

Roaming without changing your IP address

Pervasively Online

Mobile IP allows a laptop to keep its IP address, despite moving to another network. TCP connections are not even interrupted by the switch from a WLAN to a wired Ethernet. TMip (Transparent Mobile IP) does not even require modifications to the client.

BY RALF SPENNEBERG



Roaming users with laptops typically access the Internet from many different points. Depending on their location, they may have WLAN, wired Ethernet, or dial-up access. The IP changes for each access method. And this means additional effort, as each vector has its own IP address scope.

When your address changes, you also lose your active network connections. Mobile IP technologies allow you to avoid this. The client is assigned the same home address, and remains accessible via this address. Connections to

this IP address stay up, despite switching to another network. The first official RFC on this subject was published way back in 1996. The current version, RFC 3344, is from August 2002 and describes Mobile IP support in IPv4 [6]. TMip (Transparent Mobile IP) [1] is a very interesting alternative, but we will get back to that later.

Functional Principle

A mobile machine that uses Mobile IP is always identified by its home IP. This address is independent of the machine's current location. If the computer is on its home network, it can communicate normally with other computers.

If the mobile computer is on a foreign network, it needs a kind of proxy, the so-called home agent, to communicate. The agent knows the current whereabouts of the mobile machine and uses a tunnel to forward packages to it. The foreign agent is located at the opposite end of the tun-

nel and resides on the network that currently hosts the mobile machine.

When the client notices that it is on a foreign network, it calls up the foreign agent to register with its home agent. The home agent stores the IP address of the foreign agent as the care-of address. It decides whether the registration is valid and, if so, the home agent acts as a mediator. To do so, the home agent receives any packets addressed to the client and tunnels them to the care-of address, that is to the foreign agent. The

Listing 1: Secondary MLR

```
# mlrd-secondary.rc
network_name linux-magazine
port 5555
foreground false
log_file /var/log/mlrd.log
status_file /var/log/mlrd.status
log true
grant primary.linux-magazine.com
```

Ralf Spenneberg is a freelance Unix/Linux trainer and author. Last year saw the release of his first book: "Intrusion Detection Systems for Linux Servers". Ralf has also developed various training materials.



Listing 2: Primary MLR

```
# mlrd-primary.rc
network_name linux-magazine
port 6554
foreground false
log_file /var/log/mlrd.log
status_file /var/log/mlrd.status
log true
cc_mlr secondary.linux-
magazine.com:5555
```

foreign agent receives these packets and forwards them to the mobile client.

Three components are required for the Mobile IP protocol to work: a home agent, a foreign agent, and a mobile client capable of recognizing the network on which it currently resides.

Implementations

A few implementations of the Mobile IP protocol for Linux have been around for a few years now. Development of most of these programs has been discontinued. Jean Tourrilhes [4] provides a useful overview of the available resources, but has not updated his own Mobile IP implementation since 1997.

Mosquitonet [2] is a well-known example, but it supports only the Linux 2.0 and 2.2 kernels. Dynamics HUT Mobile IP [3] from the University of Helsinki, Finland, is slightly more advanced; its implementation supports kernels 2.2 and 2.4. The software even runs on Microsoft operating systems that use the Cygwin DLLs. Unfortunately, the University of Helsinki discontinued development in October 2001.

Transparent Mobile IP

Transparent Mobile IP [1] is new development, but not standards-based. TMip, which was released under the BSD license, is designed for smaller local networks where mobile clients continually switch between various WLANs and wired networks. No changes need to be made to the mobile client. TMip provides Mobile IP functionality by means of a central service. The client simply needs to use DHCP (Dynamic Host Configuration Protocol).

A TMip network comprises three components: the MLR (Mobile Location Register), CN (Correspondent Nodes)

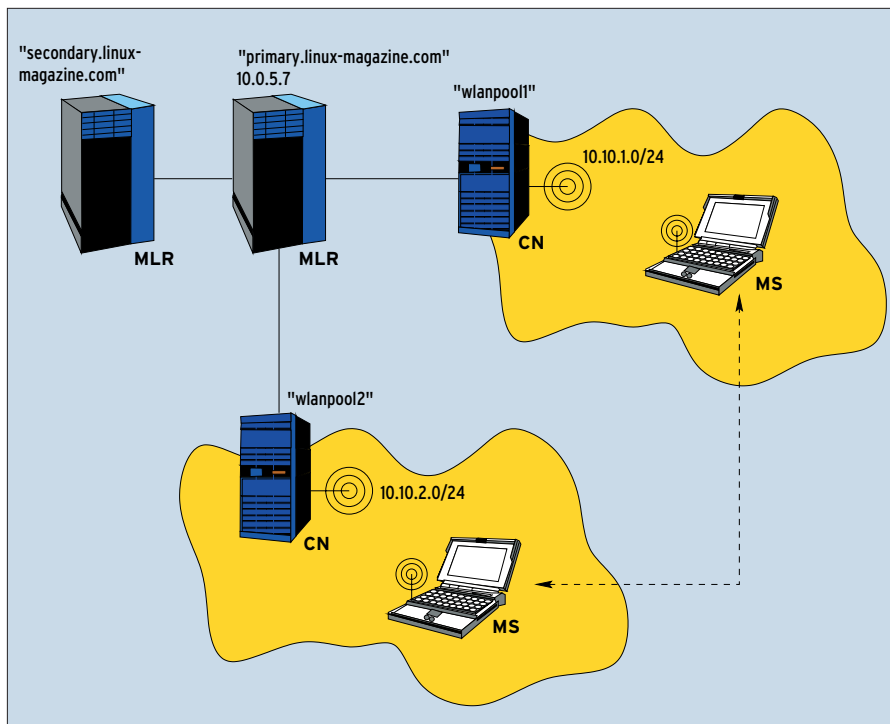


Figure 1: Transparent Mobile IP allows the client to switch arbitrarily between two networks.

and MS (Mobile Stations). The MLR stores the location of all MS at any given time. When a new mobile station logs onto its network, the CN makes a note of the network and registers the MS with a central MLR.

The MLR is like the memory of the TMip network. It handles the initial distribution of IP addresses via DHCP, stores the location of the mobile stations, and responds to the migration of a MS from one CN to another. As a failure of the MLR would have a fatal affect on the network, the MLR supports a so-called carbon copy mode. The primary MLR

automatically transfers its database to a secondary. If the primary register fails, the secondary can assume its role. When rebooted, the primary MLR will pick up its database from the secondary once more.

Mobile Location Register

Some installation and configuration steps are required to run TMip. TMip requires Libpcap – if this is not installed, *make* will complain of missing headers and exit. The first task concerns the MLR. Calling *make* in the *mlrd* directory will create the MLR daemon, *mlrd*, which then needs to be copied manually to a suitable location (such as */usr/local/sbin*). *make install* has not yet been implemented.

The next step is to create a *mlrd.rc* configuration file for the MLR daemon. Listing 1 shows an example of a file for the secondary MLR, and Listing 2 the primary MLR. These examples allow the primary and secondary MLRs to reside on the same machine, as they use different port numbers.

The *grant* instruction in Listing 1 allows the primary MLR to modify the database on the secondary MLR. The *cc_mlr* statement in Listing 2 does exactly that. It keeps the secondary daemon up to date.

Listing 3: CN Configuration

```
01 # tmipd.rc
02 mlr           primary.linux-
magazine.com
03 cn_name      wlanpool1.linux-magazine
04 cn_if        eth0
05 mobile_if    wlan0
06 network_name linux-magazine
07 addr_pool    wlan0 * *
08 dns_server   10.0.5.7
09 log_file     /var/log/tmipd.log
10 status_file  /var/log/tmipd.status
```

The secondary daemon needs to be running before the primary daemon is launched. The `-f` option tells the daemon which configuration file to use:

```
./mlrd -f mlrd-
secondary.rc
./mlrd -f mlrd-primary.rc
```

Missing access privileges to the logging directory are a common cause of errors. Among other things, the logfile indicates whether carbon copy mode is working properly. If you need to change the configuration at runtime, you can issue a sighup to the MLR to tell it to reparse the configuration file.

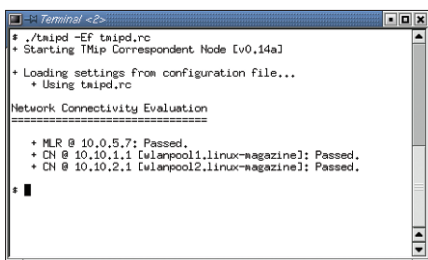
Correspondent Nodes

While the MLR is the central node for all the networks used by the mobile clients, the CN (Correspondent Node) is responsible for a single network. It handles the connection between the mobile stations and the network, typically requiring two network cards to do so. One card will be connected to the backbone (CN side), and the second will serve the mobile stations providing either wired or wireless access (WLAN).

The following example assumes two CNs (see Figure 1). Calling `make` in the `tmipd` directory compiles the CN daemon. The `tmipd` executable needs to be copied to a suitable path, such as `/usr/local/sbin`. Listing 3 shows a configuration file. `cn-if` designates the CN side interface and is used to access the other CNs and the MLR. The interface for the mobile stations is referred to as `mobil_if`.

DHCP inclusive

Typing `tmipd -f tmipd.rc` launches the TMip daemon, which in turn launches its own DHCP server. Each mobile sta-



```
Terminal <2>
# ./tmipd -E tmipd.rc
# Starting TMip Correspondent Node [v0.14a]
# Loading settings from configuration file...
# Using tmipd.rc

Network Connectivity Evaluation

+ MLR @ 10.0.5.7: Passed.
+ CN @ 10.10.1.1 [wlanpool1.linux-magazine]: Passed.
+ CN @ 10.10.2.1 [wlanpool2.linux-magazine]: Passed.
#
```

Figure 2: The `-E` option tells a CN to check whether it can reach the MLR and the other CNs.

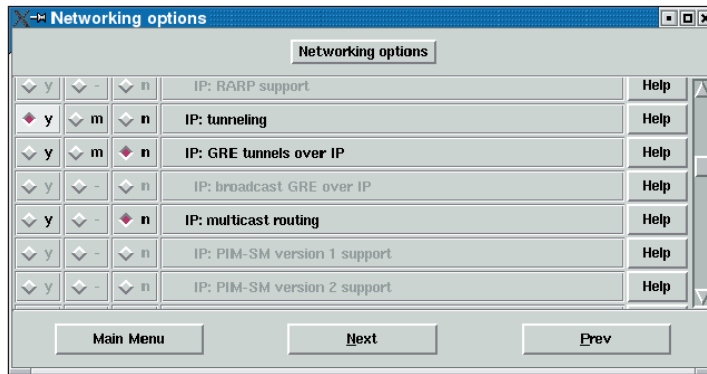


Figure 3: The kernel must support IPIP for inter-CN traffic. This option is located in the *Networking Options* and is called *IP: tunneling*.

tion that sends a DHCP request to the `wlan0` interface of the CN is assigned an IP address from this interface's scope (`addr_pool`). The DHCP server also tells the clients which DNS server is responsible for them. This address is configured in `dns_server`.

You can use the `-E` (network evaluation) option to check the connectivity between the CNs. Figure 2 shows a successful test. `tmipd` quits after performing the test. It is important that the CN can reach the MLR on TCP port 6554 and the other CNs on port 5554, and that it is capable of opening a IPIP tunnel to the other CNs. This may mean adjusting your firewall rules, and also enabling IPIP kernel support on the CN machines (see Figure 3).

Anti-spoofing protection can also cause issues. It makes sense to disable this option kernel side:

```
for i in /proc/sys/net/ipv4/
```

```
conf/*/rp_
filter; do echo
"0" > $i; done
```

In case of error, the logfile should provide detailed troubleshooting information. If this proves inadequate, you can stipulate the `-F` option when launching `tmipd` and `mlrpd`. This tells both applications to stay in the

foreground and output error messages to the screen. In this mode, you can even manage both programs by issuing commands to them. For example, `debug 4` increases the verbosity of logging. The keyboard shortcut `[Ctrl] + [D]` will terminate the applications.

Roaming

You need to enable a DHCP client on the mobile stations. Each machine is assigned an IP address by the CN in its current network – the address will come from the CN's scope. As long as the MS resides in the CN's scope (wired or wireless), TMip will act like a normal DHCP daemon. But things start to get interesting when a mobile station migrates to another network.

The aim is to let the station keep its IP address in the new network, and avoid any interruption to communications. The CN daemons records the MAC address and informs the MLR. The MLR

Listing 4: A Roaming MS

```
01 -> Mobile station detected in my cell [00:20:E0:6C:72:1E] (via IP or
    ARP activity)
02 + Establishing host's address allocation
03 + Success -> Restored from MLR
04 + Notifying MLR of host's new status
05 + Attempting mobile host handover [migrated from 10.10.1.1] parent
    10.10.1.1
06 + Contacting transaction participants
07 + Requesting tunnel between parent CN and local CN
08 + Agreed to use tunnel type [MLRP_TUN_4IN4]
09 + All OK, going for commit
10 + Using 10.10.1.9 on 10.10.2.0/255.255.255.0 gw:10.10.2.1
11 + Committing changes to MLR: Done.
12
13 -> Mobile host [00:01:f4:ee:90:44] has arrived in this cell
```

tells the CN that this station is known. As the station cannot communicate with its old IP on the new network, the CN opens up an IPIP tunnel to the home network and forwards packets to that network. At the other end, the home network sends packets destined for the MS through the tunnel to the current network. Listing 4 shows the logfile for roaming access from the perspective of the CN.

If roaming access involves a change of medium (from Ethernet to WLAN, for example), the mobile station might experience a problem. The Ethernet and wireless NICs need to use the same IP address. To allow this to work, the client itself must ascertain which of the NICs is currently enabled, and transmit packets via this card.

Laptop users can avoid this issue by using a second PCMCIA card, simply exchanging the card as required. Both cards need to report identical MAC addresses to be assigned the same IP address by DHCP. However, it is typically

quite trivial to assign a WLAN card the same MAC address as the Ethernet interface:

```
ifconfig wlan0 hw ether z
MAC address
```

TMip works fine on its own, if there is no medium change.

Conclusion

Laptop users have long sought for a way of maintaining existing connections while roaming. Mobile IP provides this. Unfortunately, the protocol does require some modifications to the mobile stations, and these modifications are available for only a few operating systems at present. In future, Mobile IP will probably be incorporated by most IP protocol stacks. If you need transparent roaming today, and without modifying your mobile stations, TMip may be exactly what you are looking for.

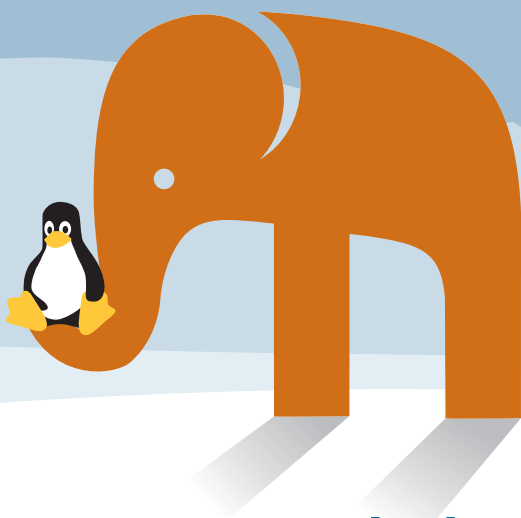
Transparent Mobile IP provides a similar service without actually honoring the

Mobile IP standard. The system automatically recognizes the migration of a mobile station from one network to another. Unfortunately, TMip does not provide encryption for the forwarded data. But standard VPN solutions such as FreeS/WAN [5] or OpenVPN provide a simple add-on. ■

INFO

- [1] Transparent Mobile IP:
http://www.slyware.com/projects_tmip.shtml
- [2] Mosquitonet Mobile IP:
<http://gunpowder.stanford.edu/mip/>
- [3] Dynamics HUT Mobile IP:
<http://www.cs.hut.fi/Research/Dynamics/>
- [4] Jean Tourrilhes Mobile IP:
http://www.hpl.hp.com/personal/Jean_Tourrilhes/MobileIP/mip.html
- [5] Ralf Spenneberg, "Virtual Private Networks with Linux 2.4 and FreeS/WAN 2.01", Linux Magazine, issue36, p.52
- [6] RFC 3344, "P Mobility Support for IPv4":
<http://www.ietf.org/rfc/rfc3344.txt>

EASY TO BUY • EASY TO SET UP • EASY TO SEE



p-p-p-pick up
a dedicated linux server

EasyVserver solutions
Debian or RedHat O/S
True "root" access
4, 6 or 8GB raid space
1 IP address
Highly secure

Make the most of Linux technology with the big name in Web registration and hosting packages. Our flexible, scalable, secure EasyVserver solutions start at just £39 per month. Back up by unrivalled support and know-how. If you want the best of Linux come along for the ride.

log on today at: www.easyspace.com

EasySpace 
your perfect partner for the web