaBytes/mes. Si lo trasladamos al plano del usuario, éste pasará de consumir 650MB en 2013 a 2.5GB en 2019 cada mes.

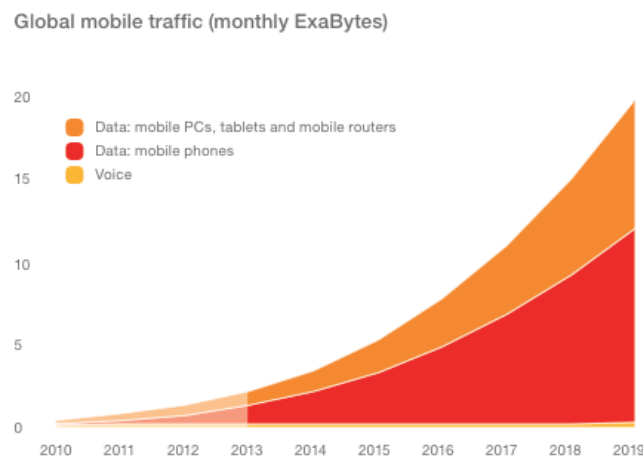


Figura 1.9: Estimación tráfico de datos global cursado en redes móviles

Estas expectativas de crecimiento harán evolucionar las redes. Este aumento sumado a los requerimientos cada vez mas exigentes por parte de las aplicaciones, son el motor que alimenta el cambio. En primer lugar la cobertura de las distintas tecnologías aumentará desde 2013

hasta 2019 en los siguientes porcentajes:

- GSM/EDGE (2G) desde un 80 % hasta un 85 % (>95 % en Europa).
- WCDMA/HSPA (3G) partiendo de un 60 % hasta el 90 %.
- LTE (4G) de una implantación del 20 % creciente hasta más del 65 % y cercana al 80 % en el caso de Europa.

Europa ha adoptado las tecnologías inalámbricas desde el minuto cero, por ello el grado de penetración es mayor aunque la contrapartida es la necesidad de proyectos de renovación y compatibilidad que otras áreas del mundo no han requerido pues han comenzado directamente con el despliegue de LTE de las que actualmente existen 288 redes repartidas en 104 países ¹.

La evolución de éstas redes se centrará en:

Agregación de portadoras Cada portadora ocupa 10Mhz, podemos combinar varias portadoras para mejorar la eficiencia espectral, desde 2 para alcanzar velocidades de 150Mbps, o agregar 3 portadoras de 20Mhz para lograr 60Mhz de espectro agregado que soporta una tasa de bajada de 450Mbps, si la categoría de los equipos involucrados es nueve. Con estas técnicas podemos pasar de una eficiencia espectral de $3.5 \frac{\text{bit}}{\text{s} \cdot \text{Hz}}$ en el enlace de subida para la versión 8, a $15 \frac{\text{bit}}{\text{s} \cdot \text{Hz}}$ para la versión 10: LTE-A.

Difusión Aunque la especificación MBMS, ya existía en redes UMTS [43], se espera que sea en la red LTE donde alcance pilotos comerciales. La aprovechamiento del espectro radica en que se entrega contenido simultáneamente a varios usuarios bajo solamente un flujo LTE, además se permitirá al operador predefinir el nivel de calidad dimensionando la tasa de *bits* para adecuarse a distintas resoluciones y tipos de contenido.

Voz de alta resolución La red LTE ha sido diseñada para la optimización de transferencia de paquetes de datos, por lo tanto deja fuera los circuitos conmutados por los que tradicionalmente iba la voz, en lugar se sustituye el núcleo de red por un sistema IMS que permite su evolución mediante actualizaciones de *software*. Sin embargo el tráfico de voz como hemos visto es insignificante pero su codificación (utilizando *chipsets* que soporten VoLTE) utiliza eficientemente el espectro y está abierta a nuevas optimizaciones.

Aprovechamiento de otras zonas del espectro Más conocido como *refarming*, es el proceso por el cual se utiliza una banda del espectro para usos diferentes de los que inicialmente fueron asignados. Para dar soporte a todo el tráfico que se prevé que cursen las redes LTE es necesario mover y utilizar las frecuencias dónde se dan las mejores condiciones de propagación para entre otros beneficios, evitar retransmisiones. El proceso se conoce como diviendo digital del que en una primera versión se estudió la banda entorno a los 900Mhz (la mejor propagación), pero para 2015 la ITU está preparando un segundo dividendo en este caso centrado sobre los 700Mhz [44].

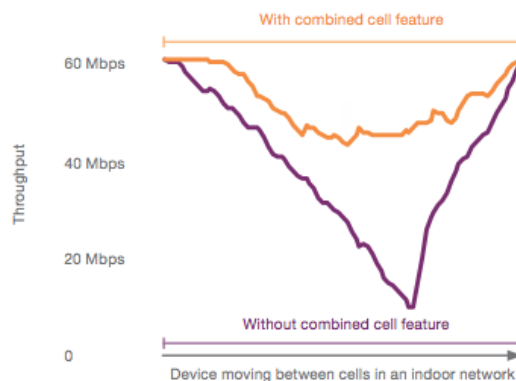
Nos movemos a un concepto de red heterogénea sin fisuras, dónde para mejorar el rendimiento y la capacidad se necesita: coordinación, formar la red de acceso con nodos que consuman poca energía (micro, pico, *Wi-Fi*) y soluciones de interior localizadas.

Una solución para conseguir este propósito pasa por evolucionar la capa de la macro célula integrando células más pequeñas. Para proveer al usuario de una experiencia sin fisuras en estas redes heterogéneas, todas la capas, dominios y nodos de red tienen que estar unidos y sincronizados con un grado sin precedentes, muy cercano al enfoque de CN. Se necesita por

¹Fuente: GSA Mayo 2014

tanto coordinación entre las pequeñas células y las macro que las contienen, gestión común de red y gestión del tráfico y movilidad de usuarios.

Coordinación a nivel radio Siguiendo las especificaciones de UMTS: traspaso suave, célula combinadas e introduciendo la tecnología de CoMP para el canal de subida, se consigue doblar las tasas de datos, en situaciones en las que varias células de distintas antenas están próximas. A la derecha podemos ver un ejemplo de usar o no esta coordinación.



Piloto LTE con software comercial

Coordinación entre la capa de células Para doblar la capacidad de la red en una ciudad, necesitaríamos añadir 9 células pequeñas por cada célula macro que tengamos. Sin embargo si existe una coordinación entre todas ellas, sólo necesitamos añadir 5 o 3 células, esto a parte de dejar sitio para incrementar la capacidad, nos resuelve el problema de posicionar tantas células y la interconexión a la red.

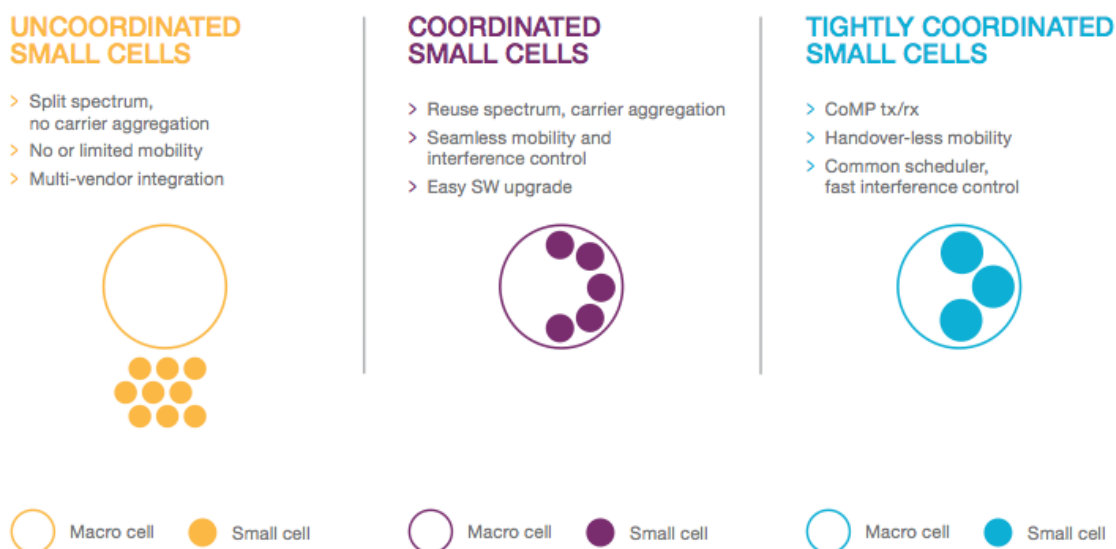
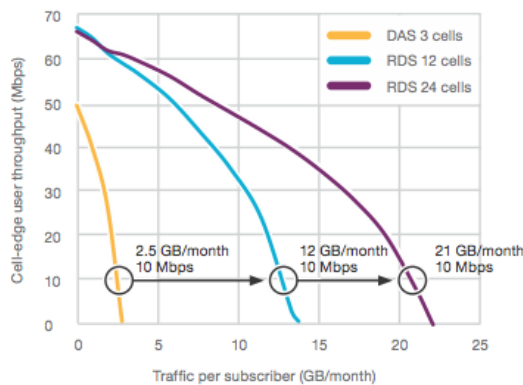


Figura 1.10: Número de células necesarias para dar la misma capacidad en función de su interoperabilidad

Gestión del tráfico y movilidad de usuarios Se incluyen tecnologías para de una manera coordinada traspasar usuarios entre tecnologías de acceso radio. A parte de las tecnologías de acceso 3GPP, se incluye *Wi-Fi* ya que es una tecnología más entre las ofertadas para banda ancha móvil. Así podemos esperar comportamientos en los que un usuario cambia de LTE a *Wi-Fi* sólo cuando le ofrece una mejor experiencia, en vez de cuando está inmediatamente disponible tal y como ocurre ahora.

Gestión de red común Obliga a tener una sola arquitectura de red y unos KPI que se recojan y sean visibles al resto.



Red acceso pico-célula: LTE FDD 20Mhz

Soluciones de interior Algunos edificios son puntos de concentración de usuarios que sobrecargan la red llegando a interferir entre ellos, con la dificultad añadida de la pobre cobertura dentro de estos recintos. Para ello se pueden instalar pico-células cuyo camino de salida sea el enlace de banda ancha fijo que posea el edificio. Estas pico-células forman lo que se conoce como un RDS. A la derecha podemos ver ejemplo de dimensionamiento en un edificio de 24 plantas con 2500m² de superficie cada una.

1.3. Redes Cognitivas

CN

1.4. Radio Cognitiva en teléfonos inteligentes

Hemos visto que la evolución de las redes 4G (LTE) prevé mecanismos para evitar la concentración de tráfico localizado mediante el uso de difusión, despliegue de células mas pequeñas, coordinación entre células y gestión centralizada del tráfico. Estas palancas de actuación aunque no parecen estar orientadas bajo una perspectiva cognitiva sí que comparten algunos de sus principios y su finalidad que es evitar el colapso del espectro. La tendencia, según los últimos congresos celebrados [45] sugieren que la infraestructura de acceso a las redes móviles tenga cada vez más características de la radio cognitiva [46]. En realidad Ericsson recomienda para la próxima generación de redes para teléfonos móviles (5G) el que haya una compartición flexible del espectro: DAS en frecuencias altas [47] que son más indicadas para la masificación y localización debido a su pobre propagación.

Al ser una nueva zona del espectro para redes telefónicas, permite realizar nuevas técnicas (como radio cognitiva) permitiendo una compatibilidad hacia LTE en la zona baja del espectro dónde se mantendrían las modulaciones y acceso al medio actuales evitando costes de actualización. En las frecuencias altas la red de transporte entre nodos de las estaciones bases se puede realizar también de forma inalámbrica en la zona alta del espectro que es más ancha aumentando la eficiencia si se comparte igualmente de forma dinámica.

Como vemos es un campo fuertemente investigado por fabricantes de equipos de telecomunicaciones, universidades y también por fabricantes de terminales como Nokia quién tiene un espacio *web* dedicado a sus investigaciones [48] de entre los que cabe destacar un artículo [49] en el que se versa sobre comunicaciones D2D mediante LTE dentro de una célula estudiando el impacto sobre el usuario primario (la célula) se demuestra que la capacidad observada en toda la célula aumenta.

Es obvio que estas nuevas técnicas afectan a las operadoras [50] con ventajas e inconvenientes. Entre las ventajas es la que incrementar parte del espectro en redes existentes, o la de acceder a nuevos mercados con mayor facilidad, ya que entre las desventajas tenemos que el valor del espectro se devalúa y aumenta la interferencia debido a los usuarios secundarios. En media

la capacidad aumenta y los costes de red disminuyen. El proyecto ‘SENDORA’ [51] utiliza una WSN para identificar zonas del espectro sin usar en ese momento.

Otros proyectos que podemos mencionar la tesis [52] en la que se hacen técnicas DAS entre una placa ‘BeagleBoard’ junto con un teléfono inteligente *Android* y un ordenador. El dispositivo *Android* habilita las funciones de sensado, pasarela y en general a realizar tareas que ésta no puede por motivos computacionales. Así como el proyecto fin de carrera [53] en la que se diseñan escenarios de cambio de tecnología de comunicación entre un emisor y un receptor ante eventos de indisponibilidad.

En la misma línea podemos mencionar el proyecto ‘Serval’ [54] que recrea la red de transporte de una operadora mediante enlaces *Wi-Fi* cifrados, habilitando las comunicaciones en caso de catástrofe o de inexistencia de red debido al alto coste.

Referencias/Bibliografía

- [1] D. Bulbrook, *WAP: A Beginner's Guide*. Network professional's library, Osborne/McGraw-Hill, 2001.
- [2] "I-Mode." <http://en.wikipedia.org/wiki/I-mode>. Último acceso el 24-9-2014.
- [3] "Google I/O 2013 highlights." <http://thenextweb.com/google/2013/05/15/google-announces-900-million-activations-of-android-in-total-to-date/>. Último acceso el 17-08-2014.
- [4] "WWDC 2014 statshot." <https://gigaom.com/2014/06/02/wwdc-statshot-800-million-ios-devices-sold-75-billion-apps-downloaded/>. Último acceso el 17-08-2014.
- [5] "WWDC 2014 statshot." <http://tnw.to/g4sdx>. Último acceso el 17-10-2014.
- [6] "Android: be together not the same." <http://www.android.com>. Último acceso el 17-8-2014.
- [7] "Android Open Source Project." <https://source.android.com/>. Último acceso el 17-8-2014.
- [8] "Samsung Developers: Samsung Mobile." <http://developer.samsung.com/samsung-mobile>. Último acceso el 17-8-2014.
- [9] "Bionic libc." [http://en.wikipedia.org/wiki/Bionic_\(software\)](http://en.wikipedia.org/wiki/Bionic_(software)). Último acceso el 17-08-2014.
- [10] "Developer guides: Tools Help." <http://developer.android.com/tools/help/index.html>. Último acceso el 17-8-2014.
- [11] "Apache Cordova." <http://cordova.apache.org/>. Último acceso el 17-08-2014.
- [12] "Developer guides: Connectivity." <http://developer.android.com/guide/topics/connectivity/index.html>. Último acceso el 17-8-2014.
- [13] "Developer guides: Wi-Fi API." <http://developer.android.com/reference/android/net/wifi/WifiManager.html>. Último acceso el 17-8-2014.
- [14] "Developer guides: Mobile Data." <http://developer.android.com/reference/android/net/ConnectivityManager.html> & <http://developer.android.com/reference/android/telephony/TelephonyManager.html>. Último acceso el 17-8-2014.
- [15] "BlackBerry Outage 2011." <http://www.theguardian.com/technology/2011/oct/14/blackberry-outage-faulty-router-suspected>. Último acceso el 17-08-2014.

- [16] “The Danger of Fetishizing BBM Security.” <http://www.christopher-parsons.com/the-danger-of-fetishizing-blackberry-messenger-security/>. Último acceso el 17-08-2014.
- [17] A. Garzia, *Quick Guide For Firefox OS App Development*. Lean Publishing, 2014.
- [18] “MDN: WebBluetooth.” https://developer.mozilla.org/en-US/docs/Web/API/Web_Bluetooth_API. Último acceso el 17-08-2014.
- [19] “MDN: WiFi Information API.” https://developer.mozilla.org/en-US/docs/Web/API/WiFi_Information_API. Último acceso el 17-08-2014.
- [20] “MDN: BluetoothAdapter.” <https://developer.mozilla.org/en-US/docs/Web/API/BluetoothAdapter>. Último acceso el 17-08-2014.
- [21] “MDN: Mobile Connection.” https://developer.mozilla.org/en-US/docs/Web/API/Mobile_Connection_API. Último acceso el 17-08-2014.
- [22] “Bugzilla@Mozilla: Implementation of 802.11 (Wi-Fi) ad hoc feature.” https://bugzilla.mozilla.org/show_bug.cgi?id=928362. Último acceso el 17-08-2014.
- [23] A. Singh, *Mac OS X Internals: A Systems Approach*. Pearson Education, 2006.
- [24] “Swift overview.” <https://developer.apple.com/swift/>. Último acceso el 17-09-2014.
- [25] “iOS Developer Library: System configuration Framework Reference.” https://developer.apple.com/library/ios/documentation/SystemConfiguration/Reference/SCNetworkReachabilityRef/index.html#//apple_ref/doc/uid/TP40007260. Último acceso el 17-08-2014.
- [26] “iOS Developer Library: Core Telephony Framework Reference.” <https://github.com/DjKira/iOS-6-Headers/blob/master/SpringBoard/SBWiFiManager.h#L46>. Último acceso el 17-08-2014.
- [27] “iOS Developer Library: Document Transfer Strategies.” https://developer.apple.com/library/ios/technotes/tn2152/_index.html#//apple_ref/doc/uid/DTS40009179-CH1-SUBSECTION13. Último acceso el 17-08-2014.
- [28] “iOS Developer Library: Core Bluetooth Programming Guide.” https://developer.apple.com/library/ios/documentation/NetworkingInternetWeb/Conceptual/CoreBluetooth_concepts/AboutCoreBluetooth/Introduction.html. Último acceso el 17-08-2014.
- [29] “Tizen.” <http://en.wikipedia.org/wiki/Tizen>. Último acceso el 24-9-2014.
- [30] “Tizen Architecture.” https://developer.tizen.org/dev-guide/2.2.1/org.tizen.gettingstarted/html/tizen_overview/tizen_architecture.htm. Último acceso el 17-09-2014.
- [31] “Tizen Net: Managing Network Connections and Communication.” https://developer.tizen.org/dev-guide/2.2.1/org.tizen.native.appprogramming/html/guide/net/net_namespace.htm. Último acceso el 17-09-2014.
- [32] “Tizen: WifiBssType.” https://developer.tizen.org/dev-guide/2.2.1/org.tizen.native.apireference/namespaceTizen_1_1Net_1_1Wifi.html#aad52b065bd137bf6879e51b46e2dbb5b. Último acceso el 17-09-2014.

- [33] “App Dev School Presentation: Get started creating Ubuntu apps.” https://docs.google.com/presentation/d/1-0cIOjMSdiLI6JdBbuhobKkxdZRaaggeA-eS0oA5cnE/edit#slide=id.g3910e7ce9_2_6. Último acceso el 17-08-2014.
- [34] “Ubuntu Touch: devices.” <https://wiki.ubuntu.com/Touch/Devices>. Último acceso el 17-08-2014.
- [35] “Ubuntu Touch: release notes.” <https://wiki.ubuntu.com/Touch/ReleaseNotes>. Último acceso el 17-08-2014.
- [36] “Develop for Lumia.” <http://developer.nokia.com/lumia>. Último acceso el 17-08-2014.
- [37] “Developing apps (XAML).” <http://msdn.microsoft.com/library/windows/apps/xaml/br229566.aspx>. Último acceso el 17-08-2014.
- [38] “Windows API reference for Windows Runtime apps.” <http://msdn.microsoft.com/en-us/library/windows/apps/xaml/br211377.aspx>. Último acceso el 17-08-2014.
- [39] “Microsoft Forums: ad hoc support, wifi control.” <https://social.msdn.microsoft.com/Forums/windowsapps/en-US/5d5da1a8-cc84-4736-8ee5-7e4cacb7f5c5/how-to-enabledisable-wifi-programmatically?forum=wpdevelop> & <http://answers.microsoft.com/en-us/winphone/forum/wp8-wpnetwork/will-windows-phone-8-accept-ad-hoc-networks/4762dd75-0fc4-42f8-a771-15abc5075778>. Último acceso el 17-08-2014.
- [40] “The World in 2014: ICT Facts and Figures.” <http://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2014-e.pdf>. Último acceso el 24-9-2014.
- [41] “Ericsson Mobility Report: on the pulse of the networked society.” <http://www.ericsson.com/res/docs/2013/emr-august-2013.pdf>. Último acceso el 24-9-2014.
- [42] “Ericsson Mobility Report: on the pulse of the networked society.” <http://www.ericsson.com/res/docs/2014/ericsson-mobility-report-june-2014.pdf> & <http://www.ericsson.com/res/docs/2014/ericsson-mobility-report-august-2014-interim.pdf>. Último acceso el 24-9-2014.
- [43] F. Hartung, U. Horn, J. Huschke, M. Kampmann, T. Lohmar, and M. Lundevall, “Delivery of broadcast services in 3g networks,” *Broadcasting, IEEE Transactions on*, vol. 53, pp. 188–199, March 2007.
- [44] “The second digital dividend: Another bite for mobile?.” <https://itunews.itu.int/En/3744-The-second-digital-dividend-Another-bite-for-mobile.note.aspx>. Último acceso el 24-9-2014.
- [45] “CROWNCOM: Cognitive Radio Oriented Wireless Networks and Communications.” <http://crowncom.org/>. Último acceso el 17-8-2014.
- [46] “9th International Conference on Cognitive Radio Oriented Wireless Networks.” <http://crowncom.org/2014/show/keynotes>. Último acceso el 17-8-2014.
- [47] Dgahkman, E.; Sachs, J.; Parkvall, S.; Mildh, G.; Selén, Y.; Peisa, J., “5G radio access,” tech. rep., Ericsson Review, June 2014.

- [48] “Nokia Research Center: Cognitive Radio.” http://research.nokia.com/cognitive_radio. Último acceso el 17-8-2014.
- [49] K. Doppler, M. Rinne, C. Wijting, C. Ribeiro, and K. Hugl, “Device-to-device communication as an underlay to lte-advanced networks,” *Communications Magazine, IEEE*, vol. 47, pp. 42–49, Dec 2009.
- [50] Grønsund, P., *Cognitive Radio from a Mobile Operator’s Perspective: System Performance and Business Case Evaluations*. PhD thesis, University of Oslo (UiO), 2013.
- [51] EURECO, “SENDORA: Study and demonstration of the Wireless Sensor Network aided Cognitive Radio concept,” in *Technology and Regulation workshop Brussels*, Seventh Framework Programme, 2010. <http://cordis.europa.eu/fp7/ict/future-networks/crw/7mercier.pdf>.
- [52] Radhakrishnan, A., “Android phone controlled Beagle board based PSCR in a Dynamic Spectrum Access environment,” Master’s thesis, Virginia Tech, 2010.
- [53] Rozas, A., *Análisis y diseño de escenarios para redes cognitivas en una plataforma móvil, PFC*. Laboratorio de Sistemas Integrados, ETSIT-UPM, sep 2011.
- [54] “Serval Project.” <http://www.servalproject.org/>. Último acceso el 17-8-2014.