Implantación de movilidad IP en el Departamento de Ingeniería Telemática: soluciones disponibles e instalación de la solución seleccionada.

Juan-Mariano de Goyeneche.
<jmseyas@dit.upm.es>

29 de Mayo de 1999

Resumen

En este documento se describen someramente las distintas soluciones estudiadas para la implantación de movilidad IP en el Departamento de Ingeniería Telemática, así como las razones que llevaron a la elección de una de ellas frente a las demás. Finalmente, se detalla el proceso de instalación de ésta con el propósito de que sirva de guía rápida de configuración y uso.

Motivaciones

La generalización del uso de ordenadores portátiles hace que cada vez sea más común el querer conectarse a la red del Departamento desde los más diversos lugares. Esto trea consigo, sin embargo, nuevos problemas derivados de los firewalls y/o de la necesidad de poner en fucionamiento distintas configuraciones (DNS, pasarela de correo, proxy...) según cuál sea el proveedor desde el que nos conectamos en cada momento.

A la hora de enviar directamente correo puede constituir también un foco de problemas el que nuestra dirección IP, seguramente asignada dinámicamente, no pertenezca al dominio al que nuestro MTA cree pertenecer.

Lo ideal sería poder tener la misma dirección IP en cualquier momento y en cualquier lugar, y poder recibir y enviar paquetes con ella. Sin embargo, debido a la asignación de direcciones y al enrutamiento de IP [Pos81] resulta imposible (o por lo menos realmente complicado, a base de rutas específicas a host) recibir datagramas en una dirección IP que se encuentre fuera de su subred.

Para conseguir que un ordenador mantenga siempre su misma dirección IP independientemente del punto de la Internet al que se encuentra conectado en un momento dado, y que pueda enviar y recibir datagramas de forma transparente, la IETF [IET00] ideó en 1996 un protocolo que lo permitía en base a encapsulación de paquetes.

Este artículo representa el estudio realizado por el autor de las diversas soluciones de movilidad IP disponibles en este momento, explicando ventajas e inconvenientes de cada una de ellas. También pretende ser una guía para la documentación del proceso de instalación y configuración de la solución finalmente elegida.

Terminología

En este documento se utilizan los siguientes términos de acuerdo al significado que les otorga el RFC 2002 [Per96a]:

- *Mobile Node*: A host or router that changes its point of attachment from one network or subnetwork to another. A mobile node may change its location without changing its IP address; it may continue to communicate with other Internet nodes at any location using its (constant) IP address, assuming link-layer connectivity to a point of attachment is available.

- *Home Agent*: A router on a mobile node's home network which tunnels datagrams for delivery to the mobile node when it is away from home, and maintains current location information for the mobile node.

- *Foreign Agent*: A router on a mobile node's visited network which provides routing services to the mobile node while registered. The foreign agent detunnels and delivers datagrams to the mobile node that were tunneled by the mobile node's home agent. For datagrams sent by a mobile node, the foreign agent may serve as a default router for registered mobile nodes.

- SPI (*Security Parameter Index*): An index identifying a security context between a pair of nodes among the contexts available in the Mobility Security Association. SPI values 0 through 255 are reserved and MUST NOT be used in any Mobility Security Association.

Soluciones estudiadas

A la hora de implantar movilidad IP en la red del Departamento, se evaluaron diversas soluciones que cumplieran unos mínimos requisitos:

- Funcionamiento en sistemas operativos UNIX, preferentemente en GNU/Linux.
- Disponibilidad de código fuente, a fin tener asegurada la posibilidad de modificación y mantenimiento en caso de que fueran necesarios posteriormente a causa de que se abandonara el mantenimiento por parte de sus autores.
- Necesidad de soluciones que permitan el establecimiento de túneles de forma directa entre *Mobile Node* y *Home Agent*, pues no resulta razonable presuponer la existencia de *Foreign Agents* en todas las redes de acceso en que se pueda encontrar el *Mobile Node*.
- Preferencia de soluciones implementadas en espacio de usuario sobre aquellas que se basaran en parches aplicados al kernel (a menos que esos parches hubieran pasado a formar parte integral del kernel estandard). La razón de esto estriba el peligro de obsolescencia (al pasar a nuevas versiones del kernel el parche dejaría de funcionar con casi toda seguridad).
- Preferencia por soluciones que se distribuyan en forma de paquetes frente a las que lo hacen con un simple `.tgz` que hay que compilar e instalar manualmente. Lo deseable es que además se soportara el máximo número de formatos de paquete para que la solución sea fácilmente instalable en el mayor número de distribuciones.

La primera solución evaluada fue *Mobile-IP for Linux* [MIP96], desarrollada por el Departamento de *Computer Science* de la *State University of New York, Binghamton*. Esta implementación, seguramente la primera que estuvo disponible para GNU/Linux (data de principios del año 96) contaba con las ventajas de estar implementada íntegramente en espacio de usuario, necesitando tan sólo la selección en el kernel de soporte de encapsulación IP en IP [Per96b] (disponible ya desde los lejanos kernels 1.3) y de distribuirse bajo la licencia libre GPL [GPL00]. Sin embargo, pronto pudimos comprobar que este software había dejado de mantenerse y que ni siquiera compilaba en entornos modernos (kernels 2.2, GNU libc). Para corregir este problema, y gracias a la disponibilidad del código fuente y a la licencia libre que lo cubría, el autor realizó un parche que hizo posible la compilación. Sin embargo, la baja calidad del software que se observó en el proceso de creación de dicho parche y la falra de preparación del sistema para la generación de paquetes a partir de él empujaron a la búsqueda de otras soluciones.

Seguidamente se estudió y consideró la *Stanford MosquitoNet Project MobileIPv4 Distribution* [Mos99] de la Universidad de Stanford. Más moderna que la solución anterior, ésta tenía el inconveniente de estar implementada a nivel de kernel. De hecho, en el momento de nuestro estudio únicamente existían parches contra los kernels 2.0.33 y 2.0.36, cuando los kernels 2.2.x hacía ya muchos meses que habían visto la luz. Puestos en contacto con uno de los líderes del proyecto, Xinhua Zhao, le preguntamos por la existencia de *ports* de esos parches para kernels 2.2. Muy amablemente nos dijo que estaban trabajando en ellos y al poco nos envió las betas de la versión 2.0 con parches para los kernels 2.2.5, 2.2.12 y 2.2.14. Además de la esperable ventaja en términos de eficiencia, desprendida del hecho de estar hecha la implementación a nivel de kernel, había otra serie de factores que convertían esta solución en atractiva, como la posibilidad de utilizar tanto túneles triangulares como túneles inversos [Mon98], o los mecanismos para activar o no la movilidad en función del tráfico (típicamente no es necesario aplicarla, por ejemplo, al tráfico Web). Sin embargo, la cercanía de los kernels 2.4 y el retraso en los *ports* a los 2.2 nos hicieron temer la posibilidad de que no continuaran con el desarrollo de MosquitoNet o que éste se dilatara demasiado en el tiempo. Otros inconvenientes añadidos, como el hecho de que este software no contara con soporte para *Foreign Agents* o que no existieran paquetes ni estuviera incluido (ni hubiera visos de que lo fuera a estar pronto) en el kernel estandard, factores que habrían hecho más fácil y cómoda su instalación, nos hicieron buscar alguna otra solución más segura y duradera a largo plazo.

Y esta solución, que colmó todas nuestras espectativas, fue *Dynamics - HUT Mobile IP* [Dyn00], desarrollado por la *Helsinki University of Technology* (HUT). Se trata de una solución moderna (pensada para los kernels 2.2, de los que usa numerosas funcionalidades, todas ellas presentes en las versiones estandard), íntegramente desarrollada en espacio de usuario, sin necesidad de ningún parche especial, libre, distribuida con licencia GPL de GNU [GNU00, FSF00], preparada para generar paquetes tanto `.deb` como `.rpm`, y con gran versatilidad: túneles triangulares e inversos, encapsulación a través de *Foreign Agents* o directa entre *Mobile Node* y *Home Agent*, soporte de extensiones inalámbricas WLAN IEEE 802.11 [IEE00]...

Arquitectura y modos de operación

Dynamics ofrece una arquitectura dinámica y jerárquica de tuneles a través de la utilización eficiente de estructuras arborescenyes de *Foreign Agents*. Sin embargo, nosotros aquí contemplaremos tan sólo el caso de túneles inversos [Mon98] entre *Mobile Node* y *Home Agent*, ya que para los propósitos de mobilidad en cualquier situación, partiendo desde cualquier red de acceso, muchas veces públicas, no se puede asumir la existencia de *Foreign Agents* que faciliten la movilidad. Por tanto es fundamental que el equipo móvil sea capaz de realizar por sí mismo la encapsulación y desencapsulación de paquetes.

Instalación

Dynamics - HUT Mobile IP requiere un kernel linux 2.2.x o superior en el que se deben seleccionar las siguientes opciones:

- `Packet socket (CONFIG_PACKET)`
- `Kernel/User netlink socket (CONFIG_NETLINK)`
- `Routing messages (CONFIG_RTNETLINK)`
- `IP: tunneling (CONFIG_NET_IPIP)`

En caso de que *IP tunneling* se seleccione como módulo del kernel, será necesario activar también `Loadable module support`.

Adicionalmente, hay otras opciones que pueden resultar necesarias o simplemente convenientes dependiendo del rol que vaya a realizar el equipo. En los *Mobile Nodes* se recomienda activar la opción `IP: Socket Filtering (CONFIG_FILTER)` para obtener mejoras en prestaciones.

En los *Foreign Agents* son necesarias otras dos opciones de compilación:

- `IP: advanced router (CONFIG_IP_ADVANCED_ROUTER)`
- `IP: policy routing (CONFIG_IP_MULTIPLE_TABLES)`

En caso de usarse redes inalámbricas, para lo que también están preparados tanto Dynamics como el kernel Linux, debe seleccionarse en los *Mobile Nodes* `Wireless LAN (non-hamradio) (CONFIG_NET_RADIO)`.

Una vez compilado el kernel con las opciones pertinentes, la instalación de Dynamics se puede hacer bien a partir del código fuente (fichero `dynamics-0.6.tar.gz`, unos 378 Kbytes) o bien instando los paquetes para la distribución favorita (existen paquetes precompilados tanto `.deb` como, si se prefieren, `.rpm`).

Tanto el código fuente como los paquetes ya compilados se encuentran en la siguiente dirección: `http://www.cs.hut.fi/Research/Dynamics/software.html`

En el caso de que se opte por la instalación de los paquetes (solución recomendada), el nombre y número de paquetes que se necesitan instalar dependerá tanto de la distribución como del rol que vaya a interpretar el equipo que se instala:

- Distribución Debian GNU/Linux[Deb00]:
  - Rol de *Mobile Node*: `dynamics-common_0.6-1_i386.deb`, `dynamics-mn_0.6-1_i386.deb`
  - Rol de *Home Agent*: `dynamics-common_0.6-1_i386.deb`, `dynamics-ha_0.6-1_i386.deb`
  - Rol de *Foreign Agent*: `dynamics-common_0.6-1_i386.deb`, `dynamics-fa_0.6-1_i386.deb`
- Distribución RedHat GNU/Linux [Red00] y derivados:
  - Rol de *Mobile Node*: `dynamics-mn-0.6-1.i386.rpm`
  - Rol de *Home Agent*: `dynamics-ha-0.6-1.i386.rpm`

– Rol de *Foreign Agent*: `dynamics-fa-0.6-1.i386.rpm`

A continuación falta por configurar, editando los ficheros correspondientes, el *Mobile Node* y el *Home Agent*, dado que en esta primera etapa de implantación de movilidad IP en el Departamento no se contempla el uso de *Foreign Agents*.

Configuración del Home Agent

El fichero de configuración del *Home Agent*, supuesto que se haya realizado la instalación a partir de los paquetes correspondientes a cada distribución, se encuentra en `/etc/dynhad.conf`.

Al final de este documento se adjunta un `/etc/dynhad.conf` de ejemplo, preparado con una configuración basada en el uso directo de túneles entre *Mobile Node* y *Home Agent*, sin intervención de *Foreign Agent*. Cada entrada de ese fichero se ha comentado en detalle para facilitar su entendimiento y posible modificación.

El proceso de configuración que se llevará a cabo más a menudo será por tanto el de adición de nuevos *Mobile Nodes* que puedan hacer uso del *Home Agent*. Para ello, en la sección `AUTHORIZEDLIST` tan sólo hay que localizar un SPI o rango de SPIs sin usar y asignárselos a la IP o rango de IPs que se quiere que establezcan túneles con ese *Home Agent*. Los rangos de SPIs y de IPs se especifican de acurdo a como aparecen en las entradas del fichero de configuración de ejemplo del final de este artículo. Seguidamente, en la opción `SECURITY` habrá que añadir una entrada con los nuevos SPIs asignados y las palabras clave (secreto compartido entre *Home Agent* y *Mobile Node*) correspondientes.

Configuración del Mobile Node

El fichero de configuración del *Mobile Node*, supuesto que se haya realizado la instalación a partir de los paquetes correspondientes a cada distribución, se encuentra en `/etc/dynmnd.conf`.

Al final de este documento se adjunta un `/etc/dynmnd.conf` de ejemplo, preparado con una configuración basada en el uso directo de túneles entre *Mobile Node* y *Home Agent*, sin intervención de *Foreign Agent*. Cada entrada de ese fichero se ha comentado en detalle para facilitar su entendimiento y posible modificación.

Para utilizar túneles directamente entre *Mobile Node* y *Home Agent* habrá que asegurarse de que la opción `EnableFADecapsulation` esté a `FALSE`. Las opciones `SPI` y `SharedSecret` se deberán rellenar con los valores que haya asignado para cada *Mobile Node* el administrador del *Home Agent*.

Puesta en marcha

En el ordenador que va a actuar de *Home Agent* debe estar lanzado el demonio `dynhad` desde el arranque del sistema.

En el que se comportará como *Mobile Node*, el demonio `dynmnd` puede estar lanzado continuamente (con la opción `--disconnect` para que no se activen los servicios de movilidad mientras no sea necesario).

El proceso de establecimiento de los túneles lo inicia el *Mobile Node*, desatando la etaoa de autentificación. Para ello se debe ejecutar, bien de forma manual, bien de forma automática asociada a los *scripts* que se ejecutan tras la obtención de una dirección vía DHCP [Dro97] o PPP [Sim94] (por ejemplo, un fichero `/etc/ppp/ip-up.d/01dynamics` en la distribución Debian GNU/Linux), la siguiente orden:

```
dynmn_tool tunnel HA
```

Referencias

[Deb00] *Debian GNU/Linux distribution.* http://www.debian.org, 2000.

[Dro97] Ralph Droms. *RFC 2131: Dynamic Host Configuration Protocol*, March 1997.

[Dyn00] *Dynamics - HUT Mobile IP. Dynamic and Hierarchical IP Tunneling System.* http://www.cs.hut.fi/Research/Dynamics, 2000.

[FSF00] *Free Software Foundation.* http://www.fsf.org, 2000.

[GNU00] *GNU's Not Unix.* http://www.gnu.org, 2000.

[GPL00] *General Public License.* http://www.gnu.org/, 2000.

[IEE00] *Institute for Electrical and Electronic Engineers.* http://www.ieee.org, 2000.

[IET00] *Internet Engineering Task Force.* http://www.ietf.org, 2000.

[MIP96] *Mobile-IP for Linux.* http://anchor.cs.binghamton.edu/~mobileip/LJ/index.html, 1996.

[Mon98] Gabriel E. Montenegro. *RFC 2344: Reverse Tunneling for Mobile IP*, May 1998.

[Mos99] *Stanford MosquitoNet Project MobileIPv4 Distribution.* http://mosquitonet.stanford.edu/mip, 1999.

[Per96a] Charles Perkins. *RFC 2002: IP Mobility Support*, October 1996.

[Per96b] Charles Perkins. *RFC 2003: IP Encapsulation within IP*, October 1996.

[Pos81] Jon Postel. *RFC 791: Internet Protocol*, September 1981.

[Red00] *RedHat GNU/Linux distribution.* http://www.redhat.com, 2000.

[Sim94] William Allen Simpson. *RFC 1661: The Point-to-Point Protocol (PPP)*, July 1994.

Ejemplo de fichero de configuración del *Home Agent*: /etc/dynhad.conf

```
# $Id: dynhad.conf,v 1.29 1999/10/05 11:39:12 jm Exp $
# Home Agent configuration file
#
# Dynamic hierarchial IP tunnel
# Copyright (C) 1998-99, Dynamics group
#
# This program is free software; you can redistribute it and/or
# modify it under the terms of the GNU General Public License
# as published by the Free Software Foundation; either version 2
# of the License, or (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA  02111-1307, USA.
#
#########################################################################
#
# NOTE!
# This is an example configuration file designed to give
# perspective to the system configuration AND to provide
# a basis for a working simple test environment.
# The values of some of the parameters may not be the
# same as the daemon's defaults, so don't get confused.
#
#
#########################################################################
#
# Home Agent IP Address
HAIPAddress 138.4.4.142

# UDP port to listen for registration requests
# The default is 434
UDPPort 434

# Socket priority for signaling sockets (UDP) can be set with SO_PRIORITY to
# allow easier QoS configuration. If this argument is set, the given value is
# used as a priority for the signaling socket. E.g. CBQ class can be used to
# make sure that signaling is not disturbed by other traffic on a congested
# link.
# This feature is still undocumented and can be left commented.
#
# SocketPriority 1

# MaxBindings can be used to restrict the maximum number of Mobile Nodes
# that are concurrently attached to this Home Agent.
# The default is 20.
MaxBindings 20
```

```
# The default tunnel lifetime is suggested also by the HA.
# The default lifetime is 500.
HADefaultTunnelLifetime 600

# The Registration error reply interval should be restricted to
# avoid system overloading situations when receiving too much
# incorrect Registration Reply messages.
# The default value for RegErrorReplyInterval is 10 seconds.
RegErrorReplyInterval 10

# SendAdvertisements variable controls whether the HA should
# or should not send advertisements.
# SendAdvertisements < TRUE | FALSE >
SendAdvertisements FALSE

# The interval with which the home agent will send agent advertisement
# messages
AdvertisementInterval 30

# Triangle tunnel means that the packages to MNs are send via the HA, but
# packages from MN are routed directly (i.e. FA use normal IP routing).
# EnableTriangleTunneling < TRUE | FALSE >
EnableTriangleTunneling FALSE

# Reverse tunnel means bi-directional tunneling in which both the packages
# from and to MN are send via HA
# EnableReverseTunneling < TRUE | FALSE >
EnableReverseTunneling TRUE

########################################################################
# The Home Agent needs to know what kind of security parameters each
# authorized Mobile Node uses. that is why there is a tbale that maps
# (in many-to-many relationship) SPI numbers, or SPI-number ranges to
# IP adresses - or IP-address ranges defined by network adresses and
# netmasks. The netmask may be defined in two ways: either in
# "bit offset notation" (the third row in the example) or in the
# "dotted decimal notation" (the fifth row in the example below).
# The list of Mobile Node information is separated between two
# keywords: AUTHORIZEDLIST_BEGIN and AUTHORIZEDLIST_END.
#
# < SPI | SPI-range       IP | network/netmask  >
# Example:

AUTHORIZEDLIST_BEGIN
# SPI           IP
#1000            192.168.240.2
#1001            192.168.240.3
#1002            0.0.0.0/0
#11000-11999     192.168.241.4
#12000           192.168.250.0/255.255.255.0
#13000-14000     192.168.251.0/28
1000      138.4.4.144
AUTHORIZEDLIST_END

# The Home Agents needs a security association for each authorized Mobile
```

```
# Node. The association includes following information.
#
# SPI (Security Parameter Index): a key for the other fields.
#
# Authentication Algorithm: 1: keyed-MD5 (RFC 2002)
#                           (this is currently the only supported algorithm)
#
# Replay Protection Method:
#    0: none
#    1: timestamps
#    2: nonces
#
# Timestamp tolerance indicates how many seconds the MN's timestamp can differ
# from the HA's clock. 7 seconds is the recommended default value. This
# tolerance is checked only when timestamps are used for replay protection.
#
# The maximum lifetime for the binding is given in seconds.
# Special case: 65535 (or more) seconds means unlimited time (the binding will
# not expire)
#
# Shared Secret: a secret data known by MN and HA. It can be given as
# a HEX code string, i.e. two characters (0-F) correspond to one octet.
# The shared secret can also be given as a character string (e.g.
# "ABCDE" corresponds to 4142434445).
# Note: RFC 2002 specifies that the default key size is 128 bits (i.e.
# 16 bytes or 32 hex 'characters'). Dynamics supports also other key lengths.
#
# The SPI is the key identificator for the rest of the security parameters
# on the same line. SPI number ranges may be assigned the same security
# parameters.
#
# The list of Mobile Node information is separated between two
# keywords: SECURITY_BEGIN and SECURITY_END.
#
SECURITY_BEGIN
#       auth.   replay  timestamp       max             shared
# SPI   alg.    meth.   tolerance       lifetime        secret
1000  1      1      120      600      "test"
#1002    1       2       60              120             01020304050607
#10000   1       1       60              300             016A352B2F235E
#10001   1       1       120             180             0EF42BD234ECCAA2
SECURITY_END
#
########################################################################
# Home Agent may have optional security associations with Foreign
# Agents. If the security association exists the session key can be
# encrypted with the help of shared secret and thus man-in-the-middle
# style attacks can be prevented. If no security association is set
# for a certain Foreign Agent - Home Agent pair, public key encryption
# (RSA) is used.
#
# The following list contains the shared secrets indexed by SPI (and
# Foreign Agent IP address). The algorithm field specifies the method
# used for key distribution (currently only one is implemented,
# i.e. 1: encryption using MD5). The format of the share secret field is
# identical to the one used with the MN-HA security association list above.
```

```
#
FA_SECURITY_BEGIN
# SPI FA IP Alg. Shared Secret
#2001 192.168.0.1 1 0123456789ABCDEF
#2002 192.168.0.2 1 "eslkfj89jr3hduh3R!as"
FA_SECURITY_END
#
# Version 0.5 did not use authentication extensions between FAs and HA even if
# there was a configured security association. Version 0.6 adds the auth. ext.
# and requires that it is present in the request and reply if the security
# association is configured. To allow operations with the v0.5 agents, it is
# possible to disable this check in newer agents. The check should be enabled
# when there is no need to support the older agents.
# RequireAuthExt < TRUE | FALSE >
RequireAuthExt TRUE
#
# The Highest FA public key can be protected from man-in-the-middle style
# attacks between the HFA and the HA with hash code. The use of this hash
# is optional, but recommented. The HA can have different ways of checking
# the hash code.
# Methods:
#    0: skip the hash code completely (not recommented)
#    1: if the hash code is received, check the public key with it
#    2: require the correct hash code for every registration message
#       with a public key (this may prevent the use of some organizations
#       which do not advertise the hash code)
PublicKeyHashMethod 1
#
##########################################################################

# The log messages are written through syslog service. The facility to be
# used defaults to LOG_LOCAL0, but it can be set with this parameter
# to any of the possible facilities (LOG_AUTHPRIV, LOG_DAEMON, and so on).
# The processing of log messages is defined in /etc/syslog.conf file.
SyslogFacility LOG_DAEMON

# Home Agents (and Foreign Agents) use unix domain sockets
# to communicate through their API interfaces.
# The group and owner must be names as strings, no groupIDs or userIDs are
# allowed. The file permissions are set in octal values like in chmod(1).
# The configuration parameters of the two API sockets are as follows:
HAAPIReadSocketPath "/var/run/dynamics_ha_read"
HAAPIReadSocketGroup "root"
HAAPIReadSocketOwner "root"
HAAPIReadSocketPermissions 0766
#
HAAPIAdminSocketPath "/var/run/dynamics_ha_admin"
HAAPIAdminSocketGroup "root"
HAAPIAdminSocketOwner "root"
HAAPIAdminSocketPermissions 0700
#
# Every configuration file must end to the keyword 'END'.
END
```

Ejemplo de fichero de configuración del *Mobile Node*: /etc/dynmnd.conf

```
# $Id: dynmnd.conf,v 1.37 1999/10/13 13:20:16 jm Exp $
# Mobile Node configuration file
#
# Dynamic hierarchial IP tunnel
# Copyright (C) 1998-99, Dynamics group
#
# This program is free software; you can redistribute it and/or
# modify it under the terms of the GNU General Public License
# as published by the Free Software Foundation; either version 2
# of the License, or (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA  02111-1307, USA.
#
########################################################################
#
# NOTE!
# This is an example configuration file designed to give
# perspective to the system configuration AND to provide
# a basis for a working simple test environment.
# The values of some of the parameters may not be the
# same as the daemon's defaults, so don't get confused.
#
# To get a minimal test working, you will need to check the
# following items:
#   * MNHomeIPAddress
#   * HAIPAddress
#   * EnableFADecapsulation
#   * HomeNetAddr and HomeNetAddrPrefixLen (if using FA decapsulation)
#   * SPI and SharedSecret
# The rest of the items should work with their preset values in
# most cases and they can be used to fine tune the operations
# after the basic operation have been tested successfully.
#
########################################################################
#
# The Mobile Nodes's IP address in the Home Network.
MNHomeIPAddress 138.4.4.144

# The IP address of Mobile Node's Home Agent.
HAIPAddress 138.4.4.142

# EnableFADecapsulation < TRUE | FALSE >. TRUE enables a mode where
# the FA decapsulates the IP-within-IP encapsulated IP packets.
# FALSE disables this mode and sets the default mode where the
# MN decapsulates the IP-within-IP encapsulated IP packets.
```

```
# With FA decapsulation the MN uses its home address in the interface even in
# the foreign network and with MN decapsulation MN needs to acquire a
# co-located care-of address from the visited network (this needs an external
# program; see man pages for more information).
# The two modes cannot be used simultaneously.
EnableFADecapsulation FALSE

# Net route in home network (HomeNetAddr/HomeNetAddrPrefixLen)
# This is used mainly with FA decapsulation. If commented, the routing entry
# is not removed nor added. The home net entry may optionally be used with
# MN decapsulation - see MNDecapsRouteHandling option below.
#
# Example: 192.168.242.0/24
HomeNetAddr 138.4.4.128
HomeNetAddrPrefixLen 26

# Home net default gateway
# This entry can be used to force a gateway that the MN uses when it is
# at home. If this is left commented, the MN tries to use the default route
# that was in use when the program was started.
#
# HomeNetGateway 192.168.242.254
HomeNetGateway 138.4.4.189

###############################################################################
# a SPI (Security Parameter Index) must be defined for every MN.
# It is used for indexing the security association at the Home Agent.
SPI 1000
#
# The SharedSecret is provided as a HEX number string. The shared secret can
# also be given as a character string
# (e.g. character string "ABCDE" corresponds to HEX number string 4142434445).
# Note: RFC 2002 specifies that the default key size is 128 bits (i.e.
# 16 bytes or 32 hex 'characters'). Dynamics supports also other key lengths.
# SharedSecret < shared secret >
# SharedSecret 016A352B2F235E
SharedSecret "test"
#
# Authentication algorithm; currently only MD5 (code 1) is implemented
AuthenticationAlgorithm 1
#
# Replay prevention method:
#   0: none
#   1: time stamps
#   2: nonces
ReplayMethod 1
#
# Mobile Node may have optional security associations with Foreign
# Agents. If the security association exists an additional Mobile Node -
# Foreign Agent Authentication Extension is added to the registration requests.
#
# The following list contains the shared secrets indexed by SPI (and
# Foreign Agent IP address). The algorithm field specifies the method
# used for key distribution (currently only one is implemented,
# i.e. 1: encryption using MD5). The format of the share secret field is
# identical to the one used with the MN-HA security association list above.
```

```
#
#FA_SECURITY_BEGIN
# SPI FA IP Alg. Shared Secret
#2001 192.168.0.1 1 0123456789ABCDEF
#2002 192.168.0.2 1 "eslkfj89jr3hduh3R!as"
#FA_SECURITY_END

###############################################################################
# TunnelingMode < 1 | 2 | 3 | 4 >
# The packets between the MN and a Correspondent Node (CN) can be routed using
# different routes. This option can be used to select, which mode will be
# selected.
# Possible values:
# 1 = automatic, prefer reverse tunnel (i.e. bi-directional tunnel)
# 2 = automatic, prefer triangle tunnel (i.e. tunnel only in CN->MN direction)
# 3 = accept only reverse tunnel
# 4 = accept only triangle tunnel
TunnelingMode 1

# When MN can get its own co-located care-of address and use reverse tunneling,
# the normal method is to set the default route to the tunnel. This means that
# all the packets destined to other networks than the current subnet in the
# visited network are send via the HA. If the co-located COA is public, it can
# be used for sessions that do not need constant IP address (e.g. most of the
# web browsing). The following configuration option specifies the routing
# operation that is used with the co-located COA.
# Possible values:
#   0 = set default route to the tunnel
#   1 = set only the home net route to the tunnel (the above HomeNetAddr
#       options must be set)
#   2 = do not change the routing entries (i.e. some external means must be
#       used to direct traffic to the tunnel, e.g. manually adding host route
#       to a specific host)
MNDecapsRouteHandling 0

# DefaultTunnelLifetime is the lifetime suggested in registration
# The lifetime is defined in seconds, default value is 300.
# The request timer will be set according to this value. If the FA's agent
# advertisment has a smaller time, it is used instead.
# Special case: 65535 (or more) seconds means unlimited time (the binding will
# not expire)
# MNDefaultTunnelLifetime [ seconds ]
MNDefaultTunnelLifetime 300

# UDP port to be used for sending registration requests
# Port 434 is allocated for Mobile IP signaling and this should not be changed
# unless the network is known to use some other port (i.e. all the FAs and HAs
# must have the same port configured).
UDPPort 434

# Socket priority for signaling sockets (UDP) can be set with SO_PRIORITY to
# allow easier QoS configuration. If this argument is set, the given value is
# used as a priority for the signaling socket. E.g. CBQ class can be used to
# make sure that signaling is not disturbed by other traffic on a congested
# link.
# This feature is still undocumented and can be left commented.
```

```
#
# SocketPriority 1

# The log messages are written through syslog service. The facility to be
# used defaults to LOG_LOCAL0, but it can be set with this parameter
# to any of the possible facilities (LOG_AUTHPRIV, LOG_DAEMON, and so on).
# The processing of log messages is defined in /etc/syslog.conf file.
SyslogFacility LOG_DAEMON

# Ignore these interfaces. No agent advertisements are received nor
# agent solicitations sent for these interfaces.
IGNORE_INTERFACES_BEGIN
lo
dummy0
tunl0
gre0
IGNORE_INTERFACES_END

# Other programs may set routing entries so that the data connection may
# fail. The MN can try to enforce the routes that it believes should be used.
# This operation should currently be used only with FA decapsulation. If the
# route enforcement is activated the MN daemon prevents certain route changes.
EnforceRoutes FALSE

##############################################################################
# Mobile Nodes use unix domain sockets to communicate through their API
# interfaces.
# The group and owner must be names as strings, no groupIDs or userIDs are
# allowed. The file permissions are set in octal values like in chmod(1).
# The configuration parameters of the two API sockets are as follows:
MNAPIReadSocketPath "/var/run/dynamics_mn_read"
MNAPIReadSocketGroup "root"
MNAPIReadSocketOwner "root"
MNAPIReadSocketPermissions 0666
#
MNAPIAdminSocketPath "/var/run/dynamics_mn_admin"
MNAPIAdminSocketGroup "root"
MNAPIAdminSocketOwner "root"
MNAPIAdminSocketPermissions 0700
#
# Every configuration file must end to the keyword 'END'.
END
```