ANEXO A.

Manual de Usuario

A.1. Introducción

Este anexo trata de completar la parte que explica la aplicación web a través de una descripción de los pasos a seguir para realizar las diferentes acciones que implementa el sistema. Estas descripciones van acompañadas de las imágenes que el usuario vería, de manera que es posible observar el aspecto de la aplicación web.

A.2. Funcionalidades de la aplicación

A.2.1. **Acceder al sistema**

Una vez se accede a la aplicación web, se observa la pantalla de bienvenida. Allí, se introducen los datos de acceso al sistema, como se observa en la figura A-1.

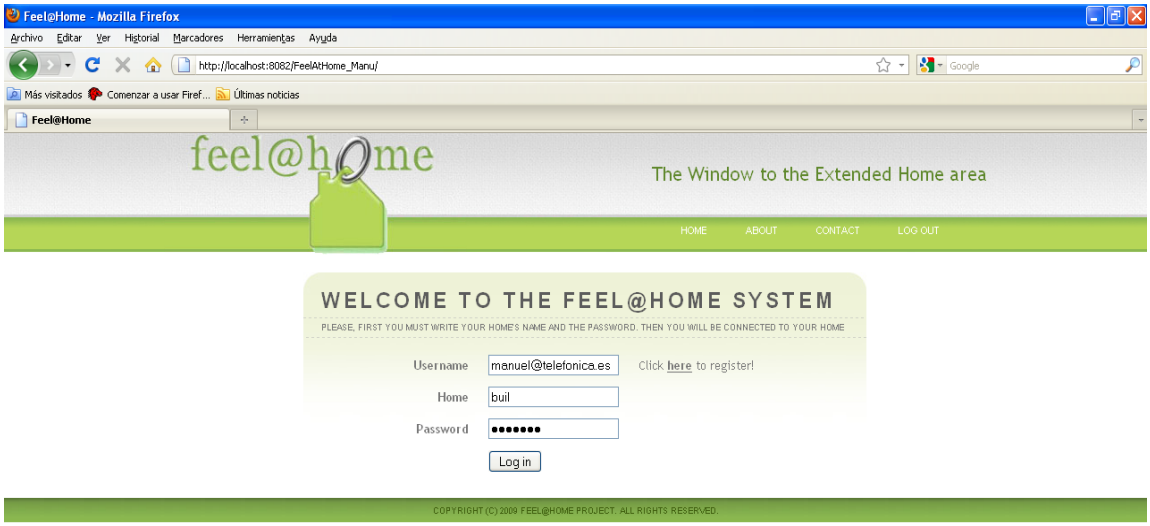


Figura A-1 : Pantalla de bienvenida

Si los datos son correctos, aparece un mensaje declarando que la autenticación es correcta y la aplicación muestra la pantalla de inicio de Feel@Home. A través de esta pantalla es posible acceder a todas las funcionalidades del sistema. En la figura A-2 se puede ver esta pantalla:

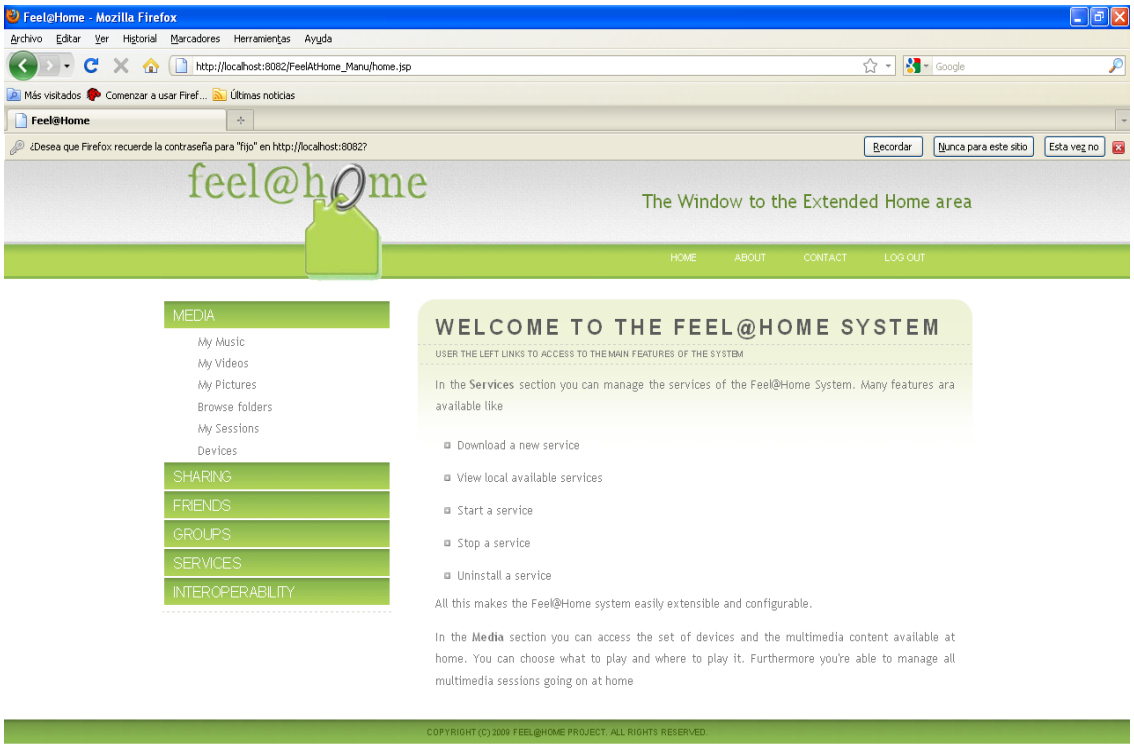


Figura A-2: Pantalla de inicio de Feel@Home

A.2.2. **Registro en el sistema**

En la figura A-1, se observa que aparece un link para registrarse a la derecha. Para registrarse en el sistema, simplemente hay que clicar en ese link existente para el registro, el cual lleva a una pantalla, figura A-3, dónde es posible registrarse.

Figura A-3 : Pantalla de registro en el sistema

A.2.3. **Búsqueda y reproducción de contenido**

A partir de la pantalla de bienvenida, es posible acceder a los contenidos descubiertos por el punto de control. Existen tres links para esto: My music, My image y My videos. La figura A-4 muestra la pantalla con los contenidos que aparecerían al clicar en uno de estos links:

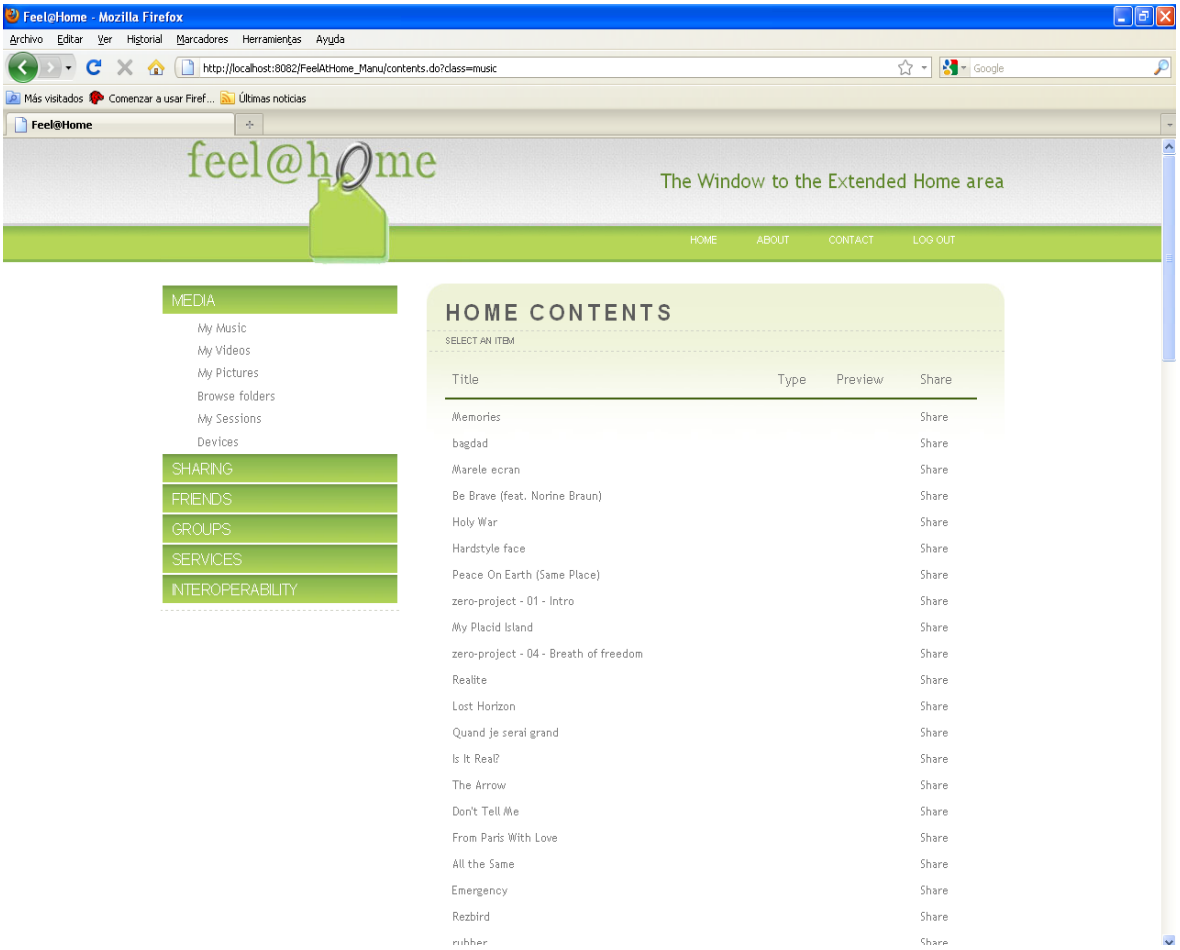


Figura A-4: Pantalla con los contenidos locales

Para reproducir el contenido, se clica en el nombre de éste. En ese momento, se pasa a la selección del dispositivo dónde se realizará la reproducción, como se observa en la figura A-5. En esta figura existen 2 dispositivos que han sido descubiertos por el punto de control como reproductores de contenido. Además, siempre existe la posibilidad de reproducir el contenido en el mismo dispositivo desde el cual se accede.

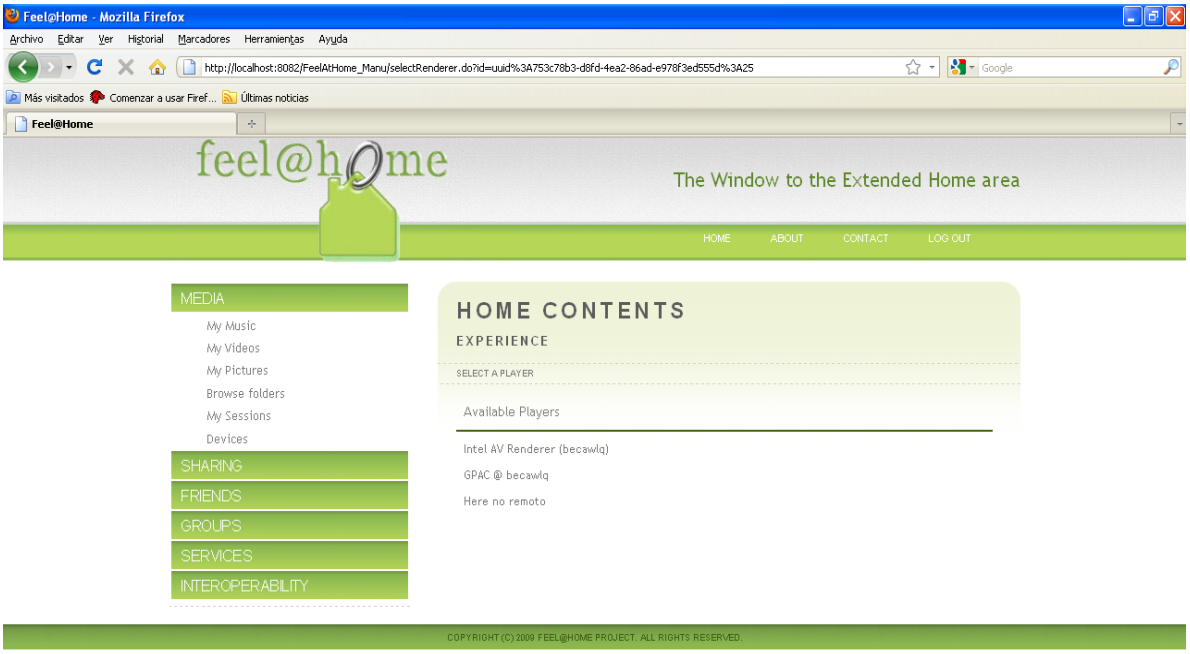


Figura A-5 : Selección de dispositivo para la reproducción

Para reproducir el contenido, es preciso clicar en el nombre de uno de los reproductores. Haciendo esto, aparece la pantalla de reproducción. En esta pantalla es posible reproducir, pausar o parar la reproducción. La figura A-6 ejemplifica esto:

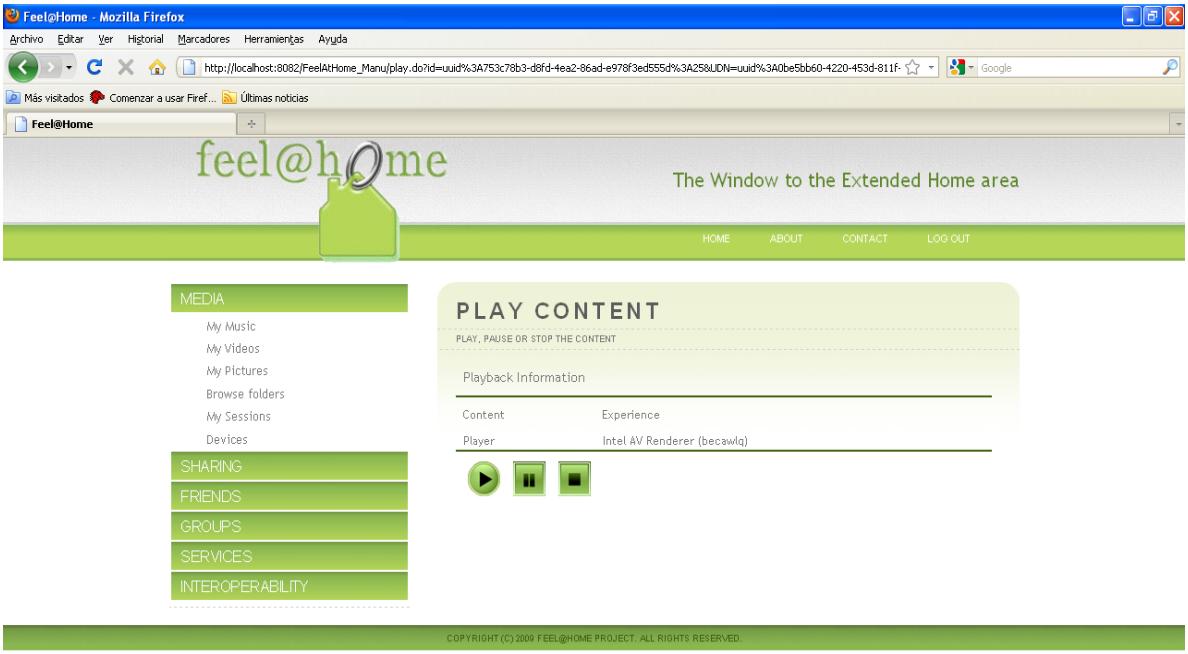


Figura A-6 : Pantalla de reproducción de contenido

A.2.4. **Modificación de las sesiones**

Cuando un contenido es reproducido, se crea una sesión en el punto de control. Estas sesiones pueden ser modificadas por el usuario a través de la aplicación web de manera que puede pausar, detener o continuar la reproducción. En la columna de la izquierda,

en la parte de MEDIA, aparece un link llamado My Sessions. Este link lleva a la pantalla representada en la figura A-7. En esta pantalla se observan dos sesiones, las cuales pueden ser pausadas, paradas o reproducidas, siempre que se hayan pausado previamente.

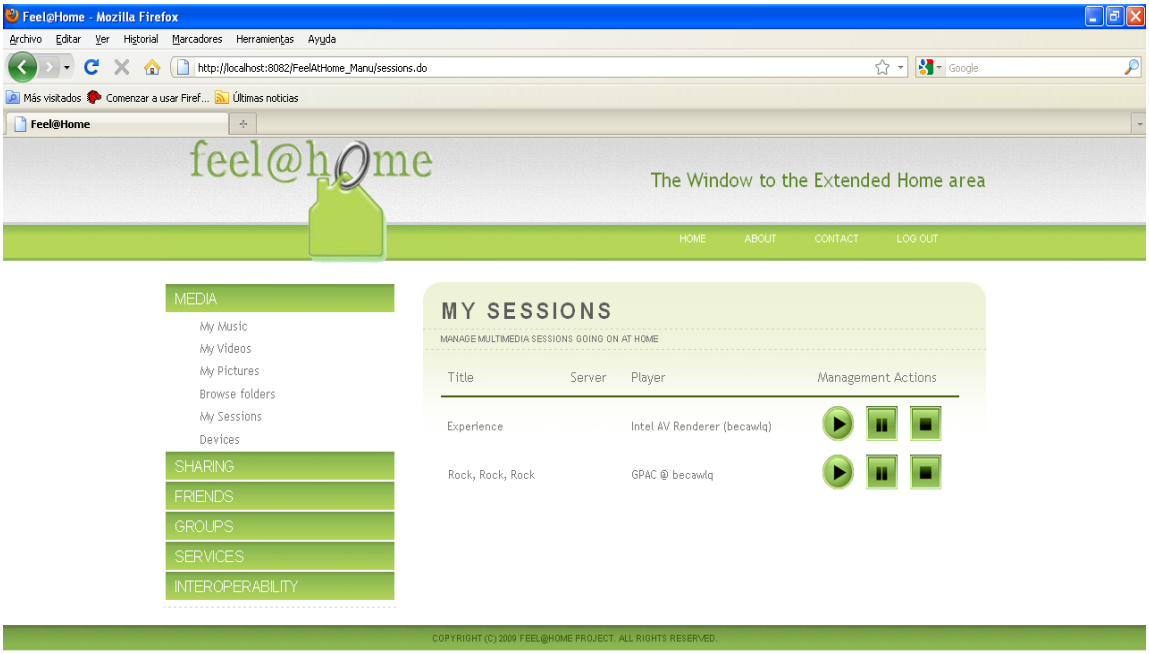


Figura A-7 : Pantalla con las sesiones

A.2.5. **Compartición de contenido**

Para compartir contenido se debe ir a la lista de contenidos ejemplificada en la figura A-4. En la parte derecha de la pantalla existe un link llamado Share para cada contenido listado. Si un contenido en concreto quiere ser compartido, se debe clicar al link Share correspondiente. Entonces, la aplicación Web preguntará si el contenido quiere ser compartido con un amigo o un grupo. Tras hacer esa elección, la aplicación web pasa a la pantalla de la figura A-8. Aquí se elige el amigo o grupo al cual se le va a compartir el contenido. Si se clica encima de uno de ellos, el contenido se compartirá y la aplicación mandará un mensaje diciendo que se ha compartido correctamente el contenido.

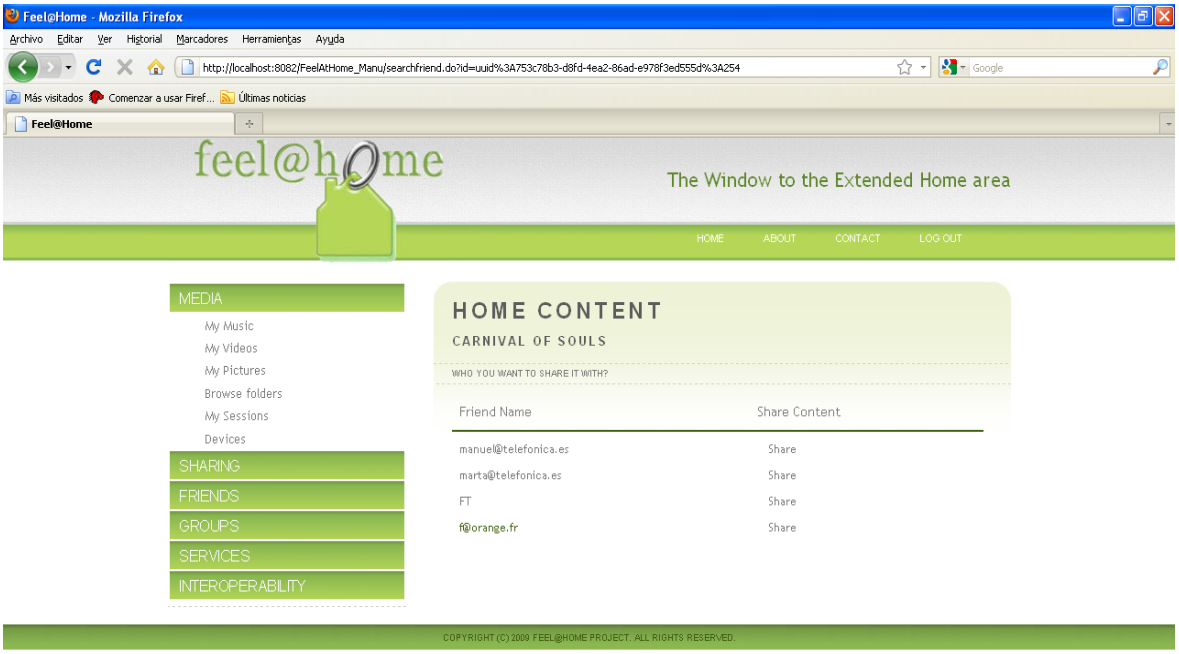


Figura A-8 : Pantalla con la lista de amigos o grupos del usuario

A.2.6. **Búsqueda de contenido por carpetas**

También es posible buscar contenido por carpetas e incluso compartir carpetas. Este método se basa en ir buscando los contenidos a través de las carpetas de los servidores existentes en el sistema. Para hacer esto, se debe ir a la barra de la izquierda y en la parte de MEDIA, clicar en Browse folders. De esta manera, la aplicación web listará los servidores existentes en el sistema. Tras la elección de uno de ellos, se podrá localizar los contenidos a través de las carpetas, tal y como se ve en la figura A-9. A la derecha de cada carpeta es posible observar un link share para compartir esa carpeta.

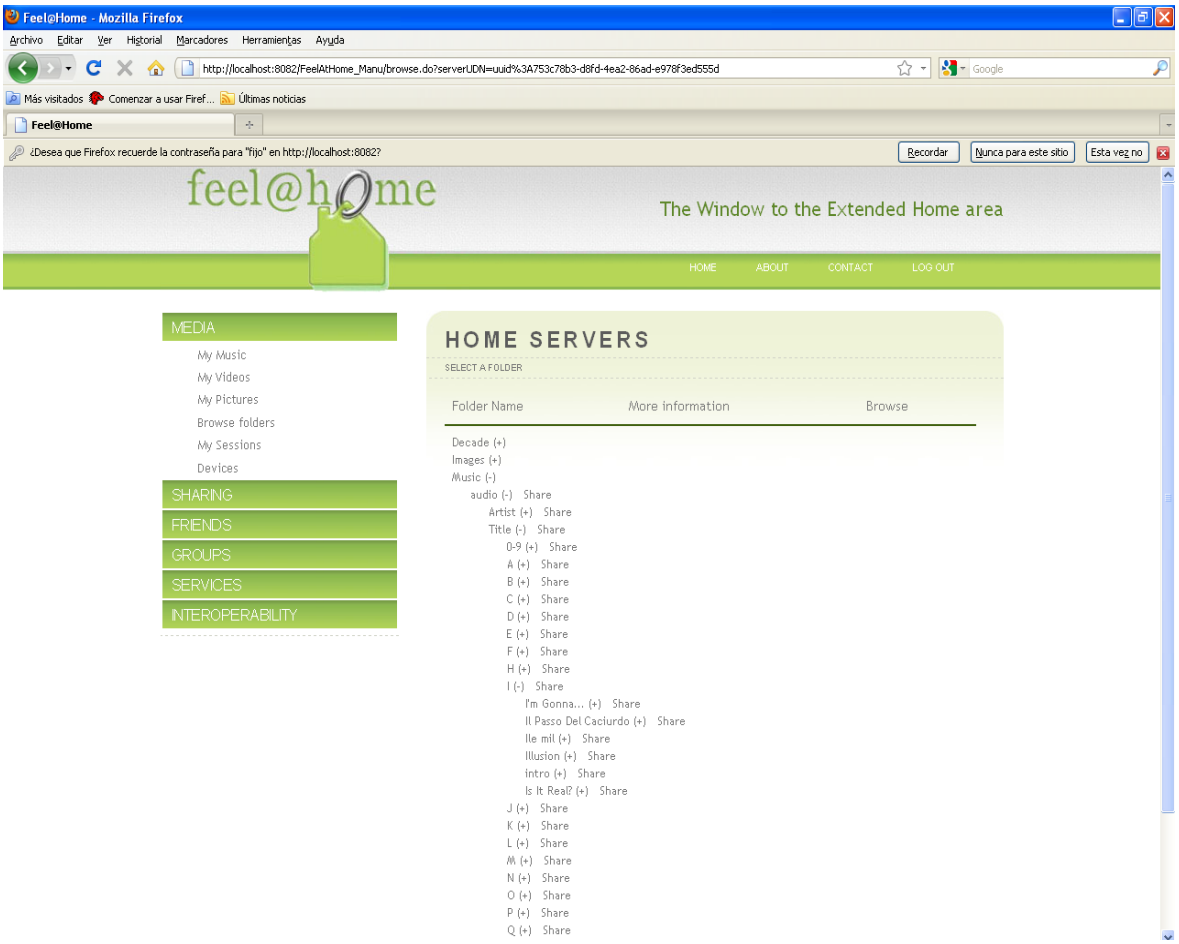


Figura A-9. Búsqueda de contenido por carpetas

A.2.7. **Comprobar y reproducir los contenidos que son compartidos con un usuario**

A la izquierda de la aplicación se observa una parte llamada SHARING. Esta etiqueta contiene dos acciones: ver el contenido que se ha compartido, dando la posibilidad de eliminar la compartición o ver quién esta compartiendo contenido contigo. Para ello, se debe clicar en el link “Users Sharing with me”. En ese momento aparecerá una pantalla con todos los usuarios y grupos que comparten algo con el usuario. La figura A-10 representa esto:

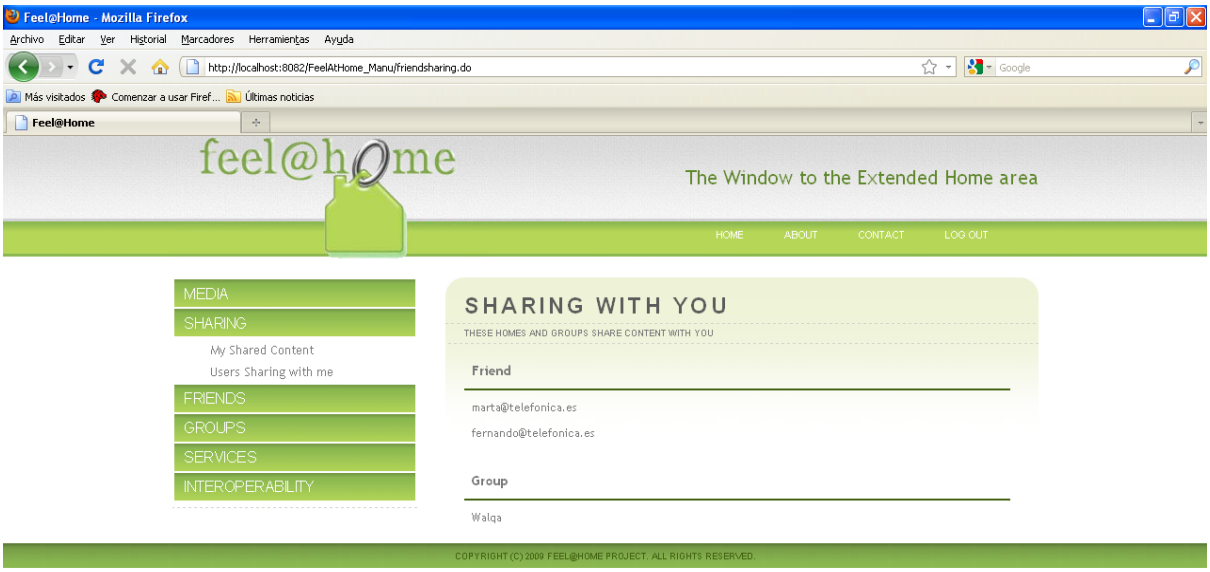


Figura A-10. Usuarios y grupos compartiendo algo con el usuario

Para ver qué contenido es compartido por uno de los usuarios o un grupo, simplemente clicar encima del usuario o grupo. En ese momento la aplicación web mostrará una lista con contenido parecida a la de la figura A-4.

A.2.8. **Añadir o eliminar un amigo**

La siguiente etiqueta que aparece en la barra a la izquierda es la llamada FRIENDS. Si clicamos en ella, aparecen 2 acciones disponibles: añadir amigo y ver los amigos. Tras clicar en añadir amigo, emerge a una pantalla dónde se da la posibilidad de añadir un amigo. Si, por el contrario, se quiere ver la lista de amigos que se tienen, aparecerá una pantalla como la de la figura A-11, en la cual se da la posibilidad de eliminar a cada uno de los amigos a través de un link.

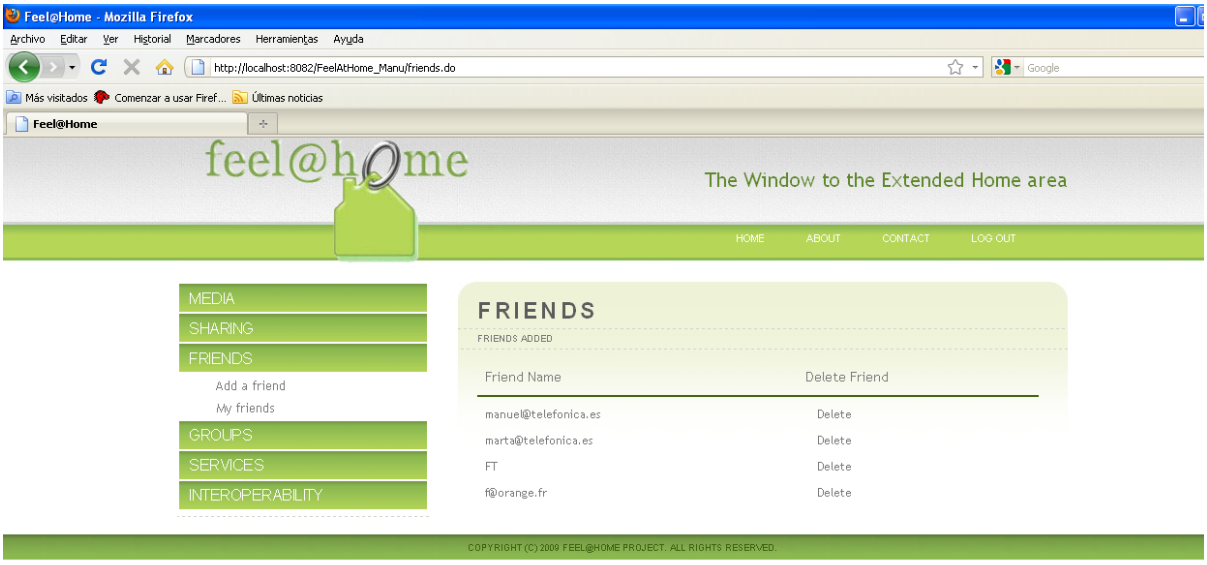


Figura A-11. Pantalla con los usuarios amigos

Debe añadirse que para poder compartir un contenido con un usuario, es necesario que este usuario pertenezca a la lista de amigos.

A.2.9. **Creación de grupos**

En la barra de la izquierda, se observa que existe una etiqueta llamada “GROUPS”. Esta etiqueta engloba todas las acciones relacionadas con los grupos. Para la creación de grupos, es necesario clicar en el link “Create a group”. Este link transporta al usuario a la pantalla A-12. En esta pantalla, el usuario debe introducir el nombre del grupo y una descripción para que los usuarios sepan el cometido de este grupo y puedan unirse a él.

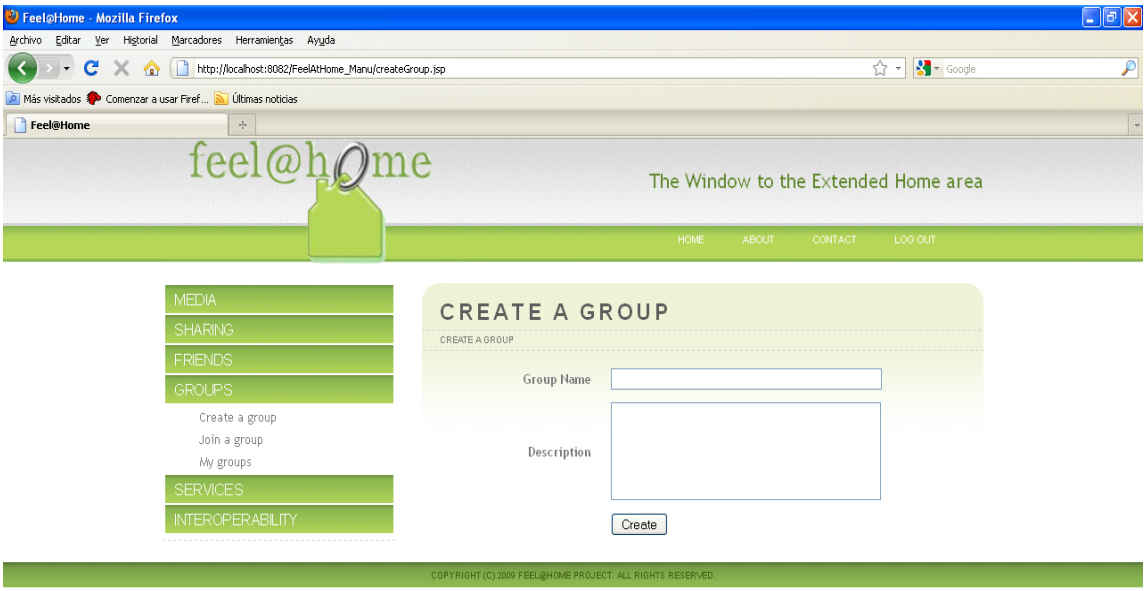


Figura A-12. Pantalla de la creación de grupos

A.2.10. **Unión a un grupo**

En la etiqueta “GROUPS” existen dos acciones más. “My Groups” lista los grupos en los que el usuario está, dando la posibilidad de dejar de pertenecer a ese grupo. “Join a Group” responde con una lista de los grupos que existen en el sistema y en los cuales el usuario no está incluido. Al lado de cada grupo, aparece un link “Join” que permite al usuario pertenecer al grupo. En la figura A-13, aparece un ejemplo de grupos que existen en el sistema.

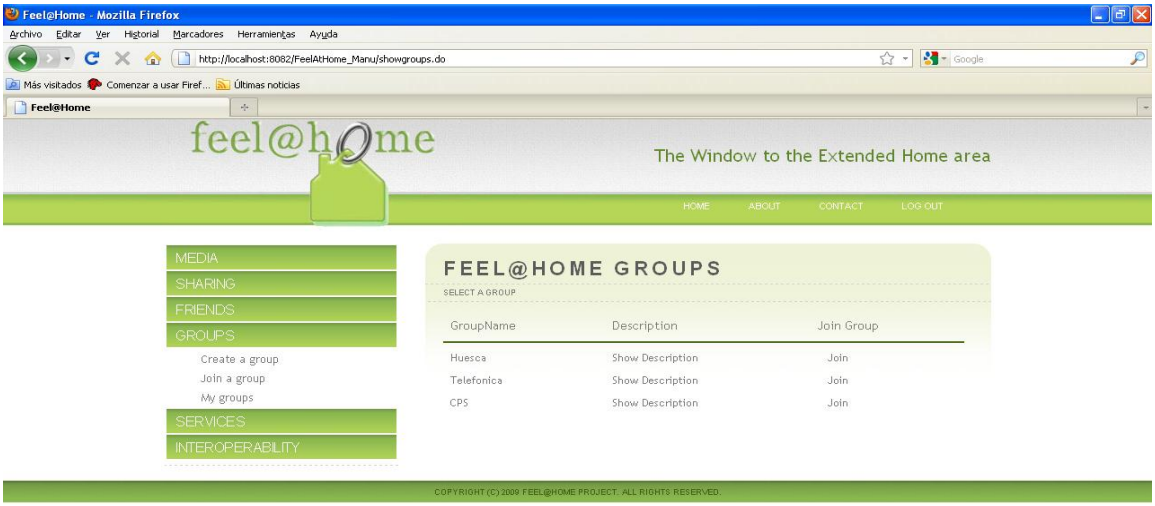


Figura A-13. Pantalla con los grupos que existen en el sistema

ANEXO B.

Configuración OpenVPN

B.1. Introducción

Este anexo trata sobre la configuración de OpenVPN. Este software permite la creación de túneles VPN, los cuales son usados para la comunicación entre los hogares y el servidor central en este proyecto.

Para describir la configuración, primero se van a mostrar los ficheros de configuración incluidos en la documentación de OpenVPN, para luego explicar, línea por línea, el significado. Como se ha apuntado previamente, OpenVPN se basa en un entorno servidor/cliente, razón por la que existen dos ficheros de configuración, uno para el servidor y otro para el cliente.

La autenticación y la seguridad serán explicadas en apartados específicos debido a la relevancia en este proyecto y complejidad que entrañan.

B.2. Configuración del servidor

Esta es la configuración más importante puesto que buena parte de la configuración de los clientes se basa en cómo se ha configurado el servidor. A continuación, se irán añadiendo y explicando cada una de las partes que tiene este fichero:

```
# Which local IP address should OpenVPN
# listen on?
local a.b.c.d
```

La primera línea del fichero de configuración pide la dirección IP en la cual el servidor OpenVPN recibirá las peticiones de los clientes para crear el túnel VPN. Es obvio que la dirección que se escriba debe pertenecer a una de las interfaces del servidor. Un ejemplo sería: local 192.168.1.32

```
# Which TCP/UDP port should OpenVPN listen on?
# If you want to run multiple OpenVPN instances
# on the same machine, use a different port
# number for each one.  You will need to
# open up this port on your firewall.
port 1194
```

Aquí se especifica el puerto que usará el software OpenVPN para crear los tuneles VPN. Es decir, el puerto por el cual viajarán las comunicaciones de la VPN. Si no es modificado, el puerto standard es el 1194 como se puede observar. Nunca tuvo que ser cambiado el puerto en este proyecto.

```
# TCP or UDP server?
```

```
;proto tcp
proto udp
```

OpenVPN soporta UDP y TCP para la comunicación de las VPN. Este parámetro puede ser importante ya que existen firewalls que no permiten uno de los dos protocolos por temas de seguridad. Esta línea permite al administrador elegir entre ambas.

```
# "dev tun" will create a routed IP tunnel,
# "dev tap" will create an ethernet tunnel.
# Use "dev tap0" if you are ethernet bridging
# and have precreated a tap0 virtual interface
# and bridged it with your ethernet interface.
# If you want to control access policies
# over the VPN, you must create firewall
# rules for the the TUN/TAP interface.
# On non-Windows systems, you can give
# an explicit unit number, such as tun0.
# On Windows, use "dev-node" for this.
# On most systems, the VPN will not function
# unless you partially or fully disable
# the firewall for the TUN/TAP interface.
;dev tap
dev tun
```

Esta opción debe elegirse depndiendo del funcionamiento que las VPN vayan a tener. La opción "dev tun" sólo permite crear túneles VPN punto a punto. Es decir, cada elemento de la red OpenVPN tendrá una dirección Ip virtual que usará para conectar con los otros dispositivos conectados a la red OpenVPN.

La opción "dev tap" permite hacer una interfaz Ethernet virtual, muy útil para hacer bridges. Esta opción se utiliza cuando quieren conectarse varias LAN entre sí de forma transparente para los usuarios de estas redes. Es decir, en vez de querer conectar dispositivos entre sí para crear una red virtual, se conectan redes privadas entre sí creando una red de redes virtual, en la que todos los elementos tienen una dirección virtual. Esta es la opción más adecuada cuando se necesitan conectar sedes alejadas geográficamente.

```
# Windows needs the TAP-Win32 adapter name
# from the Network Connections panel if you
# have more than one.  On XP SP2 or higher,
# you may need to selectively disable the
# Windows firewall for the TAP adapter.
# Non-Windows systems usually don't need this.
;dev-node MyTap
```

Cuando se ejecuta correctamente el software OpenVPN, una interfaz virtual es creada. A veces, es necesario conectar un PC a dos redes virtuales distintas cuando se usa la opción "dev tap". Esta configuración puede dar problemas en PCs con Windows por lo que OpenVPN añade está línea para solucionarlos.

```
# SSL/TLS root certificate (ca), certificate
# (cert), and private key (key).  Each client
```

```
# and the server must have their own cert and
# key file.  The server and all clients will
# use the same ca file.
#
# See the "easy-rsa" directory for a series
# of scripts for generating RSA certificates
# and private keys.  Remember to use
# a unique Common Name for the server
# and each of the client certificates.
#
# Any X509 key management system can be used.
# OpenVPN can also use a PKCS #12 formatted key file
# (see "pkcs12" directive in man page).
ca ca.crt
cert server.crt
key server.key  # This file should be kept secret
```

Los tres ficheros que se piden aquí son claves para la seguridad de las VPN. Estos ficheros son usados para la autenticación y el cifrado. En el apartado de seguridad se detallará su uso. En esta línea del fichero de configuración se pide la ruta para llegar a esos tres ficheros. Por ejemplo:

ca C:\OpenVPN\ca.crt

cert C:\OpenVPN\server.crt

key C:\OpenVPN\server.key

```
# Diffie hellman parameters.
# Generate your own with:
#   openssl dhparam -out dh1024.pem 1024
# Substitute 2048 for 1024 if you are using
# 2048 bit keys.
dh dh1024.pem
```

Esta línea también se utiliza para temas de seguridad. Concretamente describe los parámetros usados en el cifrado de las comunicaciones. Toda la información sobre este cifrado se encuentra en el apartado de seguridad. De nuevo en esta línea se requiere la ruta para llegar a él. Por ejemplo:

dh C:\OpenVPN\cifrado\dh1024.pem

```
# Configure server mode and supply a VPN subnet
# for OpenVPN to draw client addresses from.
# The server will take 10.8.0.1 for itself,
# the rest will be made available to clients.
# Each client will be able to reach the server
# on 10.8.0.1. Comment this line out if you are
# ethernet bridging. See the man page for more info.
server 10.8.0.0 255.255.255.0
```

En esta línea se define el rango de direcciones que tomarán los clientes que se conecten al servidor. Cuando un cliente pide crear un túnel VPN al servidor, éste

le proporciona una dirección. Si se opta por hacer un "dev tun", esta opción no tiene sentido ya que las direcciones son fijas para cada cliente.

```
# Maintain a record of client <-> virtual IP address
# associations in this file.  If OpenVPN goes down or
# is restarted, reconnecting clients can be assigned
# the same virtual IP address from the pool that was
# previously assigned.
ifconfig-pool-persist ipp.txt
```

Cuando se está usando "dev tun", es necesario fijar una dirección virtual para cada cliente. También puede ser útil hacer esto en diversas situaciones cuando se usa "dev tap". El fichero que se menciona en esta línea de la configuración es el que contiene esta información. Este fichero simplemente contendrá dos columnas, cliente y dirección. Para que el sistema relacione una dirección con un cliente habrá que colocar a ambos en la misma fila. Este proyecto no utilizó esta utilidad. El fichero de configuración requiere la ruta a ese fichero si esta opción va a ser usada. Por ejemplo:

Ifconfig-pool-persist C:\OpenVPN\Addresses\ipp.txt

```
# Configure server mode for ethernet bridging.
# You must first use your OS's bridging capability
# to bridge the TAP interface with the ethernet
# NIC interface.  Then you must manually set the
# IP/netmask on the bridge interface, here we
# assume 10.8.0.4/255.255.255.0.  Finally we
# must set aside an IP range in this subnet
# (start=10.8.0.50 end=10.8.0.100) to allocate
# to connecting clients.  Leave this line commented
# out unless you are ethernet bridging.
;server-bridge 10.8.0.4 255.255.255.0 10.8.0.50 10.8.0.100
```

Esta línea se refiere a la opción de hacer un bridge. En este caso se necesita especificar el cliente que hace el bridge entre la dirección virtual y la dirección física conectada a una red. Tras ello, debe detallarse el rango de direcciones que tomarán los elementos de esa red para convertirse en elementos de la red virtual. En el fichero de configuración se define un ejemplo en el cual el cliente que hace el bridge es 10.8.0.4 y el rango de direcciones es 10.8.0.50 hasta 10.8.0.100 con una máscara: 255.255.255.0. En este proyecto no se usaron bridges.

```
# Push routes to the client to allow it
# to reach other private subnets behind
# the server.  Remember that these
# private subnets will also need
# to know to route the OpenVPN client
# address pool (10.8.0.0/255.255.255.0)
# back to the OpenVPN server.
;push "route 192.168.10.0 255.255.255.0"
;push "route 192.168.20.0 255.255.255.0"
```

Existe otra forma de conectar distintas redes privadas entre sí con el objetivo de crear una red virtual de redes. Se trata de crear una red virtual en la que se incluye un elemento de cada una de las redes que se desea interconectar. Esta red sería implementada a través de la opción “dev tun”. Una vez la red funciona, cada uno de los elementos ofrece conexión a la red que hay detrás de él haciendo un broadcast con este ofrecimiento. Si observamos la figura B.1, al crearse la conexión OpenVPN con el servidor, el cliente 1 mandará un mensaje a todos los otros clientes y al servidor mediante el cual detalla que para conectarse a la red 192.168.0.0/24, deben enviar los paquetes a la 10.8.0.2. Para completar esta configuración, todos los PCs no clientes de OpenVPN deben actualizar sus tablas de rutas para poner al cliente OpenVPN que está en su red como puerta de salida para los paquetes cuya dirección sea una de las otras redes. Es decir, un PC de la red 192.168.1.0/24 deberá insertar en sus tablas: para acceder a las redes 192.168.0.0, 192.168.2.0 y 192.168.3.0 es necesario usar el PC con el cliente OpenVPN 10.8.0.3 como puerta de salida. Cuando la configuración este completa será posible conectar todas las redes sin que todos los elementos de éstas necesiten obtener una dirección virtual.

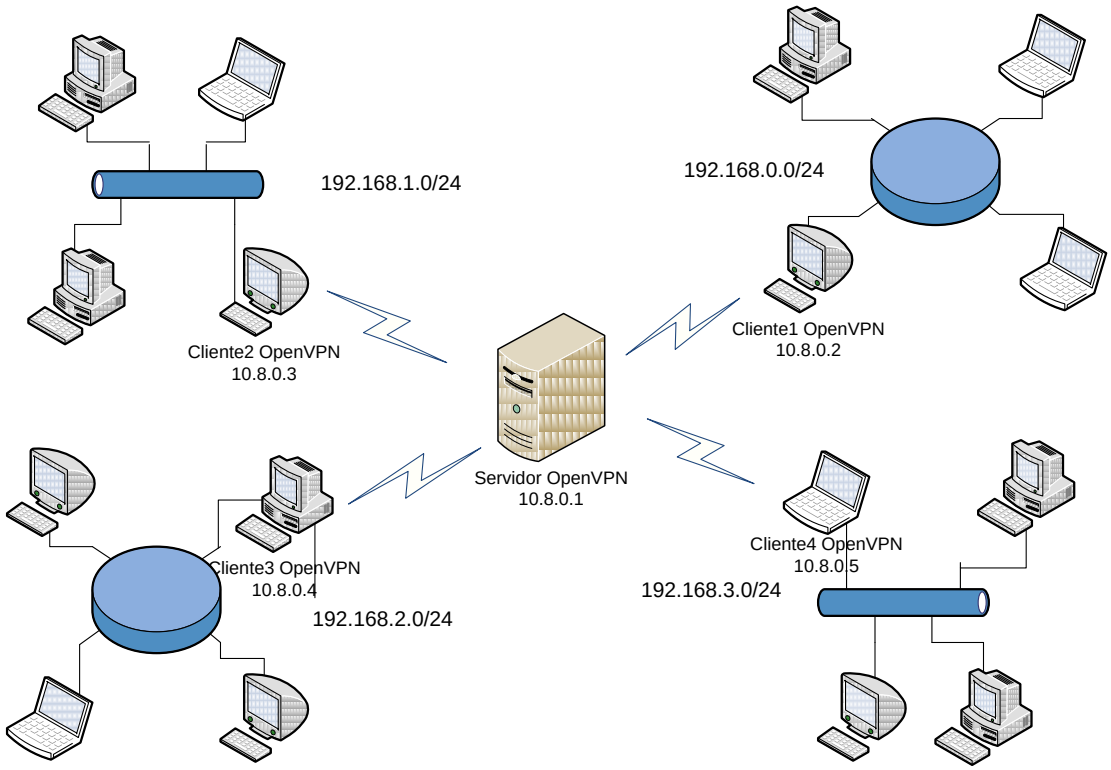


Figura B-1 : Redes interconectadas por OpenVPN

Esta línea del fichero de configuración especifica las redes que van a poder ser accedidas. En este ejemplo correspondería:

push "route 192.168.0.0 255.255.255.0"

push "route 192.168.1.0 255.255.255.0"

push "route 192.168.2.0 255.255.255.0"

push "route 192.168.3.0 255.255.255.0"

```
# To assign specific IP addresses to specific
# clients or if a connecting client has a private
# subnet behind it that should also have VPN access,
# use the subdirectory "ccd" for client-specific
# configuration files (see man page for more info).

# EXAMPLE: Suppose the client
# having the certificate common name "Thelonious"
# also has a small subnet behind his connecting
# machine, such as 192.168.40.128/255.255.255.248.
# First, uncomment out these lines:
;client-config-dir ccd
;route 192.168.40.128 255.255.255.248


# Then create a file ccd/Thelonious with this line:
#   iroute 192.168.40.128 255.255.255.248
# This will allow Thelonious' private subnet to
# access the VPN.  This example will only work
# if you are routing, not bridging, i.e. you are
# using "dev tun" and "server" directives.
```

Siguiendo con el ejemplo, para que la conexión entre redes sea correcta es necesario crear en el servidor una carpeta "ccd" donde se incluye un fichero por cada cliente. Este fichero especifica la red que tiene detrás cada cliente. En esta línea de la configuración se da la ruta hacia este fichero y de nuevo las redes a las que es posible acceder. Los ficheros deben llamarse como el cliente y añadir el comando "iroute *red_que_ofrece mascara_de_la_red_que_ofrece*". En este ejemplo:

client-config-dir C:\OpenVPN\ccd

route 192.168.0.0 255.255.255.0

route 192.168.1.0 255.255.255.0

route 192.168.2.0 255.255.255.0

route 192.168.3.0 255.255.255.0

Existirían 4 ficheros: cliente1, cliente2, cliente3, cliente4. El fichero cliente1 tendría escrito: iroute 192.168.0.0 255.255.255.0, etc.

```
# Suppose that you want to enable different
# firewall access policies for different groups
# of clients.  There are two methods:
# (1) Run multiple OpenVPN daemons, one for each
#     group, and firewall the TUN/TAP interface
#     for each group/daemon appropriately.
# (2) (Advanced) Create a script to dynamically
#     modify the firewall in response to access
```

```
#     from different clients.  See man
#     page for more info on learn-address script.
;learn-address ./script
```

Pueden surgir situaciones en las que no todos los clientes podrán compartir la misma configuración con respecto a los firewalls. Para discriminar un grupo de usuarios de forma que usen una configuración del firewall diferente a todos los otros usuarios existen dos vías: la primera es hacer que el servidor ejecute dos servidores OpenVPN cada uno con las configuraciones pertinentes. Esto es posible siempre y cuando ambos servidores no coincidan en el puerto y protocolo de comunicación (udp o tcp). La segunda vía es más compleja y supone crear un script que modifique dinámicamente la configuración dependiendo del cliente. Cuando se escoge la segunda vía, es necesario especificar la ruta dónde se encuentra este script. Esto se realiza en esta línea de la configuración. Por ejemplo:

learn-address C:\OpenVPN\script.bat

En este proyecto no se ha necesitado discriminar a un grupo de usuarios ya que todos compartían la misma configuración de firewalls.

```
# If enabled, this directive will configure
# all clients to redirect their default
# network gateway through the VPN, causing
# all IP traffic such as web browsing and
# and DNS lookups to go through the VPN
# (The OpenVPN server machine may need to NAT
# the TUN/TAP interface to the internet in
# order for this to work properly).
# CAVEAT: May break client's network config if
# client's local DHCP server packets get routed
# through the tunnel.  Solution: make sure
# client's local DHCP server is reachable via
# a more specific route than the default route
# of 0.0.0.0/0.0.0.0.
;push "redirect-gateway"
```

Esta línea, si está presente, obliga a todos los clientes OpenVPN a usar la interfaz virtual como puerta de salida para todas sus comunicaciones. Es decir, que todos los paquetes, independientemente del protocolo, puerto, etc, saldrán por esta interfaz. En este proyecto esta opción no fue activada.

```
# Certain Windows-specific network settings
# can be pushed to clients, such as DNS
# or WINS server addresses.  CAVEAT:
# http://openvpn.net/faq.html#dhcpcaveats
;push "dhcp-option DNS 10.8.0.1"
;push "dhcp-option WINS 10.8.0.1"
```

Existe la posibilidad de proveer a los clientes de la red virtual de ciertos servicios de Windows como DNS o WINS. Para ello, tal y como se puede ver en la línea de configuración, es necesario detallar la dirección virtual dónde se suministran estos servicios.

```
# Uncomment this directive to allow different
# clients to be able to "see" each other.
# By default, clients will only see the server.
# To force clients to only see the server, you
# will also need to appropriately firewall the
# server's TUN/TAP interface.
;client-to-client
```

OpenVPN permite al administrador decidir si los clientes pueden comunicarse entre sí sin la necesidad de pasar por el servidor central. Si no lo permitiera, el cliente sólo podría comunicarse con el servidor. Esta línea activa la comunicación entre clientes, la cual fue usada en el proyecto puesto que la reproducción de contenido se hace entre clientes sin pasar por el servidor central.

```
# Uncomment this directive if multiple clients
# might connect with the same certificate/key
# files or common names.  This is recommended
# only for testing purposes.  For production use,
# each client should have its own certificate/key
# pair.
#
# IF YOU HAVE NOT GENERATED INDIVIDUAL
# CERTIFICATE/KEY PAIRS FOR EACH CLIENT,
# EACH HAVING ITS OWN UNIQUE "COMMON NAME",
# UNCOMMENT THIS LINE OUT.
;duplicate-cn
```

El fichero de configuración permite que varios clientes se autentiquen con el mismo certificado. Esta directiva no es recomendable por motivos de seguridad por lo que se recomienda no usarla. En este proyecto nunca se usó esta propiedad.

```
# The keepalive directive causes ping-like
# messages to be sent back and forth over
# the link so that each side knows when
# the other side has gone down.
# Ping every 10 seconds, assume that remote
# peer is down if no ping received during
# a 120 second time period.
keepalive 10 120
```

Esta línea permite accionar un mecanismo para que los clientes sepan rápidamente si uno de los clientes ha caído de la red. Para realizar esta operación, se envían periódicamente mensajes ping entre los clientes. Si la configuración client-to-client no está activada, los ping se enviarán entre servidor y cliente. El periodo de envío se configura en esta línea. En el ejemplo se observa que se envían un mensaje ping cada 10 segundos y se considera que un cliente ha caído si a los 120 segundos no ha contestado.

```
# For extra security beyond that provided
# by SSL/TLS, create an "HMAC firewall"
# to help block DoS attacks and UDP port flooding.
#
```

```
# Generate with:
#   openvpn --genkey --secret ta.key
#
# The server and each client must have
# a copy of this key.
# The second parameter should be '0'
# on the server and '1' on the clients.
;tls-auth ta.key 0 # This file is secret
```

Tal y como se verá en el apartado de seguridad, OpenVPN permite mejorar la seguridad del sistema a través de "HMAC firewall". Para usar esta mejora, se debe incluir en el fichero de configuración la ruta al archivo ta.key. Por ejemplo, en este proyecto:

tls-auth C:\OpenVPN\ta.key

```
# Select a cryptographic cipher.
# This config item must be copied to
# the client config file as well.
;cipher BF-CBC        # Blowfish (default)
;cipher AES-128-CBC   # AES
;cipher DES-EDE3-CBC  # Triple-DES
```

OpenVPN permite usar tres tipos de métodos criptográficos: Blowfish, AES y triple-DES. Para que el sistema utilice alguno de estos debe escribir cipher y el método. En el apartado de seguridad se explicarán estos métodos. En el proyecto, se utilizó el método Blowfish.

```
# Enable compression on the VPN link.
# If you enable it here, you must also
# enable it in the client config file.
comp-lzo
```

Añadiendo esta línea se implementa la compresión de datos en la conexión VPN. Tal y como especifica en el fichero de configuración, debe ser añadida en el cliente y el servidor para que funcione. En este proyecto se utilizó la compresión.

```
# The maximum number of concurrently connected
# clients we want to allow.
;max-clients 100
```

También es posible declarar el número máximo de clientes que pueden unirse a la red virtual. Para ello, usar max-clients *número_máximo*.

```
# Output a short status file showing
# current connections, truncated
# and rewritten every minute.
status openvpn-status.log
```

Esta línea fuerza al sistema a crear un archivo log en el que se describen las conexiones que se están dando en el sistema.

```
# By default, log messages will go to the syslog (or
```

```
# on Windows, if running as a service, they will go to
# the "\Program Files\OpenVPN\log" directory).
# Use log or log-append to override this default.
# "log" will truncate the log file on OpenVPN startup,
# while "log-append" will append to it.  Use one
# or the other (but not both).
;log         openvpn.log
;log-append  openvpn.log
```

Puede ser útil tener un archivo log en el que se escriba la salida que tiene el sistema para localizar posibles errores. En estas líneas se especifica la ruta dónde debe crearse este archivo log.

```
# Set the appropriate level of log
# file verbosity.
#
# 0 is silent, except for fatal errors
# 4 is reasonable for general usage
# 5 and 6 can help to debug connection problems
# 9 is extremely verbose
verb 3
```

Esta línea especifica la verbosidad que debe tener el sistema al rellenar los archivos log o al sacar la información por pantalla.

B.3. Configuración del cliente

Ahora se describe la configuración que debe tener el cliente de OpenVPN.

```
# Specify that we are a client and that we
# will be pulling certain config file directives
# from the server.
client
```

Debe declararse en el fichero de configuración que estamos ante un cliente. Para ello escribir client.

```
# Use the same setting as you are using on
# the server.
# On most systems, the VPN will not function
# unless you partially or fully disable
# the firewall for the TUN/TAP interface.
;dev tap
dev tun
```

El cliente debe coincidir en la configuración especificada por el servidor respecto a “dev tap” o “dev tun”.

```
# Windows needs the TAP-Win32 adapter name
# from the Network Connections panel
```

```
# if you have more than one.  On XP SP2,
# you may need to disable the firewall
# for the TAP adapter.
;dev-node MyTap
```

Cuando se ejecuta correctamente el software OpenVPN, una interfaz virtual es creada. A veces, es necesario conectar un PC a dos redes distintas cuando se usa la opción "dev tap". Esta configuración puede dar problemas en PCs con Windows por lo que OpenVPN añade está línea para solucionarlos.

```
# Are we connecting to a TCP or
# UDP server?  Use the same setting as
# on the server.
;proto tcp
proto udp
```

Es necesario decidir si se usa el protocolo udp o tcp y coincidir con lo escrito en el fichero del servidor.

```
# The hostname/IP and port of the server.
# You can have multiple remote entries
# to load balance between the servers.
remote my-server-1 1194
;remote my-server-2 1194
```

El cliente debe especificar la dirección IP del servidor OpenVPN al cual debe conectarse. Para ello usa esta línea, en la cual da la dirección y el puerto que usa el servidor.

```
# Keep trying indefinitely to resolve the
# host name of the OpenVPN server.  Very useful
# on machines which are not permanently connected
# to the internet such as laptops.
resolv-retry infinite
```

Esta opción hace que el cliente intente durante toda la conexión reconectarse al servidor. Este mecanismo tiene su lógica en dispositivos que se desconectan y conectan frecuentemente ya que así se evita al administrador tener que estar alerta y reconectando frecuentemente el cliente al servidor.

```
# If you are connecting through an
# HTTP proxy to reach the actual OpenVPN
# server, put the proxy server/IP and
# port number here.  See the man page
# if your proxy server requires
# authentication.
;http-proxy-retry # retry on connection failures
;http-proxy [proxy server] [proxy port #]
```

OpenVPN puede utilizarse a través de un proxy, tal y como se puede observar en esta línea de configuración. En este proyecto no se utilizó esta propiedad.

```
# Wireless networks often produce a lot
```

```
# of duplicate packets.  Set this flag
# to silence duplicate packet warnings.
;mute-replay-warnings
```

Como se explica, en redes wireless puede ser interesante activar esta propiedad para así no enviar tantos paquetes viajando.

```
# SSL/TLS parms.
# See the server config file for more
# description.  It's best to use
# a separate .crt/.key file pair
# for each client.  A single ca
# file can be used for all clients.
ca ca.crt
cert client.crt
key client.key
```

Estos parámetros son muy importantes ya que son el mecanismo que tiene el cliente para autenticarse en el servidor OpenVPN. En esta línea de la configuración, el cliente debe indicar la ruta hacia los tres ficheros. Esta forma de autenticación se explica en el apartado de seguridad. Por ejemplo:

ca C:\OpenVPN\ca.crt

cert C:\OpenVPN\client.crt

key C:\OpenVPN\client.key

```
# Verify server certificate by checking
# that the certicate has the nsCertType
# field set to "server".  This is an
# important precaution to protect against
# a potential attack discussed here:
#  http://openvpn.net/howto.html#mitm
#
# To use this feature, you will need to generate
# your server certificates with the nsCertType
# field set to "server".  The build-key-server
# script in the easy-rsa folder will do this.
;ns-cert-type server
```

Existe un posible ataque al sistema que puede ser cubierto mediante esta opción. Este ataque supone que un servidor maligno puede hacerse pasar por servidor OpenVPN lo cual abriría una importante brecha en la seguridad. Para eliminar este riesgo es posible añadir un mecanismo para que el cliente compruebe la identidad del servidor. En el proyecto esta propiedad no se ha utilizado.

```
# If a tls-auth key is used on the server
# then every client must also have the key.
;tls-auth ta.key 1
```

En el servidor existía una propiedad para mejorar la seguridad llamada "**HMAC firewall". Si se activa, será necesario activarla en todos los clientes también.**

```
# Select a cryptographic cipher.
# If the cipher option is used on the server
# then you must also specify it here.
;cipher x
```

Aquí el cliente debe detallar el nombre del cifrado que va a ser usado. Este cifrado ya ha sido configurado en el servidor y obviamente debe coincidir (Blowfish, DES o triple-DES)

```
# Enable compression on the VPN link.
# Don't enable this unless it is also
# enabled in the server config file.
comp-lzo
```

Esta propiedad activa la compresión. Es importante destacar que la compresión debe ser activada en ambos lados: servidor y cliente.

```
# Set log file verbosity.
verb 3
```

Esta propiedad configura la verbosidad del sistema.

B.4. Autenticación y cifrado

Se definen dos modos de autenticación y cifrado: claves estáticas precompartidas y TLS/SSL más certificados para la autenticación. Además, en la última versión de OpenVPN se definen dos nuevas formas de autenticación las cuales pueden usarse junto a una de las anteriores para aumentar la seguridad o por si solas. Estas son: autenticación usuario/password y autenticación a través de una tarjeta electrónica o sistema biométrico.

B.4.1. **Claves estáticas precompartidas**

Este modo de comunicación utiliza claves que han sido compartidas a través de un canal seguro antes de crear el túnel. Estas claves son: la clave de encriptación/desencriptación y la clave HMAC.

Clave encriptación/desencriptación: Esta es la clave que se usa para encriptar el mensaje usando uno de los tres mecanismos de encriptación soportados por OpenVPN: Blowfish, AES y triple-DES. Para aumentar la seguridad, OpenVPN permite utilizar dos claves, una para encriptar y otra para desencriptar.

Para proveer al sistema de la confidencialidad necesaria, se utiliza un cifrado simétrico. Este tipo de cifrado usa la misma clave para cifrar y descifrar, tal y como se observa en la figura B.2

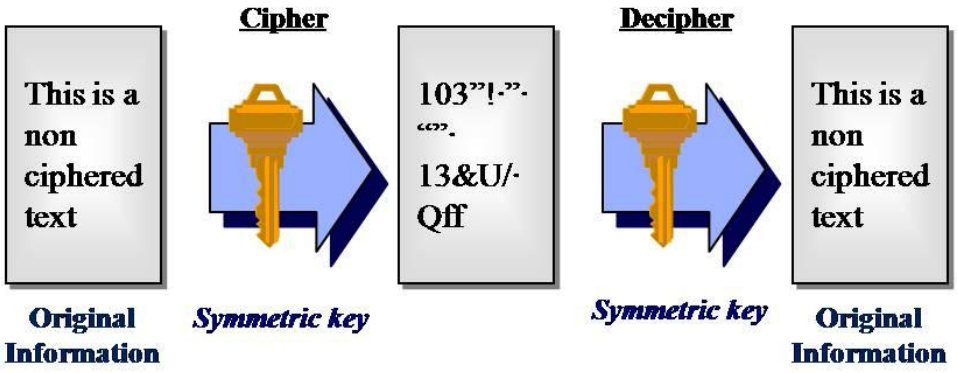


Figura B-2 – Esquema del cifrado simétrico

HMAC: OpenVPN utiliza HMAC para proteger la integridad y autenticidad del mensaje. La clave HMAC junto con el mensaje son los parámetros usados por la función de hash para crear el resultado o hash.

El algoritmo HMAC es una herramienta muy útil que garantiza la integridad del mensaje y comprueba que el emisor es quien dice ser. Esta herramienta implementa una función de hash que usa dos parámetros de entrada: el mensaje y una clave HMAC, la cual es precompartida en este modo. El parámetro de salida es una palabra resultado o hash, el cual es enviado por el emisor junto con el mensaje. El receptor repetirá la misma acción usando la misma clave y el mensaje recibido. Si el hash que obtiene es igual al que el emisor había enviado, el receptor puede estar seguro de que el mensaje no ha sido modificado y que el emisor es quien dice ser. En cambio, si el hash no fuera el mismo la conclusión que sacaría el receptor sería que o bien el mensaje ha sido modificado o bien el emisor no es quien dice ser, puesto que podría ser que no tuviera la clave HMAC, con lo cual el mensaje es descartado. La figura B.3 describe este proceso:

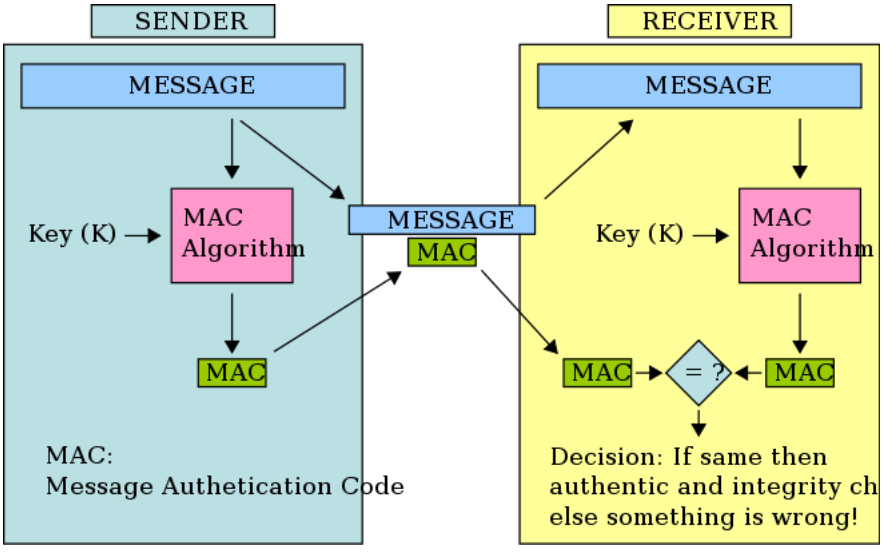


Figura B-3. Esquema de HMAC [15]

La autenticación del usuario en este modo se produce por la sencilla razón de que cada usuario obtiene unas claves distintas antes de comenzar la comunicación. El servidor

podrá saber quién es el usuario al tratar de desencriptar el primer mensaje. A partir de ese momento se relacionará a ese usuario con la Ip virtual que vaya a usar durante toda la sesión y a través de la cuál le llegarán los mensajes cifrados al servidor.

Este modo es bastante inseguro y no se recomienda su uso. El modo alternativo es mucho mejor y por eso fue el usado en este proyecto.

B.4.2. **Modo SSL/TLS**

Este es el modo usado en este proyecto. OpenVPN se basa en las librerías de OpenSSL [16] para implementar este protocolo, el cuál es usado ampliamente en internet para comunicaciones tan delicadas como la banca digital o las compras por internet.

Este protocolo basa la autenticación en PKI (Infraestructura de clave pública) y la confidencialidad de las comunicaciones en el cifrado simétrico. OpenVPN añade nuevas herramientas de seguridad que mejoran y complementan el protocolo SSL/TLS como tls-auth.

B.4.2.1 **Autenticación mediante PKI**

La infraestructura de clave pública basa todo su funcionamiento en claves públicas y privadas. Cada cliente tendrá una clave pública, conocida por todos los clientes del sistema, y una clave privada, las cuales estarán relacionadas matemáticamente, de manera que si un mensaje es cifrado usando una de ellas, sólo es posible descifrarlo a partir de la otra. Esta propiedad, unido a que la clave privada sólo la conoce el cliente y jamás es transmitida, garantiza un método de autenticación que es esquematizado en la figura B.4 y explicado a continuación.

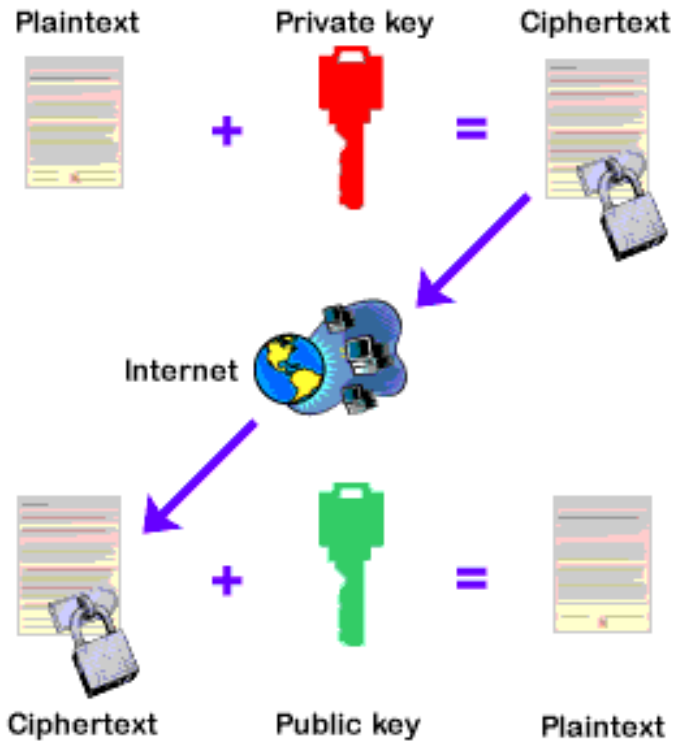


Figura B-4. Autenticación mediante PKI [17]

El usuario remitente cifra su mensaje usando la clave privada, la cual sólo tiene él. Una vez cifrado, el mensaje es enviado al destinatario, el cuál usa la clave pública del usuario remitente para descifrar el mensaje. Se observa que de esta forma la identidad del remitente queda completamente garantizada ya que la clave pública sólo podría descifrar correctamente si el remitente ha cifrado usando la clave privada. La autenticación usando PKI se hace permanentemente para cada mensaje enviado y no sólo al principio de la comunicación, lo cual incrementa la dificultad de romper la seguridad del sistema.

Uno de los problemas que pueden surgir al usar este método de claves públicas y privadas es que si existen muchos usuarios pueden surgir problemas de escalabilidad puesto que cada usuario necesita la clave pública de todos los demás usuarios. Para solucionarlo, OpenVPN utiliza una autoridad para los certificados(CA). Esta autoridad es la encargada de crear certificados para los clientes usando sus dos claves, de nuevo, una privada y una pública. El funcionamiento se explica a continuación:

1. El cliente envía su clave pública cifrada con la clave privada al CA. Es decir, en este proceso el mensaje es la clave pública del cliente.

2. El CA descifra el mensaje usando la clave pública del cliente (así comprueba su identidad) y envía de vuelta el mensaje pero ahora cifrado con la clave privada del CA.

3. El cliente recibe su mensaje (su clave pública) de vuelta pero ahora “firmado” por el CA, lo cual se llama certificado. Lo descifra usando la clave pública del CA y comprueba que coincide con su clave pública.

A partir de este momento, cuando este cliente comience una comunicación con el servidor o con otro cliente, ambos se pedirán los certificados firmados por el CA para conocer la clave pública del otro cliente. El envío de estos certificados es seguro puesto que han sido cifrados usando la clave privada del CA. De este modo, cada cliente sólo tiene que guardar la clave pública del CA para comunicarse con todos los otros clientes.

OpenVPN, además de exigir la autenticación del usuario, también exige la autenticación del servidor. De hecho, en este modo TLS/SSL, el cliente debe requerir la autenticación del servidor también. Esta autenticación se basa en la clave pública y privada también.

B.4.2.2 **Cifrado simétrico**

El cifrado simétrico ya se ha explicado en el apartado B.4.1 y simplemente supone cifrar y descifrar usando la misma clave. El problema que se extrae es que esta clave debe ser conocida por emisor y receptor y quizá no exista un canal seguro para enviar esta clave de uno a otro. En este proyecto si hubo un canal seguro y por tanto se pudo poner en conocimiento de ambas entidades esa clave. Sin embargo, para casos donde no exista un canal seguro, OpenVPN implementa el protocolo Diffie-Hellman [18] que permite a dos entidades que no se conocían previamente intercambiar una clave secreta a través de un canal inseguro.

Una vez ambas entidades tienen acceso a la clave secreta, se ejecuta el cifrado. Tal como se ha visto en los ficheros de configuración de OpenVPN existen tres algoritmos de cifrado soportados por OpenVPN: AES [19], triple-DES [20] y Blowfish [21]. En

este proyecto se utilizó indistintamente AES o Blowfish ya que triple-DES es demasiado pesado computacionalmente en comparación por ejemplo con BlowFish. Para conocer más sobre estos protocolos utilizar la bibliografía que se incorpora a este proyecto.

B.4.2.3 **Función hash**

Se ha comprobado ya que la autenticación y la confidencialidad de las comunicaciones están cubiertas. No obstante, existe un peligro que debe ser eliminado y es el que un usuario maligno pueda modificar el mensaje y lo que le llegue al receptor no sea lo que ha enviado el emisor, es decir, es necesario garantizar la integridad del mensaje. Para esto se utiliza el algoritmo HMAC el cual ya ha sido explicado en el apartado B.4.1 y que aquí de nuevo se vuelve a usar.

Simplemente, por completar estas explicaciones, se debe añadir que existen dos maneras para crear la clave de encriptación y HMAC. La primera de ellas se crea usando un método de OpenSSL llamado RAND_bytes. La segunda de ellas usando la función TLS PRF para lo cual debe añadirse **--key-method 2 en el proceso de creación de estas claves.**

B.4.2.4 **Seguridad adicional**

OpenVPN implementa una mejora en la seguridad del sistema, la cual es opcional y se indica en el fichero de configuración. En concreto se trata de la línea **`tls-auth ta.key 1`**.

Esta mejora simplemente introduce una clave que se usa para implementar un algoritmo HMAC y poder garantizar la integridad de los mensajes que se envían justo al principio de la comunicación, es decir, cuando todavía no se han comenzado a cifrar las comunicaciones. Esta clave debe de ser transmitida por un canal seguro ya que ambos, receptor y emisor, deben tenerla.

Existe otra mejora del sistema de seguridad que también es opcional y es la de la autenticación de los clientes en el servidor a través de usuario y password. Esta autenticación complementa a la anterior y provee un nuevo nivel de seguridad, ahora en la capa de aplicación. Aunque servidor y cliente se hayan autenticado mutuamente y la comunicación sea segura, no hay nada que garantice que el usuario es verdaderamente el que debería estar usando esa VPN. Para cubrir este problema, el servidor es capaz de guardar el usuario y la contraseña en un fichero de forma que justo antes de comenzar la comunicación, requerirá al usuario que introduzca estos datos para ser comprobados. Para utilizar esta herramienta, debe añadirse **auth-user-pass en los ficheros de configuración.**

OpenVPN también soporta otros métodos de autenticación vía tarjetas inteligentes o autenticación biométrica. Para ello estos sistemas deberán comunicarse con OpenVPN creando unos scripts que son explicados en la documentación de OpenVPN.

ANEXO C.

Estudio de tecnologías para la interconexión entre hogares.

C.1. Introducción

Tal como se explica en el apartado 3.3.2 del capítulo de diseño se ha elegido VPN como la tecnología más adecuada para la interconexión de los hogares. Sin embargo, existen otras tecnologías que podrían también realizar esta conexión pero que no han sido elegidas por distintas razones. En este anexo se explican las tres tecnologías que se han tenido en cuenta para la interconexión de hogares, detallando especialmente los motivos que han llevado a elegir VPN y a desechar las otras alternativas. Estas tres tecnologías son IMS, P2P y VPN.

C.2. IMS

IMS define una arquitectura funcional que permite la convergencia de video, audio, datos y toda la interconexión móvil en una infraestructura IP, puesto que se enmarca dentro de las nuevas tecnologías de red NGN fundamentadas en el *Todo-IP* [22]. Esta tecnología trata de cubrir el hueco que existe entre las comunicaciones móviles e internet de forma que a través del móvil sea posible consumir los distintos servicios que existen en la red.

IMS fue desarrollada originalmente por el 3GPP en el trabajo de estandarización de los sistemas 3G en UMTS. El principal objetivo marcado por este organismo es la integración de todos los servicios de la red de manera que facilite su acceso a los usuarios y su despliegue a los desarrolladores. Según el standard, la arquitectura de IMS define principalmente dos capas: capa de control y capa de servicio, ya que las capas inferiores coinciden con las de la tecnología UMTS. En la figura C.1 puede observarse esta arquitectura.

La capa de control incluye dos elementos: Call Session Control Function (CSCF) y Home Subscriber Server (HSS). El primero es el elemento central dentro de IMS para el funcionamiento de esta capa horizontal. Este elemento, utiliza el protocolo SIP para la comunicación con la capa de transporte a través de la cual llegan las peticiones de los usuarios y con la capa de servicio, en la que residen los diferentes servicios de la red. Este elemento procesa todos los mensajes que se envían entre usuario final y servicio, y lleva a cabo el registro de los servicios entre otras funciones. El HSS almacena en su base de datos todo tipo de información de los usuarios como por ejemplo la dirección ip, listas de amigos, etc.

La capa de servicios engloba todos los servicios ofrecidos en IMS. Está formada por servidores que alojan y ejecutan los servicios IMS a través de la capa de control. Ejemplos de servicios pueden ser la mensajería instantánea o la IPTV.

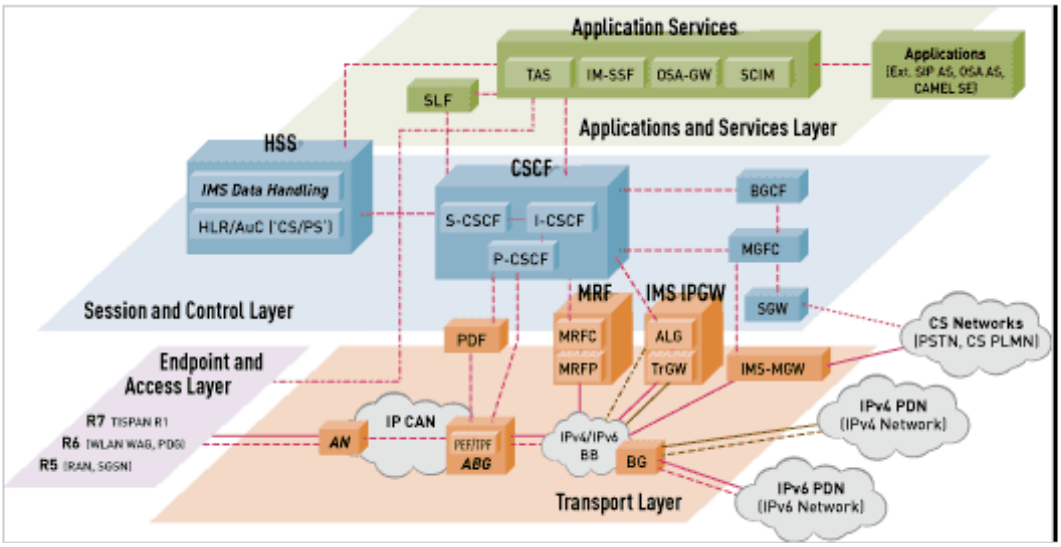


Figura C-1 Arquitectura de IMS [23]

Para conseguir el objetivo planteado por el 3GPP, IMS se basa en la capa de control, la cual es horizontal para todos los servicios. Este hecho presenta muchos beneficios entre los que destacan:

- Reducción de la complejidad, coste de los servicios y del tiempo que necesita un servicio desde su creación hasta su "venta". Al incluir una capa común para todos los servicios, es muy fácil la reutilización de componentes de diferentes servicios o usar servicios de otros desarrolladores. Además, ya que todos comparten una misma interfaz de control, la búsqueda de nuevos servicios por parte del usuario o la puesta en marcha de un servicio por parte del desarrollador es mucho más rápida y eficaz.

- Aporta valores añadidos a los diferentes servicios haciendo uso de características comunes como la señalización o la QoS. Por ejemplo, permite que mediante un solo log-in, el usuario sea autenticado en todas las aplicaciones y servicios a los que va acceder.

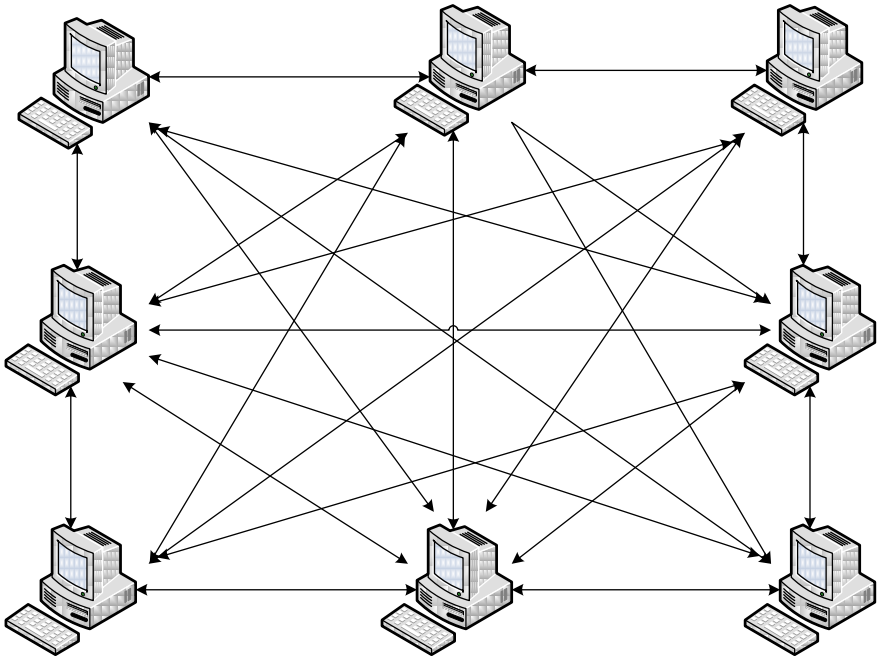
- Permite al operador tarificar más fácilmente y basándose en diferentes métodos: cantidad de bytes consumidos, cantidad de servicios consumidos pudiendo diferenciar entre tipos de datos consumidos: video, audio…

- Independientemente de la localización geográfica del usuario o del dispositivo de acceso, IMS permite la conexión entre ambos. Al basar sus comunicaciones en internet, se añade ubicuidad y el problema de acceder a los servicios en caso de *roaming* desaparecen.

Como puede observarse, esta tecnología permitiría que los dispositivos dentro de la red Feel@Home pudieran descubrir todos los servicios que existen de manera global, además de ser capaz, cada elemento, de ofrecer un nuevo servicio que podría ser demandado por otro elemento del sistema. Sin embargo, tiene una desventaja muy importante que ha sido fundamental para desechar su uso: la necesidad de muchos recursos para su funcionamiento. IMS es una tecnología relativamente nueva y muy potente que requiere muchos recursos hardware y software. Estos recursos confieren a esta tecnología de muchas funcionalidades que son inservibles para el proyecto Feel@Home y en consecuencia para este proyecto. Por eso se decidió escoger una solución mucho más sencilla que no necesitara de tantos recursos. De todos modos, como la solución de IMS es viable tecnológicamente hablando, en este proyecto se estudia e implementa una interoperabilidad entre VPN e IMS.

C.3. P2P

Peer-to-peer define una arquitectura de red distribuida basada en nodos interconectados que se organizan por si solos en distintas topologías de red. El propósito de esta tecnología es la compartición de recursos como datos, capacidad de procesamiento o ancho de banda entre los distintos nodos sin la necesidad de una entidad central que lo coordine. Las redes P2P son un modo completamente distinto de ver la compartición de contenidos comparado con la filosofía cliente-servidor puesto que en este caso todos los nodos actúan como consumidores y proveedores. En la figura C.2 se observa un esquema de esta arquitectura de red.



FiguraC-2. Esquema de la arquitectura de red P2P

Las redes P2P pueden organizarse de distintas maneras y por eso existen diferentes divisiones. Una de ellas, divide a estas redes en dos tipos: estructuradas y no estructuradas.

Las estructuradas se refieren a redes que implementan algoritmos que garantizan una comunicación entre nodos eficiente. Obviamente, estas redes son más complejas y necesitan algún tipo de elemento central que almacene una lista en la cual se relacionen todos recursos existentes con direcciones IP con el objetivo de calcular la mejor ruta.

Las no estructuradas suponen que no exista ningún tipo de algoritmo para interconectar nodos. Así, la red es mucho más sencilla y escalable aunque en este caso es menos eficiente. Para conseguir un recurso, el nodo deberá inundar toda la red con la petición hasta que algún nodo le responda. La posibilidad de no encontrar el elemento buscado aumenta considerablemente, además de cargar al sistema excesivamente.

Esta tecnología presenta dos grandes ventajas con respecto a otras como son:

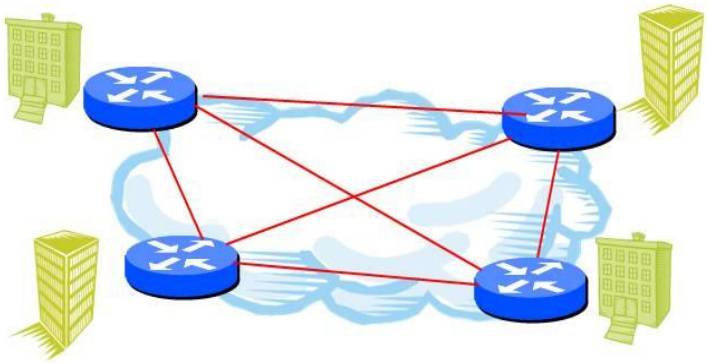
- Gran escalabilidad. Los usuarios comparten sus propios recursos y consumen recursos de otros usuarios por lo que cuantos más usuarios existan en el sistema, mejor funcionará. Además, el hecho de, en principio, no existir un elemento central descarta posibles cuellos de botella ante una gran avalancha de peticiones.

- Robustez. El hecho de que todos los nodos sean clientes y servidores permite que ante una caída de varios nodos, el sistema no se resienta. Esto muestra una clara ventaja con respecto a tecnologías que dependen de un elemento central, puesto que un mal funcionamiento de éste hace caer a todo el sistema.

P2P no fue elegida como la tecnología para interconectar hogares por ciertas desventajas. La primera es que el objetivo de los sistemas P2P difiere ostensiblemente del objetivo de este proyecto Feel@Home. Leyendo los requisitos de compartición se puede observar que en el sistema de compartición a desarrollar, un usuario va a compartir un contenido con otro usuario pero no, en principio, con el resto del sistema. Es decir, en este proyecto la compartición de contenidos va a ser privada entre uno o varios usuarios. Por lo tanto, la gran ventaja de P2P respecto a la compartición de recursos no tiene sentido en este contexto. Además, P2P adolece de ciertos problemas de seguridad ya que permite descargar contenido que podría ser peligroso enmascarado como un contenido inocuo, al no necesitar conocer al usuario que lo comparte [24].

C.4. VPN

Se trata de la solución escogida para interconectar los elementos del sistema. Las VPN (Virtual Private Network) permiten simular redes privadas que funcionan sobre redes públicas, como Internet, de una manera transparente para el usuario final. La palabra clave de las VPN es privado, puesto que aunque las comunicaciones viajan a través de la red pública, la confidencialidad es máxima al usar técnicas de cifrado de datos. Actualmente, las VPN son usadas ampliamente aunque su mayor uso se concentra en la interconexión de redes privadas alejadas geográficamente, como se observa en la figura C.3.



FiguraC-3. Distintas sedes interconectadas por VPN

De esta manera, se consigue el objetivo de que parezca que todos los elementos de las redes privadas están en la misma red, ahorrando muchos costes y sin arriesgar la privacidad y seguridad de las comunicaciones. Por este motivo esta solución es muy útil para empresas con sedes en distintos lugares del mundo.

Existen muchos tipos de VPN que se diferencian principalmente por su seguridad, arquitectura y por protocolos de enrutamiento usados. En este proyecto se busca una arquitectura centralizada, por lo que la VPN a elegir deberá soportar este tipo de arquitectura. Sobre seguridad y protocolos de enrutamiento usados, no existen preferencias, aunque siempre se va a intentar conseguir la mejor opción sin necesidad de muchos recursos.

Las ventajas de VPN que propiciaron su elección como tecnología son varias. Una de ellas es la posibilidad de conectar todos los elementos de forma que establezcan una red virtual, lo cual facilita mucho la interconexión entre los hogares. Es decir, se puede crear una red Feel@Home en la cual sólo estén incluidos los hogares que contratan este servicio independientemente del lugar donde se encuentren.

La seguridad que implementan las VPN es otro punto fuerte de esta tecnología. Aparte de la privacidad que garantiza a las comunicaciones gracias al cifrado, esta tecnología también aprovecha mecanismos de autenticación para incrementar la seguridad. Debe apreciarse la gran diferencia con respecto a P2P.

Asimismo, la tecnología VPN es bastante madura y usada, por lo que no será complicado encontrar información útil que ayude ante posibles dificultades en la implementación. Además, esta tecnología no necesita tantos recursos como IMS y por tanto es mucho más fácil de implementar. Las VPN son una solución barata y sencilla, ya que por ejemplo, todas las comunicaciones hacia el exterior del hogar usan el mismo puerto, simplificando enormemente el problema de firewalls y amenazas de seguridad producidas por puertos abiertos que podrían aparecer en los accesos remotos a los hogares.

De todos modos, VPN no es la tecnología perfecta ya que adolece de ciertas desventajas que deben ser tenidas en cuenta para evitar problemas. En este caso, el usar redes VPN en este proyecto conlleva 2 problemas graves: posibles cuellos de botella en el servidor y tener que depender de la red pública.

El primero de los problemas es debido a la arquitectura centralizada que se va a implementar. Este modelo exige una arquitectura de VPN también centralizada, lo que puede provocar cuellos de botellas ante escenarios donde existan muchas peticiones al servidor central. Consecuentemente, el problema deriva en una mala escalabilidad del sistema y en un riesgo alto de caída del sistema, ya que si el elemento central tuviera dificultades, todo el sistema estaría afectado. Estos riesgos deben ser minimizados utilizando por ejemplo redundancia en el servidor central e intentando que las peticiones sean lo más cortas y ligeras posibles.

El segundo inconveniente surge de la dependencia en una red pública. Las VPN crean redes virtuales pero en realidad viajan a través de redes públicas, tal y como se ha explicado. El hecho de usar la red pública permite reducir costes enormemente pero la dependencia con una infraestructura que no se controla y que se tiene que compartir con otros usuarios puede significar un riesgo. En primer lugar, cualquier problema que ocurra en esta red pública dejaría inservible nuestro sistema sin tener capacidad de recuperarlo, puesto que el problema sería externo. Además, se pone en riesgo la capacidad de ofrecer QoS en todo momento, ya que se depende de la carga que exista en la red pública, un dato que de nuevo es externo y por tanto imposible de controlar.