

Computación Móvil y Protocolos Ad-Hoc

Miguel Angel Ortuño Pérez

mortuno@gsysc.escet.urjc.es

Departamento de Ciencias Experimentales e Ingeniería

Universidad Rey Juan Carlos

Mayo de 2002

Índice General

1	Introducción	3
1.1	Movilidad vs Nomadicidad	3
1.2	Limitaciones de la Nomadicidad	4
1.3	La Movilidad y la Torre de Protocolos	4
1.4	Macromovilidad y Micromovilidad	4
2	Redes Inalámbricas	5
2.1	Estándars en nivel de enlace	5
2.2	Problemas de TCP en las redes inalámbricas	5
3	Redes de Area Local Inalámbricas IEEE 802.11	6
3.1	Arquitectura 802.11	6
3.2	Variantes del Estándar	6
3.3	Aspectos prácticos	7
4	Mobile IP	9
5	Cellular IP	10
6	Mobile IPv6	11
6.1	Introducción a IPv6	11
6.2	Mobile IPv6	11
7	Implementaciones Libres de Protocolos de Movilidad	13
7.1	Implementaciones de Mobile IP	13
7.1.1	Mobile IP at the National University of Singapore (NUS)	13
7.1.2	MobileIP Monarch Project	13
7.1.3	Secure Mobile	13
7.1.4	Solaris MobileIP implementation	13
7.1.5	Linux Mobile IP	14
7.1.6	MIPL: Mobile IPv6 for Linux	14
7.2	Implementaciones de Cellular IP	14
7.2.1	Columbia Cellular IPv4	14
7.2.2	Cellular IPv6	14
7.3	Implementaciones de HMIP: Hierarchical Mobile IP	14
7.3.1	Dynamics Mobile IP	14
7.3.2	HMIPv6: INRIA Hierarchical Mobile IPv6	15

8	Experimentación con Mobile IP	16
8.1	Esquema de la maqueta	16
8.2	Mediciones	16
8.3	Conclusiones	17
9	Redes Ad-Hoc	20
9.1	Aplicaciones de las Redes Ad-Hoc	20
9.2	Características de las Redes Ad-Hoc	20
9.3	Protocolos de Encaminamiento para redes Ad-Hoc	21
9.4	DSDV: Destination-Sequenced Distance-Vector	22
9.5	DSR: Dynamic Source Routing Protocol	22
9.5.1	Descripción del protocolo	23
9.5.2	Aspectos Básicos del Descubrimiento de Ruta	23
9.5.3	Aspectos Básicos del Mantenimiento de Ruta	23
9.5.4	Técnicas Adicionales para el Descubrimiento de Ruta	23
9.5.5	Técnicas Adicionales para el Mantenimiento de Ruta	24
9.5.6	Otras consideraciones sobre DSR	24
9.6	Otros Protocolos Ad-Hoc	24
9.7	Tendencias y Trabajo Futuro en Redes Ad-Hoc	25
10	Tendencias Futuras: Computación Ominipresente	26

Resumen

Los ordenadores cada vez son más pequeños. Continuamente aparecen nuevos dispositivos (portátiles, agendas electrónicas, teléfonos y un largo etcétera), de reducido tamaño, precio y peso, capaces de conectarse a la red con tecnología inalámbrica. Los equipos se mueven, eso hace aparecer nuevos requerimientos y problemas no previstos por los protocolos convencionales. Mostraremos el estado del arte al respecto así como algunos resultados de la experimentación con una implementación concreta.

En ocasiones, las estaciones móviles tienen capacidad de conectarse entre sí, pero no disponen de infraestructura de acceso a Internet. En tal caso pueden crear una red Ad-Hoc, una estructura específica para ese momento. Repasaremos algunos problemas que se presentan y soluciones actuales.

1 Introducción

Es difícil empezar un documento sobre redes de computadores sin hablar del explosivo crecimiento de internet, de las comunicaciones inalámbricas y de la proliferación de aparatos cada vez más pequeños y baratos susceptibles de conectarse a una red de comunicaciones. El *ordenador* del que hablamos hoy ya no tiene por qué ocupar ni una habitación ni una mesa, puede ser una agenda electrónica, un teléfono o un PDA que llevemos en el bolsillo. En un futuro próximo es claro que veremos dispositivos de todo tipo conectados a la red: vehículos, electrodomésticos, juguetes, equipos industriales y un largo etcétera que probablemente aún no imaginamos.

Los usuarios cada vez son más numerosos y más exigentes, pues dependen en mayor medida de estas tecnologías. Demandan continuamente mayores y nuevas prestaciones, desde cualquier sitio donde estén.

En los años 80 con la aparición de los PCs y la extensión de las redes locales cobran especial importancia los sistemas distribuidos. En los 90 los ordenadores son lo bastante pequeños como para moverse: Aparece la computación móvil. Pero actualmente los ordenadores también pueden comunicarse *mientras* se mueven. Empieza a hablarse de computación ubicua o omnipresente (*Pervasive*): Pequeños o grandes ordenadores integrados entre sí que nos envuelvan completamente hasta el punto de que prácticamente olvidemos su presencia.

Estos cambios no son sólo cuantitativos, son sustanciales. El diseño en capas de los sistemas de comunicaciones ha demostrado ser eficaz, tanto como para hacer posible que protocolos y aplicaciones de hace veinte o treinta años sigan funcionando hoy en situaciones completamente distintas. Funcionando, sí, pero infrautilizando el potencial de los ya mencionados avances en prestaciones, precio y tamaño del hardware.

1.1 Movilidad vs Nomadicidad

Los principios del protocolo IP son de los años sesenta, entre otras cosas lleva implícito el que un ordenador es algo grande y pesado y por supuesto, fijo. Hace tiempo que los ordenadores pueden moverse de un punto a otro, por ejemplo de casa al trabajo y al hotel, por citar el tópico. Es necesaria una nueva dirección IP en cada punto, que incluso no es raro que se configure a mano (a pesar de que existan alternativas como DHCP). El cambio de dirección obligará a reiniciar las conexiones, y con ello tal vez las aplicaciones. Esto basta a la mayor parte de los usuarios; es la *nomadicidad* [27], o *portabilidad* como lo denominan otros autores [20].

La aparición de la tecnología inalámbrica, supone un paso adelante muy importante: Los ordenadores pueden tener conexión incluso *mientras* se mueven. Y se desea mantener la conexión, manteniendo la dirección IP.

Una definición de movilidad la encontramos en [27] (capítulo 3.2.2). Mobility: *The ability of a node to change its point-of-attachment from one link to another while maintaining all existing communications and using the same IP address at its new link.*

Si un ordenador está quieto, no hay ningún inconveniente en que una dirección IP sea al tiempo

- Una *identificador*, distingue un equipo de otro.
- Una *localizador*, aporta la ubicación física donde han de dirigirse los datagramas.

Si el ordenador es capaz de moverse, es claro que ambos conceptos deben separarse. A partir del primero, es necesaria una entidad que obtenga el segundo en cada momento. Esta entidad se denomina *Location*

Directory o *Mobility Binding*. Todo esto es la base de la movilidad. Hay distintas formas de abordar este problema, la solución que parece más consolidada es Mobile IP, que trataremos posteriormente.

1.2 Limitaciones de la Nomadicidad

Disponer sólo de Nomadicidad y tener que cambiar la dirección IP en cada punto puede no parecer un problema tan serio. Puede bastar, el ejemplo más claro son muchos de los servicios basados en web, donde la IP no es importante.

En otros casos, mantener la dirección IP sí es relevante, algunos son evidentes, como mantener conexiones o permanecer accesibles en un misma dirección. Pero podemos citar otras situaciones no tan obvias como [27]:

- Aplicaciones cliente/servidor basadas en la IP, por la que identifican a los clientes.
- Hay motivos para pensar que en el futuro también los servidores serán móviles.
- Software propietario que en su licencia limita el uso según la dirección IP.
- Firewalls u otros elementos de seguridad pueden imponer trabajar sólo desde ciertas direcciones.
- La dificultad que conlleva la administración de un conjunto de direcciones para repartir mediante técnicas como DHCP.

1.3 La Movilidad y la Torre de Protocolos

Una cuestión importante es dónde se introduce la movilidad. La mayoría de las veces se hace en el nivel de red, pero podría ir en otros niveles, superiores o inferiores. Hacerlo en nivel de enlace puede tener alguna ventaja, (fundamentalmente la facilidad para poner otros protocolos de red por encima) pero también presenta problemas intrínsecos:

- Permiten movilidad sólo sobre un cierto tipo de medio. No se puede pasar por ejemplo de 802.11 a un modem. No permiten *heterogeneous mobility*.
- Además, hay que buscar una solución para cada nivel de enlace.
- El área geográfica es limitada, por la propia naturaleza del nivel de enlace.

1.4 Macromovilidad y Micromovilidad

La división jerárquica de la red en dominios divide a su vez el problema de la movilidad en dos sub-problemas: Macromovilidad y micromovilidad. La macromovilidad trata el movimiento de estaciones entre dominios distintos, típicamente pertenecientes a distintas entidades y a distancias relativamente altas. En micromovilidad el movimiento es dentro de un mismo dominio, lo que suele llevar consigo, reciprocamente, distancias menores (en términos físicos o de latencia) así como una única organización.

Es frecuente combinar protocolos de ambos tipos: Un mecanismo de macromovilidad encima de otro responsable de la micromovilidad. Parece claro que Mobile IP es la solución más extendida para el primer grupo, apenas se consideran otras alternativas. Para micromovilidad hay más propuestas [25].

2 Redes Inalámbricas

2.1 Estándars en nivel de enlace

Dentro del nivel físico y de enlace han aparecido varios estándares en los últimos años [1]. Entre ellos destacan:

- IrDA [29] (*InfraRed Data Association*) es un organismo internacional creado en 1993 que define y promueve un estándar para intercambio de datos por infrarojos, espectro que no está regulado. Tiene un conjunto de protocolos: IrDA Data para transmisión de datos e IrDA control para información de control. Originalmente soporta 115 kbps hasta dos metros, se modifica varias veces hasta alcanzar los 4 o 16 MBps. Tal vez la mayor limitación de esta tecnología es que no permite atravesar opacos (está basado en luz visible).
- IEEE 802.11. También conocido como *WI-FI Wavelan* o *Ethernet Inalámbrico*. Permite entre 1 y 54 Mbps, con un alcance de varios centenares de metros. Es la tecnología que parece más prometedora, la trataremos con más detalle en la próxima sección.
- Bluetooth [11] aparece en 1998, es una norma desarrollada por un consorcio de empresas con el objetivo de reemplazar IrDA y permitir comunicación inalámbrica entre dispositivos generalmente pequeños y próximos: Ordenadores, PDAs, teléfonos móviles, periféricos etc. Soporta hasta 8 dispositivos en una piconet o PAN (*Personal Area Network*). Opera en la banda de 2,56 GHz y ofrece hasta 1Mbps con un alcance máximo de 10 metros.

2.2 Problemas de TCP en las redes inalámbricas

En las implementaciones modernas de TCP hay *control de congestión*: Suponen que el nivel de enlace es fiable (así suele ser si el nivel físico es cable o fibra óptica) y que si se pierden datagramas es que algún router intermedio está congestionado y los está descartando. Entonces *educadamente* envía los paquetes más despacio. Pero lo que realmente sucede es que en nivel de enlace ahora es inalámbrico, no es tan fiable y está perdiendo tramas. TCP lo que debería hacer es todo lo contrario, insistir con mayor frecuencia en el envío de los segmentos. El resultado es una pérdida de rendimiento.

Una solución puede ser añadir fiabilidad en capa de enlace. Eso tiene el inconveniente de que los reintentos en nivel de enlace hagan que niveles superiores ya den el segmento por perdido y se reenvie de todos modos. Esto podría paliarse comprobando si se están reenviando segmentos, pero no es posible en el caso de comunicaciones encriptadas.

Otras propuestas para mejorar TCP van en la línea de que el nivel de enlace aporte información al nivel de transporte, pero esto va frontalmente en contra del modelo de capas.

3 Redes de Area Local Inalámbricas IEEE 802.11

3.1 Arquitectura 802.11

El comité IEEE 802.11 publica en 1997 un conjunto de normas para redes de area local inalámbricas [3]. Las estaciones operan en la banda de 2.4 GHz ISM (Instrumentation, Scientific and Medical) , frecuencias donde no es necesaria licencia. En su arquitectura (figura 1) el bloque fundamental es el BSS (Basic Services Set), lo que se conoce habitualmente como celda. Consta de un grupo de estaciones que ejecutan el mismo protocolo MAC y compiten por el acceso al medio compartido. Una celda puede operar de dos modos:

- Modo *Ad-Hoc*, es el esquema más sencillo. Todas las estaciones son iguales.
- El modo *Infrastructure*, que sigue el esquema cliente-servidor donde hay una estación principal, el *access point*.

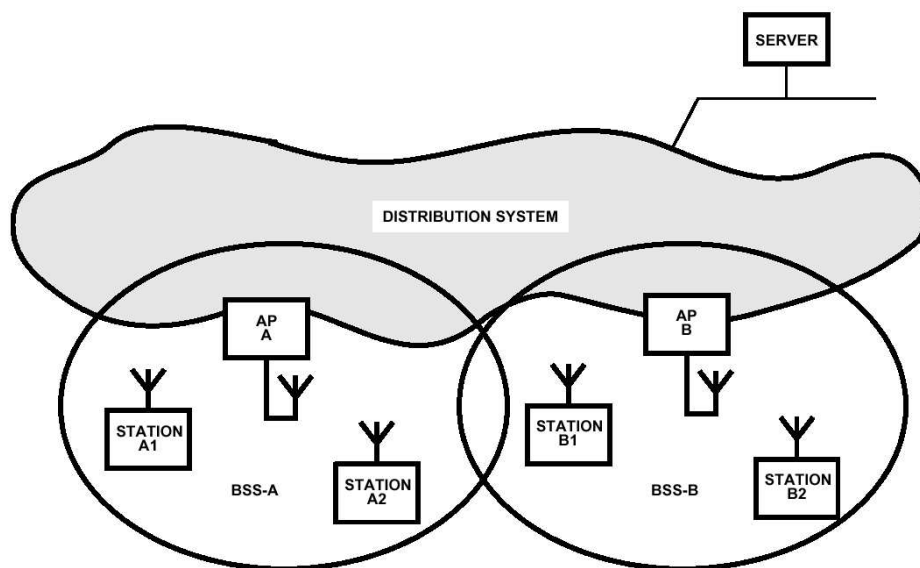


Figura 1: Arquitectura 802.11

El área geográfica cubierta por el BSS es el BSA, *Basic Service Area*. Los BSS pueden estar aislados o conectados a otra red. Varias celdas conectadas entre sí, (generalmente por otra LAN cableada), se pueden agrupar en un ESS (*Extended Services Set*). El nivel de control de enlace lógico (LLC) ofrece el conjunto como una sola LAN lógica. El estándar define movilidad, de tres tipos:

- Sin transición, las estaciones se mueven dentro de la misma celda.
- Transición BSS, las estaciones se mueven de una celda a otra pero dentro del mismo ESS.
- Transición ESS, entre distintos ESS. En este caso se produce una interrupción del servicio.

3.2 Variantes del Estándar

Dentro de 802.11 hay varios estándares, algunos aún en desarrollo:

- IEEE 802.11. Es la norma original, publicada en junio de 1997. Soporta 1 o 2 Mbps.

- IEEE 802.11b *High Rate* [15] [5], aparecido en septiembre de 1999 es el formato que podemos encontrar actualmente en el mercado. Consigue un máximo de 11Mbps, es más resistente a interferencias y compatible con el estándar anterior. Ocupa las mismas frecuencias, la mejora en las prestaciones se consigue con la técnica de modulación. (CCK, *Complimentary Code Keying*)
- 802.11a. En 1999 se define con el ánimo de que sea el sucesor de 802.11b. Llega hasta los 54 Mbps. Pero entre otros problemas, [16] opera en la banda de 5 GHz, con lo que no es compatible con 802.11b.
- 802.11g. Con prestaciones similares a 802.11a, pero compatible con 802.11b.
- 802.11i. Su definición se espera a lo largo de 2002. Llevará un estándar de seguridad, que debe ser implementado en hardware.

Una red 802.11 se integra bien con 802.3, por ejemplo comparten tamaño de trama y de direcciones. De hecho también se le conoce como *Ethernet Inalámbrico*, pero este es fundamentalmente por motivos de marketing. El acceso al medio usado en IEEE 802.3 CSMA/CD: *CSMA with Collision Detection* no es aplicable en esta tecnología. No es posible detectar todas las colisiones:

- El hardware es *Half Duplex* (*Full Duplex* haría los equipos demasiado caros).
- El problema del nodo oculto: Dadas las estaciones ABC, A quiere transmitir a B, y no puede detectar que C lo está haciendo en ese instante, mientras que B, que la tiene más próxima, sí.

Por tanto el acceso al medio es CSMA/CA (*Carrier Sense Multiple Access with colision Avoidance*). Cuando una estación está a la escucha puede hacer detección de portadora *física*. Además hay una detección de portadora *virtual*: Antes de emitir se hace una petición RTS (*Request To Send*), a la que se responde con CTS (*Clear To Send*). Así, aunque una estación no detecte al nodo emisor, sí escuchará el CTS.

Los 1518 bytes de una trama ethernet 802.3 resultan demasiado para un medio inalámbrico. Pero 802.11 debe soportarlo. Así que se fragmenta y defragmenta, con parada y espera.

El protocolo incluye técnicas para ahorro de energía, aspecto muy importante cuando lo normal es que los equipos se alimenten con baterías. También está previsto el cifrado en este nivel mediante WEP (*Wired Equivalent Privacy*) que si bien puede ser roto con un esfuerzo moderado, cumple su objetivo de ofrecer una privacidad análoga a la del cable.

3.3 Aspectos prácticos

El enlace más eficiente lo conseguiríamos con un solo cliente contra una estación base, con visibilidad directa y a pocos metros. Así podríamos tener los 11 Mbps teóricos. Si las condiciones no son tan buenas, el ancho de banda disminuirá. Pero por suerte esta tecnología es bastante tolerante a situaciones no tan favorables.

En un entorno de oficina, con una estación base tarjetas convencionales podemos tener cobertura en un círculo de unas cuantas decenas de metros de radio y con unas cuantas decenas de usuarios. Para cubrir áreas mayores, se pueden usar varios puntos de acceso, cada uno en un canal distinto y evitando en lo posible los solapamientos.

Modificando las antenas y en espacios abiertos, con visibilidad directa, se pueden alcanzar varios kilómetros. De hecho empieza a crecer en muchas ciudades comunidades interesada en esta tecnología, levantando redes inalámbricas gratuitas de ámbito metropolitano. Podemos citar MadridWireless¹ y BarcelonaWireless² El punto más interesante para introducir mejoras es precisamente éste, las antenas. El uso de amplificadores es posible pero presenta obstáculos técnicos que hacen que sea una solución cara. También puede implicar problemas legales. Y no debe olvidarse que siendo una comunicación bidireccional, no serviría de nada amplificar solamente en una de las estaciones. Los principales obstáculos para estas frecuencias son [6]:

- Objetos que absorban microondas, como obstáculos geográficos, edificios, árboles y personas.
- Objetos que reflejen o difuminen la señal, como metal, vallas, tuberías, pantallas o simplemente agua.

¹<http://madridwireless.net>

²<http://barcelonawireless.net>

- Fuentes de ruido en la banda de 2.4 GHz, como hornos microondas, teléfonos inalámbricos u otras redes 802.11.

La norma IEEE 802.11 define un total de 14 canales, 14 frecuencias. En Estados Unidos todos estos canales son de libre uso sin licencia, En Japón el único permitido es el 14. en Europa, del 1 al 13. ³ Un aspecto muy importante es tener en cuenta que el canal representa sólo la frecuencia central que usa el transmisor. Por ejemplo 2,412 GHz para el canal 1 y 2,417 GHz para el 2. Sólo hay 5 MHz de separación entre los canales, mientras que la señal ocupa unos 30 MHz del espectro, 15 MHz por encima y 15 MHz por debajo de la frecuencia centra. Por tanto, sólo hay como mucho tres canales que se pueden usar sin interferencias entre ellos. La mayoría de los usuarios no modifican los parámetros de sus tarjetas y estaciones base. Así, es de esperar que el canal 7, que suele ser la configuración por defecto, sea el más ocupado.

canal	frecuencia central	Lim inf	Lim Sup
1	2.412	2.397	2.427
2	2.417	2.402	2.432
3	2.422	2.407	2.437
4	2.427	2.412	2.442
5	2.432	2.417	2.447
6	2.437	2.422	2.452
7	2.442	2.427	2.457
8	2.447	2.432	2.462
9	2.452	2.437	2.467
10	2.457	2.442	2.472
11	2.462	2.447	2.477
12	2.467	2.452	2.482
13	2.472	2.457	2.487
14	2.477	2.462	2.492

Tabla 1: Canales y Frecuencias (GHz) en IEEE 802.11

³La normativa española al respecto la podemos encontrar en el suplemento del nº 164 del Boletín Oficial del Estado, 10 de Julio de 1991, revisado el 25 de Junio de 1993, documentos ETS 300-328, ETS 300-339.

4 Mobile IP

IP móvil (*Mobile IP*) [17] [23] [20] es una modificación de IP para permitir macromovilidad. El objetivo es procesar datagramas de forma continua a un host (el *Mobile Node*, MN) que se mueve por la red. Con tal de que siga teniendo acceso a nivel de enlace con algún punto de conexión, el MN puede cambiar su ubicación física en la red manteniendo la misma dirección IP, puede seguir comunicándose de forma ininterrumpida con otros hosts en Internet, los denominados *Correspondent Nodes* (CN). Mobile IPv4 se puede describir como tres conjuntos de tareas: Supongamos una estación que sale de su red (*Home Network*) y llega a una distinta (*Foreign Network*).

- Descubrimiento de Agente: El MN llega a la *foreign network*, donde descubre un router denominado *Foreign Agent* (FA).
- Registro: El FA asigna al MN una dirección IP para su ubicación actual, una *care-off address*. El MN le comunicará a su *Home Agent* (HA). El HA es un router en la *Home Network* del MN que cooperará con el FA para que el MN reciba los datagramas que vayan dirigidos a su dirección original.
- Tunneling: El HA crea un túnel a la Care-Off address del MN.

Los datagramas pueden llegar desde el HA hasta el MN de dos formas

- Túnel hasta el FA. El HA toma los datagramas, los lleva en un túnel al FA que los desencapsula y envía al MN. El MN sigue usando su home address
- Túnel hasta el MN. El túnel llega hasta el propio MN. El MN necesita una co-located care-off address, esto es, una dirección real de la Foreign Network. Típicamente la habrá obtenido por DHCP.

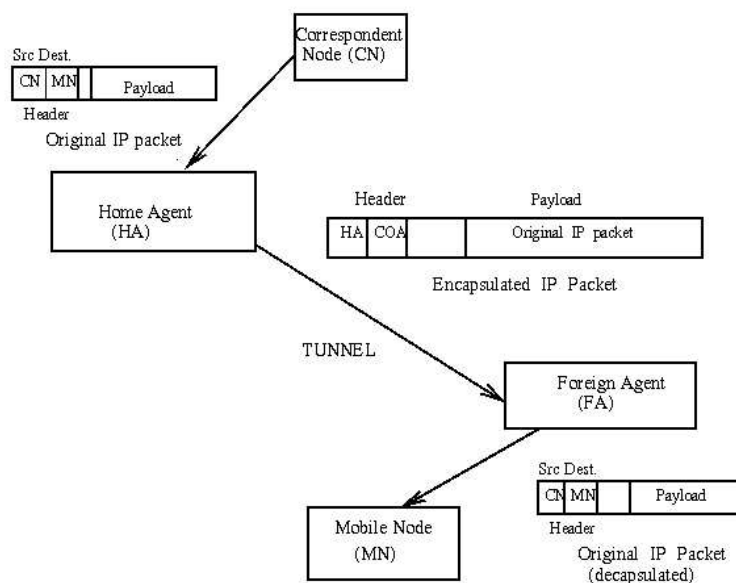


Figura 2: Esquema de Mobile IP

En caso de que el MN esté en su propia red, en la fase de descubrimiento de agente el MN descubre su Home Agent, en la fase de registro se hace saber al HA que la dirección actual es la IP en la Home Network, con lo que pueden deshacerse los túneles que se hubieran levantado previamente.

5 Cellular IP

Cellular IP [2] Es un concepto distinto a Mobile IP: No busca ser la base de movilidad global, sino permitir transiciones eficientes entre puntos de acceso en zonas geográficas restringidas. En determinada area de coberturas los usuario móviles tienen acceso inalámbrico. Aunque no sean demasiados, controlar todas las estaciones al tiempo consume demasiados recursos. Con el enfoque de Cellular IP, las estaciones sólo se registran cuando tienen llamadas activas. Los terminales ociosos tienen *conectividad pasiva*, hacen de vez en cuando *paging*, envían un mensaje cuyo significado es *hola estoy aquí*, de forma que cuando hace falta localizarlos, se tiene una idea aproximada de su ubicación.

La piedra angular de Cellular IP es la *base station*, el punto de acceso inalámbrico. Encamina paquetes y que desempeña las funciones que suelen hacer los MSC (*Mobile Switching Centers* y BSC Base Station Controllers) en la telefonía celular GSM. La red Cellular IP está conectada a Internet por un encaminador pasarela (*Gateway Router*.) Típicamente las configuraciones tendrán uno de estos encaminadores y varias estaciones base.

Los paquetes que parten del host se encaminan salto a salto por las estaciones base, hasta el router. Recordando por dónde han llegado estos paquetes, el gateway lleva los datagramas a los hosts. La movilidad entre los gateways la gestiona Mobile IP, reservándose a Cellular IP la movilidad dentro de la red. Los host móviles usan la dirección del gateway como care-of address. En este caso, el gateway entonces equivale al LFA (Lowest Foreign Agent) de Mobile IP.

Dentro de la red celular, las estaciones son identificadas por su *home address*, y se encamina directamente del gateway al host. Lo mismo ocurre con los datagramas del host móvil hacia internet. Los paquetes que envían las estaciones les sirven a las estaciones para tenerlos localizados y saber dónde enviarles posteriormente los paquetes. Los gateways de vez en cuando envían una baliza (*beacon*) para que las base station sepan dónde están.

6 Mobile IPv6

6.1 Introducción a IPv6

El protocolo de la internet actual, IPv4 [28] [24] está basado en el estado del arte en 1975. Cada estación se identifica por una dirección de cuatro bytes, lo que en su día parecía asegurar un número sobradamente alto de direcciones para todos los nodos que fueran a conectarse a la red. Pero el imprevisible éxito del protocolo, así como el ineficiente reparto de las direcciones en clases hace que ya a finales de los ochenta, principios de los noventa se perciba claramente la insuficiencia de este espacio de direcciones.

Las modificaciones mínimas consistirían en aumentar el tamaño de las direcciones, pero vista la exigencia de un nuevo protocolo, se le añaden los avances de quince años de experiencia en una nueva versión: IPv6 [4] [14] [13].

El cambio más importante es el tamaño de las direcciones, que pasa a ser de 4 a 16 bytes. Las estimaciones más pesimistas en cuanto a la eficiencia de la distribución de direcciones afirman que esto permite unos 1500 ordenadores por metro cuadrado de la superficie terrestre. Las más optimistas, hablan de varios trillones por metro cuadrado.

El formato del datagrama es muy similar al de IPv4, con tres simplificaciones fundamentales:

- Todas las cabeceras tienen el mismo formato, al que se le añaden extensiones si hace falta.
- IPv4 tenía un checksum en la cabecera. A pesar de que tanto el nivel superior (transporte) como el inferior (enlace) realiza sus propios chequeos, esto en su día tenía sentido: Memoria RAM defectuosa o errores de programación. Ambos motivos se consideran obsoletos y en IPv6 se eliminan los checksum.
- En IPv4 se segmentaba salto a salto: Si en un salto concreto una trama del nivel de enlace era insuficiente para un datagrama, se dividía en varias tramas. Esto se demuestra ineficiente: Por un lado el tiempo perdido en fragmentar y recomponer. Pero también, la probabilidad de que el datagrama se pierda aumenta notablemente; basta un fallo en una sola trama para obligar a desechar todas las demás. Así que en IPv6, si es necesario fragmentar, se hace extremo a extremo.

IPv6 incluye un conjunto de direcciones más rico que IPv4. Estas son:

- *Unicast*. Un datagrama a una dirección de este tipo se entrega al interfaz asociado a ella. Hay tres categorías
 - *Link local*. Un paquete que contenga una dirección de este tipo, no debe salir de su segmento de red.
 - *Site-Local*. Reservadas para intranets. Un router no debe sacar fuera de la organización un datagrama con una de estas direcciones.
 - *Global Address*. Dirección global, sin restricciones.
- *Multicast*. Identifica un conjunto de interfaces. A todas ellas se les debe entregar el datagrama.
- *Anycast*. Identifica un conjunto de interfaces, pero sólo se hará llegar el datagrama a una de ellas.

Se observa la ausencia de direcciones broadcast, que pasa a ser un caso particular de multicast.

6.2 Mobile IPv6

Todo lo contado sobre IPv4 Móvil es aplicable a IPv6 móvil (*Mobile IPv6*): Se trata de conseguir que un ordenador en una red que no es la suya pueda enviar y recibir datagramas como si no hubiera cambiado su localización física [22], [27] capítulo 12. La aproximación seguida en Mobile IPv6 es muy similar, con misma terminología y esquemas de funcionamiento. Los principales cambios en IPv6 que afectan a Mobile IP son:

- En IPv4 era necesaria la presencia de *Foreign Agents* que facilitasen a los *Mobile Nodes* las *co-located care-of addresses*. De esta manera, la dirección IP de un mismo FA servía múltiples MNs. Pero en IPv6 resulta muy fácil facilitar a las estaciones visitantes una dirección válida en esa red, hay muchísimas disponibles. Además, en IPv6 todos los routers tienen que implementar un *router discovery*. Ambos motivos hacen que los FA resulten innecesarios en Mobile IPv6.

- En IPv4 cualquier datagrama con opciones debía ser consultado por todos los routers. Esto resultaba ineficiente. En IPv6 se agrupan por un lado las direcciones que deben consultarse salto a salto, por otro las que tienen sentido extremo a extremo.
- Hay una nueva cabecera *routing header* que facilita el enrutado, dice por dónde deben pasar los datagramas.
- IPv4 no incluía mecanismos de autenticación, por lo que IPv4 móvil necesitaba su propias cabeceras para este fin. Mientras que los mecanismos de autenticación propios de IPv6 son suficientes para IPv6 móvil.

El mecanismo por el que un nodo mantiene su dirección de red en Mobile IPv6 es el siguiente:

- Primero el MN comprueba si está en su *Home Network*. Lo hace con una llamada ICMPv6 de descubrimiento de hosts.
- Luego obtiene una care-of address, una IP válida en la nueva red. Lo puede hacer de dos formas, o con DHCP v6 o con *Stateless Address Autoconfiguration*, que consiste en hacerse sobre la marcha una dirección concatenando un prefijo de esa red con la dirección de nivel de enlace. [27]
- Le comunica a su HA su nueva dirección
- El HA le hace llegar a algunos CN *Correspondent Node* esta dirección.
- Los CN que conocen la nueva co-located care-off address envían directamente allí los datagramas, usando una cabecera de enrutado IPv6 donde la nueva dirección es la penúltima. Los CN que no conozcan la dirección, envían los datagramas al HA, que los hará llegar al CN por un túnel. (El Routing Header también existía en IPv4 pero se usaba poco porque resultaba costoso consultar las cabeceras con opciones).
- Los datagramas que vuelven desde el MN al CN, pueden enrutarse sin ningún mecanismo especial, aunque también pueden ir a través del túnel.

7 Implementaciones Libres de Protocolos de Movilidad

Hemos hecho una búsqueda de implementaciones de los protocolos de macro y micromovilidad, con el objetivo de localizar aquellas que sean *libres*. Esta condición la entendemos imprescindible para poder usarlas como base de futuro trabajo. Libre significa que el autor permite su uso sin restricciones, acceso al fuente y su modificación, así como su redistribución [10].

7.1 Implementaciones de Mobile IP

7.1.1 Mobile IP at the National University of Singapore (NUS)

Se desarrolla para estudiar el rendimiento del protocolo tal y como lo describe la RCF 2002. Además de la movilidad, esta implementación incluye reserva de recursos y calidad de servicio. Está disponible en <http://opensource.nus.edu.sg/projects/mobileip/mip.html>

	SO	Version	License	Progr Lang	Rel Year	Pro Active	Size
MobileIPv4	Linux 2.0.34	3.0beta	GPL	C	1999	?	?
MobileIPv6	Linux 2.1.59	1.0alpha	GPL	C	1999	?	?

7.1.2 MobileIP Monarch Project

Desarrollado por el proyecto *Monarch* de *School of Computer Science at Carnegie Mellon University*

	SO	Version	License	Progr Lang	Rel year	Pro Active	Size
MobileIPv4	NetBSD 1.1, FreeBSD 2.2.2	1.1.0	BSD-like	¿?	1998	?	?
MobileIPv6	FreeBSD 2.2.2	1.0	BSD-like	¿?	1997	?	?

<http://www.monarch.cs.cmu.edu/software.html>

7.1.3 Secure Mobile

Es parte de un proyecto de la Portland State University. Su objetivo es integrar Mobile-IP e IPSEC , aportando seguridad a todas las comunicaciones. Está desarrollado para FreeBSD, y del Mobile Node también hay implementación Linux.

	SO	Version	License	Progr Lang	Rel year	Pro Active	Size
MobileIPv4	FreeBSD 4.5, Linux 2.2.12-20	4.5	FreeBSD-like	C	2001	Y	?
MobileIPv6	-	-	-	-	-	-	-

<http://www.cs.pdx.edu/research/SMN>

7.1.4 Solaris MobileIP implementation

Su autor es el grupo Mobile IP de los laboratorios Sun. Está diseñado para Solaris, se supone que debe funcionar sobre Linux. Sun ya no da soporte a esta implementación, al formar parte de Solaris 8.

	SO	Version	License	Progr Lang	Rel Year	Pro Active	Size
MobileIPv4	Solaris 2.5.1, 2.6	?	Solaris License	¿?	1999	N	C
MobileIPv6	-	-	-	-	-	-	-

<http://playground.sun.com/pub/mobile-ip/sunlabs>

7.1.5 Linux Mobile IP

Es un proyecto de MosquitoNet Mobile Computing Group en la Stanford University.

	SO	Version	License	Progr Lang	Rel Year	Pro Active	Size
MobileIPv4	Linux 2.2.x	2.0.2beta	FreeBSD-like	C	2000	N	?
MobileIPv6	-	-	-	-	-	-	-

<http://gunpowder.stanford.edu/mip/>

7.1.6 MIPL: Mobile IPv6 for Linux

Esta implementación parte de la desarrollada por el Helsinki University of Technology. Es para Linux y tiene licencia GPL.

	SO	Version	License	Progr Lang	Rel Year	Pro Active	Size
MobileIPv4	-	-	-	-	-	-	-
MobileIPv6	Linux 2.4.x	0.9.1	GPL	C	2001	Y	?

<http://www.mipl.mediapoli.com/>

7.2 Implementaciones de Cellular IP

Pertenece a Ericsson y la Columbia University

7.2.1 Columbia Cellular IPv4

Hay versiones para Linux http://www.comet.columbia.edu/cellularip/linux_src_code.htm así como FreeBSD http://www.comet.columbia.edu/cellularip/fbsd_src_code.htm

	SO	Version	License	Progr Lang	Rel Year	Pro Active	Size
CellularIP	Linux 2.2.14,FreeBSD 3.2	1.1(Linux), 1.0(BSD)	Prop	C	2000		?

<http://www.comet.columbia.edu/cellularip/>

7.2.2 Cellular IPv6

Sus autores son la empresa griega INTRACOM y el VTT Electronics de Finlandia

<http://cipv6.intranet.gr/>

7.3 Implementaciones de HMIP: Hierarchical Mobile IP

Es una extensión de Mobile IP para permitir micromovilidad: Los FA se configuran en árboles de forma que los túneles son segmentados: Si un Mobile Node se desplaza entre dos FA del mismo árbol, sólo cambia el último segmento del túnel, sin necesidad de notificar nada al Home Agent.

7.3.1 Dynamics Mobile IP

Esta desarrollado en la Helsinki University of Technology (HUT). Es escalable y jerárquico. Funciona sobre linux, y está parcialmente portado a Microsoft Windows.

	SO	Version	License	Progr Lang	Rel Year	Pro Active	Size
Hierarchical MobileIP	Linux 2.2.x, 2.4.x	0.8.1	GPL	C	2000	Y	?

<http://www.cs.hut.fi/Research/Dynamics/>

7.3.2 HMIPv6: INRIA Hierarchical Mobile IPv6

HMIPv6 es un Mobile IPv6 jerárquico desarrollado en la universidad de California. Soporta Macromovilidad y Micromovilidad.

	SO	Version	License	Progr Lang	Rel Year	Proj Active	Size
Hierarchical MobileIP IPv6	FreeBSD 3.4	2.0	FreeBSD-like	C	2000	N	?

<http://www.inrialpes.fr/planete/people/bellier/hmip.html>

8 Experimentación con Mobile IP

8.1 Esquema de la maqueta

Analizadas todas las implementaciones de Mobile IP disponibles, resulta claro que la más interesante es es Dynamics Mobile IP de la Helsinki University of Technology (HUT), montamos una maqueta con la versión 0.8.1 para poder estudiar su comportamiento. Tenemos como referencia los resultados de sus autores, [9] [8] [7] que intentaremos repetir.

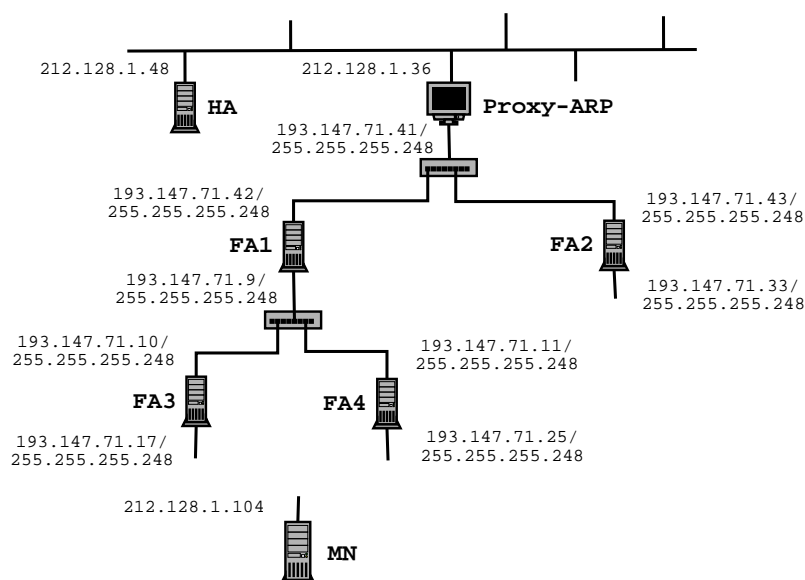


Figura 3: Esquema de la Maqueta Mobile IP

Hut dynamics Mobile IP hace especial énfasis en los túneles jerárquicos, para poder experimentar con ello instalamos una maqueta de PCs con varias subredes, cuyo esquema vemos en la figura 3; Los interfaces de red son IEEE 802.3, excepto en el MN y los MN de los extremos finales, que son tarjetas inalámbricas IEEE 802.11 configuradas en modo Ad-Hoc.

En cada subred habrá un FA. A cada FA llegará un túnel. Cuando el MN se mueva p.e. entre la subred 3 y la 5, será necesario deshacer completamente un túnel y levantar otro nuevo. Mientras que si el movimiento es entre el nodo 3 y el 4, los túneles jerárquicos permiten conservar el fragmento de tunel que llega hasta la subred 2, y modificar sólo el último tramo.

8.2 Mediciones

El objetivo más importante es que el rendimiento caiga lo menos posible a pesar de los handovers. Usando las herramientas que aporta esta implementación, forzaremos el handover cada cierto intervalo de tiempo.

Lo primero es tener en cuenta la naturaleza de la tecnología inalámbrica, muy sensible a factores externos. En la figura 4 podemos ver un ejemplo de la fluctuación de señal y ruido en un intervalo de 7 horas, sin actividad en la maqueta.

En la figura 5 sobre el eje X representamos el nº de segundos entre handover y handover, en el eje Y, el ancho de banda de una transmisión TCP entre el MN y un *Correspondet Node*. (Para conocer este dato usamos el benchmark *NetPerf*⁴). Tal y como era previsible, el rendimiento es más alto cuando las transiciones se hacen entre FA en la misma jerarquía.

⁴<http://www.netperf.org>

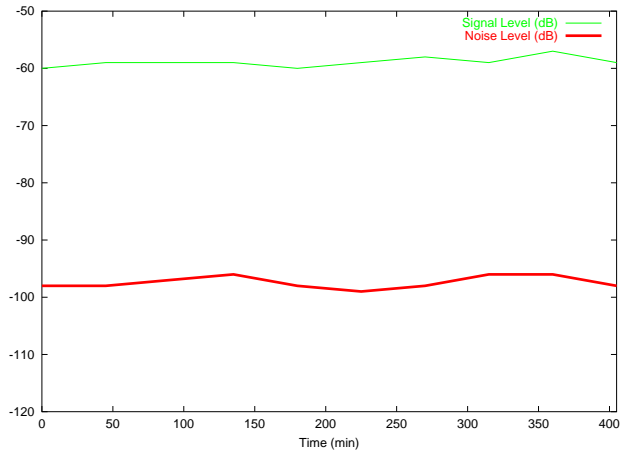


Figura 4: Ejemplo de Señal y Ruido con la maqueta en reposo

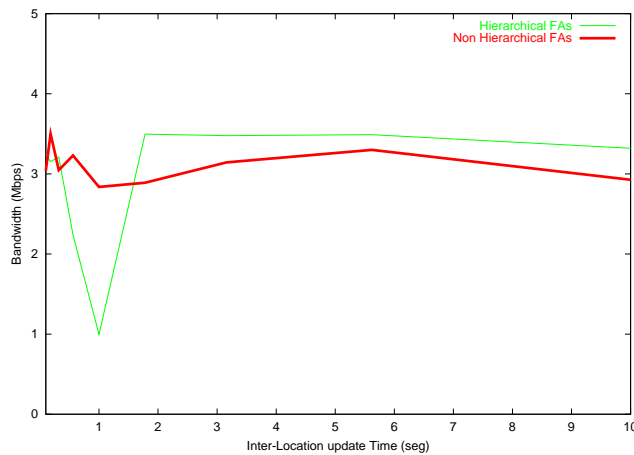


Figura 5: Ancho de banda en función del tiempo entre handovers

Para aumentar el realismo de la maqueta, se simula el alejamiento entre la *Foreign Network* y la *Home Network*, introduciendo retardos con la herramienta NistNet⁵. En la figura 6 se muestran los resultados, una sensible caída del ancho de banda.

Además del *benchmarking* con NetPerf, buscamos comprobar el número de paquetes que se pierden en una transmisión UDP. Con la ayuda de la librería LowerLayer⁶ realizamos un pequeño cliente y servidor para poder medirlo. En la figura 7 se observa cómo aumenta la pérdida del n^o de paquetes al aumentar su tamaño. En la figura 8 se repiten las mediciones, simulando el alejamiento del Home Agent 100ms.

8.3 Conclusiones

Tras la experimentación, parece quedar claro porqué este protocolo se considera asentado, y esta implementación, la más extendida. Los resultados son satisfactorios, se ajustan a lo prometido por sus autores: Un sistema al que apenas le afectan las transiciones entre puntos de acceso. Es flexible y relativamente sencillo de configurar. Sigue en desarrollo, está mantenido por sus autores a quienes se lee con frecuencia en la lista de correo del proyecto y cuenta con un grupo de usuarios bastante activo.

⁵<http://snad.ncsl.nist.gov/itg/nistnet>

⁶http://gsync.escet.urjc.es/simple_com

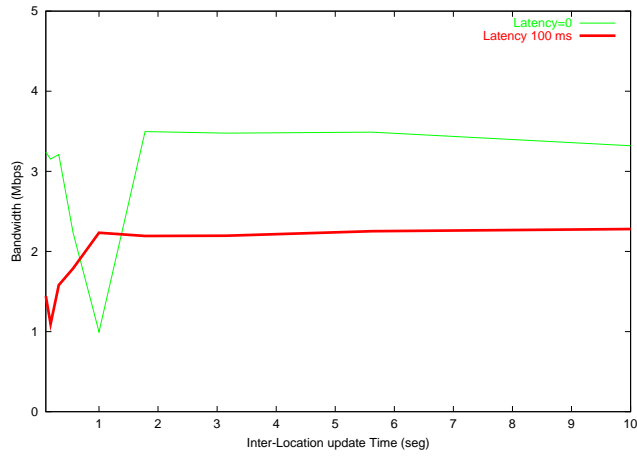


Figura 6: Ancho de banda introduciendo latencia

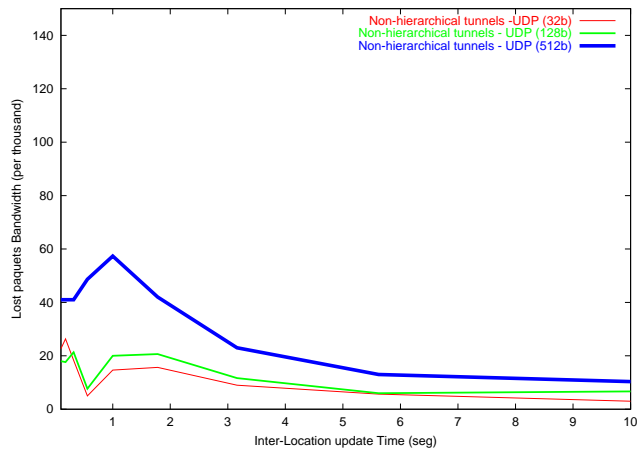


Figura 7: Paquetes Perdidos. Latencia=0

En cuanto a los inconvenientes podemos citar la documentación, que puede ser suficiente pero no se corresponde con la exigible a un producto terminado. El principal obstáculo para usar la implementación probablemente no esté relacionado directamente con Mobile IP sino con las tarjetas 802.11: Configurarlas correctamente en Linux para integrarlas con este software no es paso trivial, se requiere una comunicación a bastante bajo nivel con el driver. Si bien al ser esta un area de creciente interés, son de esperar rápidas mejoras.

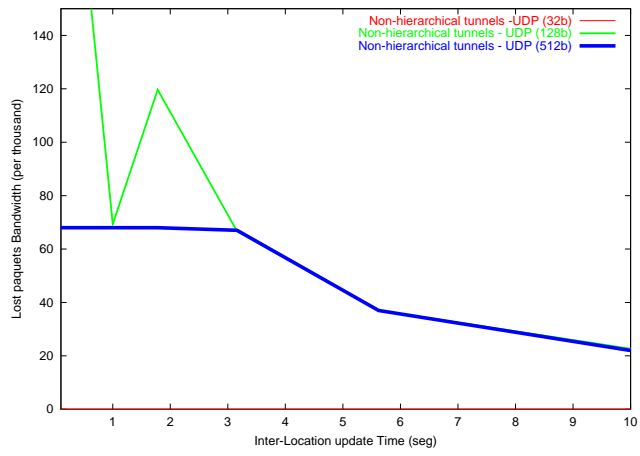


Figura 8: Paquetes Perdidos. Latencia=100 ms

9 Redes Ad-Hoc

Una red *ad-hoc* se define como una red de comunicaciones que se forma cuando se necesita, sin precisar de ninguna infraestructura externa como pueda ser la actual Internet. [21] pg 4. Están formadas por nodos similares que cooperan entre sí. Estas redes no dependen de infraestructuras como cableado, puntos de acceso, routers pre-existentes o ni siquiera alimentación. La comunicación es típicamente inalámbrica y usando baterías, lo que hará la topología muy cambiante. Otro aspecto muy importante es que no deben necesitar administración por parte del usuario.

Si todos los nodos ven a todos los demás, el problema del encaminamiento está resuelto. Pero esto es muy infrecuente y probablemente un derroche de energía y ancho de banda. Cada nodo tendrá accesible directamente los nodos más próximos, accederá a los más distantes a través de otros que actuarán de encaminadores.

Uno de los servicios que un nodo puede ofrecer a los demás es precisamente el acceso a internet. IP móvil resulta entonces especialmente adecuado, un nodo se convierte en un Foreign Agent ofreciendo conectividad a cualquiera de los demás.

9.1 Aplicaciones de las Redes Ad-Hoc

Es fácil encontrar situaciones donde se ve el interés de las redes Ad-Hoc. Uno de los ejemplos más claros puede ser una reunión de trabajo: Un grupo de personas con ordenadores portátiles o PDAs. Son de distintas empresas y por tanto sus direcciones son distintas. Tal vez en la sala haya acceso a internet y puedan usar por ejemplo IP móvil, pero ¿para qué pasear sus datagramas por toda la ciudad o todo el país cuando están en la misma habitación?. Sus equipos probablemente estén dotados de puertos de infrarojos que les permitan formar una red para la ocasión.

Podemos pensar en una familia, donde hay un ordenador de sobremesa, el portátil del padre, de la madre y un juguete capaz de conectarse a la red. Se podrían configurar para que todos estén en la misma red. Los portátiles volverían a la red del trabajo al volver a la oficina. Pero esta configuración manual es lo que desea evitarse.

En otros casos, simplemente no habrá infraestructuras de apoyo. Pensemos en poblaciones aisladas o de orografía difícil, situaciones de emergencia, desastres naturales o en un campo de batalla. Pensemos también en las denominadas PAN (*Personal Area Networks*) o pico-nets : Una red formada por los dispositivos de una persona: Su reloj, su agenda, su teléfono móvil. Una red así puede querer entrar en contacto con la red de otra persona que en ese momento está enfrente.

Y aunque se puedan pensar muchos usos, la *killer ap* puede ser cualquier otra aplicación que hoy no imaginamos. Probablemente no sea necesario. La única cuestión es que con la tecnología inalámbrica, necesitar infraestructuras remotas es algo que pertenece al pasado.

9.2 Características de las Redes Ad-Hoc

En internet, el enrutado se hace a partir de las direcciones. Históricamente se comenzó encaminando a partir de las clases, hoy se hace usando CIDR (*Classless Inter-Domain Routing*). Pero el principio es el mismo: A partir de cierto prefijo, de tamaño fijo o tamaño variable, se localiza el destino. Para las redes ad-hoc inicialmente se intentó una aproximación similar, pero resultó inadecuado, demasiado costoso: En una de estas redes no hay relación entre una dirección de red y una ubicación física, por tanto no sirve de nada agregar prefijos en las direcciones. Al no poderse agregar, la escalabilidad es un problema serio: Las tablas de enrutado corren el riesgo de hacerse inmanejables, al exigir una entrada por dirección y no por grupo de direcciones.

Una red donde los nodos se mueven, aparecen y desaparecen, puede generar muchos mensajes de control. Es importante encontrar un equilibrio: Demasiados mensajes consumirán el ancho de banda sólo en mantener la red. Un número muy bajo, hará la información de las tablas obsoleta. Los mecanismos deben ser incrementales: Sería muy ineficiente que el cambio de un solo nodo obligase a recalcular la información de toda la red

Un asunto actualmente en investigación es el multicast, que puede ahorrar drásticamente ancho de banda en distribución masiva. Lo que no está claro es si debe incluirse en este nivel de los algoritmos. Por un

lado, los problemas de multidistribución pueden ser de naturaleza lo bastante específica como para que no merezca la pena mezclarlos con otras cuestiones. Pero por otra parte, los protocolos que se dedique mucho esfuerzo a conocer el estado de nodos intermedios que cambian continuamente. Si el multicast se convierte en n-unicast, se puede estar derrochando recursos.

Estas redes se basan en la cooperación de buena fe entre nodos. Un problema potencial es que esto no suceda: Por ejemplo la batería es un bien escaso y preciado. Si el nivel está bajo puede ser legítimo no gastarlo retransmitiendo para los demás, o limitarse a los mensajes más urgentes. Pero esto puede dar lugar a abusos.

Otro problema es que hay una gran variedad de medios físicos distintos, todos incompatibles entre si. Cuando todos los nodos emplean la misma tecnología en este nivel 1, pueden trabajar juntos. Pero sólo en ese caso. Esto es un serio inconveniente para llegar a un protocolo estándar.

Un aspecto muy importante a tener en cuenta es si los enlaces son simétricos. Si la estación A puede transmitir a la estación B, ¿B puede hacer lo mismo con A?. Depende de cada tecnología en concreto, lo más habitual es que sí. Si no es el caso, todos los algoritmos se complican notablemente.

Otra cuestión es qué nivel colocar la movilidad. Normalmente se hace en nivel de red. Esto presenta circunstancias favorables: En las tablas de enrutado se trabaja con direcciones de red, que las aplicaciones resuelven en direcciones de enlace: En la dirección de enlace del propio nodo destino si accesible directamente, o bien la dirección de enlace del nodo que hace de router y que se lo hará llegar. Si la movilidad está en el nivel de enlace, entre otras cosas hay que resolver direcciones de enlace con otras direcciones de enlace, lo que resulta poco natural.

Usando enlaces sin cables, el ancho de banda es típicamente un orden de magnitud inferior al cable. Sería deseable que las aplicaciones fueran conscientes de ello, y en cada caso adapten la información enviada: Tal vez reduciendo el número de frames por segundo en vídeo, el número de colores o la resolución en una imagen u omitiendo la animación en flash de utilidad dudosa.

La seguridad es otra cuestión delicada. Por un lado lo que está *en el aire* puede ser capturado fácilmente. La solución naturalmente es cifrar las comunicaciones. Pero la seguridad y el cifrado se basan en una distribución segura de claves y en una autoridad certificadora centralizada. Esto último es casi una contradicción en términos con una red ad-hoc, sin hablar del ancho de banda consumido.

9.3 Protocolos de Encaminamiento para redes Ad-Hoc

Muchos protocolos *convencionales* de internet en principio son aplicables a las redes Ad-Hoc: Son capaces de funcionar sin configuración inicial (son *self-starting*), se adaptan a cambios en la red y ofrecen múltiples rutas para un destino. Estos algoritmos de enrutado generales se dividen en dos grupos:

- Estado del enlace, como OSPF. Cada nodo conoce el estado de toda la red y luego calcula ruta óptima al destino aplicando el algoritmo de Dijkstra. En otros protocolos de estado del enlace no se distribuye la información completa sino en forma estadística, centrándose en las zonas que parecen más interesantes.
- Vector de Distancias (DV, *Distance Vector*), también conocidos como DBF *Distributed-Bellman-Ford*, donde cada nodo lo único que conoce de la ruta para llegar a otro es el primer salto y la distancia hasta el destino. Estos algoritmos presentan el inconveniente del salto al infinito: Supongamos una ruta que pase por los nodos ABCD. Supongamos que B quiere encaminar un paquete hasta D, pero el enlace BC está roto (tiene peso infinito). Entonces B intentará encaminar hasta D pasando por A, porque A le ofrece un camino. Pero no puede saber que ese camino precisamente acabará pasando por el mismo enlace roto, circunstancia de la que A no tiene noticia aún. Algunas técnicas paliativas para este problema son las conocidas como *split-horizon* y *poisoned-reverse*).

Los protocolos de estado del enlace no son adecuados para redes Ad-Hoc: Intentar que cada nodo guarde en todo momento el grafo completo no tiene sentido en un entorno que tan dinámico como los son estas redes. Así, los protocolos para redes Ad-Hoc se basan en vector de distancias. Todo lo que se conoce sobre protocolos de enrutado convencionales se puede aplicar aquí, lo que supone una gran ventaja. Pero es necesaria una adaptación, ya que precisamente el peor comportamiento de estos protocolos se da cuando los nodos se mueven con frecuencia. Centrándonos en protocolos Ad-Hoc, se pueden clasificar en:

- Proactivos o guiados por tablas. Guardan información de en todo momento sobre las rutas a todos los nodos. (Se trata de protocolos de vector de distancias, esta información como mencionamos en el apartado anterior es la distancia y el primer salto). Esta aproximación es la primera históricamente, usada a mediados de los 90. Pero guardar información sobre todas las rutas es caro. Se consume mucho ancho de banda en tablas de rutas que se levantan y caen, cuando es posible que el nodo no las necesite. La principal ventaja es que cuando una aplicación necesita enviar datos, la latencia es baja pues ya se dispone de la ruta.
- Reactivos, también conocidos como *bajo demanda*. Sólo buscan una ruta cuando va a ser necesaria. Este enfoque tiene menor coste, pero aumenta la latencia del primer paquete a enviar.
- Mixtos, que llegan a un compromiso entre ambos enfoques. Por ejemplo, Multipath Routing.

9.4 DSDV: Destination-Sequenced Distance-Vector

DSDV. *Destination-Sequenced Distance-Vector* ([18]). Es uno de los protocolos más clásicos. Está anticuado, pero sigue teniendo interés por motivos históricos y como referencia. Permite trabajar en nivel de enlace o en nivel de red. Si el direccionamiento se hace en el nivel de enlace, en el nivel superior se puede tener cualquier otro protocolo.

Toma como punto de partida el protocolo de vector de distancias RIP. Los paquetes se transmiten entre las estaciones usando tablas de encaminamiento que almacena cada nodo. Cada tabla en todo momento almacena para todos los destinos posibles: Cuál es el primer salto, número total de saltos y un nº de secuencia de esa información, número que ha asignado el nodo destino. Este último campo permite mantener la consistencia de la información: Los nodos no tendrán una hora global, este contador será la *hora lógica* por la que podré saber qué información es más reciente. Por convenio, este contador es un número par.

Cada estación retransmite por broadcast su tabla a sus vecinos, que la agregan a las suyas. Cuando aparece un camino mejor para cierto destino, el camino previo se puede desechar o se puede conservar como camino alternativo.

Para reducir el tráfico, se definen dos tipos de paquetes para transmitir las tablas: Volcado completo o incremental. Un paquete incremental debería caber en un solo paquete del nivel inferior. Los cambios más importantes se transmiten antes, en actualizaciones incrementales.

Información poco prioritaria puede ser la actualización de una ruta que no modifica su coste. O bien la actualización de un host que sabemos que aún no ha estabilizado su posición. ¿Cómo suponemos que está estable? Guardando el tiempo transcurrido entre la *primera* ruta y la mejor. (Sería desastroso que el movimiento de un nodo causase una tormenta de broadcast)

Un enlace roto puede ser detectado por el nivel de enlace. O puede ser deducido si transcurre cierto tiempo sin recibir noticias de un nodo que previamente era vecino. Cuando un salto se rompe, cualquier ruta que pase por él también está rota, así que esta es una información de la máxima prioridad. En este caso, el número de secuencia de esta información es impar, indicando que no ha sido el nodo destino quien lo ha marcado.

9.5 DSR: Dynamic Source Routing Protocol

DSR (*Dynamic Source Routing Protocol* [18]) es uno de los protocolos *bajo demanda* más puros. Las rutas se obtienen sólo cuando se necesitan, y es el nodo origen quien las fija completamente, así, cuando un paquete sale de su punto de partida, tiene la lista completa y ordenada de nodos por los que debe pasar. Una consecuencia inmediata de esto es que se garantiza la ausencia de bucles. Como todos los protocolos este tipo, la latencia es alta para el primer paquete.

Este protocolo asume ciertas premisas: Todos los nodos están deseando participar, de buena fe. El diámetro de la red (máximo mínimo para ir de extremo a extremo) será más bien pequeño (5 o 10 nodos). Los nodos se mueven, pero no tanto como para pensar que cada paquete vendrá de un punto distinto. Cada nodo, aunque tenga varios interfaces, usará sólo una dirección IP, su *Home Address*. El protocolo podrá aplicar ciertas optimizaciones si el interfaz puede trabajar en modo promiscuo (escuchar mensajes no dirigidos a él), aunque no es imprescindible (esto consume cpu y batería). También funcionará mejor si los enlaces son bidireccionales, pero no es imprescindible.

9.5.1 Descripción del protocolo

El protocolo está compuesto de dos mecanismos que trabajan coordinadamente:

- Descubrimiento de Rutas (*Route Discovery*). Si el nodo S quiere enviar un paquete al nodo D y no conoce ruta, la busca.
- Mantenimiento de Ruta (*Route Maintenance*). Si un nodo está enviando paquetes a un destino, y le llega un mensaje de error indicando que esa ruta está rota, si conoce una ruta alternativa la usa, si no, lanza un nuevo descubrimiento de ruta.

Y todo esto siempre bajo demanda, no hay ninguna operación que se realice periódicamente. Cuando el nodo emisor se mueva o cambie la topología de la red, el algoritmo percibe los cambios y se adapta a ello, pero sólo para las rutas que se estén usando.

9.5.2 Aspectos Básicos del Descubrimiento de Ruta

La mayor parte de las veces, cuando un nodo deba enviar un paquete, tendrá la ruta en caché. Si no es así, la busca. Para ello envía por *broadcast* una petición (*Route Request*), que tendrá un identificador (*request id*). Cuando un nodo recibe una de estas peticiones, si preguntan por él, responde a quien lo envió. En otro caso, se vuelve a retransmitir (a menos que sea la segunda vez que se escucha esa misma petición). Este paquete contiene una lista de todos los nodos por los que va pasando, lista que se almacena, contiene valiosa información sobre rutas.

Normalmente los enlaces son bidireccionales, con lo que la respuesta sigue exactamente el camino inverso a la petición. Si los enlaces son unidireccionales, hay que generar una nueva petición. (marcándola con un flag *Route Reply* para evitar una cascada de peticiones)

Mientras un nodo busca rutas para los paquetes a enviar, los almacena con una marca de tiempo en el *Send Buffer*. Si transcurrido cierto tiempo no llega una ruta, se reintenta. Tras varios intentos sin éxito, se descarta el paquete. También debe tenerse en cuenta que si hay una petición en curso, y llega un nuevo paquete el mismo destino, no debe generarse una nueva petición.

9.5.3 Aspectos Básicos del Mantenimiento de Ruta

Cuando un nodo retransmite un paquete, necesita confirmación de que ha llegado al siguiente salto. Este confirmación puede aportarla el nivel de enlace, (como sucede con IEEE802.11). O bien puede ser un ACK *pasivo*: Si el nodo A envía algo a B, y luego escucha a B retransmitirlo a C, tiene la certeza de que no hubo problemas. Si ninguno de estos mecanismos está disponible, se sube un flag en la cabecera pidiendo acuse de recibo explícito, si el enlace es bidireccional la respuesta vuelve por el mismo camino, si no lo es, será necesaria una nueva ruta.

Naturalmente, una ruta puede romperse. Supongamos un nodo C que recibe un paquete de A con la ruta ABCDE. Si no es posible alcanzar E, se responde a A con un mensaje *Route Error*. En este caso, A no lanza nuevas peticiones de ruta ni reintenta el envío por caminos alternativos, esto corresponde al nivel de transporte.

9.5.4 Técnicas Adicionales para el Descubrimiento de Ruta

Todos los paquetes contienen el historial de nodos por los que va pasando, así que cualquier estación que capture el paquete, bien para reenviarlo, bien por estar en modo promiscuo, puede almacenar esta información. En este segundo caso, la ruta en general no incluirá a la estación que lo ha escuchado.

A un *Route Request* se puede contestar sin ser el nodo final: Puede tenerse información en caché de una ruta para ese destino, con lo que la respuesta se obtiene concatenando el camino seguido por el paquete con la ruta en caché. En este caso, hay una aparente *mejora* que el protocolo prohíbe explícitamente:

Supongamos que el nodo F recibe una petición de ruta para el nodo E. Esta petición ha pasado por ABCF. F tiene almacenada la ruta FCDE, así que concatenando ambas subrutinas, obtiene ABCFCDE. Es fácil *caer en la tentación* de intentar optimizar la ruta omitiendo el bucle en F y devolviendo ABCDE. Pero

esto lo prohíbe el protocolo por ser potencialmente una fuente de problemas: Si un nodo devuelve una ruta donde él mismo no está, y esa ruta se cae, el mensaje *Route Error* no pasará por él.

Además, cuando un nodo obtiene una ruta a partir de lo que tiene en caché, probablemente sus vecinos también puedan hacerlo, con lo que debe esperar un intervalo de tiempo aleatorio o habrá muchas respuestas simultáneas que provocarán colisiones en el medio. Mientras, escuchará las respuestas, si alguna es mejor que la suya, cancelará su respuesta.

Una posible mejora es incluir en cada petición un una cota al número de saltos que puede tener la ruta (*hop limit*). Si se le da valor 0, es análogo a un ARP convencional. Con otros valores, se puede intentar localizar rutas cortas, para ir incrementando el valor gradualmente. Aunque este mecanismo aumenta el tiempo necesario para localizar una ruta.

9.5.5 Técnicas Adicionales para el Mantenimiento de Ruta

- Puede hacerse lo que se llama *salvar paquetes*: Supongamos la ruta ABCDE. El enlace DE se rompe y D envía a A *Route Error*. C lo retrasmitirá hacia A, pero si conoce una un camino alternativo, intentará usarlo para hacer llegar el paquete. Este camino alternativo puede reemplazar la ruta completa o sólo el segundo tramo. Cuando un paquete se está salvando, se activa un flag en su cabecera, para que no se intente salvar de nuevo provocando *salvamentos recursivos*.
- Las rutas también pueden acortarse automáticamente. Sea una ruta ABCD. Si C escucha a A enviando un paquete a B para que luego pase por C, puede evitarse el nodo B. Así que C envía un *gratuitous Route Reply*.
- Si un nodo recibe la información de que una ruta está rota, eso es de gran interés para los nodos vecinos. Así que se incluye dentro de la siguiente petición que se difunda.
- Hay otras posibles optimizaciones que el protocolo considera pero no implementa. Por ejemplo almacenar información negativa. Si un nodo tiene noticia de que se ha caído cierto enlace, puede rechazar cualquier ruta que pase por él. Pero sólo durante cierto tiempo, porque puede recuperarse.

9.5.6 Otras consideraciones sobre DSR

Alguno de los nodos puede tener más de un interfaz, típicamente el segundo tendría mayor alcance pero menor ancho de banda, y estar conectado a internet. En este caso puede ofrecer acceso a la red a todos los demás, puede ser un *Foreign Agent* de Mobile IP. Como premisa de partida se exigía que aunque el nodo tenga varios interfaces, usa siempre la misma IP: En caso de varios interfaces, se indexan con un número que fija el propio host. Hay números reservados para los Foreign Agent.

En cuanto a su ubicación en la torre de protocolos, los autores inicialmente lo colocaban en el nivel de enlace; por un lado por motivos históricos, ya que los trabajos uniciales partieron como mejoras a ARP. Pero fundamentalmente por permitir que por encima fuera cualquier otro protocolo de red. Además, si tenemos un sistema basado en IEE E802.11 o similar, tener las estaciones actuando como repetidores extiende de forma sencilla el alcance de un punto de acceso. Sin embargo, las implementaciones finalmente optan por colocarse en el nivel de red, por ser el único sitio donde de forma natural puede haber nodos con distintos interfaces y de distintos tipos.

9.6 Otros Protocolos Ad-Hoc

En las dos secciones anteriores hemos visto con cierto detalle dos protocolos: Un ejemplo de protocolo proactivo, otro reactivo. Pero hay más ejemplos relevantes, entre los que podemos citar:

- AODV. Ad Hoc on Demand distance-vector protocol [19]. Es otro protocolo reactivo, parte de DSDV pero lo hace trabajar bajo demanda. A diferencia de DSR donde el nodo origen marca la ruta, ésta se va estableciendo dinámicamente en los nodos intermedios.
- Basados en clusters. Tratan de organizar los nodos en clusters, de tal forma que algún nodo pueda representar el cluster completo. eligen Cluster Based

- ZRP. *Zone Routing Protocol* [12]. Solución intermedia entre DSDV y los protocolos bajo-demanda. Ofrece soluciones de compromiso entre ambos enfoques, y se puede parametrizar, de tal forma que en los extremos se comporta como alguno de los dos tipos expuestos.

9.7 Tendencias y Trabajo Futuro en Redes Ad-Hoc

La escalabilidad es tal vez el principal problema. Hoy con apenas 50 o 100 nodos los resultados empiezan a ser insatisfactorios. Incluso es difícil simular 10.000 nodos. También sigue sin resolver la calidad de servicio: Hasta ahora tratamos que los enlaces estén en pie o no, pero no hay expresividad para indicar cambios en la calidad.

El modelo cliente-servidor es el habitual en internet, pero no es adecuado aquí, ya no hay una entidad bien conocida que ofrezca servicios. Precisamente otra pregunta es *¿Dónde están los servicios?*, lo que resulta difícil en entornos tan cambiantes. Hay propuestas que apuntan a integrar el descubrimiento de servicios con el descubrimiento de rutas: Un nodo busca cierto servicio y hace para ello una petición broadcast en el nivel de red, con los inconvenientes que esto plantea: Para empezar, va contra el modelo en capas, cuya bondad está suficientemente probada. Además puede no ser posible pedir el servicio de forma que encaje en una petición de red o tomar decisiones sobre autorizaciones en este nivel.

Otra forma de localizar servicios puede ser basarse en direcciones multicast bien conocidas. Esto puede ser adecuado para servicios básicos como DNS o DHCP, pero en redes Ad-Hoc es un tema abierto. También lo son la seguridad, así como la gestión eficiente de energía.

10 Tendencias Futuras: Computación Ominipresente

Algunos autores [26] hablan de una clasificación donde el primer paso son los sistemas distribuidos, el segundo la computación móvil y el tercero la computación ominipresente (*Pervasive Computing*).

Es una tecnología que envuelva completamente al humano, coordinada entre sí, sin que el humano apenas la note. Se busca que se usen bien los espacios, que sea invisible, que escale pero teniendo en cuenta lo que está cerca (la mesa, la habitación), y que enmascare situación heterogéneas (distintos dispositivos, aplicaciones, fabricantes, estándares etc).

Referencias

- [1] ARORA, R. Recent advances in wireless data networking.
- [2] CAMPBELL, A. T., GOMEZ, J., AND VALKÓ, A. G. An overview of cellular IP. *IEEE Wireless Communications and Networks Conference 1999 (WCNC'99) 2* (1999), 606–611.
- [3] CROW, B. P. IEEE 802.11 wireless local area networks. *IEEE Communications Magazine* (september 1997).
- [4] DEERING, S. Internet protocol, version 6 (ipv6). <http://www.ietf.org/rfc/rfc2460.txt>, Dec. 1998.
- [5] FAKATSELIS, J. Upgrade to high rate physical layer by ieee802.11. In *International IC Taipei'99* (1999).
- [6] FLICKENGER, R. *Building Wireless Community Networks*. O'Reilly, 2002.
- [7] FORSBERG, D. Communication availability with mobile ip in wireless lans. Master's thesis, Helsinki University of Technology, 2000.
- [8] FORSBERG, D., MALINEN, J., , MALINEN, J., AND KARI, H. Increasing communication availability with signal-based mobile controlled handoffs. In *IP based Cellular Networks (IPCN2000)* (2000). 004.
- [9] FORSBERG, D., MALINEN, J., MALINEN, J., WECKSTROM, T., AND TIUSANEN, M. Distributing mobility agents hierarchically under frequent location updates, 1999. 003.
- [10] GONZÁLEZ-BARAHONA, J. M., AND DADDARA, C. Free software/open source: Information society opportunities for europe? <http://eu.conecta.it/paper/>, April 2000. 001.
- [11] HAARTSEN, J. C. The bluetooth radio system. *IEEE Personal Communications Magazine* (feb 2000), 28–36.
- [12] HAAS, Z. J. *Ad Hoc Networking*. Addison-Wesley, 2001, ch. ZRP A Hybrid Framework for Routing in Ad Hoc Networks.
- [13] HINDEN, R. M. IP Next Generation overview. *Communications of the ACM* 39, 6 (1996), 61–71.
- [14] HUITEMA, C. *IPv6. The New Internet Protocol. 2nd ed.* Prentice Hall, 1997.
- [15] IEEE STANDARDS ASSOCIATION. IEEE 802.11b.
- [16] KANELLAKIS, K. Enterasys on standard's confusion. http://www.80211-planet.com/tutorials/article/0,4000,10724_981611,00.html, Feb. 2002.
- [17] PERKINS, C. RFC 2002 IP Mobility Support. <http://www.ietf.org/rfc/rfc2002.txt>, Oct. 1996.
- [18] PERKINS, C., AND BHAGWAT, P. Highly dynamic destination-sequenced distance-vector routing (DS-DV) for mobile computers. In *ACM SIGCOMM'94 Conference on Communications Architectures, Protocols and Applications* (1994), pp. 234–244.
- [19] PERKINS, C. E. Ad-hoc on-demand distance vector routing, 1997.
- [20] PERKINS, C. E. *Mobile IP Design Principles and Practices*. Addison-Wesley, 1998. 002.
- [21] PERKINS, C. E. *Ad Hoc Networking*. Addison-Wesley, 2001.
- [22] PERKINS, C. E., AND JOHNSON, D. B. Mobility support in ipv6. In *Mobile Computing and Networking* (1996), pp. 27–37.
- [23] PERKINS, C. E., AND MYLES, A. Mobile IP. *Proceedings of International Telecommunications Symposium* (1994), 415–419. 005.
- [24] POSTEL, J. Internet Protocol. <http://www.ietf.org/rfc/rfc791.txt>, Sept. 1981.

- [25] REINBOLD, P., AND BONAVENTURE, O. A Comparison of IP Mobility Protocols, 2001.
- [26] SATYANARAYANAN, M. Pervasive computing: Vision and challenges.
- [27] SOLOMON, J. D. *Mobile IP. The Internet Unplugged*. Prentice Hall, 1998.
- [28] STEVENS, W. R. *TCP/IP Illustrated, Volume 1*. Addison-Wesley, 1994.
- [29] WILLIAMS, S. Irda: Past, present and future. *IEEE Personal Communications Magazine* (feb 2000), 11–19.